NONNEGATIVE TENSOR FACTORIZATION FOR SOURCE SEPARATION OF LOOPS IN AUDIO

Jordan B. L. Smith and Masataka Goto

National Institute of Advanced Industrial Science and Technology (AIST), Japan

ABSTRACT

The prevalence of exact repetition in loop-based music makes it an opportune target for source separation. Nonnegative factorization approaches have been used to model the repetition of looped content, and kernel additive modeling has leveraged periodicity within a piece to separate looped background elements. We propose a novel method of leveraging periodicity in a factorization model: we treat the two-dimensional spectrogram as a three-dimensional tensor, and use nonnegative tensor factorization to estimate the component spectral templates, rhythms and loop recurrences in a single step. Testing our method on synthesized loop-based examples, we find that our algorithm mostly exceeds the performance of competing methods, with a reduction in execution cost. We discuss limitations of the algorithm as we demonstrate its potential to analyze larger and more complex songs.

Index Terms— nonnegative tensor factorization, source separation, loop-based music, repetition

1. INTRODUCTION

Repetition is a defining aspect of music, and occurs at multiple timescales [1]. Many approaches to source separation take advantage of this: for example, nonnegative matrix factorization (NMF) can model a sustained note with a single spectral template and a time-varying activation function. Nonnegative matrix factor deconvolution (NMFD) can model non-stationary notes with a single time-varying spectrogram excerpt and a sparse activation function indicating the onsets of pattern instances [2]. Different notes often lead to the same spectral templates, only shifted in pitch; extensions to NMFD anticipate these cases [3].

None of the algorithms in this family of nonnegative approaches take advantage of another kind of repetition, periodicity, which is the core of the REPET algorithm [4]. In it, a spectrogram fragment corresponding to an individual measure of music is compared to the spectrogram's median values across multiple measures, enabling separation of a repeating background and an independent foreground.

In this work, we introduce a similar technique to model periodic repetition in music using a nonnegative approach. Like with REPET, we divide the spectrogram into measure-length slices. Our key insight is that by stacking these slices in a new dimension, we can apply nonnegative tensor factorization (NTF) to estimate spectral, metric, and measure-based templates (see Fig. 1). In this way, we leverage many kinds of repetition at once: the short-term repetitions of NMF, the medium-term repetitions of NMFD, and the long-term repetitions of REPET. NTF was previously used to model the gain of elements common to both channels in a stereo signal [5], and the NMF variants cited above, including shift-invariant NMFD, can be formulated as NTF. However, we are not aware of any previous use of NTF to model periodicity.

This work was supported in part by JST ACCEL Grant Number JPMJAC1602, Japan.

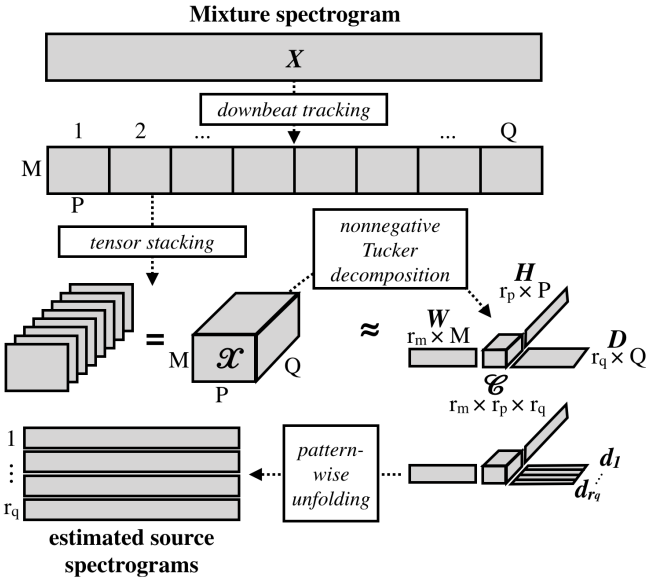

Fig. 1. Algorithm pipeline: the mixture magnitude spectrogram X is divided into measure-length fragments; these are stacked into a tensor $\mathcal{X}$; the tensor is factorized; the pattern activations in D are unfolded one at a time into r_q estimated source spectrograms.

