Timbre-sensitive drum transcription using non-negative matrix factorization methods

Shayan Gupta
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
shayang@andrew.cmu.edu

Jaxter Kim
School of Music
Carnegie Mellon University
Pittsburgh, PA 15213
jaxterk@andrew.cmu.edu

Dong-Hyun Lee
School of Music
Carnegie Mellon University
Pittsburgh, PA 15213
donghyu3@andrew.cmu.edu

Davis Polito
School of Music
Carnegie Mellon University
Pittsburgh, PA 15213
dpolito@andrew.cmu.edu

Abstract

Automatic drum transcription is the process of converting drum audio into a musical notation. We use non-negative matrix factorization to split an input audio drum loop into its component parts. The model trains the basis vectors and factors on a set of training samples using the multiplicative update rules derived from KL divergence. It then uses the same rules to update the factors on the input audio. We then split the basis vectors and recombine them with the factors in order to produce a set of output spectrograms, which can easily be turned into output audio tracks. From here, we use onset detection to convert the audio into a musically meaningful transcription. Our median accuracy with this method was regularly above 0.6 for all the types of drums tested.

1 Problem statement

The current research methods for ADT are primarily concerned with analysis of drum tracks and their rhythmic properties of time-keeping as a more general way to determine features such as tempo and genre. They explore drum tracks in the context of larger polyphonic mixtures, such as source separation and event detection. Using our system, experienced drummers could communicate with a computer by simply playing an acoustic drum set rather than use dedicated controllers, which can be expensive, unexpressive, and unfamiliar. We also explore potential avenues for improving the accuracy and reliability of ADT systems, for example by improving the feature extraction.

2 Literature Research

2.1 Approach 1: Modeling Timbre Distinction on the Auditory System

Timbre can be characterized by a relatively broad nature of features that are perceived by the human auditory system. These cues include spectral shape, temporal modulations in spectrum, and onset dynamics. The auditory system applies a cascade of processing layers to more finely distinguish on different configurations of the extracted features with each layer [7]. Timbre distinction, in a fashion similar to the auditory system, can be accomplished by cortical multiscale decomposition

Preprint. Under review.

along the spectral dimension, an algorithm that incorporates the cepstrum with local and complex phase information. These cortical representations can then be fed into a non-supervised hierarchical clustering of musical timbres according to their spectral similarities using the Linde-Buzo-Gray (LBG) algorithm or other Tree-Structured Vector Quantization algorithms [6].

2.2 Approach 2: Automatic Transcription with NMF

A deep learning model takes in two lists: the filenames of the drum sample dataset and a one-hot-encoded list outlining the class of drum hits. The drums are classified as either a kick, snare, or hat, but we plan to use more categories. The audio samples correspond to the expected input, and the one-hot list corresponds to the expected output. In the feature extraction step, the audio data is converted into a 3D spectrogram matrix using an STFT, where one dimension is the number of samples, and the other two correspond to the maximum STFT size. The input data is processed to offset the mean of each vector to zero and scale each vector to have a unit variance. A training/testing data split of 75% to 25% was used, with the training and testing data both have proportional amounts of each drum class [5]. NMF with a gamma mixture model has been used to classify drum beats over periods of time. Unlike classification-based approaches to drum detection, this approach also provides amplitude information. This model also essentially detects beat and aligns the detected drums with the beat grid. This beat detection can be extended to beat tracking, where the algorithm picks the most reliable period hypothesis from multiple measurements and can label the classified drum hits with their respective beats [1].

2.3 Approach 3: Drum Analysis Using NN, KNN, and SVM

Pre-processing was done with window sizes of 512, 1024, and 2048 samples. The following time-domain features we used: Temporal Centroid, Attack Time, RMS, Zero Crossing Rate, Subband Analysis.

Spectral domain features used included: Spectral Flux, Spectral Rolloff, Spectral Centroid, Spectral Kurtosis, Spectral Skewness, Mel-Frequency Cepstrum Coefficients, Linear Predictive Coding Coefficients, Energy in nine wavelet bands, Variance from the mean in each wavelet band. The seven classes, each a different snare drum stroke were: Rimshot, Brush stroke, Center, Near center, Halfway, Near edge and edge. For classification, a simple dense linear layer neural network, as well as KNN and SVM, were compared. The classifications were performed on 4 different subsets of classes: All classes; Center, Near Center, Halfway, Near Edge Edge (only 5); Rimshot, Brush, Edge (RBE); and Center, Halfway, Edge (CHE). KNN achieved above 90% accuracy for all subsets. SVM main performed poorly with the larger sets of classes using time-domain features. The neural network had the best classification over all subsets, notably with 99.8% accuracy on the RBE subset. The time-domain features performed nearly as well as the full feature set when classifying the groups with only three classes but significantly less effective when classifying multiple classes. Window sizing did not contribute significantly [4].

