2007 IEEE International Symposium
on Signal Processing and Information Technology

SQUARED EUCLIDEAN DISTANCE BASED CONVOLUTIVE NON-NEGATIVE MATRIX FACTORIZATION WITH MULTIPLICATIVE LEARNING RULES FOR AUDIO PATTERN SEPARATION

Wenwu Wang

Centre for Vision, Speech and Signal Processing
University of Surrey, Guildford, United Kingdom
E-mail: w.wang@surrey.ac.uk

ABSTRACT

A novel algorithm for convolutive non-negative matrix factorization (NMF) with multiplicative rules is presented in this paper. In contrast to the standard NMF, the low rank approximation is represented by a convolutive model which has an advantage of revealing the temporal structure possessed by many realistic signals. The convolutive basis decomposition is obtained by the minimization of the conventional squared Euclidean distance, rather than the Kullback-Leibler divergence. The algorithm is applied to the audio pattern separation problem in the magnitude spectrum domain. Numerical experiments suggest that the proposed algorithm has both less computational loads and better separation performance for auditory pattern extraction, as compared with an existing method developed by Smaragdis.

1. INTRODUCTION

Non-negative matrix factorization (NMF) is an emerging technique for data analysis [1] [2]. By representing a data matrix with a product of low rank matrices, NMF provides an effective way for finding latent structures or features in original data. Under the non-negative constraint, NMF also gives "parts" based representation [2]. These promising properties have made NMF very useful for many applications in signal and image processing [2] [6] [5] [7] [8].

The standard NMF model given in [2] has shown to be satisfactory and sufficient for a number of signal processing tasks. For the time-frequency analysis of audio signals, however, the obtained single basis via decomposition may not be adequate to capture the temporal dependency of the repeating patterns within the signal. This is especially true for the signals whose frequencies vary with time. To address this issue, Smaragdis has presented a convolutive model of NMF in [6] and [7], where the input data matrix is denoted as the convolution of a group of shifted matrices. Smaragdis has further developed a multiplicative learning algorithm based on the minimization of Kullback-Leibler (KL) divergence, by extending the multiplicative rules in [2] for standard NMF to those for the convolutive case.

In this paper, we develop a novel convolutive NMF algorithm with multiplicative forms based exclusively on the minimization of the Euclidean distance. By applying this algorithm to the separation of audio objects (repeating patterns), whose frequencies vary with time, we demonstrate its superior performance to the standard NMF. Compared with the algorithm in [6] and [7], the proposed algorithm is computationally less complex, while having better separation performance, and thereby provides a good alternative to Smaragdis's algorithm.

The remainder of the paper is organized as follows. The next section briefly reviews the standard NMF. The proposed convolutive NMF algorithm based on the Euclidean distance is detailed in section 3. In section 4, we show how to prepare the non-negative input matrix, i.e., audio magnitude spectrum matrix in our application. Section 5 demonstrates its performance using numerical examples. And finally, section 6 concludes the paper.

2. STANDARD NMF

Given an $M \times N$ non-negative matrix $\mathbf{X} \in \mathbb{R}_+^{M\times N}$, the goal of NMF is to find nonnegative matrices $\mathbf{W} \in \mathbb{R}_+^{M\times R}$ and $\mathbf{H} \in \mathbb{R}_+^{R\times N}$, such that

$$\mathbf{X} \approx \mathbf{WH} \tag{1}$$

where R is the rank of the factorization, generally chosen to be smaller than M (or N), or akin to $(M+N)R < MN$, which results in the extraction of some latent features whilst reducing some redundancies in the input data. To find the optimal choice of matrices $\mathbf{W}$ and $\mathbf{H}$, we should minimize the reconstruction error between $\mathbf{X}$ and $\mathbf{WH}$. Several error functions have been proposed for this purpose [1]-[5]. For instance, an appropriate choice is to use the criterion based

