SOUND SOURCE SEPARATION USING SHIFTED NON-NEGATIVE TENSOR FACTORISATION

Derry FitzGerald[*]*, Matt Cranitch*

Dept. of Electronic Engineering
Cork Institute of Technology
Rossa Avenue, Bishopstown, Cork, Ireland
derry.fitzgerald@cit.ie

Eugene Coyle

Dept. of Control Engineering
Dublin Institute of Technology
Kevin Street, Dublin, Ireland
eugene.coyle@dit.ie

ABSTRACT

Recently, shifted Non-negative Matrix Factorisation was developed as a means of separating harmonic instruments from single channel mixtures. However, in many cases two or more channels are available, in which case it would be advantageous to have a multichannel version of the algorithm. To this end, a shifted Non-negative Tensor Factorisation algorithm is derived, which extends shifted Non-negative Matrix Factorisation to the multi-channel case. The use of this algorithm for multi-channel sound source separation of harmonic instruments is demonstrated. Further, it is shown that the algorithm can be used to perform Non-negative Tensor Deconvolution, a multi-channel version of Non-negative Matrix Deconvolution, to separate sound sources which have time evolving spectra from multi-channel signals.

1. INTRODUCTION

In recent years, matrix factorisation techniques have been used as means of attempting single channel sound source separation. These methods include the use of Independent Subspace Analysis (ISA), Sparse Coding (SC), and Non megative Matrix Factorisation (NMF) [1, 2, 3]. These techniques attempt to factorise a magnitude spectrogram $\mathbf{X}$ into matrix factors $\mathbf{A}$ and $\mathbf{S}$ such that $\mathbf{X} \approx \mathbf{AS}$, where $\mathbf{X}$ is an n x m spectrogram, where n is the number of frequency bins, and m is the number of frames in the spectrogram, $\mathbf{A}$ is an n x r matrix, and $\mathbf{S}$ is an r x m matrix, with r smaller than n or m, where r is the chosen rank of the decomposition. The columns of $\mathbf{A}$ contain frequency basis functions, while the associated rows of $\mathbf{S}$ contain corresponding amplitude envelopes for the frequency basis functions. Individual elements of $\mathbf{A}$ and $\mathbf{S}$ can then be used to attempt resynthesis of individual components or sources in the input data.

All of the above methods have a shortcoming in that it cannot be assumed that natural sounds will have fixed spectra over time. In an attempt to overcome this problem, convolutive forms of SC and NMF have been proposed [4, 5]. Both of these methods attempt to describe a source as a sequence of successive spectra and a corresponding amplitude envelope across time.

Another shortcoming in the standard factorisation techniques such as SC and NMF is that a single basis function is typically required to represent each note of a given instrument. This means that for source separation of instruments playing melodies, some method of clustering the basis functions together is required. Methods for clustering basis function have been proposed by Casey and Virtanen [1, 2]. However, it is difficult to obtain a correct clustering in many situations for reasons discussed in [6].

As a result of this, shifted Non-Negative Matrix Factorisation has recently been proposed to deal with the situation where different notes from the same instrument occur over the course of a spectrogram [7]. Shifted Non-Negative Matrix Factorisation assumes that the notes belonging to a single source consist of translated versions of a single frequency basis function which represents the typical frequency spectrum of any note played on the instrument in question. Using this assumption requires that the chosen time-frequency representation has logarithmic frequency resolution, such as the Constant Q transform [8]. If the center frequencies in the representation are set so that $f_i = f_{i-1}2^{1/12}$, where f_i is the center frequency of band i, then the spacing between center frequencies will match that of the even-tempered tuning system [9]. As a result, translating a frequency basis function of a note up by one bin is equivalent to a pitch change of one semitone.

For the remainder of this paper the following conventions are used. Indexing of elements within a matrix or tensor, usually denoted by $\mathbf{X}_{i,j}$ is instead notated as $\mathbf{X}(i,j)$. Tensors are denoted by calligraphic uppercase letters, such as $\mathcal{T}$. Contracted product multiplication of two tensors is defined as follows. If $\mathcal{W}$ is a tensor of size $I_1 \times \cdots \times I_N \times J_1 \times \cdots \times J_M$ and $\mathcal{Y}$ is a tensor of size $I_1 \times \cdots \times I_N \times K_1 \times \cdots \times K_P$ then contracted product multiplication of the two tensors along the

[*]This work was funded by the Irish Research Council for Science, Engineering and Technology

first N modes is given by:

$$\langle \mathcal{W}\mathcal{Y} \rangle_{\{1:N,1:N\}}(j_1,\ldots,j_m,k_1,\ldots,k_p) = \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} \mathcal{W}(i_1,\ldots,i_N,j_1,\ldots,j_M) \mathcal{Y}(i_1,\ldots,i_N,k_1,\ldots,k_P)$$

Using this notation, the modes to be multiplied are specified in the subscripts that follow the angle brackets, in line with the conventions adapted by Bader and Kolda in [10]. Outer product multiplication is denoted by $\circ$, $./$ denotes elementwise division and $.*$ denotes elementwise multiplication.

