Sparseness Constraints on Nonnegative Tensor Decomposition

Na Li
nali@clarkson.edu

Department of Mathematics
Clarkson University
Potsdam, New York 13699, USA

Carmeliza Navasca
cnavasca@clarkson.edu

Department of Mathematics
Clarkson University
Potsdam, New York 13699, USA

November 24, 2010

Abstract

The Nonnegative Tensor Factorization (NTF) method, has been shown to separate the mixture of several sound sources reasonably well. Based on the sparsity of power spectrogram of signals, we propose to add sparseness constraints to one factor matrix, which contains frequency basis, to obtain a sparse representation of this nonnegative factor. Two methods are presented for the sparse and nonnegative factor matrix. The algorithms are tested to separate the instrument sources from a two channel mixtures containing a clarinet, a piano and a steel drum. Some comparison results of these two NTF-based methods are then presented.

1 Introduction

The purpose of this paper is to use a new numerical technique for tensor decomposition. Given a tensor $\mathcal{T} \in \mathbb{R}^{I\times J\times K}$, we find the decomposition of $\mathcal{T}$ into matrix factors $\mathbf{A} \in \mathbb{R}^{I\times R}$, $\mathbf{S} \in \mathbb{R}^{J\times R}$ and $\mathbf{G} \in \mathbb{R}^{K\times R}$, such that,

$$\mathcal{T} \approx \widehat{\mathcal{T}} = \sum_{r=1}^{R} \mathbf{A}_r \circ \mathbf{S}_r \circ \mathbf{G}_r$$

where $\widehat{\mathcal{T}}$ is an approximation to $\mathcal{T}$. Here $\mathbf{G_r}$, $\mathbf{A_r}$ and $\mathbf{S_r}$ are the rth column of the matrices $\mathbf{G}$, $\mathbf{A}$ and $\mathbf{S}$, respectively. We use the CANDECOMP/PARAFAC (CP) decomposition that is due to Harshman [11] and Carol and Chang [6]. In the CP decomposition, the matrices $\mathbf{A} \in \mathbb{R}^{I\times R}$, $\mathbf{S} \in \mathbb{R}^{J\times R}$, and $\mathbf{G} \in \mathbb{R}^{K\times R}$ are assumed to exist in which the original tensor $\mathcal{T}$ is defined as

$$\mathcal{T}_{ijk} = \sum_{r=1}^{R} \mathbf{A}_{ir}\mathbf{S}_{jr}\mathbf{G}_{kr}\,, \qquad (1)$$

for $(i, j, k) \in [1, I] \times [1, J] \times [1, K]$.

A specific application of CP in signal processing is in sound source separation. Sound source separation refers to the problem of synthesizing source signals given a mixture of those same source signals. For example, the original tensor $\mathcal{T} \in \mathbb{R}^{I\times J\times K}$ contains the power spectrogram of the multi-channel mixed signals [10]. The unknown factors are the following: the entries of matrix $\mathbf{G}$ are the gains of each independent source from R channels and each matrix $\mathbf{A_r} \circ \mathbf{S_r}^T$ is a power spectrogram of a source signal. Due to some properties of the power spectrogram of signals [10], the data has an inherent tensor structure and hence, tensor decomposition methods are applied. The Alternating Least-Squares (ALS) method introduced in [11, 6] is a well-known method for approximating the factors of the original tensor $\mathcal{T}$ described in 1. For sound source separation, it is typical to add a nonnegativity constraint on all three factors due to the nature of the data. Thus, each of the subproblem of the ALS is replaced by a nonnegative matrix factorization subproblem. This method is called the Nonnegative Tensor Factorization (NTF); see [16, 3] for the implementation. There are also various methods for sound source separation; see the references [7, 18, 10, 17, 8].