978-1-4244-1835-0/07/$25.00 ©2007 IEEE

on the squared Euclidean distance,

$$(\hat{\mathbf{W}}, \hat{\mathbf{H}}) = \underset{\mathbf{W},\mathbf{H}}{\arg\min} \left\| \mathbf{X} - \hat{\mathbf{X}} \right\|_F^2 \quad (2)$$

where $\hat{\mathbf{W}}$ and $\hat{\mathbf{H}}$ are the estimated optimal values of $\mathbf{W}$ and $\mathbf{H}$, $\|\cdot\|_F$ denotes the Frobenius norm, and $\hat{\mathbf{X}}$ is given by

$$\hat{\mathbf{X}} = \mathbf{W}\mathbf{H} \quad (3)$$

Alternatively, we can also minimize the error function based on the extended KL divergence,

$$(\hat{\mathbf{W}}, \hat{\mathbf{H}}) = \underset{\mathbf{W},\mathbf{H}}{\arg\min} \sum_{m=1}^{M} \sum_{n=1}^{N} \mathbf{D}_{mn} \quad (4)$$

where $\mathbf{D}_{mn}$ is the mn-th element of the matrix $\mathbf{D}$ which is given by

$$\mathbf{D} = \mathbf{X} \odot \log[\mathbf{X} \oslash \hat{\mathbf{X}}] - \mathbf{X} + \hat{\mathbf{X}} \quad (5)$$

where $\odot$ and $\oslash$ denote the Hadamard (element-wise) product and division respectively. Although gradient decent and conjugate gradient approaches can both be applied to minimize these cost functions, we are particularly interested in the multiplicative rules developed by Lee and Seung [2] [3]. In compact form, the multiplicative update rules for minimizing criterion (2) can be re-written as

$$\mathbf{H}^{q+1} = \mathbf{H}^q \odot ((\mathbf{W}^q)^T \mathbf{X}) \oslash ((\mathbf{W}^q)^T \mathbf{W}^q \mathbf{H}^q) \quad (6)$$

$$\mathbf{W}^{q+1} = \mathbf{W}^q \odot (\mathbf{X}(\mathbf{H}^{q+1})^T) \oslash (\mathbf{W}^q \mathbf{H}^{q+1} (\mathbf{H}^{q+1})^T) \quad (7)$$

where q is the iteration index, and $(\cdot)^T$ is the matrix transpose operator. Comparatively, these rules are easy to implement and also have good convergence performance. Additionally, a step size parameter which is normally required for gradient algorithms, is not necessary in these rules.

3. PROPOSED CONVOLUTIVE NMF BASED ON SQUARED EUCLIDEAN DISTANCE

To take into account the potential dependency between the neighboring columns of the input data matrix $\mathbf{X}$, the standard (instantaneous) NMF model, i.e., equation (1), is extended to a convolutive form:

$$\mathbf{X} \approx \sum_{p=0}^{P-1} \mathbf{W}(p) \overset{p\rightarrow}{\mathbf{H}} \quad (8)$$

where $\mathbf{W}(p) \in \mathbb{R}_+^{M\times R}$, $p = 0, \cdots, P-1$, are a set of bases, $\mathbf{H} \in \mathbb{R}_+^{R\times N}$ is a weighting matrix, and $\overset{p\rightarrow}{\mathbf{H}}$ shifts the columns of $\mathbf{H}$ by p spots to the right, with the columns shifted in from outside the matrix set to zero. Analogously, $\overset{\leftarrow p}{\mathbf{H}}$ shifts the columns of $\mathbf{H}$ by p spots to the left. These notations will also be used for the shifting operations of other matrices throughout the paper. Note that, $\overset{0\rightarrow}{\mathbf{H}} = \overset{\leftarrow 0}{\mathbf{H}} = \mathbf{H}$. Essentially, $\mathbf{W}\mathbf{H}$ in equation (1) is replaced with a convolutive operation $\sum_{p=0}^{P-1} \mathbf{W}(p) \overset{p\rightarrow}{\mathbf{H}}$ in equation (8), i.e., a sum of shifted matrix products. With the convolutive model, the temporal continuity possessed by many audio signals can be expressed more effectively in the time-frequency domain, especially for those signals whose frequencies vary with time (see numerical examples in simulations).

