On Connection between the Convolutive and Ordinary Nonnegative Matrix Factorizations

Anh Huy Phan[1], Andrzej Cichocki[1,⋆], Petr Tichavský[2,⋆⋆], and Zbyněk Koldovský[3]

[1] Lab for Advanced Brain Signal Processing, Brain Science Institute - RIKEN, Japan
[2] Institute of Information Theory and Automation, Prague, Czech Republic
[3] Faculty of Mechatronics, Informatics and Interdisciplinary Studies, Technical University of Liberec, Studentská 2, 461 17 Liberec, Czech Republic

Abstract. A connection between the convolutive nonnegative matrix factorization (NMF) and the conventional NMF has been established. As a result, we can convey arbitrary alternating update rules for NMF to update rules for CNMF. In order to illustrate the novel derivation method, a multiplicative algorithm and a new ALS algorithm for CNMF are derived. The experiments confirm validity and high performance of our method and of the proposed algorithm.

Keywords: nonnegative matrix factorization, convolutive nonnegative matrix factorization, nonnegative quadratic programming, ALS, music analysis.

1 Introduction

Expression of a nonnegative data matrix by set of basis patterns (objects) shifting along a direction (horizontal or vertical) of a given data following the convolutive model has recently attracted considerable interest from the view point of applications such as music analysis, image deconvolution [1–5, 9, 11]. This decomposition model is called the convolutive nonnegative matrix factorization (CNMF), and is considered as an extension of nonnegative matrix factorization (NMF). While there is a vast literature on algorithms for NMF [2], algorithms for CNMF are still very limited in the literature. All existing CNMF algorithms are based on the multiplicative update rules which minimize the least-squares error [1, 6, 9] or the Kullback-Leiber divergence [1, 4], or the generalized alpha- or beta- divergences [2, 3]. We note that the multiplicative algorithms have a relatively low complexity of each iteration but they are characterized by rather slow convergence and they sometimes converge to spurious local minima [7, 8].

Blind deconvolution of a given nonnegative data $\mathbf{Y} \in \mathbb{R}_+^{I\times J}$ is to find P basis patterns (objects) $\mathbf{A}^{(p)} = [\boldsymbol{a}_1^{(p)}, \boldsymbol{a}_2^{(p)}, \ldots, \boldsymbol{a}_{R_p}^{(p)}] \in \mathbb{R}_+^{I\times R_p}$, $p = 1, 2, \ldots, P$ and a location matrix $\mathbf{X} \in \mathbb{R}_+^{P\times J}$, each p-th row vector $\boldsymbol{x}_{p:}$ representing location and intensity of $\mathbf{A}^{(p)}$. For simplicity, assuming that all basis patterns $\mathbf{A}^{(p)}$ have the same size $R_p = R, \forall p$, otherwise they can be padded with zeros to the right. P basis patterns $\mathbf{A}^{(p)}$ are lateral slices of a

⋆ Also affiliated with the EE Dept., Warsaw University of Technology and with Systems Research Institute, Polish Academy of Science, Poland.

⋆⋆ The work of P. Tichavsky was supported by Ministry of Education, Youth and Sports of the Czech Republic through the project 1M0572 and by Grant Agency of the Czech Republic through the project 102/09/1278.

F. Theis et al. (Eds.): LVA/ICA 2012, LNCS 7191, pp. 288–296, 2012.

3-D tensor $\mathcal{A} \in \mathbb{R}^{I\times P\times R}$, i.e., $\mathcal{A}(i,p,r) = \mathbf{A}^{(p)}(i,r)$, $i = 1,\ldots,I$, $r = 1,\ldots,R$. Frontal slices $\mathbf{A}_r = [\boldsymbol{a}_r^{(1)}\, \boldsymbol{a}_r^{(2)} \cdots \boldsymbol{a}_r^{(P)}] \in \mathbb{R}^{I\times P}$ are component matrices, $r = 1,2,\ldots,R$. The mode-1 matricized version of $\mathcal{A}$ is denoted by $\mathbf{A}_{(1)} = [\mathbf{A}_1\, \mathbf{A}_2 \cdots \mathbf{A}_R] \in \mathbb{R}_+^{I\times RP}$.

