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

Chih-Wei Wu, Alexander Lerch
Georgia Institute of Technology, Center for Music Technology
{cwu307, alexander.lerch}@gatech.edu

ABSTRACT

In this paper, a template adaptive drum transcription algorithm using partially fixed Non-negative Matrix Factorization (NMF) is presented. The proposed method detects percussive events in complex mixtures of music with a minimal training set. The algorithm decomposes the music signal into two dictionaries: a percussive dictionary initialized with pre-defined drum templates and a harmonic dictionary initialized with undefined entries. The harmonic dictionary is adapted to the non-percussive music content in a standard NMF procedure. The percussive dictionary is adapted to each individual signal in an iterative scheme: it is fixed during the decomposition process, and is updated based on the result of the previous convergence. Two template adaptation methods are proposed to provide more flexibility and robustness in the case of unknown data. The performance of the proposed system has been evaluated and compared to state of the art systems. The results show that template adaptation improves the transcription performance, and the detection accuracy is in the same range as more complex systems.

1. INTRODUCTION

Being one of the most intensively researched areas in Music Information Retrieval (MIR), automatic music transcription is often considered the core technology that would enable high-level representations of music signals with the potential of improving virtually any MIR system. A complete transcription system comprises many 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 mixtures 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 enables applications in active listening [27], music education, and interactive music performance.

This study explores the application of the popular transcription method NMF for drum transcription in polyphonic music. A standard NMF approach for music transcription decomposes a signal into a dictionary matrix, which consists of multiple pre-defined templates, and an activation matrix, which contains the activity of the corresponding templates. In this paper, we propose to transcribe drum events using a signal-adaptive method based on NMF.

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

Drum transcription is a task that requires instrument identification and onset detection for percussive sounds. To transcribe signals containing only drum sounds, standard approaches with a feature extractor and a subsequent classifier are able to produce results with high accuracy [11]. For most use cases, however, a drum transcription system is expected to work on mixtures of percussive and harmonic sound sources. Gillet and Richard propose to categorize automatic drum transcription systems into three categories:(i) *segment and classify* [4, 7, 22], for which the audio signal is segmented into a series of events using onset detection, and each event is classified based the extracted temporal or spectral features, (ii) *separate and detect* [1,6,15,17], which assumes music to be a superposition of different sound sources; by decomposing the signal into source templates with corresponding activation functions, the content can be transcribed by analyzing the activities of each template, and (iii) *match and adapt* [28, 29], identifying the drum events using a template matching method in which the templates are searched for the closest match and adapted in an iterative process.

Methods extended from these three types of approaches have been presented as well. 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, the method needs to train on multiple drum sequences, thus, a large dataset is needed to obtain 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 addi-

tional information of the bar locations and a large dictionary, which can be impractical in some use cases.

Among the above mentioned methods, the second type of approaches (*separate and detect*), frequently using NMF-related methods, has the advantage of joint estimation of multiple instruments and easy interpretation of the results. However, when NMF is applied to the task of drum transcription, 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 determining the rank r of the dictionary. This problem would be less severe when the sound sources of the target signal are given [14]. However, in most cases, this prior information is difficult to acquire. One solution 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 simultaneously, its robustness against unknown sources still needs to be evaluated.

Second, without any prior knowledge, it can be hard to identify the corresponding instrument of every template in the dictionary matrix [26]. This problem becomes more severe when the rank is selected too high or too low. Helen and Virtanen trained an SVM to separate drum templates from harmonic templates; the rank number was derived empirically during the factorization process [10]. The identified drum templates and their corresponding activation 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 templates in the polyphonic signal. This approach ensures that the drum templates in both dictionary matrices are estimated only from the drum track, resulting in proper isolation of the harmonic templates from the drum templates. Since their system aims at drum separation, they can work at higher ranks. For drum transcription, however, this approach is not directly applicable because the corresponding instrument of the templates 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 NMF cost function that will facilitate the different properties (e.g., the sparseness) in the resulting activation matrix. Virtanen proposed to use constraints for temporal continuity and sparseness [24]. He 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.