Our approach is especially suited to pieces of music based on loops. Loops are short clips of audio that are repeated frequently, often in a way that defines the metre of the piece. We will compare our approach to two methods of source separation tailored to loop-based music: López-Serrano et al.'s NMFD approach [6] and Seetharaman and Pardo's iterative NMF algorithm [7]. López-Serrano et al. showed that a standard NMFD algorithm, provided with the number of expected loop templates and their length, was effective for discovering the activation patterns of the loops.

In contrast, the iterative NMF approach exploits a typical compositional technique in electronic dance music (EDM): the iterative building up of layers of loops. An example layout, based on the synthetic data used in [6] and re-used in this work, is shown in Fig. 2. Assuming the target track has such a structure, iterative NMF first trains an NMF model on a small initial excerpt. This model is used to reconstruct the rest of the piece; with reference to Fig. 2, this is like modeling the first loop 'A' in order to subtract 'A' from the rest of the mix. The algorithm detects the onset of the next new loop as

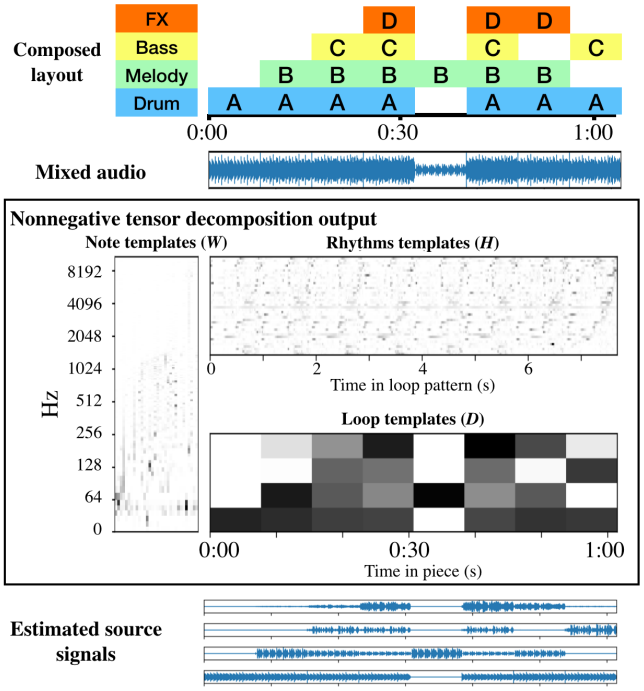

Fig. 2. Example input and output of the system. Top: *composed* layout of a loop-based piece of music. Middle: estimated templates corresponding to the notes, rhythms, and loop types in the audio. Bottom: source-separated signals.

a spike in the reconstruction error, and repeats the same steps.

In contrast to iterative NMF, our NTF approach does not require the composition to unfold in an additive manner. We show that our approach is faster to compute than a competitive implementation of NMFD and gives a more compact model, since rhythms shared by multiple spectral templates are encoded together. And in contrast to foreground/background separation with REPET, NTF can simultaneously separate multiple independent signals.

2. METHOD

2.1. Tensor construction

Beginning with the mixture signal, we apply the Short-Term Fourier Transform to obtain a power spectrogram $X \in \mathbb{R}^{M \times N}$. From there, the pipeline of our algorithm is illustrated in Fig. 1. We perform downbeat estimation using the madmom package [8], obtaining the frame index at which each measure starts.

We cut X at the downbeats into segments of size $M \times P$, where P is the maximum downbeat period, and stack the Q segments on top of each other to create a three-dimensional tensor $\mathcal{X} \in \mathbb{R}^{M \times P \times Q}$. Since the segments will vary slightly in length, we zero-pad measures shorter than P frames. In future work, it may be helpful to instead time-warp each segment to improve alignment.

