DRUM TRANSCRIPTION USING PARTIALLY FIXED NON-NEGATIVE MATRIX FACTORIZATION WITH TEMPLATE ADAPTATION

Chih-Wei Wu
Georgia Institute of Technology
Center for Music Technology
cwu307@gatech.edu

Alexander Lerch
Georgia Institute of Technology
Center for Music Technology
alexander.lerch@gatech.edu

ABSTRACT

In this paper, a drum transcription algorithm using partially fixed non-negative matrix factorization with template adaptation is presented. The proposed method allows users to identify percussive events in complex mixtures of music with a minimal training set. The algorithm decomposes the music signal into two parts: percussive part with predefined drum templates and harmonic part with undefined entries. The harmonic part is able to adapt to the music content, allowing the algorithm to work in polyphonic mixtures. Drum events can be simply picked from the percussive activation matrix with onset detection. The algorithm also adapts the drum templates to each signal individually, providing a more flexible use case against unknown data. The performance of the proposed system has been evaluated and compared with other systems. The results show that template adaption helps to improve the transcription performance, and the evaluation results are comparable with state of the arts systems.

1. INTRODUCTION

As being one of the most intensively researched areas in Music Information Retrieval (MIR), Automatic Music Transcription (AMT) is often considered the core technology that would enable high-level representations of music signals with the potential of improving virtually any MIR task. A complete transcription system comprises many related sub-tasks such as multi-pitch detection, onset detection, instrument recognition, and rhythm extraction [2]. While the main focus is mostly on pitched instruments, a considerable amount of publications deal with the transcription of percussive sounds in mixture of tonal and percussive instruments. The drum track in popular music conveys information about tempo, rhythm, style, and possibly the structure of a song. A drum transcription system alone enables applications in active listening [27], music education, and interactive music performance [25].

This study explores the application of the popular transcription method of Non-Negative Matrix Factorization (NMF) for drum transcription in polyphonic music. The proposed method addresses the problem of unknown sources in NMF with content and template adaptability. The paper is structured as follows: Section 2 provides an overview of the research in this area. In Section 3 we present our approach; evaluation results are being presented and discussed in Section 4. Section 5 provides a summary, conclusion, and directions of future work.

2. RELATED WORK

Automatic drum transcription systems, as summarized by Gillet and Richard [10], can be divided into three categories: (i) *segment and classify* [4, 8, 22], (ii) *separate and detect* [1, 6, 7, 15, 17], and (iii) *match and adapt* [28, 29].

Recently, more systems are built based on the second type of approaches (*separate and detect*) as NMF-related methods become more and more popular. In this type of approaches, the music signal is assumed to be a superposition of multiple sound sources. By decomposing the signal into source templates with corresponding activation functions, the system is able to transcribe the musical events by analyzing the activation of different source templates. When NMF is applied to the task of music transcription, typically the following challenges have to be faced:

First, the number of sound sources and notes within a music recording is usually unknown. To optimally decompose a signal, this number is necessary for determine the size of the dictionary and activation matrix. Without this information, it is difficult to determine a suitable rank r setting for the decomposition process. This problem would be less severe when the sound sources in the target signal are available [14]. However, in most cases, this prior information is difficult to acquire. Another approach is to build a dictionary that contains more source templates than the target signal. Benetos et al. used a probabilistic extension of NMF (Probabilistic Latent Component Analysis, PLCA) to jointly transcribe pitched and unpitched sounds in polyphonic music with a relatively large pre-trained dictionary [3]. Although this method can provide harmonic and percussive contents of the music simultaneous, its robustness against unknown sources still needs to be evaluated with larger datasets.

Second, it can be hard to identify the corresponding in-

Attribution: Chih-Wei Wu, Alexander Lerch. "Drum Transcription Using Partially Fixed Non-Negative Matrix Factorization with Template Adaptation", 16th International Society for Music Information Retrieval Conference, 2015.

strument of every component in the dictionary matrix W. This problem becomes more severe when the rank is selected too high or too low. Helen and Virtanen trained an SVM to separate drum components from the harmonic components; the rank number was derived empirically during the factorization process [11]. The identified drum components and their corresponding activities could later be used to reconstruct the drum signal, resulting in a system for drum source separation. Their approach requires a significant amount of training data for the classifier and, more importantly, the results can be expected to be very susceptible to choice of rank. Yoo et al. proposed a co-factorization algorithm [26] to simultaneously factorize a drum track and a polyphonic signal. They used the dictionary matrix from the drum track to identify the drum components in the polyphonic signal. This approach ensures that the drum components in both dictionary matrices are estimated only from the drum track, resulting in proper isolation of the harmonic components from the drum components. Since their system aims at drum separation they can work at very high ranks. For drum transcription, however, this approach is not directly applicable because of the association between the instruments and the entries in the dictionary matrix is unknown.