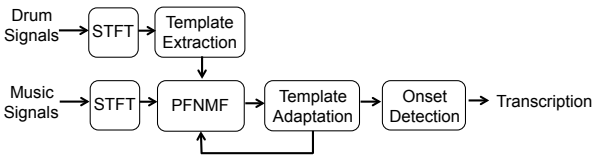

Figure 1. Flowchart of the drum transcription system

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 semi-adaptive bases during the NMF decomposition process [5]. However, their results indicate that the semi-adaptive process did not improve the performance of the transcription accuracy compared to fixed bases. Furthermore, no results were reported for the transcription performance in polyphonic mixtures.

3. METHOD

3.1 Implementation

Figure 1 shows the flow chart of the implemented system. The STFT of the signals will be calculated using a Hann window with a window length and a hop size of 2048 and 512, respectively, and the sample rate is 44.1 kHz. The resulting magnitude spectrogram is used as the input representation. A pre-trained dictionary matrix W_{D} will be constructed from the training set, which consists of isolated drum sounds. Next, the initial drum dictionary will be used in the partially fixed NMF (PFNMF) process and updated by the selected template adaptation methods described in Section 3.3. Finally, the activation matrix H_{D} is processed to determine the onset positions and their corresponding classes.

The initial drum dictionary matrix W_{D} is generated from a subset of the ENST dataset, which contains audio tracks of 5 to 6 single hits for each drum, performed by three drummers. For every drum class, one track per drummer is collected as training data. The onset position of these single hits was determined using the annotated ground truth. The template spectrum is a median spectrum of all individual events of one drum class in the training set. 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 a signal-adaptive threshold for peak picking [13]. In this paper, the window length and the offset coefficient λ of the median adaptive threshold are set to be 0.1 s and 0.12 for every track. The Matlab implementation of the presented system is available online. [1]

[1] https://github.com/cwu307/NmfDrumToolbox

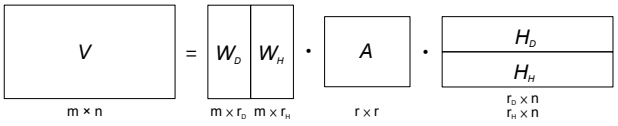

Figure 2. Illustration of the factorization process. W: dictionary matrix, H: activation matrix; Subscript $_{\mathrm{D}}$: drum, subscript $_{\mathrm{H}}$: harmonic components. A is the weighting matrix.

3.2 Algorithm Description

The basic concept of NMF can be expressed as $V \approx WH$ with non-negativity constraints, in which V is a $m \times n$ matrix, W is a $m \times r$ dictionary matrix, and H is a $r \times n$ activation matrix, with r being the rank of the NMF decomposition. In most audio applications, V is the spectrogram with m frequency bins and n frames, 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 signal adaptive method 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 slightly different from the target source, (ii) we propose to use a small dictionary matrix which is both efficient and easily interpretable, and (iii) the proposed method is able to adapt to different content in the polyphonic mixtures.

PFNMF [25] is a method inspired by [26] for drum transcription task. Figure 2 visualizes the concept: 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, the algorithm initializes the matrix W_{D} with drum templates and does not modify it during the factorization process. The matrices W_{H}, H_{H}, and H_{D} are initialized randomly. The rank r_{D} of W_{D} and H_{D} depends on the number of templates (i.e., instruments) provided, and the rank r_{H} can be arbitrarily chosen. The total rank $r = r_{\mathrm{D}} + r_{\mathrm{H}}$. A is a $r \times r$ diagonal weighting matrix, which contains weighting coefficients for every template to balance the drum and harmonic dictionaries in the NMF cost function (as discussed in Section 4.3.1). In our experiment, the coefficients are set to be $\alpha = (r_{\mathrm{D}} + r_{\mathrm{H}})/r_{\mathrm{D}}$ for each drum template and $\beta = r_{\mathrm{H}}/(r_{\mathrm{D}} + r_{\mathrm{H}})$ for each harmonic template. This setting is to increase the weighting of drum templates and slightly decrease the weighting of harmonic templates as r_{H} becomes larger. When $r_{\mathrm{H}} = 0$, the algorithm reduces to the original NMF.