We call the tensor $\mathcal{X}$ a "spectral cube". Slicing the cube in different dimensions reveals different patterns, as illustrated in Fig. 3. The matrix $\mathcal{X}_{[:,:,n]}$ returns the spectrogram of the n^{th} measure, an $M \times P$ matrix. $\mathcal{X}_{[:,n,:]}$ selects a single time index n within the period—i.e., a single metrical position—and shows the spectrum at that position across all measures, an $M \times Q$ matrix. Finally, $\mathcal{X}_{[n,:,:]}$ selects a single frequency bin n, and shows how it evolves throughout each measure (from bottom to top) throughout the piece (from left to right), a $P \times Q$ matrix. In Fig. 3, we observe redundancies in each dimensional slice: steady-state tones within measures (Fig. 3a); groups of tones that recur with the same period (Fig. 3b); and rhythmic patterns that recur throughout the piece (Fig. 3c).

2.2. Tensor decomposition

In standard NMF, we estimate the rank-r decomposition of the spectrogram X as the outer product of W and H:

$$X \approx \hat{X} = W \circ H \tag{1}$$

where $W \in \mathbb{R}^{M \times r}$ is a matrix of r spectral templates and $H \in \mathbb{R}^{r \times N}$ is a matrix of r activation functions. The templates and activation functions are paired: the outer product of the i^{th} pair, $W_{[:,i]} \circ H_{[i,:]}$, is a matrix the size of the original spectrum giving the contribution of the i^{th} template.

Nonnegative tensor factorization extends NMF to higher dimensions, allowing one to find dependencies across all dimensions (see [9] for a detailed reference). We discuss one version of NTF, the nonnegative Tucker decomposition, which we compute using TensorLy [10]. Whereas we choose a single rank r for NMF, the Tucker decomposition requires three ranks, (r_m, r_p, r_q). It models the tensor $\mathcal{X}$ as:

$$\mathcal{X} \approx \hat{\mathcal{X}} = \mathcal{C} \circ (W \circ H \circ D) \tag{2}$$

where $\mathcal{C} \in \mathbb{R}^{r_m \times r_p \times r_q}$ is the "core tensor", which can be "unfolded" by taking the outer product with the three template matrices W, H and D to obtain an approximation of $\mathcal{X}$. These template matrices are: $W \in \mathbb{R}^{M \times r_m}$, a set of r_m spectral templates ('notes'), like before; $H \in \mathbb{R}^{P \times r_p}$, a set of r_p activation patterns, but now expressed as a function of metrical position instead of time ('rhythms'); and $D \in \mathbb{R}^{Q \times r_q}$, a set of r_q repetition patterns, each indicating how a sound may repeat across the piece ('loops'). The patterns are not paired off like with NMF; instead, the pixel $\mathcal{C}_{[i,j,k]}$ encodes how any note W_i repeats with rhythm H_j in loop D_k. An example of W, H and D are shown in Fig. 2.

The loop templates in D directly model the structure of the piece, similar to the activation functions in NMFD, but presuming a consistent period. To estimate the signal associated with the k^{th} loop type, unfold a single repetition pattern in D to get the estimated source spectrum $\hat{X}_k = \mathcal{C}_{[:,:,k]} \circ (W \circ H \circ D_{[:,k]})$. We use soft masking with power = 2 to estimate the source signal (see librosa [11] for details).

The Tucker decomposition is a very compact representation. To model the song in Fig. 2, NMFD required over 1.36 million values $(4(M \cdot P + N))$; iterative NMF required 148,928 values $(4r_m \cdot (M+N))$; and NTF required 22,704 values $(r_m M + r_p P + r_q Q + r_m r_p r_q)$, with $M = 1024, P = 330, Q = 8, r_m = 32, r_p = 40, r_q = 4$.