In this paper, a two-channel mixture of the source signals from a clarinet, a piano and a steel drum is considered. A tensor data consists of the spectrogram of the input channel signal in one channel. Therefore, in the case of two channels, we obtain a tensor of size $I \times J \times 2$. By using the ALS algorithm, we can obtain the factor $\mathbf{A}$, $\mathbf{S}$ and $\mathbf{G}$ with no constraints imposed on the factors. Thus, we propose two *hybrid* nonnegative tensor decompositions which add sparseness constraint to one of the nonnegative factors. Unlike the application of the NTF in [10], all three factors are subjected to nonnegativity constraints. In our application, the CP decomposition (1) is used while assuming that the matrix $\mathbf{A}$ is sparse since it represents the signals in the frequency basis.

The proposed methods are optimization based methods relying on NMF [15, 16] methods and sparsity methods, namely, compressed sensing for recovering sparse signals [5, 4] and projection-based technique to handle sparseness constraints [13].

2 Preliminaries

We denote the scalars in $\mathbb{R}$ with lower-case letters $(a, b, \ldots)$ and the vectors with bold lower-case letters $(\mathbf{a}, \mathbf{b}, \ldots)$. The matrices are written as bold upper-case letters $(\mathbf{A}, \mathbf{B}, \ldots)$ and the symbol for tensors are calligraphic letters $(\mathcal{A}, \mathcal{B}, \ldots)$. The subscripts represent the following scalars: $(\mathcal{A})_{ijk} = a_{ijk}$, $(\mathbf{A})_{ij} = a_{ij}$, $(\mathbf{a})_i = a_i$ and $\mathbf{A}_r$ is the rth column of $\mathbf{A}$. The superscripts indicate the length of the vector or the size of the matrices. For example, $\mathbf{b}^K$ is a vector with length K and $\mathbf{B}^{N\times K}$ is a $N\times K$ matrix. In addition, the lower-case superscripts on a matrix indicate the mode in which it has been matricized. For example, $\mathbf{T}_{(n)}$ is the mode-n matricization of the tensor $\mathcal{T} \in \mathbb{R}^{I\times J\times K}$ for $n = 1, 2, 3$.

Definition 2.1 *The Kronecker product of matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *is defined as*

$$\mathbf{A}\otimes\mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}.$$

Definition 2.2 *The Khatri-Rao product is the "matching columnwise" Kronecker product. Given matrices* $\mathbf{A} \in \mathbb{R}^{I\times K}$ *and* $\mathbf{B} \in \mathbb{R}^{J\times K}$*, their Khatri-Rao product is denoted by* $\mathbf{A}\odot\mathbf{B}$*. The result is a matrix of size* $(IJ \times K)$ *defined by*

$$\mathbf{A}\odot\mathbf{B} = [\mathbf{A}_1\otimes\mathbf{B}_1 \quad \mathbf{A}_2\otimes\mathbf{B}_2 \quad \ldots].$$

If $\mathbf{a}$ *and* $\mathbf{b}$ *are vectors, then the Khatri-Rao and Kronecker products are identical, i.e.,* $\mathbf{a}\otimes\mathbf{b} = \mathbf{a}\odot\mathbf{b}$.

Definition 2.3 (Mode-n matricization) *Matricization is the process of reordering the elements of an* N*th order tensor into a matrix. The mode-n matricization of a tensor* $\mathcal{T} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ *is denoted by* $\mathbf{T}_{(\mathbf{n})}$ *and arranges the mode-n fibers, the vectors obtained from fixing every index with the exception of the nth mode, as the columns of the resulting matrix.*

Definition 2.4 (Rank-one tensor) *An* N*th order tensor* $\mathcal{T} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ *is a rank-one if it can be written as the outer product of* N *vectors, i.e.,*

$$\mathcal{T} = \mathbf{a}^{(1)}\circ\mathbf{a}^{(2)}\circ\cdots\circ\mathbf{a}^{(N)},$$

where $\mathbf{a}^{(r)} \in \mathbb{R}^{I_r\times 1}, 1 \le r \le N$*. The symbol "$\circ$" represents the vector outer product. This means that each element of the tensor is the product of the corresponding vector elements:*

$$t_{i_1 i_2\cdots i_N} = a^{(1)}_{i_1} a^{(2)}_{i_2}\cdots a^{(N)}_{i_N}, \quad \textit{for all } 1 \le i_n \le I_n.$$