The distance measure used is KL-divergence, in which $D_{\mathrm{KL}}(x \mid y) = x \cdot \log(x/y) + (y - x)$. The NMF 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):

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

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

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

To summarize, the presented method before template adaptation 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}.

3.3 Template Adaptation

Previous approaches to include template adaptation in drum transcription process can be found in [5, 29]. These approaches usually start with seed templates and gradually adapt them to the optimal templates. In this paper, we propose two methods for template adaptation with PFNMF. Both methods have the same criterion to stop iterating when the error between two consecutive iterations changes by less than 0.1% or the number of iterations exceeds 20. However, the adaptation process typically converges after 5–10 iterations.

3.3.1 Method 1: Complementary Update

In the first method (referred to as AM1), the drum dictionary W_{D} is updated based on the cross-correlation between the activations H_{H} and of each individual drum in H_{D}. PFNMF starts by randomly initializing a W_{H} with rank r_{H}. Although W_{H} tends to adapt to the harmonic content, it may still contain entries that belong to percussive instruments due to a mismatch between the initialized drum templates and the target sources. This will result in cross-talk (simultaneous activation) between H_{H} and H_{D} and generate a less pronounced activation. However, these harmonic templates 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)

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

where x and y represent different activation vectors, and n is the number of samples in the activation vectors. A threshold ρ_{thres} is defined for identification of related 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. Since a low ρ_{thres} can introduce too much adaptation and vice versa, a ρ_{thres} = 0.5 is chosen heuristically. The amount of adaptation also depends on the coefficient $\gamma = \frac{1}{2^k}$, which decreases as iteration number k increases.

$$W'_{\mathrm{D}} = (1-\gamma)W_{\mathrm{D}} + \gamma\frac{1}{S}\sum_{i=1}^{S}\left(\rho^{(i)}W_{\mathrm{H}}^{(i)}\right) \qquad (6)$$

3.3.2 Method 2: Alternate Update

In the second method (referred to as AM2), the drum template W_{D} is adapted by alternatively fixing W_{D} and H_{D} during the decomposition process. The adaptation process starts by fixing W_{D}, and PFNMF will try to fit the 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 detected activation H_{D}. The update rule for W_{D} is shown in Eq. (7).

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

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 drum dataset [8]. This dataset 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 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 [9]. The accompaniments are mixed with their corresponding drum tracks using a scaling factor of 1/3 and 2/3 in order to reproduce the evaluation settings as used in [16].

The second 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 for training. However, in our experiments, the isolated sounds are not used.

4.2 Evaluation Procedure

We evaluate the proposed system for both monophonic (drum only) and polyphonic mixtures. The same set of audio tracks is used with and 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, and complete mixtures from the third drummer are used to test the system. The process repeats three times to test every drummer in the dataset. This process is the same as described in [16], and 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 for each drum class. This is similar to the real-world use case, where the users may have access only to a limited number of training samples.

The evaluation metrics follow the standard calculation of the precision (P), recall (R), and F-measure (F). To be consistent with [9], an onset is considered to be a match with the ground truth if the time deviation in between is less or equal to 50 ms. It should be noted that some authors use more restrictive settings, compare e.g. the 30 ms as used in [16].

4.3 Evaluation Results

4.3.1 Rank Independence

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 polyphonic signals with and without a weighting matrix. As shown in Figure 3, a general trend of decreasing performance can be observed when $r_{\mathrm{H}} > 5$ without a weighting matrix. With a weighting matrix, however, the performance slightly increases for both HH and SD, and slightly decreases for BD as the r_{H} increases. The results demonstrate the robustness of the proposed system against the rank selection when a weighting matrix is introduced.

By increasing the rank r_{H}, a larger W_{H} will be initialized to better adapt to the target signal, however, this unbalanced increase in templates would also decrease the weight of the drum templates in the optimization process, thus reducing the impact of the percussive templates on the NMF cost function. This effect is reduced by the weighting matrix A which balances the weights between drum and harmonic templates.