Translation is carried out by means of a translation matrix. For an n x 1 vector, an n x n translation matrix can be used. This translation matrix can be obtained by permuting the columns of the identity matrix. To achieve a shift up by one position, the required translation matrix can be obtained from $\mathbf{I}(:,[n,1:n-1])$ where $\mathbf{I}$ denotes the identity matrix, and where the ordering of the columns is defined in the square brackets. For k possible translations, the k translation matrices are grouped into a translation tensor $\mathcal{T}$ of size n x k x n, where $\mathcal{T}(:,k,:)$ contains the kth translation matrix.

For r sources, the frequency basis functions representing each source can be grouped into an n x r tensor, denoted $\mathcal{A}$. Following from this, a spectrogram $\mathbf{X}$ can be decomposed as:

$$\mathbf{X} \approx \hat{\mathbf{X}} = \langle \langle \mathcal{T}\mathcal{A} \rangle_{\{3,1\}} \mathcal{S} \rangle_{\{2:3,1:2\}}$$

where $\mathcal{S}$ is a size k x r x m tensor containing the time envelopes associated with each translation of each source.

For the decomposition described above, the cost function proposed by Lee and Seung [11], called the generalised Kullback-Liebler divergence, was used:

$$D(\mathbf{X}\|\hat{\mathbf{X}}) = \sum_{ij} \left(\mathbf{X}(i,j) log \frac{\mathbf{X}(i,j)}{\hat{\mathbf{X}}(i,j)} - \mathbf{X}(i,j) + \hat{\mathbf{X}}(i,j) \right)$$

From this, update equations for $\mathcal{A}$ and $\mathcal{S}$ were derived, and the algorithm was shown to be capable of separating mixtures of harmonic instruments from single channel recordings.

2. SHIFTED NON-NEGATIVE TENSOR FACTORISATION

As noted above, matrix factorisation techniques have been used for sound source separation of single channel mixtures. However, most musical recordings from the past 40 years are two channel recordings created from linear mixtures of individual instrument recordings. Therefore, for any given instrument, the only difference between the channels lies in the intensity of the instrument. As a result, the same instrument basis function could be used to describe a given instrument in either channel, the only difference being the gain of the basis function in each channel. It is proposed to learn a single set of instrument basis functions which can be used to describe both channels of the input signal, a corresponding set of amplitude basis functions, and a set of corresponding gains which decide how loud a given set of instrument and amplitude basis functions are in each channel, thus takes advantage of the fact that certain instruments will be louder in some channels over others. Though aimed at two channel recordings, the derivation given below can deal with any number of linearly instantaneously mixed channels.

The signal model can then be described as:

$$\mathcal{X} \approx \hat{\mathcal{X}} = \sum_{t=1}^{r} \langle \langle (\mathcal{T} \circ \mathcal{G}(:,t)) \mathcal{A}(:,t) \rangle_{\{3,1\}} \mathcal{S}(:,t,:) \rangle_{\{2,1\}}$$

where $\mathcal{X}$ is a tensor of size n x l x m, containing the spectrograms of the l channels, and $\hat{\mathcal{X}}$ is an approximation to $\mathcal{X}$. $\mathcal{T}$ is a translation tensor as described previously, $\mathcal{G}$ is a tensor of size l x r, containing the gains for each instrument in each channel, and $\mathcal{A}$ and $\mathcal{S}$ are as before.