Some limitations of the NTF algorithm can be noted immediately. First, the method depends on there being a close alignment between the spectra of each measure. If the downbeat tracking fails, or the measures are of varying length to begin with, NTF will not model the signal effectively. Second, while the factorization strives to estimate templates with independent columns (i.e., so that the columns of D look different), redundancies in the core tensor can occur. That is, different estimated loops can end up sounding similar. For example, in analyzing the song in Fig. 2, instead of learning templates that match A, B, C and D, it could learn templates for $A, B, (A + C)$

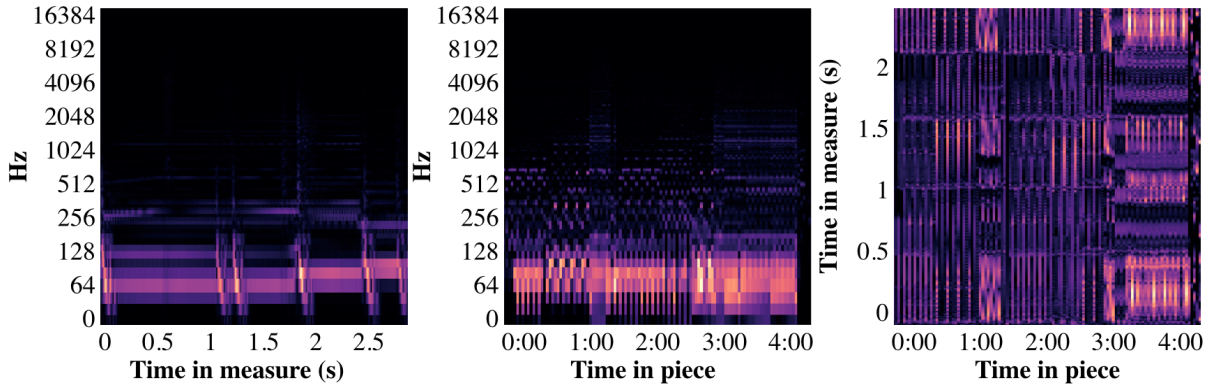

Fig. 3. Three slices of the spectral cube $\mathcal{X}$ constructed for the song "El Pico" by Ratatat. From left to right: (a) a single measure; (b) a single metrical position (the second beat onset); (c) a single frequency bin (pointing to the E above middle C).

and $(B + D)$; this decomposition also models the piece perfectly, but not in the desired way. However, this is also a shortcoming of NMFD. Modifying the NTF algorithm to enforce independence between loops is a goal for future work.

3. EVALUATION

We test our proposed NTF approach against two published algorithms: NMFD [12][1] and iterative NMF [7], which we re-implemented. Moreover, we test standard and guided versions of each algorithm; i.e., versions with or without knowledge of the ground truth downbeat locations.

We evaluate the source separation quality of all algorithms using the standard trio of metrics, signal-to-distortion ratio (SDR), signal-to-interference ratio (SIR) and signal-to-artifact ratio (SAR), computed with mir_eval [14]. For NTF and NMFD, we also evaluate the quality of the estimated map of sample activations using two metrics: correlation, also reported by [6]; and accuracy (which requires binarizing the estimated activations).

All metrics (SDR, SIR, SAR, correlation, accuracy) compare two audio tracks or two activation functions. To compare a set of 4 estimated tracks to the 4 ground truth tracks, we test all permutations and report the highest result. Iterative NMF sometimes estimates more sources, but we kept only the first 4 for evaluation. Every other algorithm was explicitly informed with the correct number of loops, 4. Testing how sensitive each algorithm is to changes in this parameter is an important task for future work. Additional details:

- NTF (proposed): as described in Section 2, with $r_m = 32, r_p = 40, r_q = 4$. We picked values for r_m and r_p that were much less than $M = 1024$ and $P \approx 330$, respectively, and which gave fair results for songs like that in Fig. 3.
- NTF (guided): the true downbeats are used instead of the madmom beat tracking output.
- Iterative NMF: as described in [7], with $r = 8$ (for each of the 4 NMF model).
- Iterative NMF (guided): we provide the correct 'next boundary' time at which a new loop is introduced.
- NMFD: as described in [6], using log-frequency spectrogram; we provide P, the true template size.