4.3.2 Threshold Selection

The transcription results can be obtained after applying onset detection on each drum activation (see Section 3.1). 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 offset coefficient λ on IDMT-SMT-Drums dataset is shown in Figure 4. A general trend of parabolic curve can be observed. This is in agreement with the findings of Dittmar et al. [5]. One major difference is that in most regions of the curve, both AM1 and AM2 outperform

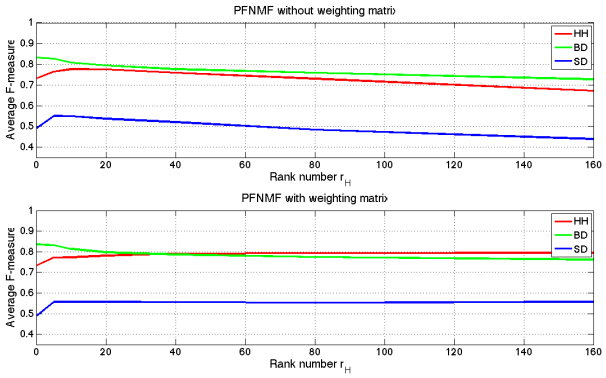

Figure 3. Average F-measure versus harmonic rank r_{H} in (Top) without weighting matrix (Bottom) with weighting matrix

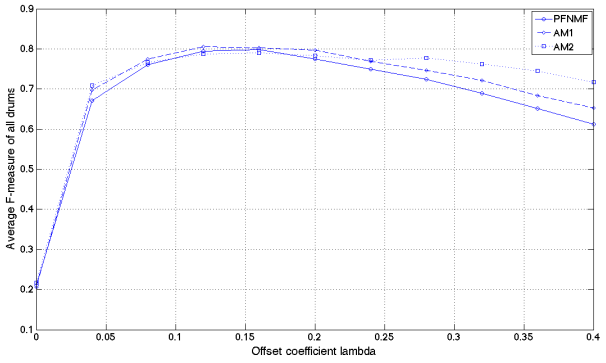

Figure 4. Evaluation results for IDMT-SMT-Drums dataset using (a) PFNMF (Solid circle) (b) AM1 (Dash diamond)(c) AM2 (Dotted square)

PFNMF. This verifies that template adaptation process does help the algorithm in the case of the unknown sounds (templates and the test signals are from two different datasets). The overall performance is slightly lower than [5] due to the mismatch in templates and target signals. However, the F-measures of AM1 can reach 74.0%, 93.2% and 73.4% for HH, BD, SD, respectively, which indicates the applicability of the proposed method across datasets.

4.3.3 Results

Table 1 shows the evaluation results on ENST drum dataset *minus one* subset *without accompaniments*. For comparison, we also list the results of Gillet et al. [9] and Paulus et al. [16]. All the compared methods use the same dataset with identical mixing settings (1/3 for accompaniments and 2/3 for drum tracks). Since the target signals contain only drum sounds, the rank r_{H} can be small. 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 of 77.9% using AM2. This result is higher than the 73.8% reported in [9], and at the same level as reported in [16].

Table 2 shows the evaluation results on ENST drum dataset *minus one* subset *with accompaniments*. The compared methods are the same as described above. Since the target signals contain both percussive and harmonic parts, r_{H} is set to 50. The results show that our proposed method achieves an average F-measure = 72.2% using AM2, which is higher than 67.8% [9] and at a similar range as the 72.7%, reported in [16].

In general, our methods outperform [9] for all instruments except the snare drum. The possible reason is that many of the playing technique variations are applied to the snare (e.g., ghost note, rim shot, with/without snare on), and a single snare drum template cannot cover all the possibilities even with template adaptation. In the polyphonic dataset, our proposed methods perform better on BD and SD but slightly worse on HH compared to the HMM based method [16]. Since Paulus et al. [16] trained and tested their system using the same ENST dataset, the music played by all three drummers is highly correlated because of the same accompaniments used. This may lead to a tendency of overfitting the transition probability in this dataset. For all the methods, the performances drop from the monophonic to the polyphonic dataset, especially for BD and SD. This is an unsurprising trend. The less prominent decrease for HH might be due to the fact that the typical frequency range of HH is more separated from other instruments than BD and SD, thus is more robust against the presence of tonal sounds. In the case of template adaptation, a general trend of increase in precision and decrease in recall can be observed. One explanation is that once a better representation of the drum templates is found, the system might become more selective, leading toward a reduction in both false positives and true positives.