Third, a suitable penalty term or sparsity constraint for detecting percussive instruments still needs to be investigated. In general, these constraints are the additional terms in the cost function that will facilitate the sparseness or different properties in the resulting activation matrix. Virtanen proposed to use additional constraints for temporal continuity and sparseness [24]. It is reported that by using the temporal continuity criterion, the detection accuracy and SNR of the pitched sounds can be improved in the source separation task, whereas no significant improvement is shown with the sparseness constraint.

Another issue is the adaptability of the extracted templates. When using supervised NMF, the algorithm loses its adaptability and might fail when the target signal is very different from the pre-trained dictionary. Dittmar and Gartner proposed to use adaptive bases during the NMF decomposition process [5]. However, the results indicated that the adaptive process did not improve the performance of the transcription accuracy. Also, whether this method can work in the polyphonic mixtures remains to be shown.

Transcription methods other than NMF can avoid the above mentioned issues, but they can offer different challenges at the same time. Paulus and Klapuri proposed to use hidden markov models (HMM) for drum transcription [16]. This method models temporal connections between drum events and detect the drum based on the probabilistic model. However, its complexity may increase drastically when more instruments are to be detected. Additionally, a larger dataset would be needed for this method to train a generic model. Another recent approach is to use bar information to classify the audio signal into different predefined drum patterns [23]. This approach requires additional information of the bar locations and a large dictionary, which can be impractical in some use cases.

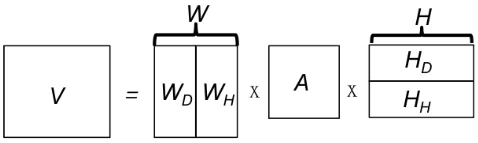

Figure 1. Illustration of the factorization process. Subscript D: drum components H: harmonic components. A is the weighting matrix.

3. METHOD

3.1 Algorithm Description

The basic concept of NMF is to approximate a matrix V with matrices W and H as $V \approx WH$ with non-negativity constraints. Given a $m \times n$ matrix V, NMF will decompose the matrix into the product of a $m \times r$ dictionary matrix W and an $r \times n$ activation matrix H, with r being the rank of the NMF decomposition. In most audio applications, V is the spectrogram to be decomposed, W contains the magnitude spectra of the salient components, and H indicates the activation of these components with respect to time [20]. The matrices W and H are estimated through an iterative process that minimizes a distance measure between the target spectrogram V and its approximation [12].

In this paper, we propose a method using partially-fixed NMF (here we refer it as PFNMF) to transcribe drum events in polyphonic signals. The idea of using NMF with prior knowledge of the target source within the mixture has been applied to source separation tasks [21], and multipitch analysis [18]. The method described here is based on similar ideas but with different emphasis: (i) we focus on a real world scenario in which users only have limited amount of training samples that are possibly different from the target source, and (ii) we propose to use a small dictionary matrix which is both efficient and easily interpretable. (iii) the proposed method is able to adapt to different contents in the polyphonic mixtures

To allow NMF to adapt the music content, a method inspired by [26] is proposed for drum transcription task. Figure 1 visualizes a similar concept from the work of Yoo et al.: the matrices W and H are split into the matrices W_D and W_H, and H_D and H_H, respectively. Instead of using co-factorization, however, we propose to initialize the matrix W_D with drum templates and to not modify it during the factorization process. Matrices W_H, H_H, and H_D are initialized with random numbers. The rank r_D of W_D and H_D depends on the number of templates provided, and the rank r_H depends on the user's input. The total rank $r = r_\mathrm{D} + r_\mathrm{H}$.