We denote a shift matrix $\mathbf{S}_r$ of size $J \times J$ which is a binary matrix with ones only on the r-th superdiagonal for $r > 0$, or on the r-th subdiagonal for $r < 0$, and zeroes elsewhere. $\overset{r\rightarrow}{\mathbf{X}} = \mathbf{X}\mathbf{S}_r$ is an r column shifted version of $\mathbf{X}$ to the right, with the columns shifted in from outside the matrix set to zero. The relation between $\mathbf{Y}$, $\mathbf{A}^{(p)}$ and $\mathbf{X}$ can be expressed as

$$\mathbf{Y} = \sum_{r=1}^{R} \mathbf{A}_r \mathbf{X} \mathbf{S}_{r-1} + \mathbf{E} = \sum_{r=0}^{R-1} \mathbf{A}_{r+1} \overset{r\rightarrow}{\mathbf{X}} + \mathbf{E}. \tag{1}$$

Most CNMF algorithms were derived by considering (1) as R NMFs [3, 4, 6, 9]

$$\mathbf{Y} = \mathbf{A}_{r+1} \overset{r\rightarrow}{\mathbf{X}} + \left(\sum_{s\neq r} \mathbf{A}_{s+1} \overset{s\rightarrow}{\mathbf{X}} \right) + \mathbf{E} = \mathbf{A}_{r+1} \overset{r\rightarrow}{\mathbf{X}} + \mathbf{E}_{r+1}, \quad r = 0,1,\ldots,R-1. \tag{2}$$

For example, the multiplicative algorithms [3, 9] update $\mathbf{A}_r$ and $\mathbf{X}_r$

$$\mathbf{A}_{r+1} \leftarrow \mathbf{A}_{r+1} \circledast \left(\overset{r\leftarrow}{\mathbf{Y}}\, \mathbf{X}^T \right) \oslash \left(\overset{r\leftarrow}{\widehat{\mathbf{Y}}}\, \mathbf{X}^T \right), \quad r = 0,1,\ldots,R-1, \tag{3}$$

$$\mathbf{X}_{r+1} \leftarrow \mathbf{X} \circledast \left(\mathbf{A}_{r+1}^T \overset{r\leftarrow}{\mathbf{Y}} \right) \oslash \left(\mathbf{A}_{r+1}^T \overset{r\leftarrow}{\widehat{\mathbf{Y}}} \right), \tag{4}$$

where symbols "$\circledast$" and "$\oslash$" denote the Hadamard element-wise product and division. The coding matrix $\mathbf{X}$ is averaged over R estimations $\mathbf{X}_r$ in (4), i.e., $\mathbf{X} = \frac{1}{R}\sum_{r=1}^{R} \mathbf{X}_r$. Although the approach is simple and quite direct, its average update rule for $\mathbf{X}$ is not optimal. The reason is that the factorization (2) does not consider other shifts $\overset{s\rightarrow}{\mathbf{X}}(s \neq r)$ existing in $\mathbf{E}_{r+1}$. Moreover, practical simulations show that the average rules are not stable and converge slowly.

In the sequel, we present a connection between CNMF and NMF. Based on this, an arbitrary alternating update rule for NMF can be conveyed to CNMF. In order to illustrate the novel derivation method, a multiplicative algorithm and a robust ALS algorithm for CNMF are proposed.

2 A Novel Derivation for CNMF Algorithms

In general, update rules for $\mathcal{A}$ and $\mathbf{X}$ can be derived by minimizing a cost function which can be the Frobenius norm of the approximation error

$$D(\mathbf{Y}\|\widehat{\mathbf{Y}}) = \frac{1}{2}\|\mathbf{Y} - \widehat{\mathbf{Y}}\|_F^2 = \frac{1}{2}\|\mathbf{Y} - \sum_{r=1}^{R} \mathbf{A}_r \mathbf{X} \mathbf{S}_{r-1}\|_F^2. \tag{5}$$

From (1), the approximation of $\mathbf{Y}$ can be expressed as an NMF with rank PR, that is,

$$\mathbf{Y} = \left[\mathbf{A}_1\, \mathbf{A}_2 \cdots \mathbf{A}_R \right] \begin{bmatrix} \mathbf{X} \\ \overset{1\rightarrow}{\mathbf{X}} \\ \vdots \\ \overset{(R-1)\rightarrow}{\mathbf{X}} \end{bmatrix} + \mathbf{E} = \mathbf{A}_{(1)}\, \mathbf{Z} + \mathbf{E}, \tag{6}$$

or as an approximation of $\text{vec}(\mathbf{Y})$

$$\text{vec}(\mathbf{Y}) = \text{vec}\left(\sum_{r=0}^{R-1} \mathbf{A}_{r+1}\,\mathbf{X}\,\mathbf{S}_r + \mathbf{E}\right) = \mathbf{F}\ \text{vec}(\mathbf{X}) + \text{vec}(\mathbf{E})\,, \tag{7}$$