[1] http://www2.imm.dtu.dk/pubdb/p.php?4499 [13]

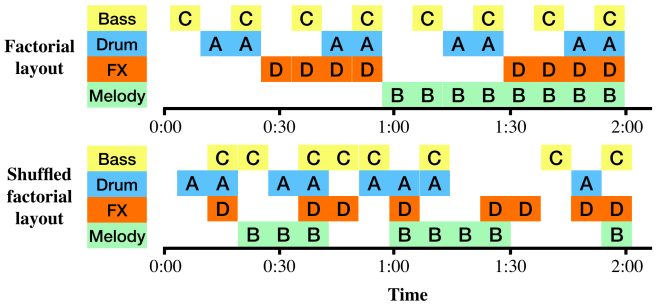

Fig. 4. New layouts tested besides the *composed* layout in Fig. 2.

- NMFD (guided): we also initialize the activation matrix H to be 1 at the true downbeat frames (and their immediate neighbours) and zeros elsewhere.

We apply all the algorithms to the same audio dataset created by [6], which was published openly.[2] The set contains songs in seven styles of EDM, each with the same compositional layout of four loops, as shown in Fig. 2. This *composed* layout fits the assumption that tracks will be introduced iteratively, and as stated above it can be decomposed into four templates in at least two valid ways. To compare the algorithms when the iterative assumption does not hold and when the loops are less linearly dependent, we created two new layouts, illustrated in Fig. 4, using the same audio.[3] The *factorial* layout contains all 15 possible combinations of the loops, arranged iteratively. The *shuffled factorial* layout contains the same combinations of loops, but in a shuffled order. The loops ('A' to 'D') range from 6 to 8 seconds long, so the *composed* clips range in length from 50–64 seconds and the *factorial* clips from 93–120 seconds.

4. RESULTS AND DISCUSSION

The source separation results for the unguided algorithms are reported in Fig. 5a. Comparing the unguided algorithms (darker colours), NMFD performs the best overall, but the proposed algorithm (NTF) achieves high SIR, indicating the estimated sources are