By increasing the r_H, a larger W_H will be initialized to better adapt to the target signal. However, this unbalanced increase in components will also reduce the weight of the drum components in the optimization process, causing a significant drop in the performance. Therefore, a weighting matrix A is introduced to balance the weight between drum and harmonic components. The weighting matrix A is a $r \times r$ diagonal matrix, which contains r_D coefficients α and

r_{H} coefficients β. In the paper, the coefficients are set to be $\alpha = (r_{\mathrm{D}} + r_{\mathrm{H}})/r_{\mathrm{D}}$ and $\beta = r_{\mathrm{H}}/(r_{\mathrm{D}} + r_{\mathrm{H}})$

The distance measure used in this paper is KL-divergence, in which $D_{\mathrm{KL}}(x \mid y) = x \cdot \log\left(x/y\right) + (y - x)$. The cost function as shown in Eq. (1) is minimized by applying gradient decent and multiplicative update rules.

$$J = D_{\mathrm{KL}}(V \mid \alpha W_{\mathrm{D}} H_{\mathrm{D}} + \beta W_{\mathrm{H}} H_{\mathrm{H}}) \tag{1}$$

The matrices W_{H}, H_{H}, and H_{D} will be updated according to Eqs. (2)–(4).

$$H_{\mathrm{D}} \leftarrow H_{\mathrm{D}} \frac{\alpha W_{\mathrm{D}}^T (V/(\alpha W_{\mathrm{D}} H_{\mathrm{D}} + \beta W_{\mathrm{H}} H_{\mathrm{H}}))}{\alpha W_{\mathrm{D}}^T} \tag{2}$$

$$W_{\mathrm{H}} \leftarrow W_{\mathrm{H}} \frac{(V/(\alpha W_{\mathrm{D}} H_{\mathrm{D}} + \beta W_{\mathrm{H}} H_{\mathrm{H}}))\beta H_{\mathrm{H}}^T}{\beta H_{\mathrm{H}}^T} \tag{3}$$

$$H_{\mathrm{H}} \leftarrow H_{\mathrm{H}} \frac{\beta W_{\mathrm{H}}^T (V/(\alpha W_{\mathrm{D}} H_{\mathrm{D}} + \beta W_{\mathrm{H}} H_{\mathrm{H}}))}{\beta W_{\mathrm{H}}^T} \tag{4}$$

To summarize, the method consists of the following steps:

1. Construct a $m \times r_{\mathrm{D}}$ dictionary matrix W_{D}, with r_{D} being the number of drum components to be detected.

2. Given a pre-defined rank r_{H}, initialize a $m \times r_{\mathrm{H}}$ matrix W_{H}, a $r_{\mathrm{D}} \times n$ matrix H_{D} and a $r_{\mathrm{H}} \times n$ matrix H_{H}.

3. Normalize W_{D} and W_{H}.

4. Update H_{D}, W_{H}, and H_{H} using Eqs. (2)–(4).

5. Calculate the cost of the current iteration using Eq. (1).

6. Repeat step 3 to step 5 until convergence.

The time positions of the drum events can then be extracted by applying a simple onset detection on the rows of matrix H_{D}.

3.2 Template Adaptation

While PFNMF retains the identity of each individual instrument by using pre-defined drum templates, it may fail to deal with the variations in the unknown data. One approach to address this issue is using template adaption method during the process. Previous approaches to include template adaptation in drum transcription process can be found in [29], [5]. These approaches usually start with seed templates and gradually adapt them to the target sources. In this paper, we propose two methods for template adaption with PFNMF.

3.2.1 Method 1: Cross-correlation Based Update

In the first method, the drum template W_{D} is updated based on the cross-correlation between activation H_{H} and H_{D} for each individual drum. As described in section 3.1, PFNMF starts by randomly initializing a W_{H} with rank r_{H}. Although W_{H} tends exclude drum part and adapt to the harmonic content, it may still contain entries that belongs to percussive instruments due to the mismatch between the drum templates and the target sources. This will result in cross-talk between H_{H} and H_{D} and potentially decrease the performance. However, these entries may also provide complementary information to the original drum templates. To identify these entries, the normalized cross-correlation between H_{H} and H_{D} for each individual drum is computed using Eq.(5), where x and y represents different activation vectors, and N is the number of samples in the activation vectors. A threshold ρ_{thres} = 0.5 is defined for finding the complementary entries, and the drum template W_{D} can be updated using Eq.(6), where $W_{\mathrm{H}}^{(i)} (i = 1, ..., S)$ are the entries with their corresponding $\rho_{x,y}$ higher than ρ_{thres}, and S is the number of the selected entries. The adaptation coefficient $\gamma = \frac{1}{2^k}$, where k is the number of iteration.