where $\mathbf{F} = \sum_{r=0}^{R-1}\left(\mathbf{S}_r^T \otimes \mathbf{A}_{r+1}\right) \in \mathbb{R}_+^{IJ\times PJ}$, '$\otimes$' denotes the Kronecker product. From (5), (6) and (7), we can alternatively update $\mathcal{A}$ or $\mathbf{X}$ while fixing the other according to the following procedure

$$\mathbf{X} = \arg\min_{\mathbf{X}} \|\,\text{vec}(\mathbf{Y}) - \mathbf{F}\,\text{vec}(\mathbf{X})\,\|_2^2, \quad \text{subject to } \mathbf{X} \geq \mathbf{0} \text{ with fixed } \mathcal{A}, \tag{8}$$

$$\mathcal{A} = \arg\min_{\mathbf{A}_{(1)}} \|\mathbf{Y} - \mathbf{A}_{(1)}\mathbf{Z}\|_F^2, \quad \text{subject to } \mathcal{A} \geq \mathbf{0} \text{ with fixed } \mathbf{X}. \tag{9}$$

Note that we can employ any update rules for NMF to update $\mathcal{A}$ and $\mathbf{X}$. For example, by employing the multiplicative update rules [7] we can update $\mathcal{A}$

$$\mathbf{A}_{(1)} \leftarrow \mathbf{A}_{(1)} \circledast \left(\mathbf{Y}\,\mathbf{Z}^T\right) \oslash \left(\widehat{\mathbf{Y}}\mathbf{Z}^T\right) = \mathbf{A}_{(1)} \circledast \left[\mathbf{Y}\,\mathbf{S}_r^T\,\mathbf{X}^T\right]_{r=0}^{R-1} \oslash \left[\widehat{\mathbf{Y}}\,\mathbf{S}_r^T\,\mathbf{X}^T\right]_{r=0}^{R-1}, \tag{10}$$

which can be rewritten for component matrices $\mathbf{A}_r$, $r = 1, 2, \ldots, R$

$$\mathbf{A}_r \leftarrow \mathbf{A}_r \circledast \left(\mathbf{Y}\,\mathbf{S}_{r-1}^T\,\mathbf{X}^T\right) \oslash \left(\widehat{\mathbf{Y}}\,\mathbf{S}_{r-1}^T\,\mathbf{X}^T\right), \quad r = 1, 2, \ldots, R. \tag{11}$$

The multiplicative Least-Squares update rule for $\mathbf{X}$ is given by

$$\begin{aligned}\text{vec}(\mathbf{X}) &\leftarrow \text{vec}(\mathbf{X}) \circledast \left(\mathbf{F}^T\,\text{vec}(\mathbf{Y})\right) \oslash \left(\mathbf{F}^T\,\text{vec}\left(\widehat{\mathbf{Y}}\right)\right) \\ &= \text{vec}(\mathbf{X}) \circledast \text{vec}\left(\sum_{r=0}^{R-1} \mathbf{A}_{r+1}^T \mathbf{Y}\,\mathbf{S}_r^T\right) \oslash \text{vec}\left(\sum_{r=0}^{R-1} \mathbf{A}_{r+1}^T \widehat{\mathbf{Y}}\,\mathbf{S}_r^T\right)\end{aligned}$$

or in the matrix form

$$\mathbf{X} \leftarrow \mathbf{X} \circledast \left(\sum_{r=0}^{R-1} \mathbf{A}_{r+1}^T \mathbf{Y}\,\mathbf{S}_r^T\right) \oslash \left(\sum_{r=0}^{R-1} \mathbf{A}_{r+1}^T \widehat{\mathbf{Y}}\,\mathbf{S}_r^T\right). \tag{12}$$

The update rules in (11) and (12) are particular cases of the multiplicative algorithm for CNMF2D [1]. However, its derivation is much simpler than that in [1]. Similarly, it is straightforward to derive update rules for the multiplicative Kullback-Leiber algorithms, the ALS algorithms. In addition, (6) and (7) also lead to condition on the number of patterns and the number of components $PR \leq min(I, J)$.