To find a decomposition with the form of equation (8), Smaragdis has developed multiplicative learning rules based on the extended KL divergence (4). With our notational convention, these rules can be re-written as

$$\mathbf{H}^{q+1} = \mathbf{H}^q \odot (((\mathbf{W}^q(p))^T \overset{\leftarrow p}{\tilde{\mathbf{X}}^q}) \oslash ((\mathbf{W}^q(p))^T \boldsymbol{\Xi})) \quad (9)$$

$$\mathbf{W}^{q+1}(p) = \mathbf{W}^q(p) \odot ((\tilde{\mathbf{X}}^q (\overset{p\rightarrow}{\mathbf{H}^{q+1}})^T) \oslash (\boldsymbol{\Xi} (\overset{p\rightarrow}{\mathbf{H}^{q+1}})^T)) \quad (10)$$

where $\boldsymbol{\Xi}$ is an $M \times N$ matrix whose elements are all set to 1, $\tilde{\mathbf{X}}^q = \mathbf{X} \oslash \hat{\mathbf{X}}^q$, and $\hat{\mathbf{X}}^q$ is an estimate of $\mathbf{X}$ given by (12).

Instead of using (4), we use the squared Euclidean distance and develop an alternative multiplicative algorithm. Essentially, we aim at extending the learning rules (6) and (7) to their convolutive forms. Therefore, the objective function we are interested is

$$(\hat{\mathbf{W}}(p), \hat{\mathbf{H}}) = \underset{\mathbf{W}(p),\mathbf{H}}{\arg\min} \left\| \mathbf{X} - \hat{\mathbf{X}} \right\|_F^2 \quad (11)$$

where $\hat{\mathbf{X}}$ is the estimate of $\mathbf{X}$ given by

$$\hat{\mathbf{X}} = \sum_{p=0}^{P-1} \mathbf{W}(p) \overset{p\rightarrow}{\mathbf{H}}. \quad (12)$$

To minimize this cost function, we can effectively treat the convolutive model as a collection of a group of standard NMF problems. This implies that $\mathbf{H}$ and a set P of $\mathbf{W}(p)$, i.e., totally $P + 1$ matrices, are required to be updated in each iteration. With appropriate yet simple matrix shifting operations, our new update equations for minimizing (11) can be formulated as

$$\mathbf{W}^{q+1}(p) = (\mathbf{W}^{q+1}(p) \odot ((\mathbf{X} (\overset{p\rightarrow}{\mathbf{H}^q})^T) \oslash (\hat{\mathbf{X}}^q (\overset{p\rightarrow}{\mathbf{H}^q})^T)) \quad (13)$$

$$\mathbf{H}^{q+1} = \mathbf{H}^q \odot (((\mathbf{W}^{q+1}(p))^T \overset{\leftarrow p}{\mathbf{X}}) \oslash ((\mathbf{W}^{q+1}(p))^T \overset{\leftarrow p}{\hat{\mathbf{X}}^q})) \quad (14)$$

where $p = 0, \cdots, P-1$. Note, that the above equations may lead to a biased estimate of $\mathbf{H}$, as all $\mathbf{W}(p)$ share the same $\mathbf{H}$. In order to mitigate this effect, we can update all $\mathbf{W}(p)$ first, and then take the average of all the updates for

$\mathbf{H}$, that is

$$\mathbf{H}^{q+1} = \frac{1}{P}\sum_{p=0}^{P-1}\mathbf{H}^{q}(p) \tag{15}$$

where $\mathbf{H}^{q}(p)$ is given by

$$\mathbf{H}^{q}(p) = \mathbf{H}^{q} \odot (\mathbf{W}^{q+1}(p)^{T}\overset{\leftarrow p}{\mathbf{X}}) \oslash (\mathbf{W}^{q+1}(p)^{T}\overset{\leftarrow p}{\hat{\mathbf{X}}^{q}}). \tag{16}$$

From equations (13) and (14), it is clear that the updates of $\mathbf{W}(p)$ and $\mathbf{H}$ both rely on the update of $\hat{\mathbf{X}}$, which, on the other hand, depends on the instantaneous values of $\mathbf{W}(p)$ and $\mathbf{H}$, according to equation (12). This means that $\hat{\mathbf{X}}$ should be updated correspondingly once each $\mathbf{W}(p)$ is updated. Nevertheless, updating the whole equation (12) is computationally inefficient if only an individual $\mathbf{W}(p)$ has a new value. Therefore, instead of directly using equation (12), we use the following simpler formulation