$$\rho_{x,y} = \frac{\sum_{n=1}^{N} x(n) \cdot y(n)}{\|x\|_2 \cdot \|y\|_2} \tag{5}$$

$$W_{\mathrm{D}}' = (1 - \gamma) W_{\mathrm{D}} + \gamma \frac{1}{S} \sum_{i=1}^{S} (\rho^{(i)} W_{\mathrm{H}}^{(i)}) \tag{6}$$

The method can be summarized in the following steps:

1. Normalize H_{D} and H_{H}.

2. Compute normalized cross-correlation between every entries of H_{D} and H_{H} using Eq.(5).

3. Select entries $i = 1, ..., S$ with $\rho >= \rho_{thres}$.

4. Update W_{D} using Eq.(6).

5. Randomly re-initialize $W_{\mathrm{H}}^{(i)}$.

6. Perform PFNMF using W_{D}'.

7. Repeat step 1 to step 6 until convergence.

3.2.2 Method 2: Alternate Update

In the second method, the drum template W_{D} is adapted by alternatively fixing W_{D} and H_{D} during the process. The adaptation process starts by fixing W_{D}, and PFNMF will try to fit a best activation H_{D} to approximate the drum part in the music. Once H_{D} is determined, a new iteration of PFNMF can be started by fixing H_{D} and allow W_{D}, W_{H} and H_{H} to update. This constraint will guide the algorithm to fit better drum templates based on the current detected activation H_{D}. The update rule for W_{D} is shown in Eq.(7).

$$W_{\mathrm{D}} \leftarrow W_{\mathrm{D}} \frac{(V/(\alpha W_{\mathrm{D}} H_{\mathrm{D}} + \beta W_{\mathrm{H}} H_{\mathrm{H}}))\alpha H_{\mathrm{D}}^T}{\alpha H_{\mathrm{D}}^T} \tag{7}$$

The method can be summarized in the following steps:

1. Perform PFNMF with fixed W_{D}, update H_{D}, W_{H} and H_{H}.

2. Randomly re-initialize W_{H} and H_{H}.

3. Perform PFNMF with fixed H_{D}, update W_{D}, W_{H} and H_{H}.

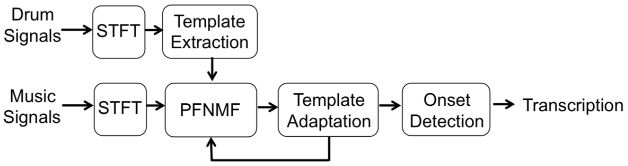

Figure 2. Flowchart of the drum transcription system

4. Randomly re-initialize H_D, W_H and H_H .
5. Repeat step 1 to step 4 until convergence.

Both of the above mentioned methods have the same criteria that will stop iterating when the error change between two consecutive iterations is smaller than 0.1%. The maximum iteration number is set to 20. However, the adaptation process usually converges after 5 to 10 iterations.

3.3 Implementation

Figure 2 shows the flow chart of the implemented system. The STFT of the signals will be calculated using a window size and a hop size of 2048 and 512 with a sampling frequency of 44.1 kHz. A pre-trained dictionary matrix will be constructed from the training set, which consists of isolated drum sounds. Next, the PFNMF will be performed with the initial drum dictionary and rank $r = r_\mathrm{D} + r_\mathrm{H}$ as described in section 3.1. In each iteration, the drum dictionary will be updated using template adaptation methods described in section 3.2. Finally, the activation Matrix H_D is evaluated to determine the onset positions and their corresponding classes.

The dictionary matrix W_D is created by extracting a template spectrum from isolated training drum samples. The template spectrum is a median spectrum of all individual events of one drum class in the training set. The length of each event is approximately 80 ms. The templates are extracted for the three classes: Hi-Hat (HH), Bass Drum (BD) and Snare Drum (SD).

High values in the activation matrix H_D indicate the presence of a drum event. More specifically, the activity difference of each row of the activation matrix could be considered as the onset novelty function of each individual drum. We use a median filter as a standard approach to create an signal-adaptive threshold for peak picking [13]. In this paper, the length b and the offset coefficient λ of the median adaptive threshold are set to be 0.1 s and 0.12 for every track; b is the order of the filter, and λ is a constant shift.