The generalised Kullback-Liebler divergence, extended to 3 dimensions, is again used as a cost function. Eliminating terms in $\mathcal{X}$ which are constant, and taking the derivative with respect to $\mathcal{G}$ yields:

$$\mathcal{G}(:,t) = \mathcal{G}(:,t) + \lambda \{ \langle \mathcal{P}\mathcal{S}(:,t,:) \rangle_{\{[1,3],[1,3]\}} - \langle \mathcal{B}\mathcal{S}(:,t,:) \rangle_{\{[1,3],[1,3]\}} \}$$

where

$$\mathcal{D} = \mathcal{X}./\hat{\mathcal{X}}$$
$$\mathcal{P} = \langle \langle \mathcal{T}\mathcal{D} \rangle_{\{1,1\}} \mathcal{A}(:,t) \rangle_{\{2,1\}}$$
$$\mathcal{B} = \langle \langle \mathcal{T}\mathcal{O} \rangle_{\{1,1\}} \mathcal{A}(:,t) \rangle_{\{2,1\}}$$

and where $\mathcal{O}$ is an all ones tensor of size equal to $\mathcal{D}$. The update equation for $\mathcal{G}$ can be converted to a multiplicative update rule by setting $\lambda = \mathcal{G}./\langle \mathcal{B}\mathcal{S}(:,t,:) \rangle_{\{[1,3],[1,3]\}}$, yielding:

$$\mathcal{G}(:,t) = \mathcal{G}(:,t).* \left[\langle \mathcal{P}\mathcal{S}(:,t,:) \rangle_{\{[1,3],[1,3]\}} ./ \langle \mathcal{B}\mathcal{S}(:,t,:) \rangle_{\{[1,3],[1,3]\}} \right]$$

Update equations can be derived for $\mathcal{A}$ and $\mathcal{S}$ in a similar manner. The update equation for $\mathcal{A}$ is given by:

$$\mathcal{A}(:,t) = \mathcal{A}(:,t).* \left[\langle \mathcal{Q}(:,:,t,:) \mathcal{S}(:,t,:) \rangle_{\{[1,4],[1,3]\}} ./ \langle \mathcal{C}(:,:,t,:) \mathcal{S}(:,t,:) \rangle_{\{[1,4],[1,3]\}} \right]$$

where

$$\mathcal{Q} = \langle (\mathcal{T} \circ \mathcal{G}) \mathcal{D} \rangle_{\{[1,4],[1,2]\}}$$
$$\mathcal{C} = \langle (\mathcal{T} \circ \mathcal{G}) \mathcal{O} \rangle_{\{[1,4],[1,2]\}}$$

The update equation for $\mathcal{S}$ is given by:

$$\mathcal{S} = \mathcal{S}.* \left[\langle \mathcal{R}\mathcal{D} \rangle_{\{[1,3],[1,2]\}} ./ \langle \mathcal{R}\mathcal{O} \rangle_{\{[1,3],[1,2]\}} \right]$$

where

$$\mathcal{R}(:,:,:,t) = \langle (\mathcal{T} \circ \mathcal{G}(:,t)) \mathcal{A}(:,t) \rangle_{\{3,1\}}$$

Once G, A, and S are randomly initialised to positive values, the use of multiplicative updates ensures that the factorisation will be non-negative. Although the convergence proofs used for NMF (see [11]) no longer apply, in practice it has been observed that the algorithm converges reliably.

The algorithm was implemented in Matlab using the Matlab Tensor Classes available from [12]. Fig. 1 shows the original waveforms of flute,viola and piano respectively, while Fig 2 shows a two-channel mixture of these sources. Fig 3 shows the separated waveforms obtained with the number of instruments set to 3 and the number of translations set to 7. It can be seen that the sources have been separated well, and on listening, the individual source melodies are clear with some small interference from the other instruments. The timbres of the instruments have been captured reasonably well, though the attack of the piano notes is missing. This demonstrates that the algorithm is capable of separating harmonic instruments from multichannel mixtures. However, the algorithm is sensitive to the chosen number of translations, which is analogous to the rank of the decomposition.