3 Optimization Methods

The CP decomposition is the simplest factorization for representing a given tensor $\mathcal{T} \in \mathbb{R}^{I\times J\times K}$ into sums of rank-one tensors; i.e. solve for three factor matrices $\mathbf{A} \in \mathbb{R}^{I\times R}$, $\mathbf{S}^{J\times R}$ and $\mathbf{G}^{K\times R}$ for a fixed R in the optimization problem:

$$\min_{\mathbf{A},\mathbf{S},\mathbf{G}} \quad \|\mathcal{T} - \sum_{r=1}^{R} \mathbf{A}_r\circ\mathbf{S}_r\circ\mathbf{G}_r\|_F^2 \tag{2}$$

Through tensor matricization via the Khatri-Rao products, the problem (2) is reformulated into subproblems:

$$\begin{aligned}
\mathbf{A}^{k+1} &= \operatorname*{argmin}_{\widehat{\mathbf{A}}\in\mathbb{R}^{I\times R}} \|\mathbf{T_{(1)}}^{I\times JK} - \widehat{\mathbf{A}}(\mathbf{G}^k\odot\mathbf{S}^k)^{\mathrm{T}}\|_F^2,\\
\mathbf{S}^{k+1} &= \operatorname*{argmin}_{\widehat{\mathbf{S}}\in\mathbb{R}^{J\times R}} \|\mathbf{T_{(2)}}^{J\times IK} - \widehat{\mathbf{S}}(\mathbf{G}^k\odot\mathbf{A}^{k+1})^{\mathrm{T}}\|_F^2,\\
\mathbf{G}^{k+1} &= \operatorname*{argmin}_{\widehat{\mathbf{G}}\in\mathbb{R}^{K\times R}} \|\mathbf{T_{(3)}}^{K\times IJ} - \widehat{\mathbf{G}}(\mathbf{S}^{k+1}\odot\mathbf{A}^{k+1})^{\mathrm{T}}\|_F^2.
\end{aligned} \tag{3}$$

This is the Alternating Least-Squares method. This iterative technique requires initial guesses $\mathbf{A}^0$, $\mathbf{S}^0$ and $\mathbf{G}^0$.

For sound separation applications, we require additional constraints on each of the subproblems in the ALS algorithm: $\mathbf{S}$ and $\mathbf{G}$ are nonnegative matrices while $\mathbf{A}$ is a sparse nonnegative matrix. To obtain nonnegative matrices $\mathbf{S}$ and $\mathbf{G}$, the last two subproblems in (3) is replaced by the optimization problem of finding nonnegative factors $\mathbf{W}$ and $\mathbf{H}$ from a nonnegative matrix $\mathbf{V}$:

$$\text{minimize}\|\mathbf{V} - \mathbf{W}\mathbf{H}\|_F^2, \quad \text{subject to } \mathbf{W},\mathbf{H}\succeq 0 \tag{4}$$

where $\succeq$ denotes the nonnegative entries of $\mathbf{W}$ and $\mathbf{H}$. This is called Nonnegative Matrix Factorization (NMF). Several algorithms have been proposed for NMF, namely, a gradient-based method by Paatero [16] and a multiplicative updating algorithm by Lee and Seung [15].

3.1 NMF-Sparse vs ℓ_1-NMF

For factor $\mathbf{A}$, we have to deal with two constraints: sparsity and nonnegativity. We implement two methods differing in the order of how the constraints are imposed: NMF with sparse constraints [13] (NMF-Sparse) and ℓ_1-minimization with nonnegative constraints (ℓ_1-NMF).