$$\hat{\mathbf{X}}^{q} = \hat{\mathbf{X}}^{q} - \mathbf{W}^{q}(p)\overset{p\rightarrow}{\mathbf{H}^{q}} + \mathbf{W}^{q+1}(p)\overset{p\rightarrow}{\mathbf{H}^{q}} \quad (p = 0, \cdots, P-1) \tag{17}$$

where $\hat{\mathbf{X}}^{q}$ is updated recursively to accommodate the new values of each $\mathbf{W}(p)$ (inside the P loops), and the initial value of $\hat{\mathbf{X}}^{q}$ $(q > 1)$ in the right hand side (RHS) of equation (17) is obtained at the end of $(q-1)$-th iteration (outside the P loops), when the recursions are completed. For $q = 1$, $\hat{\mathbf{X}}^{q}$ in the RHS of equation (17) is still calculated via equation (12). In practice, we found that the non-negative property of $\hat{\mathbf{X}}^{q}$ may not be guaranteed, due to the subtraction operation and small numerical errors. The small negative values can be prevented by using the projection operation:

$$\hat{\mathbf{X}}^{q} = \max(\epsilon, \hat{\mathbf{X}}^{q}) \tag{18}$$

where $\max(\cdot)$ takes the maximum value of its arguments, and ϵ is a trivial constant, typically, $\epsilon = 10^{-9}$ in our implementation. The algorithm stops iterations when the following criterion is satisfied,

$$\frac{\left\|\hat{\mathbf{X}}^{q+1} - \hat{\mathbf{X}}^{q}\right\|_{F}}{\left\|\hat{\mathbf{X}}^{q}\right\|_{F}} < \zeta \tag{19}$$

where ζ is a small constant.

In summary, the adaptation of equations (13), (17), (18), (16), (15) and (19) in order represents our proposed algorithm. If $P = 1$, it basically reduces to the standard NMF algorithm represented by equations (6) and (7). If $P > 1$, the computational load of the proposed algorithm is approximately P times that of the standard NMF. However, compared with the multiplicative algorithms in [6] and [7], the proposed algorithm is computationally less complex, due to the following two reasons. First, there is one less matrix Hadamard division in our algorithm for computing $\mathbf{W}^{q}(p)$ and $\mathbf{H}^{q}$ at each loop p, which amounts to a reduction of $2PMN$ element divisions in each iteration. Second, a simpler computing method, i.e. equation (17), is adopted for the update of $\hat{\mathbf{X}}$, which amounts to approximately $(P-2)(2R-1)MN$ less element operations (multiplications and additions) in each iteration.

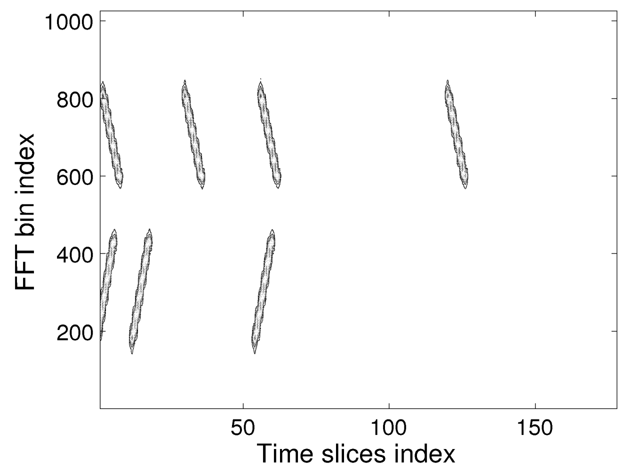

Fig. 1. The contour plot of the magnitude spectrum matrix $\mathbf{X}$.