3 Alternative Least Squares Algorithm for CNMF

The alternative least squares (ALS) algorithm and its variations are commonly used for nonnegative matrix factorizations (see Chapter 4 [2]). For CNMF, it is straightforward to derive from (8) and (9) two ALS update rules given by

$$\mathbf{A}_{(1)} \leftarrow \left[\mathbf{Y}\,\mathbf{Z}^T\left(\mathbf{Z}\mathbf{Z}^T\right)^{-1}\right]_+, \qquad \text{vec}(\mathbf{X}) \leftarrow \left[\mathbf{Q}^{-1}\,\boldsymbol{b}\right]_+, \tag{13}$$

where $[x]_+ = \max(x, \varepsilon)$ is the element-wise rectifier which converts negative input to zero or a small enough value, and

$$\mathbf{Q} = \mathbf{F}^T\mathbf{F} = \sum_{r=0}^{R-1}\sum_{s=0}^{R-1}\left(\mathbf{S}_r\,\mathbf{S}_s^T \otimes \mathbf{A}_{r+1}^T\mathbf{A}_{s+1}\right) \quad \in \mathbb{R}_+^{L\times L}, L = JP, \tag{14}$$

$$\boldsymbol{b} = \mathbf{F}^T\,\mathrm{vec}(\mathbf{Y}) = \mathrm{vec}\left(\sum_{r=0}^{R-1}\mathbf{A}_{r+1}^T\mathbf{Y}\,\mathbf{S}_r^T\right) \in \mathbb{R}^L. \tag{15}$$

Although the ALS algorithm (13) is simple, it is not stable for sparse nonnegative data as illustrated in Section 5 for decomposition of spectrogram of sound sequences. In the sequel, a robust ALS algorithm is proposed for CNMF. From (5), (8), we consider the least-squares cost function which leads to a nonnegative quadratic programming (NQP) problem

$$D = \frac{1}{2}\|\,\mathrm{vec}(\mathbf{Y}) - \mathbf{F}\;\mathrm{vec}(\mathbf{X})\,\|_2^2 = \frac{1}{2}\|\mathbf{Y}\|_F^2 + \frac{1}{2}\boldsymbol{x}^T\mathbf{Q}\boldsymbol{x} - \boldsymbol{b}^T\,\boldsymbol{x}, \tag{16}$$

where $\boldsymbol{x} = \mathrm{vec}(\mathbf{X})$.

We denote $\tilde{\boldsymbol{x}} = [\tilde{x}_1\;\tilde{x}_2\;\cdots\;\tilde{x}_L]^T = \mathbf{Q}^{-1}\,\boldsymbol{b}$ the solution of the gradient $\nabla D(\boldsymbol{x}) = \mathbf{Q}\,\boldsymbol{x} - \boldsymbol{b}$, and $\mathcal{I}_+ \subset \{1, 2, \ldots, L\}$ a set of $L_1 \le L$ nonnegative entries, i.e. $\tilde{\boldsymbol{x}}_{\mathcal{I}_+} \ge \mathbf{0}$. If $L_1 = L$, then $\tilde{\boldsymbol{x}}$ is solution of (8) as in (13). Otherwise, the $(L - L_1)$ negative entries $\boldsymbol{x}_{\mathcal{I}_-}$, $\mathcal{I}_- = \{1, 2, \ldots, L\}\backslash\mathcal{I}_+$ are set to zeros by the rectifier according to (13). Hence, from (16), the rest L_1 variables $\boldsymbol{x}_{\mathcal{I}_+}$ are solutions of a reduced problem of a lower order L_1, that is

$$D = \frac{1}{2}\|\mathbf{Y}\|_F^2 + \frac{1}{2}\boldsymbol{x}_{\mathcal{I}_+}^T\,\mathbf{Q}_{\mathcal{I}_+}\,\boldsymbol{x}_{\mathcal{I}_+} - \boldsymbol{b}_{\mathcal{I}_+}^T\,\boldsymbol{x}_{\mathcal{I}_+}, \tag{17}$$

where $\mathbf{Q}_{\mathcal{I}_+}$ and $\boldsymbol{b}_{\mathcal{I}_+}$ are parts of $\mathbf{Q}$ and $\boldsymbol{b}$ whose row and column indices are specified by $\mathcal{I}_+$, respectively. If $\tilde{\boldsymbol{x}}_{\mathcal{I}_+} = \mathbf{Q}_{\mathcal{I}_+}^{-1}\,\boldsymbol{b}_{\mathcal{I}_+}$ has $L_2 < L_1$ nonnegative entries, we solve the subproblem of (17) of the lower order L_2. The procedure is recursively applied until there is not any negative entry $\tilde{x}$, i.e. $\mathcal{I}_- = \emptyset$ (see the subfunction `nqp` in Algorithm 1).