3.1.1 ℓ_1-NMF

The first subproblem in (3) is reformulated by first vectorizing the equation $\mathbf{T_{(1)}}^{I\times JK} = \widehat{\mathbf{A}}(\mathbf{G}^k\odot\mathbf{S}^k)^{\mathrm{T}}$ into $\mathbf{t} = \mathbf{Q}\widehat{\mathbf{a}}$ via column stacking with $\mathbf{Q} = \mathbf{I}^{I\times I}\otimes(\mathbf{G}^k\odot\mathbf{S}^k) \in \mathbb{R}^{IJK\times IR}$, $\mathbf{t} \in \mathbb{R}^{IJK}$ and $\widehat{\mathbf{a}} \in \mathbb{R}^{IR}$. Then the least-squares subproblem is replaced by an ℓ_1-minimization with equality constraints [12]:

$$\min\|\widehat{\mathbf{a}}\|_{\ell_1} \quad \text{subject to } \mathbf{t} = \mathbf{Q}\widehat{\mathbf{a}}. \tag{5}$$

Note that if $\mathbf{x} \in \mathbb{R}^n$, then $\|\mathbf{x}\|_{\ell_1} = \sum_{i=1}^{n} |x_i|$. The idea behind this model (5) is to construct sparse and exact solution vector $\widehat{\mathbf{a}}$ which matricizes into a sparse factor matrix $\widehat{\mathbf{A}}$ satisfying the CP decomposition.

The ℓ_1 minimization problem can be recast as linear program [2]:

$$\min \mathbf{1}'\widehat{\mathbf{a}} \quad \text{subject to } \ \widehat{\mathbf{a}} \geq 0 \text{ and } \mathbf{Q}\widehat{\mathbf{a}} = \mathbf{t} \tag{6}$$

with both equality and inequality constraints.

The link between ℓ_1 minimization and linear programs has been known since the 1950's in the paper of [14]. Moreover, numerical techniques for solving linear programs have been well studied. In our codes, we implement Matlab's linear programming algorithm based on the simplex and interior-point methods. To impose the nonnegative constraint on $\widehat{\mathbf{A}}$, the approximation from the linear program is further refined by a nonnegative least-squares method.

3.1.2 NMF-Sparse

In [13], based on the method proposed in [16] and [15], Hoyer added a sparseness constraint on the NMF algorithm. Below we summarize the algorithm found in [13]: for the nonnegative matrix $\mathbf{V} = {\mathbf{T}_{(1)}}^{k-1}$

1. Initialize $\mathbf{W} = \mathbf{A}^{k-1}$ calculated in a previous ALS iteration and set $\mathbf{H} = (\mathbf{G}^{k-1} \odot \mathbf{S}^{k-1})^T$
2. $\mathbf{W} \leftarrow \mathbf{W} - \mu_{\mathbf{W}}(\mathbf{WH} - \mathbf{V})\mathbf{H}^T$
3. Project each column of $\mathbf{W}$ to a vector that is nonnegative with same ℓ_2 norm, but the ℓ_1 norm is set to achieve the desired sparseness

where $\mu_{\mathbf{W}} = \frac{\mathbf{W}_{ia}}{(\mathbf{WHH}^T)_{ia}}$ is small positive multiplicative step from [15]. In [13], Hoyer provided the projection operator algorithm find the closest (in the Euclidean sense) nonnegative vector $\mathbf{s}$ for any vector $\mathbf{x}$ constrained to a given the ℓ_1 norm and a given ℓ_2 norm. The desired sparsity of the factors, S_ω and S_h, are defined as

$$\text{sparseness}(\mathbf{W}_i) = S_\omega, \quad \text{sparseness}(\mathbf{H}_i) = S_h,$$

where

$$\text{sparseness}(\mathbf{x}) = \frac{\sqrt{n} - \frac{\|\mathbf{x}\|_{\ell_1}}{\|\mathbf{x}\|_{\ell_2}}}{\sqrt{n} - 1}.$$

for a vector $\mathbf{x}$.

4 Numerical Experiments

A two-channel mixture of the sound source signals from a clarinet, a piano and a steel drum is the time-domain input signal (see Figure 1). Each channel is a 91136 time sampled signal in 2 seconds and is divided into 89 frames, the power spectrum is calculated within each frame using the discrete Fourier transform (DFT) for each channel (see Figure 2a). Then the two spectrograms are stacked into a tensor format (see Figure 2b). The two-channel mixture of the signals is tensor $\mathcal{T}$ of size $1024 \times 89 \times 2$.