4. NON-NEGATIVE DECOMPOSITION OF MAGNITUDE SPECTRA

In our problem, the non-negative matrix $\mathbf{X}$ is generated as the magnitude spectra of the input audio data. We denote the original audio signal as $s(t)$, where t is the time instant. Using a T-point windowed DFT, a time-domain signal $s(t)$ can be converted into a frequency-domain time-series signal as

$$S(f, k) = \sum\nolimits_{\tau=0}^{T-1} s(k\delta + \tau)w(\tau)e^{-j2\pi f\tau/T} \tag{20}$$

where $w(\tau)$ denotes a T-point window function, $j = \sqrt{-1}$, δ is the time shift between the adjacent windows, and f is a frequency index, $f = 0, 1, \cdots, T-1$. Clearly, the time index k in $S(f, k)$ is generally not a one-to-one mapping to the time index t in $s(t)$. If the whole signal has, for instance, L samples, then the maximum value of k, i.e. K, is given as $K = \lfloor (L-T)/\delta \rfloor$, where $\lfloor\cdot\rfloor$ is an operator taking the maximum integer no greater than its argument[1]. Let $\tilde{S}(f, k)$ be the absolute value of $S(f, k)$, we can then generate $\mathbf{X}$ by packing $\tilde{S}(f, k)$ together, where $f = 0, \cdots, T/2+1$

[1]In practice, zero-padding may be required to allow the remaining r $(0 \leq r < \delta)$ samples at the end of the signal to be covered by the analysis window.

(i.e., only considering non-negative frequency bins due to the symmetrical property of the spectrum). The dimension of $\mathbf{X}$, i.e. $M \times N$, then becomes $(T/2+1) \times K$ [8].

Upon the convergence of the algorithm proposed in section 3, $\mathbf{X}$ can be effectively decomposed into the convolution of P non-negative matrices, denoted as $\mathbf{W}^o(p) \in \mathbb{R}_+^{(T/2+1)\times R}$ and $\mathbf{H}^o \in \mathbb{R}_+^{R\times K}$, i.e., the corresponding local optimum values of $\mathbf{W}(p)$ and $\mathbf{H}$ respectively. An advantage of exploiting spectral matrix is that both the obtained basis matrices $\mathbf{W}^o(p)$ and $\mathbf{H}^o$ have meaningful interpretation. That is, $\mathbf{H}^o$ is a dimension-reduced matrix which contains the bases of the temporal patterns while $\mathbf{W}^o(p)$ contains the frequency patterns of the original data. All P set of $\mathbf{W}^o(p)$ together contain both frequency and temporal information of time-frequency patterns (i.e., audio objects) of the original audio signal.

5. NUMERICAL EXPERIMENTS

We apply the proposed algorithm to the separation of audio objects with repeating patterns, for both artificial and real music audio signals.

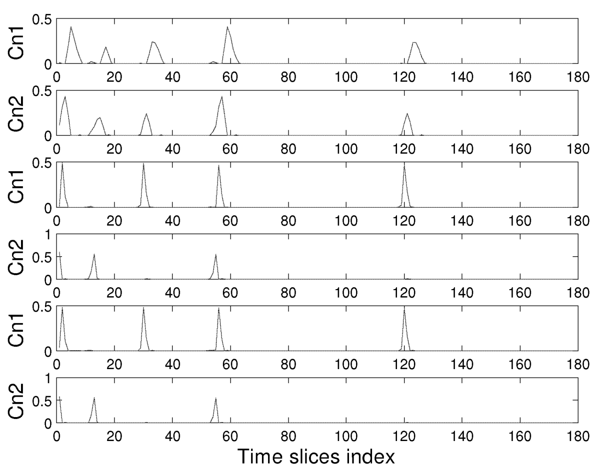

Fig. 2. Visualization of matrix $\mathbf{H}^o$. "Cn1 (or Cn2)" denotes Column 1 (or 2) of $\mathbf{H}^o$. Results were obtained by using the standard NMF (the above two plots), Smaragdis's algorithm (the middle two plots), and our proposed algorithm (the below two plots).

5.1. Artificial Audio

Two audio signals are generated. One contains 3 repeating patterns with frequencies changing linearly with time from 123Hz to 223Hz, and the other contains 4 repeating patterns whose frequencies change linearly from 600Hz to 515Hz. These two signals are added together to generate a mixed signal. The sampling frequency f_s for both signals is 1500Hz. The whole signal has $L = 45000$ samples with a length of 30 seconds. This mixture is transformed into the frequency domain by the procedure described in Section 4, where the frame length T of the Fast Fourier transform (FFT) is set to 2048 samples, i.e., the frequency resolution is approximately 0.73Hz. The signal is segmented by a Hamming window with the window size being set to 600 samples (400ms), and the time shift δ to 250 samples (approximately 167ms), that is, an overlap between the neighboring frames is used. The small size of the signal segments is chosen to guarantee a sufficient time resolution, and each segment is then zero-padded to have the same size as T for FFT operation. The generated matrix $\mathbf{X}$ is visualized in Figure 1.