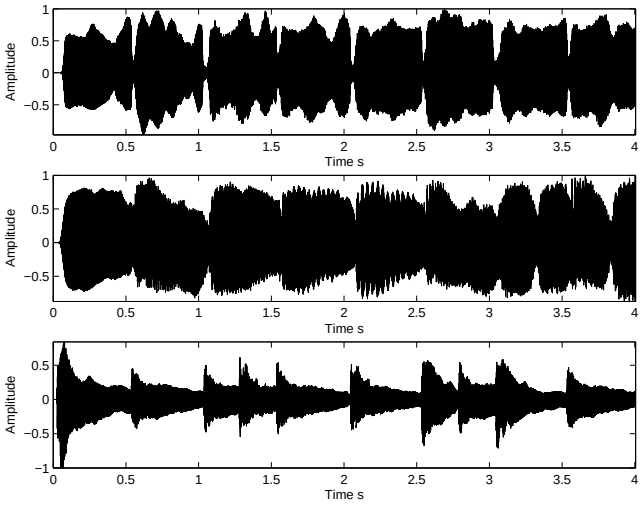

Fig. 1. *Original waveforms of flute, viola and piano*

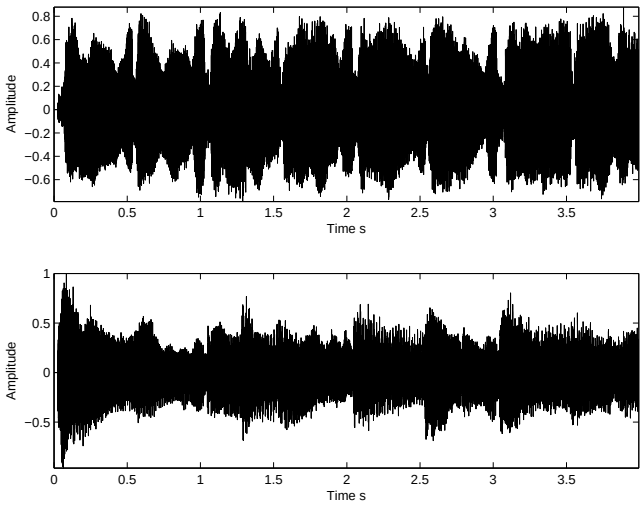

Fig. 2. *Two-channel mixture of flute, viola and piano*

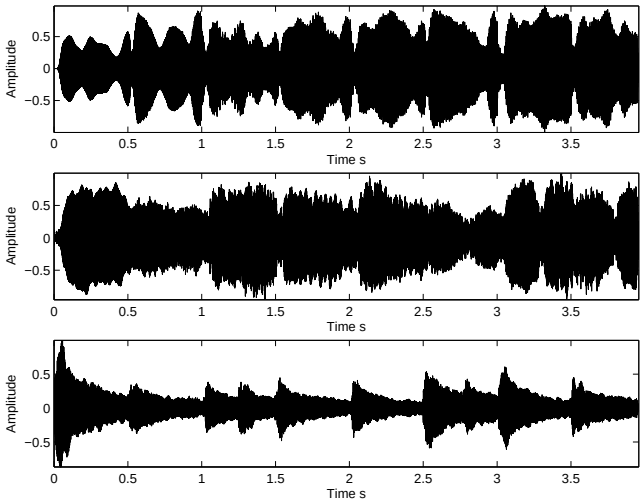

Fig. 3. *Separated waveforms of flute, viola and piano*

3. NON-NEGATIVE TENSOR DECONVOLUTION

The shifted Non-negative Tensor Factorisation algorithm can also be used to perform a multichannel version of convolutive NMF. This can be achieved by presenting $\mathcal{X}$ to the algorithm in a different way. The dimensions of $\mathcal{X}$ are rearranged so that instead of an n x l x m tensor being presented, a tensor of size m x l x n is input, or in other words the frequency and time dimensions are swapped around. As a result, $\mathcal{A}$ now recovers source amplitude envelopes which are shifted across time, while $\mathcal{S}$ recovers a sequence of successive spectra corresponding to the shifted envelopes. In a similar way, shifted Non-negative Matrix Factorisation can also be used to perform Non-negative Matrix Deconvolution. Therefore, the algorithm can be used to capture sound sources which have spectra which evolve with time, such as drums.

As an example of the use of the algorithm, Fig. 4 shows a two channel mixture of snare drum, bass drum and hi-hat. Magnitude spectrograms were obtained for each channel and combined into a tensor, which was re-arranged as described above. The number of sources was set to 3 and the number of shifts set to 10. Fig. 5 shows the recovered waveforms of the separated sources. It can be seen that the drum sounds have been successfully separated using the algorithm, though at the loss of part of the attack transients of the sources.