Similarly, the cost function (9) can also be expressed as an NQP problem to update $\mathbf{A}_{(1)}$ or horizontal slices $\mathbf{A}_{i::}$ defined as $\mathbf{A}_{i::}(p, r) = \mathcal{A}(i, p, r)$, $i = 1, 2, \ldots, I$

$$D = \frac{1}{2}\|\mathbf{Y}\|_F^2 + \frac{1}{2}\,\mathrm{vec}\left(\mathbf{A}_{(1)}^T\right)^T\left(\mathbf{I}_I \otimes \left(\mathbf{Z}\mathbf{Z}^T\right)\right)\mathrm{vec}\left(\mathbf{A}_{(1)}^T\right) - \mathrm{vec}\left(\mathbf{Z}\mathbf{Y}^T\right)^T\,\mathrm{vec}\left(\mathbf{A}_{(1)}^T\right) \tag{18}$$

$$= \frac{1}{2}\|\mathbf{Y}\|_F^2 + \sum_{i=1}^{I}\left(\frac{1}{2}\;\mathrm{vec}(\mathbf{A}_{i::})^T\,\left(\mathbf{Z}\mathbf{Z}^T\right)\mathrm{vec}(\mathbf{A}_{i::}) - \left(\boldsymbol{y}_{i:}\,\mathbf{Z}^T\right)\mathrm{vec}(\mathbf{A}_{i::})\right), \tag{19}$$

where $\boldsymbol{y}_{i:}$ denotes the i-th row vector of $\mathbf{Y}$. Finally, pseudo code of the (Q)ALS algorithm is described in Algorithm 1.

4 Initialization for CNMF Algorithms

In general, patterns $\mathbf{A}^{(p)}$ and coding matrix $\mathbf{X}$ can be initialized by nonnegative random values over multiple runs. The final solution can be chosen among them. Practical experiments show that although this simple method can produce acceptable solution, it needs

Algorithm 1. QALS Algorithm

Input: $\mathbf{Y}$: nonnegative matrix $I \times J$
P, R: number of patterns and components
Output: $\mathbf{A}^{(p)} \in \mathbb{R}_+^{I \times R}$, $p = 1, 2, \ldots, P$ and $\mathbf{X} \in \mathbb{R}_+^{R \times J}$
begin
  Initialize $\mathcal{A}$ and $\mathbf{X}$
  repeat
    $\text{vec}(\mathbf{X}) = \text{nqp}(\mathbf{Q}, \boldsymbol{b})$ // $\mathbf{Q}$ in (14), $\boldsymbol{b}$ in (15)
    for $i = 1$ **to** I **do** $\text{vec}(\mathbf{A}_{i::}) = \text{nqp}\left(\mathbf{Z}\mathbf{Z}^T, \mathbf{Z}\,\mathbf{y}_{i:}^T\right)$; // Update $\mathcal{A}$
  until *a stopping criterion is met*
end

function $\boldsymbol{x} = \text{nqp}(\mathbf{Q}, \boldsymbol{b})$ // $\mathbf{Q} \in \mathbb{R}_+^{L \times L}$, $\boldsymbol{b} \in \mathbb{R}^L$
begin
  $\mathcal{I}_+ = \{1, 2, \ldots, L\}$
  repeat
    $\tilde{\boldsymbol{x}}_{\mathcal{I}_+} = \mathbf{Q}_{\mathcal{I}_+}^{-1}\,\boldsymbol{b}_{\mathcal{I}_+}$; $\mathcal{I}_- = \{l \in \mathcal{I}_+ : \tilde{x}_l < 0\}$; $\mathcal{I}_+ = \mathcal{I}_+ \backslash \mathcal{I}_-$;
  until $\mathcal{I}_- = \emptyset$
  $\boldsymbol{x} = \max\{0, \tilde{\boldsymbol{x}}\}$
end