The factorization rank R is set to 2, i.e., exactly the same as the total number of the signals in the mixture. The matrices $\mathbf{W}(p)$ and $\mathbf{H}$ are initialized as the absolute values of random matrices. P is set[2] to 6, and ζ to 0.0001. All tests were running on a computer whose CPU speed is 1.8GHz. The proposed algorithm is compared with the standard NMF ($P = 1$) and Smaragdis's algorithm (with corresponding parameters set identical to those in our algorithm). We plot $\mathbf{H}^o$ and $\mathbf{W}^o(p)$ in Figure 2 and Figure 3 respectively. It is clear from these figures that the

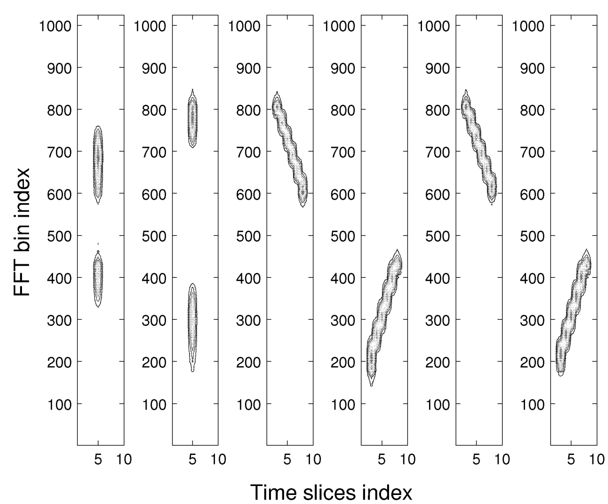

Fig. 3. Visualization of all matrices $\mathbf{W}^o(p)$, $p = 0, ...5$, with the first and second rows of all $\mathbf{W}^o(p)$ plotted on the left and right side respectively. Results were obtained by using the standard NMF (the left two plots), Smaragdis's algorithm (the middle two plots), and our proposed algorithm (the right two plots)

[2]In order for the object to be separated, P should be big enough to cover the length of the object in the audio signal.

audio objects with repeating patterns are successfully separated by both our algorithm and Smaragdis's algorithm, with $\mathbf{W}^o(p)$ being the time-frequency representation of the repeating patterns, and $\mathbf{H}^o$ containing the temporal structure of these patterns, i.e., the happening time of individual patterns. The standard NMF described by the learning rules (6) and (7), however, totally fails for separating the audio objects in these tests. We have extensively tested the algorithm for different set-ups of the parameters, including other randomly initialized matrices $\mathbf{W}$ and $\mathbf{H}$, and found such similar separation performance. To show the computational efficiency of our proposed algorithm, we have performed another experiment. We keep the set-up of the parameters in the above experiments unchanged, and we run both our and Smaragdis's algorithms 300 iterations. We plot the computing time against the number of iterations in Figure 4, where a variant algorithm of Smaragdis, i.e., using equation (16) to calculated $\hat{\mathbf{X}}$ (other steps remain unchanged), is also tested. Clearly, our algorithm consumes considerably less computing time than Smaragdis's algorithm. If a larger P is required (for extracting larger objects), the computational efficiency of our proposed algorithm can be more significant.

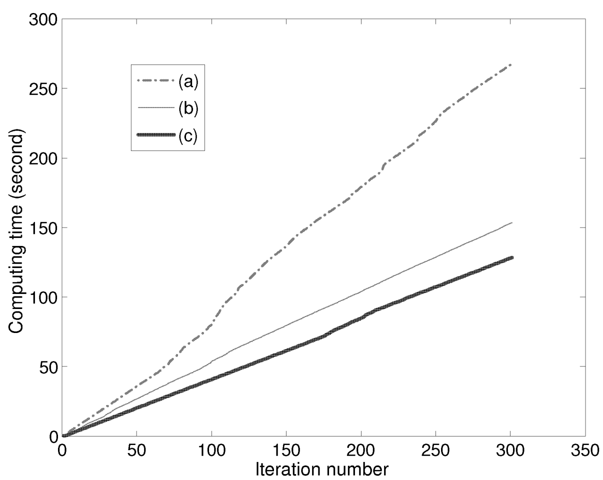

Fig. 4. Comparison of the required computing time changing with the number of iterations for (a) Smaragdis's algorithm, (b) a variant of Smaragdis's algorithm in which the calculation of $\hat{\mathbf{X}}$ is replaced by equation (17), (c) the proposed algorithm.