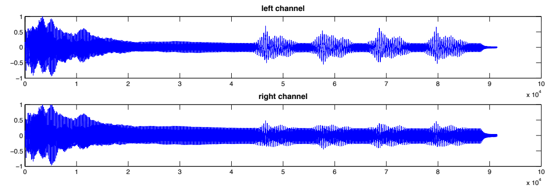
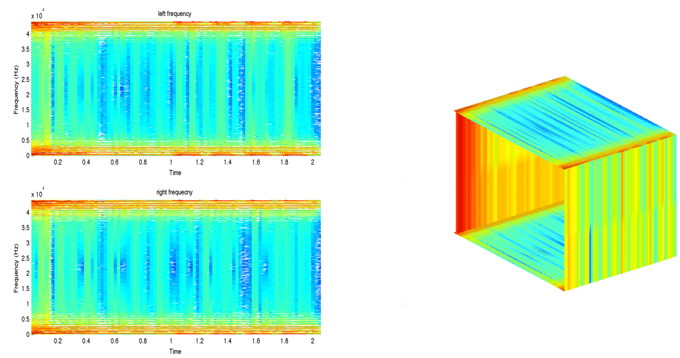


Figure 1: The two-channel sound signal mixtures

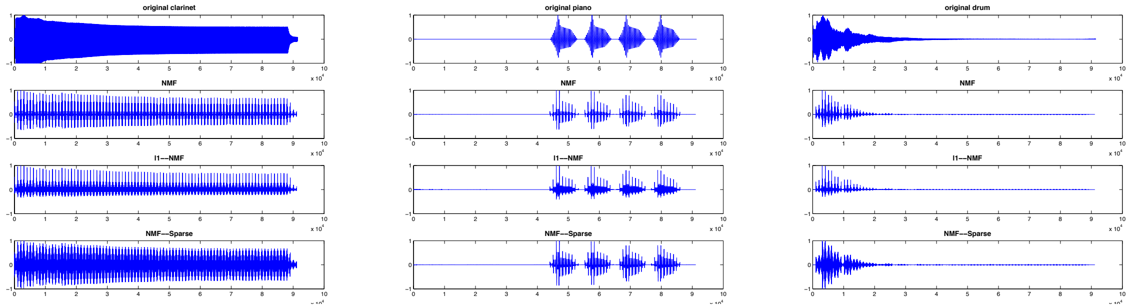


(a) The frequency of the two channels (b) Tensor

Figure 2: Tensor structure: the left frequency is the top slice and the right frequency is the bottom slice, obtaining a rotated tensor $\mathcal{T} \in \mathbb{R}^{1024 \times 89 \times 2}$

According to the previous discussion in Section 3.1.1 and 3.1.2, below we describe the algorithm allowing different constraints on factor $\mathbf{A}$. The algorithm is the following:

1. CP decomposition is applied to $\mathcal{T}$ via ALS (3) to obtain $\mathbf{A}$, $\mathbf{S}$ and $\mathbf{G}$. Choose one from the following three methods for factor $\mathbf{A}$ and apply the nonnegative matrix factorization method for factors $\mathbf{S}$ and $\mathbf{G}$.
   - (NMF)–For nonnegative $\mathbf{A}$, solve the first subproblem of (3) by NMF.
   - (ℓ_1-NMF)–For nonnegative sparse matrix $\mathbf{A}$, use ℓ_1 minimization with nonnegative constraints (5) as the method for the first least-squares subproblem.
   - (NMF-Sparse)–For nonnegative sparse $\mathbf{A}$, use NMF with sparseness constraints as a method by using the algorithm in Section (3.1.2).
   - Nonnegative factor matrices $\mathbf{S}$ and $\mathbf{G}$ are solved by NMF.



(a) Separated waveforms for clarinet (b) Separated waveforms for piano (c) Separated waveforms for steel drum

Figure 3: Waveforms of original signal, NMF, ℓ_1 nonnegative minimization and NMF with sparseness

2. For each $1 \leq r \leq R$, construct the source spectrogram, $\mathbf{F}_r = \mathbf{A}(:, r)\mathbf{S}(:, r)^T$.

3. The matrix $\mathbf{G}$ gives the ratios of each signal of the instrument in two channels. let a vector of size R, $H = \log \frac{\mathbf{G}(1,:).}{\mathbf{G}(2,:)}$. Using k-means cluster method [9], H is divided into 3 clusters, where each is from an instrument. So, H_r corresponds to $\mathbf{F}_r$ which should be in the same cluster.