4. EVALUATION

4.1 Dataset Description

The experiments have been conducted on two different datasets. The first one is the *minus one* subset from the ENST public drum data set [9]. This data set consists of recordings from three different drummers performing on their own drum kits. The set for each drummer contains individual hits, short phrases of drum beats, drum solos, and short excerpts played with the accompaniments. The minus one subset has 64 tracks of polyphonic music, and the sampling rate of every track is 44.1 kHz. Each track in this subset has a length of approximately 70 s with varying style. More specifically, the subset contains various drum playing techniques such as ghost notes, flam, and drag; these techniques are considered difficult to identify with existing drum transcription systems. The accompaniments are mixed with their corresponding drum tracks using a scaling factor of 1/3 and 2/3, which are the same settings as in [16].

Another dataset used for cross-dataset validation is IDMT-SMT-Drums [5]. This dataset consists of 95 drum loop recordings from three drum kits (RealDrum, WaveDrum and TechnoDrum). The sampling rate of every track is 44.1 kHz, and the total duration of the dataset is approximately two hours. This dataset also contains isolated drum hits that are used to synthesize the drum loops. However, in our experiments, we only use the drum loops for testing the robustness of our system against unknown sounds.

The drum templates have been generated from a different part of the ENST dataset which only contains single hits performed by the same group of drummers. Each track contains 5 to 6 single hits on different drums for each drummer. For every instrument (HH, BD, SD), one track per drummer is collected as training data. The onset position of these single hits was determined using the annotated ground truth.

4.2 Evaluation Procedure

We evaluate the proposed system in both monophonic and polyphonic mixtures, which are the same set of audio tracks with or without accompaniments. A three-fold cross-validation is applied to the evaluation process. Single drum hits collected from two drummers are used to train the system, complete mixtures from the third drummer are used to test the system, and the process repeats three times to test every drummer in the dataset. This process is the same as described in [16], the purpose is to prevent the system from seeing the test data. Note that the training data used in the system are single drum hits, and the number of onsets is significantly fewer than the test data. Typically, the training data only consists of 10 to 12 single hits of each drum class. This is similar to the real-world use case, where the users may only access to a limited number of training samples.

The evaluation metrics follow the standard calculation of the precision (P), recall (R), and F-measure (F). An onset is considered to be a match with the ground truth if the time deviation between the annotated and detected onset is less or equal to 50 ms.

4.3 Evaluation Results

4.3.1 Rank Estimation

In an initial test to determine the rank r_H of the PFNMF, $r_\mathrm{H} = 5, 10, 20, 40, 80, 160$ have been tested in both monophonic and polyphonic signals. For the monophonic case, the same tracks as described in Section 4.1 are used except

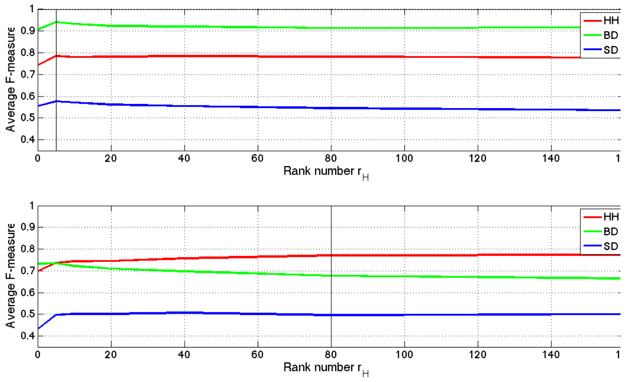

Figure 3. Average F-measure versus harmonic rank r_{H} in (Top) monophonic (Bottom) polyphonic dataset

for removing the accompaniments. The resulting individual F-measures are shown in Figure 3. In the monophonic case, a general trend of increasing performance with increasing r_{H} can be observed up to $r_{\mathrm{H}} = 5$; As the r_{H} goes higher, the performance will slightly decrease and eventually reach a saturation point. In the polyphonic case, however, the saturation point seems to shift toward $r_{\mathrm{H}} = 80$ as indicated by the black line.