5.2. Real Music Audio

Two music audio signals with each containing repeating musical notes G4 and A3 played by a guitar are mixed together. The mixed signal is approximately 6.8s sampled at $f_s = 22050$Hz. $P = 105$. Other parameters and the method for generating $\mathbf{X}$ are the same as those in the above experiments. Spectrogram matrix $\mathbf{X}$ is visualized in Figure 5. $\mathbf{W}(p)$ and $\mathbf{H}$ in both algorithms are initialized randomly.

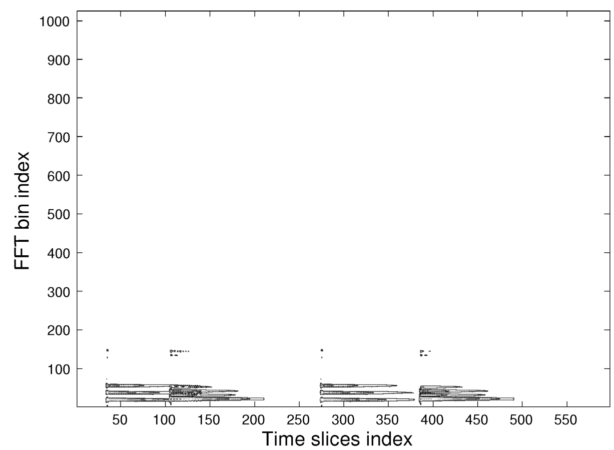

Fig. 5. The contour plot of the magnitude spectrum matrix $\mathbf{X}$ of the real music audio signal.

As an example, the resulted $\mathbf{H}^o$ and $\mathbf{W}^o(p)$ by applying the proposed learning algorithm are plotted in Figure 6 and 7 respectively.

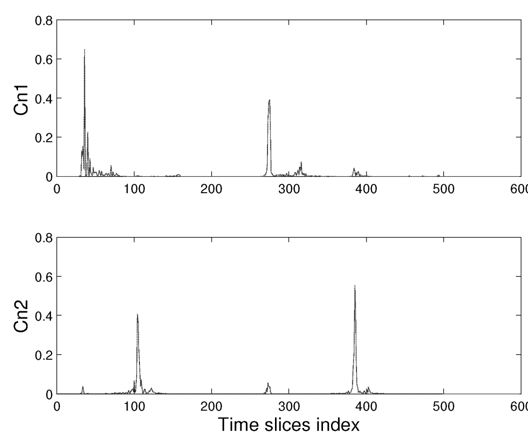

Fig. 6. Visualization of the factorized $\mathbf{H}^o$.

To evaluate the performance more accurately, we use two performance indices. One is the rejection ratio (RR). If we denote $\hat{\mathbf{X}} = \sum_{i=1}^{R} \hat{\mathbf{X}}(i)$, i.e., splitting $\hat{\mathbf{X}}$ into R factorized

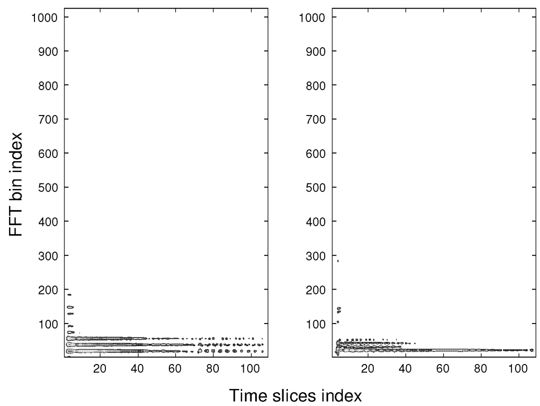

Fig. 7. Visualization of the factorized $\mathbf{W}^o(p)$, $p = 0, ..., 104$. The left plot represents note G4, and the right denotes note A3.