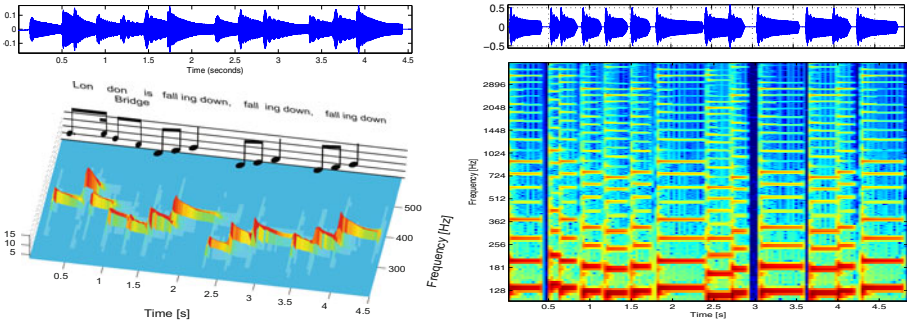
a large number of iterations and several (many) runs from different initial conditions to minimize probability of being stucked in false local minima instead of the global minimum. Noting that from the connection (6), $\mathbf{A}_{(1)}$ in approximation $\|\mathbf{Y} - \mathbf{A}_{(1)}\mathbf{Z}\|_F$ without nonnegativity constraints must comprise PR leading left singular components of $\mathbf{Y}$. Therefore, an SVD-based initialization method is proposed for $\mathbf{A}^{(p)}$ by taking in account that these leading singular components should be distributed among patterns $\mathbf{A}^{(p)}$. That is the first component matrix $\mathbf{A}_1$ takes R leading left singular components, the next R leading left components are for $\mathbf{A}_2$, and so on. Similarly, $\mathbf{X}$ can be initialized by P leading right singular vectors of $\mathbf{Y}$. Moreover, due to nonnegativity constraints, absolute values of singular vectors are used.

5 Simulations

In this section, we compare CNMF algorithms including QALS, the average multiplicative algorithm (aMLS) in (4) [9], the simultaneous multiplicative algorithm (MLS) in (12) [1] through decomposition of two music sequences into basic notes. For the first sequence, the sampled song "London Bridge" composed of five notes D4, E4, F4, G4 and A4 was played on a piano for 4.5 seconds illustrated in Fig. 1(a) (see Chapter 3 [2]). The signal was sampled at 8 kHz and filtered by using a bandpass filter with a bandwidth of 240 − 480 Hz. The magnitude spectrogram $\mathbf{Y}$ of size 257 frequency bins × 141 time frames is shown in Fig. 1(a), in which each rising part corresponds to the note actually played. It means $\mathbf{Y}$ is very sparse.

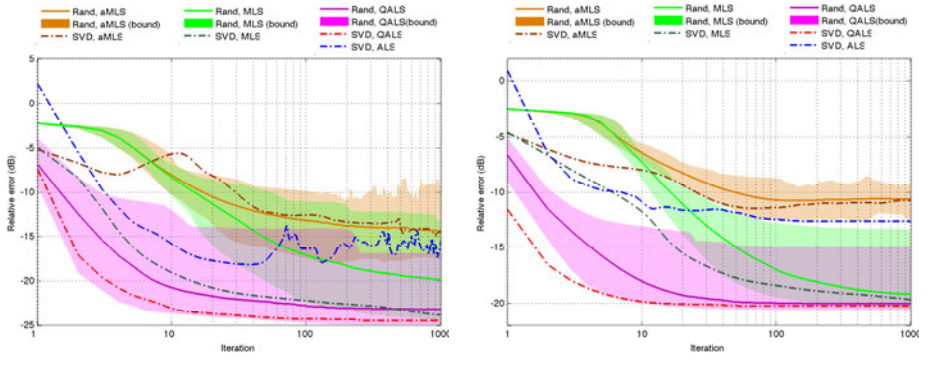
The second sequence was recorded from the same song but composed of five notes A3, G3, F3, E3 and D3 played on a guitar for 5 seconds (see Chapter 3 [2]). The log-frequency spectrogram $\mathbf{Y}$ (364 × 151) illustrated in Fig. 1(b) was converted from the

linear-frequency spectrogram with a quality factor $Q = 100$ and in the frequency range from $f_0 = 109.4$ Hz (bin 8) to $f_I = f_s/2 = 4000$ Hz (bin 257) [10]. The lowest approximation error for this spectrum is 27.56 dB when there was no decomposition.



(a) Sequence 1 (b) Sequence 2

Fig. 1. Waveforms and spectrograms of the two sequences "London Bridge"



(a) Sequence 1 (b) Sequence 2

Fig. 2. Convergence behavior of CNMF algorithms as function of the number of iterations for decomposition of two music sequences. Min-max bounds to the relative errors are shown shaded for random initialization.