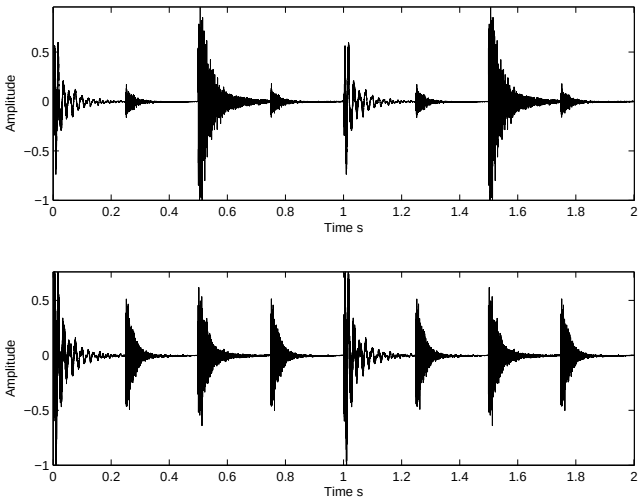

Fig. 4. *Mixture waveforms of snare, bass drum and hi-hat*

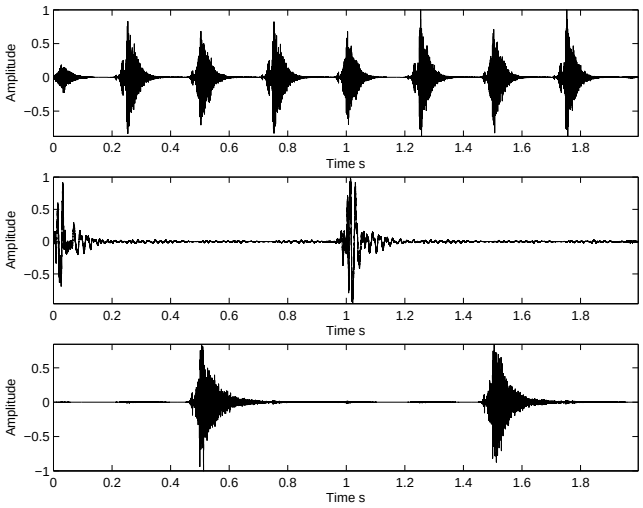

Fig. 5. *Separated waveforms of snare, bass drum and hi-hat*

4. CONCLUSIONS

An algorithm which performs shifted Non-negative Tensor Factorisation is presented, extending shifted Non-negative Matrix Factorisation to the multi-channel case. This is shown to be capable of separating harmonic instruments from multichannel recordings. The algorithm can also perform Non-negative Tensor Deconvolution, an extended version of Non-negative Matrix Deconvolution, which captures sound sources with time-varying spectra. However, there are problems with the method, namely separation degrades as the number of sources increases for a given number of channels. Also, the algorithm can be sensitive to the chosen number of translations. This remains an open issue in all matrix or tensor factorisation algorithms, and is an item for future research.

5. REFERENCES

[1] M. Casey and A. Westner, "Separation of mixed audio sources by independent subspace analysis," in *Proc. of International Computer Music Conference*, Berlin, Germany, Aug 2000, pp. 154–161.

[2] T. Virtanen, "Sound source separation using sparse coding with temporal continuity objective," in *Proc. of International Computer Music Conference*, Singapore, Oct 2003.

[3] P. Smaragdis and J. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2003, pp. 177–180.

[4] T. Virtanen, "Separation of sound sources by convolutive sparse coding," in *Proc. of ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing*, 2004.

[5] P. Smaragdis, "Discovering auditory objects through non-negativity constraints," in *Proc. of ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing*, 2004.

[6] D. FitzGerald, "Automatic drum transcription and source separation," Ph.D. dissertation, Conservatory of Music and Drama, Dublin Institute of Technology, Dublin, Ireland, 2004.

[7] D. FitzGerald, M. Cranitch, and E. Coyle, "Shifted non-negative matrix factorisation for sound source separation," in *IEEE Workshop on Statistical Signal Processing*, Bordeaux, France, 2005.

[8] J. Brown, "Calculation of a constant q spectral transform," *Journal of the Acoustic Society of America*, vol. 90, pp. 60–66, 1991.

[9] E. M. Burns, "Intervals, scales, and tuning," in *The Psychology of Music*, D. Deutsch, Ed. Academic Press, 1999.

[10] B. Bader and T. Kolda, "MATLAB tensor classes for fast algorithm prototyping, technical report SAND2004-5187, Sandia National Laboratories, Livermore, California," 2004.

[11] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*, T. Leen, T. Dietterich, and V. Tresp, Eds. MIT Press, 2001, vol. 13, pp. 556–562.

[12] T. Kolda and B. Bader, "Matlab tensor classes, SAND2004-5189," 2004. [Online]. Available: http://csmr.ca.sandia.gov/ tgkolda/