[2] https://www.audiolabs-erlangen.de/resources/MIR/2016-ISMIR-EMLoop

[3] https://github.com/jblsmith/icassp2018

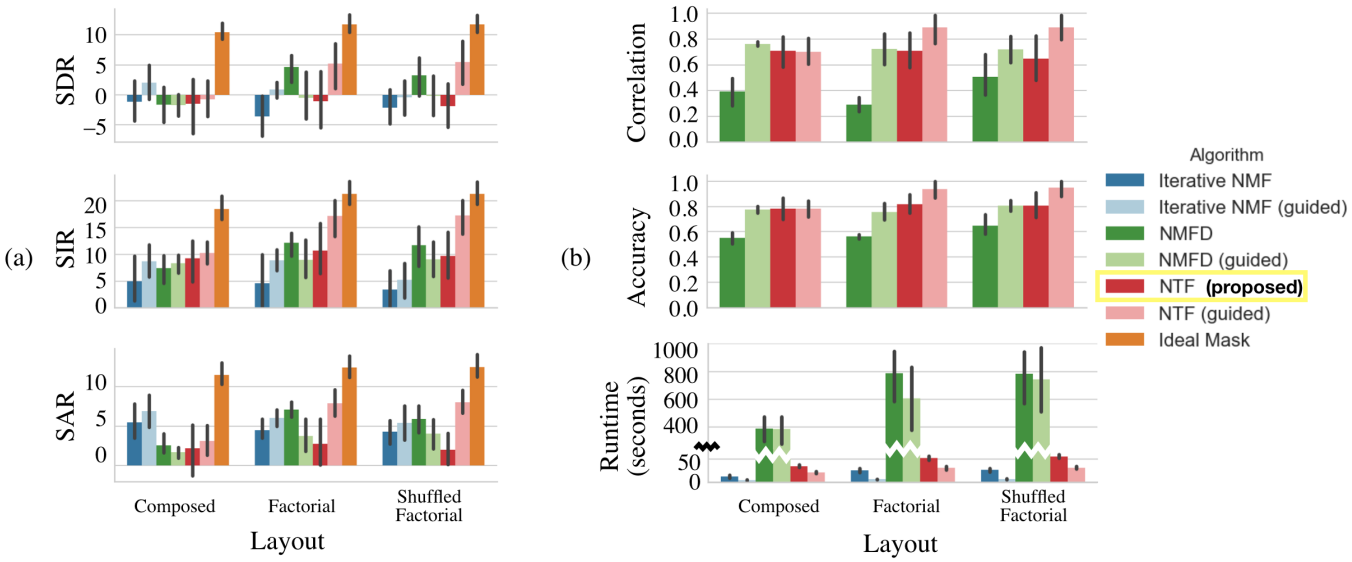

Fig. 5. (a) Source separation quality (higher is better). (b) Layout prediction correlation and accuracy (higher is better), and algorithm runtime (lower is better).

very independent. However, comparing the algorithms guided with the correct downbeats (paler colours), the NTF approach takes a clear lead overall, especially in SIR, where it approaches the quality of the ideal mask on the *factorial* layouts. Among the nine setups (3 metrics $\times$ 3 datasets), guided NTF is surpassed twice: by iterative NMF, for SDR and SAR, in the *composed* layout it is tailored to.

Listening to the output of our algorithm in cases where it failed, our algorithm seemed to devote excess modeling power to the drum track. The drum was usually the loudest loop, so perhaps focusing on the drum track was the optimal way, in the least-squares sense, to model the signal.

Analyzing the loop structure of a piece is of interest for source separation, but also as a type of transcription. The quality of the loop activation transcriptions is shown in Fig. 5b. (For NTF, this transcription is D; for NMFD, it is the activation function at the downbeat samples and their neighbours, summed together.) We measure the correlation between these real-valued transcriptions and the ground truth layout. We also binarize the transcriptions by scaling each column d_k in D between 0 and 1, and keeping all values above 0.5. We can then measure the simple accuracy of the binarized transcriptions. On both metrics, the performance of NTF surpasses NMFD. Also, the quality of the NTF loop transcription appears to be less sensitive to downbeat estimation errors (i.e., less need for guidance). Finally, we note the mean running time for each algorithm (Fig. 5b). While the average runtime was under 30 seconds for both NTF and iterative NMF, the average for NMFD was over 10 minutes. However, each system relies on code components from different public sources, so there is no guarantee they each is comparably optimized.

Compared to NMFD, NTF strikes the following compromise: reduced runtime (and model size), and increased source independence (SIR) and layout accuracy—at the cost of more artefacts (SAR) and greater reconstruction error (SDR). Compared to iterative NMF, NTF has slightly greater runtime, improved independence (SIR) and similar reconstruction error (SDR)—at the cost of more artefacts (SAR).

We observed the high quality SIR of NTF compared to other metrics. Also, while NTF was clearly best for describing loop layout, it required accurate downbeats to exceed the others on source separation. Both results are explained by noting that NTF gives a *conservative* estimate of the sources: it only groups together sounds which recur precisely. Thus, if the downbeats are more precise, it can model more of the power spectrum. While this is a limitation for source separation, it may be the source of its advantage in layout description: by conservatively modeling only the precisely-recurring sounds, it estimates a much cleaner map of the loop activations, even when the downbeats are not provided. This highlights the potential to apply NTF to structure analysis.

An important caveat is that we did not study the effect of varying r_q, the number of loop types anticipated; instead, we informed each algorithm with the correct rank, 4. A key feature of iterative NMF is that it obviates the need to pre-select this rank. Thus, future evaluations of NTF should demonstrate robustness to incorrect values, or propose a method of estimating the rank automatically.

5. CONCLUSION

We have introduced a nonnegative approach to source separation that exploits the periodicity of musical signals. Although the quality of its source separation depends on the quality of the downbeat estimation, it is a robust method of estimating loop activations.