Table 1. Performance comparison for various CNMF algorithms

Algorithm	Sequence 1			Sequence 2		
	SNR (dB)		RTime (secs)-SNR (dB)	SNR (dB)		RTime (secs) - SNR (dB)
	Random	SVD-based		Random	SVD-based	
aMLS	15.00 ± 1.98	15.49		11.52 ± 0.54	11.81	
MLS	19.75 ± 4.99	25.08	6.54 - 25.08	19.30 ± 2.18	19.42	13.10 - 19.42
QALS	**24.22± 3.23**	**25.74**	**1.37 - 25.14**	**20.25 ± 0.90**	**20.38**	**3.22 - 20.33**
ALS		18.12			15.07	

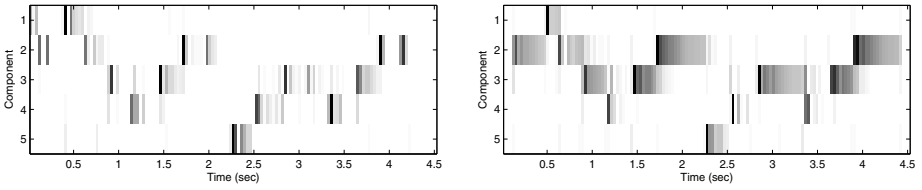
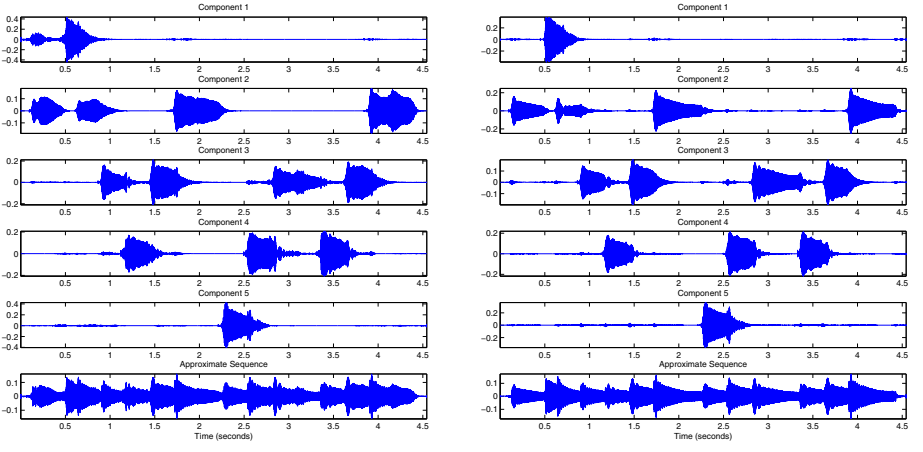


Fig. 3. Coding matrices $\mathbf{X}$ estimated by aMLS (left) and QALS (right) using SVD-based initialization, and matched with the piano roll for the sequence 1



(a) Basis and reconstructed sequences by aMLS, SNR = 15.49 dB.

(b) Basis and reconstructed sequences by QALS, SNR = 25.74 dB.

Fig. 4. Waveforms of basis spectral patterns $\mathbf{A}^{(p)}$ and the corresponding coding vectors $\boldsymbol{x}_{p:}$ estimated by aMLS and QALS. The reconstructed sequences (in the bottom) are summation of basis sequences.

For both sequences, CNMF algorithms were applied to extract 5 patterns $\mathbf{A}^{(p)} \in \mathbb{R}_+^{I\times 10}$, and to explain the observed sequence through basis audio sequences. The approximate signals were reconstructed from basis patterns, and normalized to have the same energy as the original signal. Algorithms were initialized by the nonnegative random values over 100 times or by absolute values of leading singular vectors extracted from $\mathbf{Y}$. The relative approximation errors $20\log_{10}\left(\dfrac{\|\mathbf{Y}-\widehat{\mathbf{Y}}\|_F}{\|\mathbf{Y}\|_F}\right)$ (dB) with different initializations are illustrated as function of the number of iterations in Fig. 2. Moreover, Table 1 provides the signal-to-noise (SNR) ratio between the original audio sequence and its approximate signal $-20\log\left(\dfrac{\|\boldsymbol{y}-\hat{\boldsymbol{y}}\|_2}{\|\boldsymbol{y}\|_2}\right)$ (dB). Running time (seconds) and SNR as the algorithm converged are given in columns 3 and 5 in Table 1, respectively.

As seen in Fig. 2, aMLS (orange shading) is not stable for both sequences, and often gets stuck in local minima by random initialization. Although SVD-based initialization

can improve its performance (brown dash-dot lines), the cost values of aMLS did not always decrease. This was caused by the average rule (4) which is not optimal here. MLS which simultaneously updates $\mathbf{X}$ has better convergence (green shading) than aMLS. Among algorithms with random initialization, QALS mostly achieved the lowest approximation error (magenta shading and magenta dash lines). Moreover, QALS reached the converged values ealier, after 100 iterations, than MLS and aMLS.