4. Apply phase information [1] for the spectrogram obtained in 2 where the corresponding source signal up to the clusters is dominant to $\mathbf{F}_r$. Invert the spectrogram to obtain the time domain waveforms. See Figures 3a–3c.

Figure 1 shows the two channel sound signal mixture input formed into a tensor. We compared these methods (NMF-Sparse, ℓ_1-NMF) and NMF to the original waveforms in terms of the time domain waveform plots and frequency plots. Figures 3a, 3b and 3c show the comparison results, the waveform (top) in each figure is the original sound signal waveform. The rest of the waveforms (second from the top to bottom) in Figures 3a, 3b and 3c are NMF, ℓ_1-NMF and NMF-Sparse, respectively. It can be seen that all three methods can capture the main characteristics of the sources.

We measure the distance between two signals w and v in a least squared error sense [1] by the following equation:

$$E = \sum_{i=1}^{N} (|w_i| - |v_i|)^2 \tag{7}$$

where N is the length of the signals. In Figure 4, set 1 is the comparison of the separated clarinet signals, set 2 and set 3 are the separated piano and steel drum signals, respectively. It shows that ℓ_1-NMF (the green bar) is better than the other two methods in all three sets.

Although both of ℓ_1-NMF and NMF-Sparse obtain a sparse and nonnegative factor matrix $\mathbf{A}$, we have seen that ℓ_1-NMF is better in the least squared error sense.

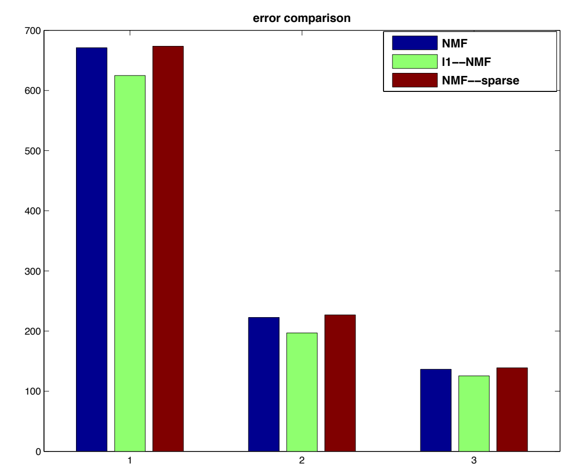


Figure 4: Error comparison of the different methods

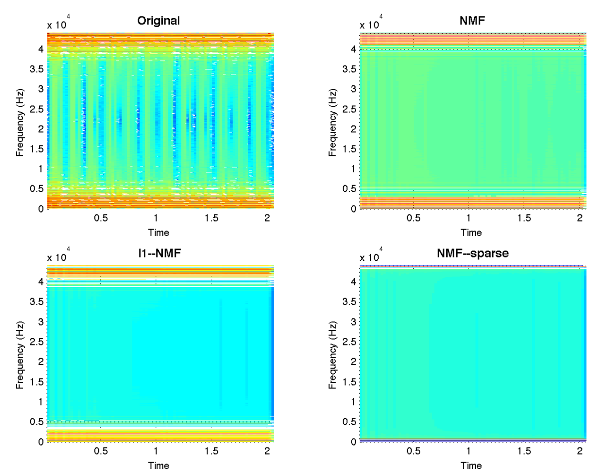


Figure 5: Spectrograms of the clarinet for the varying methods, the top left one is the original spectrogram

Let us look at the clarinet spectrogram plots of the separated and original signals. Figure 5 shows that the

NMF-Sparse method does not capture the spectral density at frequency between 0 to 0.5×10^4 and over 4×10^4 while the higher densities appear in the spectrograms of the NMF and ℓ_1-NMF methods.