AM1 seems to perform better than AM2 on BD in both monophonic and polyphonic dataset. One possible explanation is that bass drum usually appears on the downbeats, which tends to have higher correlation with other entries in harmonic activation matrix. This means BD has a higher chance of being adapted to better templates using AM1. AM2 uses a more generalized adaptation process and performs better on HH and SD. However, it is more computationally demanding since it adapts the templates constantly, whereas AM1 only adapts when the correlation is above the threshold. To sum up, both template adaptation methods perform at the similar level, and the best fit of either method for specific types of music still needs to be investigated.

5. CONCLUSION

We have presented a drum transcription system for both monophonic and polyphonic music using partially fixed NMF with template adaptation. The system is robust against rank changes, and the evaluation results show that the two presented template adaptation methods improve the precision of the system, leading toward better performance. The proposed method is able to achieve average F-measures of 77.9% and 72.2% in monophonic and polyphonic music respectively for detecting 3 classes of drums.

The presented method has the following advantages:

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**
AM1	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**
AM2	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. [9]	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**
AM1	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**
AM2	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. [9]	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

First, the system only requires a few training samples for template extraction, and these templates can adapt toward the target sources gradually. This makes the system more applicably to the real world use case. Second, adjustment of the parameter r_{H} allows the algorithm to work with polyphonic music, and the use of a weighting matrix prevents the performance from dropping as r_{H} increases. Third, the cross-dataset evaluation results indicate a robustness against template mismatches, possibly allowing the application in situations with minimum prior knowledge. Last but not least, the evaluation results indicate that the F-measure of the proposed methods is at the same level as state-of-the art systems with a lower model complexity.

Possible directions for future work include the automatic estimation of r_{H} for any given signal using a probabilistic approach similar to [19]; this might be a solution for the system to optimally select the rank. Furthermore, a more detailed analysis of playing techniques might be necessary toward a more complete drum transcription system. Finally, different penalty terms for the NMF cost function, such as sparsity, temporal continuity [24], or rank r_{H} might be taken into account for better adjustment of the current method.

6. REFERENCES

[1] David S Alves, Jouni Paulus, and José Fonseca. Drum transcription from multichannel recordings with non-negative matrix factorization. In *Proc. 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 *Proc. of the International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2014.

[4] Christian Dittmar. Drum detection from polyphonic audio via detailed analysis of the time frequency domain. In *Proc. of the Music Information Retrieval Evaluation eXchange (MIREX)*, 2005.

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

[6] Derry FitzGerald, Bob Lawlor, and Eugene Coyle. Drum transcription in the presence of pitched instruments using prior subspace analysis. In *Proc. of the*

Irish Signals & Systems Conference (ISSC), Limerick, 2003.

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

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

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

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

[11] Perfecto Herrera, Amaury Dehamel, and Fabien Gouyon. Automatic labeling of unpitched percussion sounds. In *Proc. of the 114th Audio Engineering Society Convention*. AES, March 2003.

[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*. John Wiley & Sons, 2012.

[14] Henry Lindsay-Smith, Skot McDonald, and Mark Sandler. Drumkit Transcription via Convolutive NMF. In *Proc. of the International Conference on Digital Audio Effects (DAFX)*, pages 15–18, 2012.

[15] Arnaud Moreau and Arthur Flexer. Drum transcription in polyphonic music using non-negative matrix factorisation. In *Proc. 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*, pages 1–9, 2009.

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

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

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

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

[21] Paris Smaragdis, Bhiksha Raj, and Madhusudana Shashanka. Supervised and semi-supervised separation of sounds from single-channel mixtures. In *Proc. 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 *Proc. 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 *Proc. 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] Chih-Wei Wu and Alexander Lerch. Drum Transcription using Partially Fixed Non-Negative Matrix Factorization. In *Proc. of the European Signal Processing Conference (EUSIPCO)*, Nice, France, 2015.

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

[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 *Proc. 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. *IEEE transactions on Audio, Speech and Language Processing*, 15(1):333—345, January 2007.