Fig. 2 also indicates that SVD-based initialization improved performance compared with random initialization. QALS (dash-dot red lines) converged after 20 iterations in 1.37 seconds and in 3.22 seconds for two sequences, respectively. Whereas MLS (dash-dot green lines) run at least 1000 iterations to achieve similar approximation errors in 6.54 seconds and 13.10 seconds respectively. Running time was measured on a computing server which has 2 quadcore 3.33 GHz processors and 64 GB memory. Therefore, although complexity per iteration of QALS is higher than that of MLS, QALS may converge earlier than MLS due to significantly less computation iterations.

Fig. 3 illustrates two coding matrices $\mathbf{X}$ estimated by QALS and aMLS for the sequence 1 after matching with its piano roll. The coding map $\mathbf{X}$ by QALS is more similar to the piano roll than that of aMLS. The patterns appear continually as the notes played in the piano roll. In addition, waveforms constructed from the basis spectral patterns $\mathbf{A}^{(p)}$ and the corresponding coding row vectors $\boldsymbol{x}_{p:}$, $p = 1, \ldots, 5$, are illustrated in Fig. 4 for aMLS and QALS. aMLS achieved a reconstruction error of 15.49 dB. Whereas QALS obtained much higher performance with an approximation error of 25.74 dB. The standard ALS achieved an error of 18.20 dB. More comparisons between the algorithms are given in Table 1, which confirms the superior performance of QALS.

6 Conclusions

A connection between CNMF and NMFs is presented and allows us to straightforwardly extend arbitrary alternating NMF update rules to CNMF. The novel derivation method has been illustrated by two simple CNMF algorithms. In addition, a novel (Q)ALS algorithm is proposed and has been confirmed to give higher performance than those of the multiplicative algorithms in the sense of convergence, and reconstruction error. Moreover, based on the new connection, an SVD-based initialization method has been proposed for CNMF algorithms.

References

1. Schmidt, M.N., Mørup, M.: Nonnegative Matrix Factor 2-D Deconvolution for Blind Single Channel Source Separation. In: Rosca, J.P., Erdogmus, D., Principe, J.C., Haykin, S. (eds.) ICA 2006. LNCS, vol. 3889, pp. 700–707. Springer, Heidelberg (2006)
2. Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.: Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation. Wiley, Chichester (2009)
3. O'Grady, P.D., Pearlmutter, B.A.: Discovering speech phones using convolutive non-negative matrix factorization with a sparseness constraint. Neurocomput. 72, 88–101 (2008)
4. Smaragdis, P.: Convolutive speech bases and their application to supervised speech separation. IEEE Transactions on Audio, Speech and Language Processing 15(1), 1–12 (2007)

5. Ozerov, A., Févotte, C.: Multichannel nonnegative matrix factorization in convolutive mixtures with application to blind audio source separation. In: ICASSP 2009, USA, pp. 3137–3140 (2009)
6. Wang, W., Cichocki, A., Chambers, J.A.: A multiplicative algorithm for convolutive non-negative matrix factorization based on squared Euclidean distance. IEEE Transactions on Signal Processing 57(7), 2858–2864 (2009)
7. Lee, D.D., Seung, H.S.: Algorithms for Nonnegative Matrix Factorization, vol. 13. MIT Press (2001)
8. Berry, M., Browne, M., Langville, A., Pauca, P., Plemmons, R.: Algorithms and applications for approximate nonnegative matrix factorization. Computational Statistics and Data Analysis 52(1), 155–173 (2007)
9. Smaragdis, P.: Non-Negative Matrix Factor Deconvolution; Extraction of Multiple Sound Sources from Monophonic Inputs. In: Puntonet, C.G., Prieto, A.G. (eds.) ICA 2004. LNCS, vol. 3195, pp. 494–499. Springer, Heidelberg (2004)
10. Ellis, D.: Spectrograms: Constant-q (log-frequency) and conventional (linear) (May 2004), http://labrosa.ee.columbia.edu/matlab/sgram/
11. FitzGerald, D., Cranitch, M., Coyle, E.: Extended Nonnegative Tensor Factorisation Models for Musical Sound Source Separation. In: Computational Intelligence and Neuroscience, vol. 2008, Article ID 872425, 15 pages (2008), doi:10.1155/2008/872425