The model we presented may be extended by adding new dimensions (e.g., by stacking harmonics in the manner of [15]'s HCQT, or by hierarchically adding a within-beat time dimension). Although we analyzed the pieces at a single time-scale, a spectral cube may be created for any timescale. Managing structures discovered at multiple timescales remains a task for future work, along with enforcing independence between the loop types.

Also, when using NMF for source separation, one may set r much higher than the expected number of sources, and try to group the templates according to timbral or temporal correlation [16]. This approach could be used after NTF to further separate sources within each loop pattern.

6. REFERENCES

[1] Elizabeth Margulis, *On Repeat*, Oxford University Press, Oxford, UK, 2014.

[2] Paris Smaragdis, "Non-negative matrix factor deconvolution: Extraction of multiple sound sources from monophonic inputs," in *Independent Component Analysis and Blind Signal Separation*, vol. 3195 of *Lecture Notes in Computer Science*, pp. 494–499. Springer-Verlag, Berlin, Heidelberg, 2004.

[3] Derry FitzGerald, Matt Cranitch, and Eugene Coyle, "Extended nonnegative tensor factorisation models for musical sound source separation," *Computational Intelligence and Neuroscience*, vol. 2008, 2008.

[4] Zafar Rafii, Antoine Liutkus, and Bryan Pardo, "REPET for background/foreground separation in audio," in *Blind Source Separation*, G. R. Naik and W. Wang, Eds., Signals and Communication Technology, pp. 395–411. Springer-Verlag, 2014.

[5] Derry FitzGerald, Matt Cranitch, and Eugene Coyle, "Sound source separation using shifted non-negative tensor factorisation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Toulouse, France, 2006, vol. 5.

[6] Patricio López-Serrano, Christian Dittmar, Jonathan Driedger, and Meinard Müller, "Towards modeling and decomposing loop-based electronic music," in *Proceedings of the International Society for Music Information Retrieval Conference*, New York, NY, USA, 2016, pp. 502–508.

[7] Prem Seetharaman and Bryan Pardo, "Simultaneous separation and segmentation in layered music," in *Proceedings of the International Society for Music Information Retrieval Conference*, New York, NY, USA, 2016, pp. 495–501.

[8] Sebastian Böck, Filip Korzeniowski, Jan Schlüter, Florian Krebs, and Gerhard Widmer, "madmom: a new Python Audio and Music Signal Processing Library," in *Proceedings of the ACM International Conference on Multimedia*, Amsterdam, The Netherlands, November 2016, pp. 1174–1178.

[9] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shunichi Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to exploratory multiway data analysis and blind source separation*, Wiley, 2009.

[10] Jean Kossaifi, Yannis Panagakis, and Maja Pantic, "TensorLy: Tensor learning in python," *arXiv preprint arXiv:1610.09555*, 2016.

[11] Brian McFee, Colin Raffel, Dawen Liang, Daniel Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the Python in Science Conference*, 2015.

[12] Mikkel N. Schmidt and Morten Mørup, "Nonnegative matrix factor 2-d deconvolution for blind single channel source separation," in *International Conference on Independent Component Analysis and Signal Separation*. Springer, 2006, pp. 700–707.

[13] Mikkel N. Schmidt and Morten Mørup, "Matlab demo: Nonnegative matrix factor 2-D deconvolution for blind single channel source separation," 2006.

[14] Colin Raffel, Brian McFee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel PW Ellis, "mir_eval: A transparent implementation of common MIR metrics," in *Proceedings of the International Society for Music Information Retrieval Conference*, Curitiba, Brazil, 2014, Citeseer, pp. 367—372.

[15] Rachel M. Bittner, Brian McFee, Justin Salamon, Peter Li, and Juan Pablo Bello, "Deep salience representations for f_0 estimation in polyphonic music," in *Proceedings of the International Society for Music Information Retrieval Conference*, 2017, pp. 63–70.

[16] Martin Spiertz and Volker Gnann, "Source-filter based clustering for monaural blind source separation," in *Proceedings of the 12th International Conference on Digital Audio Effects*, 2009.