One explanation could be that when the signal contains no harmonic components, the extra templates might not benefit the decomposition process. Instead, cross talk between W_{D} and W_{H} might happen and introduce noise to the H_{D}. On the other hand, if the signal contains harmonic components, increasing r_{H} could actually help the algorithm to better adapt to the signal, especially for high frequency parts of the drums (HH and SD). As a trade-off between accuracy and efficiency, a rank number $r_{\mathrm{H}} = 50$ is chosen as a general setting for transcribing polyphonic music in our experiments.

4.3.2 *Threshold Selection*

The transcription results can be obtained after applying onset detection on each drum activation (please see section 3.3). However, the performance varies according to the selection of the signal-adaptive threshold. To evaluate the influence of different thresholds, the average F-measure of all drums with different λ on IDMT-SMT-Drums dataset is shown in Figure [CITE FIGURE HERE]. A general trend as reported in [5] can be observed as well. One major difference is that both PFNMF with adaptation method 1 (referred as APFNMF1) and adaptation method 2 (referred as APFNMF2) outperform the original algorithm. This verifies the assumption that template adaptation process does help the algorithm against the unknown sounds. The overall performance is slightly lower than [5] due to the fact that the templates are extracted using ENST dataset. However, the average F-measure of APFNMF1 is 80.2%, which indicates the robustness of the proposed method outside of dataset.

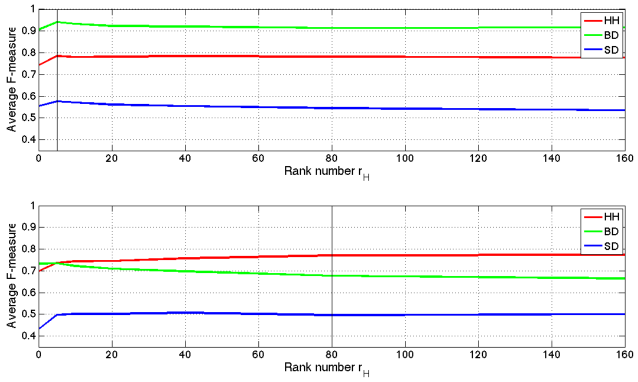

Figure 4. Average F-measure versus offset coefficient λ using (a) PFNMF (Solid line) (b) APFNMF1 (Dotted line)(c) APFNMF2 (Diamond dotted line)

4.3.3 *Evaluation*

Table 1 shows the evaluation results on ENST drum dataset *minus one* subset without accompaniments. The compared methods consist of Gillet et al. [10] using *segment and classify* approach with late decision fusion and Paulus et al. [16] using HMM with maximum likelihood linear regression (MLLR) for acoustic adaptation. All the compared methods use the same dataset with the same mixing settings. Since the target signals contain only drum components, the r_{H} can be set to be to a small number. In this experiment, r_{H} is set to 10 for absorbing drum sounds other than HH, BD and SD. The results show that our proposed method is able to transcribe drum events with an average F-measure = 77.9% using APFNMF2. This result is higher than 73.8% as reported in [10], and at the same level of 77.9% as [16].

In order to investigate the performance of the proposed system in the complete mixtures of music, another set of tracks as mentioned in Section 4.2 has been used. The results, listed in Table **??**, show that the proposed system has better performances comparing with the reference system in [10]. The average F-measures using the original training samples are 78.0%, 76.5%, and 56.1% for HH, BD, SD, respectively. Comparing with the reported F-measures of 77.7%, 65.0%, and 64.8%, our system shows great improvement in BD, and comparable accuracy in HH and SD. Another result using training samples from 200 Drum Machines shows a similar trend in both BD and SD, but slight worse result in HH. In the first case, only a few training samples are used to construct the dictionary; In the second case, the training samples are completely different from the target signals. The results from these cases indicate that the presented algorithm is relatively robust in polyphonic music. This would allow to construct a template from different sound sources independent of the recording to be analyzed allowing more general applications.