5 Conclusion

Two methods for hybrid tensor nonnegative decomposition are presented in the application of sound source separation. The two methods, the ℓ_1-NMF and NMF-Sparse, are proposed to implement the hybrid tensor decomposition to obtain a sparse and nonnegative factor due to the sparsity of power spectral. The numerical examples show the effectiveness of these techniques. Moreover, the error comparison plots show the ℓ_1 nonnegative minimization (ℓ_1-NMF) is better than NMF. The results of ℓ_1-NMF shows that it performs well with respect to the error and the resulting spectrogram plots of the separated signals which coincide with the actual sound. The NMF-Sparse method is the worst according to the error comparison. More study is needed to compare the two methods. Furthermore, we plan to develop more efficient techniques for sound source separation with better clustering methods.

6 Acknowledgements

C.N. and N.L. are both in part supported by the U.S. National Science Foundation DMS-0915100. C.N. and N.L. are very grateful to Jason Holloway for his assistance. N.L. would like to thank Prof. Dr.-Ing. Gerald Schuller for useful conversations at the 44th IEEE Asilomar Conference on Signals, Systems and Computers.

References

[1] D. Barry, R. Lawlor and E. Coyle, "Comparison of Signal Reconstruction Methods for the Azimuth Discrimination and Resynthesis Algorithm", 118^{th} *AES Convention Barcelona*, Spain, 2005.

[2] S. Boyd and L. Vandenberghe. "Convex Optimization," *Cambridge University Press*, Cambridge, 2004.

[3] R. Bro and S. De Jong, "A fast non-negativity-constrained least squares algorithm," *Journal of Chemometrics,* vol. 11, pp. 393-401, 1997.

[4] E. Candès, J. Romberg and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications of Pure and Applied Mathematics*, vol. 59, pp. 1207-1223, 2005.

[5] E. Candès and T. Tao, "Decoding by linear programming," *IEEE Trans. Inform. Theory*, vol. 52, no.12, pp. 4203-4215, 2005.

[6] J. Carrol and J. Chang, "Analysis of Individual Differences in Multidimensional Scaling via an N-way Generalization of "Eckart-Young" Decomposition" *Psychometrika*, vol. 9, pp. 267-283, 1970.

[7] M. A. Casey and A. Westner, "Separation of Mixed Audio Sources By Independent Subspace Analysis", *Proc. Of ICMC 2000*, pp. 154-161, Berlin, Germany.

[8] L. De Lathauwer, B. De Moor and J. Vandewalle, "An introduction to independent component analysis", *J. Chemometrics*, vol. 14, pp. 123-149, 2000.

[9] L. Eldén, "Matrix Methods in Data Mining and Pattern Recoqnition", *Fundamentals of Algorithms 4*, SIAM, 2007.

[10] D. Fitzgerald, M. Cranitch and E. Coyle, *Non-Negative Tensor Factorization for Sound Source Separation*, Dublin, Sept. 1-2, ISSC 2005.

[11] R. A. Harshman, "Foundations of the PARAFAC procedure: Model and Conditions for an "Explanatory" Mutli-code Factor Analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1-84, 1970.

[12] J.R. Holloway and C. Navasca. "Recovering tensor data from incomplete measurement via compressive sampling." *Proc. IEEE Asilomar Conference on Signals, Controls and Systems 2009* Pacific Grove, California, November, 2009.

[13] P. O. Hoyer, "Non-negative Matrix Factorization with Sparseness Constraints," *Journal of Machine Learning Research*,5, 2009, 1457-1469.

[14] A. Karmarkar, "A new polynomial-time algorithm for linear programming," *Combinatorica*, vol. 4, 1984.

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

[16] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics,* vol. 5 , pp. 111-126, 1994.

[17] P. Smaragdis and J.C. Brown, "Non-negative Matrix Factorization for Polyphonic Music Transcription", *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 177-180, October 2003.

[18] T. Virtanen, "Sound Source Separation Using Sparse Coding with Temporal Continuity Objective", *Proc. of International Computer Music Conference (ICMC2003)*, Singapore, 2003.