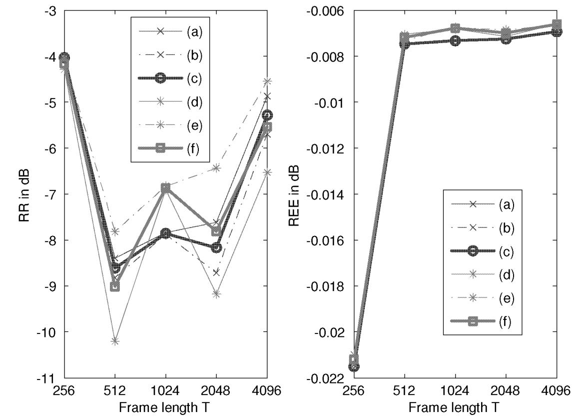

Fig. 8. The comparison of the performance indices (i.e., RR and REE) varying with T between the two algorithms. (a), (b), and (c) are the results of the two random tests and their average respectively by applying our algorithm. (d), (e), and (f) are the corresponding results obtained by applying Smaragdis's algorithm.

components, we can define RR as

$$RR(\text{dB}) = 10\log_{10}\left[\sum_{\forall j \neq i} cor(\hat{\mathbf{X}}(i), \hat{\mathbf{X}}(j))\right] \quad (21)$$

where cor denotes the correlation. This index can measure how accurate the separation performance is, and a lower value represents a better performance. The other index is the relative estimation error (REE) defined as

$$REE(\text{dB}) = 10\log_{10}\left(\left\|\mathbf{X} - \hat{\mathbf{X}}\right\|_F / \|\mathbf{X}\|_F\right) \quad (22)$$

which measures the accuracy of the factorization, a lower value representing a better performance.

We run both the proposed algorithm and Smaragdis's algorithm twice for each T, where T is set to be 256, 512, 1024, 2048, and 4096 respectively. The results of these two tests, together with their average are shown in Figure 8. From the plot of REE, we see that the proposed algorithm is consistently better than Smaragdis's algorithm. From the plot of RR, although the improvement is trivial, the proposed algorithm seems less sensitive to different random initializations. These results indicate that the proposed algorithm is highly competitive to Smaragdis's algorithm.

6. CONCLUSIONS

A new multiplicative learning algorithm for convolutive NMF has been presented. The algorithm is featured with the novel learning rules derived from the squared Euclidean distance, together with an efficient method for computing the estimate of the low-rank approximation. The proposed algorithm has advantages over both Smaragdis's algorithm and the standard NMF technique in the context of audio object and feature separation. The proposed algorithm can be a useful tool for a wide range of applications including the analysis of potentially more complex auditory scenes, which, together with the theoretical proof of its convergence, remains as our future interests.

7. REFERENCES

[1] P. Paatero, "Least squares formulation of robust non-negative factor analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 37, pp. 23-35, 1997.

[2] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788-791, 1999.

[3] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *Advances in Neural Information Processing 13* (in Proc. NIPS 2000), MIT Press, 2001.

[4] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, no. 5, pp. 1457-1469, 2004.

[5] A. Cichocki, R. Zdunek, and S. Amari, "Csiszar's divergences for non-negative matrix factorization: family of new algorithms," *Spinger Lecture Notes in Computer Science*, vol. 3889, pp. 32-39, 2006.

[6] P. Smaragdis, "Non-negative matrix factor deconvolution, extraction of multiple sound sources from monophonic inputs," in *Proc. 5th Int. Conf. on Independent Component Analysis and Blind Signal Separation*, Granada, Spain, Sept. 22-24, Lecture Notes on Computer Science (LNCS 3195), pp.494-499, 2004.

[7] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Trans. Audio Speech and Language Processing*, vol. 15, no. 1, pp. 1-12, 2007.

[8] W. Wang, Y. Luo, S. Sanei, and J. A. Chambers, "Non-negative matrix factorization for note onset detection of audio signals," in *Proc. IEEE Int. Workshop on Machine Learning for Signal Processing*, pp. 447-452, Maynooth, Ireland, Sept.6-8, 2006.