5. CONCLUSION

We have presented a drum transcription system for both monophonic and polyphonic music using partially fixed

Method	Metric	HH	BD	SD	Mean
PFNMF	P	0.918	0.886	0.825	0.876
	R	0.705	0.938	0.453	0.698
	F	0.797	0.911	0.585	**0.764**
APFNMF1	P	0.909	0.955	0.837	0.900
	R	0.682	0.927	0.473	0.694
	F	0.779	**0.940**	0.604	**0.774**
APFNMF2	P	0.928	0.914	0.854	0.898
	R	0.703	0.927	0.483	0.704
	F	0.799	0.920	0.617	**0.779**
Gillet et al. [10]	P	0.736	0.798	0.710	0.748
	R	0.865	0.700	0.642	0.735
	F	0.795	0.745	**0.674**	**0.738**
Paulus et al. [16]	P	0.838	0.941	0.750	0.806
	R	0.849	0.921	0.567	0.843
	F	**0.843**	0.930	0.645	**0.779**

Table 1. Evaluation results for ENST drum dataset *minus one* subset **without** accompaniments

Method	Metric	HH	BD	SD	Mean
PFNMF	P	0.902	0.714	0.684	0.766
	R	0.706	0.862	0.464	0.677
	F	0.792	0.781	0.552	**0.708**
APFNMF1	P	0.904	0.781	0.758	0.814
	R	0.679	0.856	0.45	0.661
	F	0.775	**0.816**	0.564	**0.719**
APFNMF2	P	0.908	0.774	0.726	0.802
	R	0.694	0.855	0.466	0.671
	F	0.786	0.812	0.567	**0.722**
Gillet et al. [10]	P	0.702	0.744	0.619	0.688
	R	0.818	0.653	0.552	0.674
	F	0.755	0.695	**0.583**	**0.678**
Paulus et al. [16]	P	0.847	0.802	0.663	0.770
	R	0.826	0.815	0.453	0.698
	F	**0.836**	0.808	0.538	**0.727**

Table 2. Evaluation results for ENST drum dataset *minus one* subset **with** accompaniments

NMF. This method uses a partially pre-trained dictionary matrix to decompose the target signal and to estimate the activation matrix. The initial test shows that increasing the r_{H} does help the algorithm to adapt to the harmonic components and increase the performance in polyphonic mixtures. The evaluation results show that this method is able to achieve average F-measures of 89.6% and 70.4% in monophonic and polyphonic music respectively for detecting 3 classes of drums.

The presented method has the following advantages: First, the fixed dictionary matrix in the model makes it easier to interpret the corresponding activation matrix for transcription tasks. Second, simultaneous sounds can be detected separately without the need of training extra classes. Third, adjustment of the parameter r_{H} allows the algorithm to adapt to different different types of polyphonic music. Fourth, evaluation results indicate a robustness against template mismatches, possibly allowing the application in situations with minimum prior knowledge. Last but not least, the approach only requires a few drum samples to train the dictionary matrix, and the evaluation results indicate that the performance is comparable with state-of-the art methods at lower algorithmic complexity.

Possible directions for future work are: investigation into means to iteratively adapt the template during the decomposition as a way of improving the current method. Furthermore, the automatic estimation of r_{H} for any given signal using a probabilistic approach similar to [19] might be a solution to rank selection. Finally, different penalty terms for the cost function, such as sparsity, temporal continuity [24], or rank r_{H} might be taken into account for better adjustment of the current method. To reach the goal of a complete drum transcription system for polyphonic music, however, more factors such as playing techniques and more drum classes still need to be addressed in the future.

6. REFERENCES

[1] David S Alves, Jouni Paulus, and José Fonseca. Drum transcription from multichannel recordings with non-negative matrix factorization. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, Glasgow, 2009.

[2] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri. Automatic music transcription: Challenges and future directions. *Journal of Intelligent Information Systems*, 41(3):407—434, December 2013.

[3] Emmanouil Benetos, Sebastian Ewert, and Tillman Weyde. Automatic transcription of pitched and unpitched sounds from polyphonic music. In *Proceedings of the International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2014.

[4] Christian Dittmar. Drum detection from polyphonic au-

dio via detailed analysis of the time frequency domain. In *Proceedings of the Music Information Retrieval Evaluation eXchange (MIREX)*, 2005.

[5] Christian Dittmar, I Fraunhofer, and D Gärtner. Real-time Transcription and Separation of Drum Recording Based on NMF Decomposition. In *Proceedings of the International Conference on Digital Audio Effects (DAFX)*, pages 1–8, 2014.

[6] Derry FitzGerald and Bob Lawlor. Sub-band independent subspace analysis for drum transcription. In *Proceedings of the International Conference on Digital Audio Effects (DAFX)*, Hamburg, 2002.

[7] Derry FitzGerald, Bob Lawlor, and Eugene Coyle. Drum transcription in the presence of pitched instruments using prior subspace analysis. In *Proceedings of the Irish Signals & Systems Conference (ISSC)*, Limerick, 2003.

[8] Olivier Gillet and Gaël Richard. Automatic transcription of drum loops. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages iv–269–iv–272 vol.4, May 2004. 00062.

[9] Olivier Gillet and Gaël Richard. ENST-Drums: an extensive audio-visual database for drum signals processing. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Victoria, 2006.

[10] Olivier Gillet and Gaël Richard. Transcription and separation of drum signals from polyphonic music. *Transactions on Audio, Speech, and Language Processing*, 16(3):529—540, March 2008.

[11] Marko Helen and Tuomas Virtanen. Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, Talya, 2005.

[12] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, 2000.

[13] Alexander Lerch. *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics*, volume 5. John Wiley & Sons, 2012.

[14] H Lindsay-Smith, S McDonald, and M Sandler. Drumkit Transcription via Convolutive NMF. In *Proceedings of the International Conference on Digital Audio Effects (DAFX)*, volume 1, pages 15–18, 2012.

[15] Arnaud Moreau and Arthur Flexer. Drum transcription in polyphonic music using non-negative matrix factorisation. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 353—354, 2007.

[16] Jouni Paulus and Anssi Klapuri. Drum Sound Detection in Polyphonic Music with Hidden Markov Models. *EURASIP Journal on Audio, Speech, and Music Processing*, 2009:1–9, 2009.

[17] Jouni Paulus and Tuomas Virtanen. Drum transcription with non-negative spectrogram factorisation. In *Proceedings of the 13th European Signal Processing Conference (EUSIPCO)*, page 4, Talya, 2005.

[18] Stanislaw A. Raczyski, Nobutaka Ono, and Shigeki Sagayama. Multipitch analysis with harmonic nonnegative matrix approximation. In *Proceedings of International Conference on Music Information Retrieval (ISMIR)*, 2007.

[19] Mikkel N. Schmidt and Morten Mø rup. Infinite nonnegative matrix factorization. In *European Signal Processing Conference (EUSIPCO)*, 2010.

[20] Paris Smaragdis and Judith C Brown. Non-negative matrix factorization for polyphonic music transcription. In *Proceedings of the Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA))*, New Paltz, 2003. IEEE.

[21] Paris Smaragdis, Bhiksha Raj, and Madhusudana Shashanka. Supervised and semi-supervised separation of sounds from single-channel mixtures. In *Proceedings of the 7th international conference on Independent component analysis and signal separation*, pages 414–421, 2007.

[22] Koen Tanghe, Sven Degroeve, and Bernard De Baets. An algorithm for detecting and labeling drum events in polyphonic music. In *Proceedings of the Music Information Retrieval Evaluation eXchange (MIREX)*, 2005.

[23] Lucas Thompson, Matthias Mauch, and Simon Dixon. Drum Transcription via Classification of Bar-Level Rhythmic Patterns. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2014.

[24] Tuomas Virtanen. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE transactions on audio, speech, and language processing*, 15(3):1066–1074, 2007.

[25] Gil Weinberg, Aparna Raman, and Trishul Mallikarjuna. Interactive jamming with shimon: a social robotic musician. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pages 233—234. ACM, 2009.

[26] Jiho Yoo, Minje Kim, Kyeongok Kang, and Seungjin Choi. Nonnegative matrix partial co-factorization for drum source separation. In *Proceedings of the International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 1942—1945, Dallas, 2010. IEEE.

[27] Kazuyoshi Yoshii, Masataka Goto, and Kazunori Komatani. Drumix: An audio player with real-time drum-part rearrangement functions for active music listening. *IPSJ Digital Courier*, 3:134—144, 2007.

[28] Kazuyoshi Yoshii, Masataka Goto, and Hiroshi G. Okuno. Automatic drum sound description for real-world music using template adaptation and matching methods. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Barcelona, 2004.

[29] Kazuyoshi Yoshii, Masataka Goto, and Hiroshi G Okuno. Drum sound recognition for polyphonic audio signals by adaptation and matching of spectrogram templates with harmonic structure suppression. *Transactions on Audio, Speech and Language Processing*, 15(1):333—345, January 2007.