3 Dataset

We are currently using IDMT-SMT-Drums [2] as our main dataset. It contains 608 WAV files which is about 2:10 hours long combined. There are 104 polyphonic drum set recordings (drum loops) and each of them has 3 training files: kick drum, snare drum, and hi-hat. It comes from three different sources: real-world recordings on acoustic drum sets, drum sample libraries, and drum synthesizers. All of the drum loop comes with the transcription annotation (onset and offset marking). Below is a more detailed description of the methodology in which the data was collected:

In order to capture the individual characteristics, the audio data was recorded such that the drums were hit separately with varying velocity to produce the varied sounds that could be mixed. For standardized recording, we made sure to use a dataset where the recording microphone was at a fixed distance from the sound source. The recordings were made with 10 different drum kits, which allowed for the capture of drums with different sizes and various composite materials. The kick drum size ranges from 18 - 24 inches in diameter, and depths of 16 - 22 inches. Kick drum composite materials recorded consisted of birch, mahogany or maple. The snare drums were 14 inches in diameter and 6.5 inch in depth and kick drums of different composite materials - metal, wood or

acrylic - were recorded. The sizes for hi-hat ranged from 13 to 15 inches in diameter. An additional subset was generated using sample-based drum sets from the BFD6 plug-in. A third subset featuring purely synthetic drum kits was generated using Steinberg's Groove Agent7 plug-in. The onsets were transcribed manually by an experienced drummer using the software Sonic Visualiser.

4 Methodology

4.1 Non-negative Matrix Factorization

Creating a suitable basis for the drum sound is an important requirement for the NMF method. We implement the basic NMF model with Kullback-Leibler divergence to the convolutive version of NMF [3] to see the importance of the impulse response of the drum sound.

$$M \approx BH$$

$$D\left(M \| BH\right) = \sum_{ij} \left(M_{ij} \log \left(\frac{M_{ij}}{(BH)_{ij}} \right) - M_{ij} + (BH)_{ij} \right) \tag{1}$$

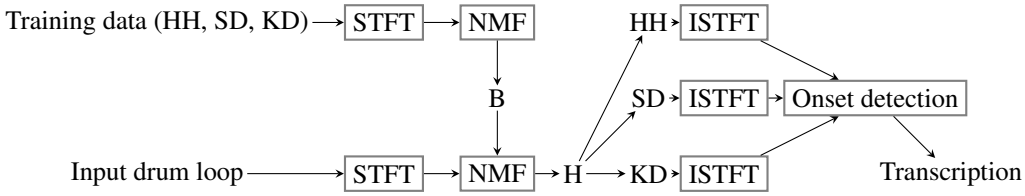
$$B = B \otimes \frac{\left(\frac{M}{BH}\right) H^{\top}}{1_{p \times q} H^{\top}} \tag{2}$$

$$H = H \otimes \frac{B^{\top} \left(\frac{M}{BH}\right)}{B^{\top} 1_{p \times q}} \tag{3}$$

where:

M is a $p \times q$ non negative matrix
B is a $p \times k$ non-negative matrix
H is a $k \times q$ non-negative matrix

Using a set of training samples (preprocessed with an STFT), we build a set of vectors that will serve as the basis B for the factorization of the drum audio. We update the values of B and H during training by minimizing the KL divergence (1) using the multiplicative update rules (2, 3) for each of the training sets. Then, once we find these values of B and H, we fix B, introduce the mixed audio file as M, and update H using (3). From these values of B and H, we can reconstruct each individual track separately, post-process with an iSTFT, and use onset detection to create a transcription.



4.2 Onset Detection

To perform a transcription on the separated sound, we came up with a simple onset detection algorithm. It works by locating note onset events by picking peaks in an onset strength envelope. Since drum sounds have a fast attack and decay, simply picking peaks was enough for the transcription. However, we performed post-processing on the separated drum sound to eliminate some residual noise from the separation. We used root mean square (RMS) amplitude and it worked well since residual sounds were low level signals compared to the much louder actual drum sounds. Applying RMS amplitude drastically reduced the false negative rate of the transcription.

4.3 Time-domain

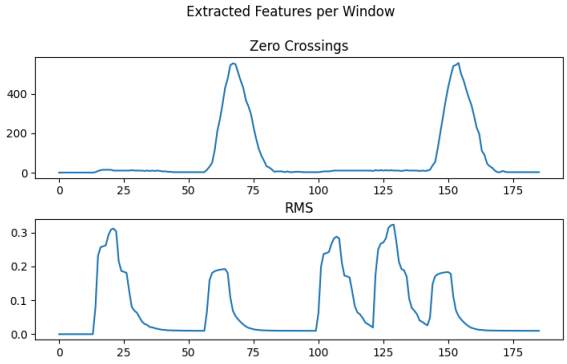
We have seen in the literature that extracted time domain features could potentially be useful in gestural classification. We wanted to find out whether time-domain based classification likewise

produces acceptable results, and if combining this approach with our working NMF approach could improve robustness.

The existing literature recommends the following time-domain features for classification: Temporal Centroid, Attack Time, RMS amplitude, Zero Crossing Rate, and Subband Analysis. Using similar frame parameters as we did for NMF, the above time-domain features are extracted from each frame to create a matrix of features.

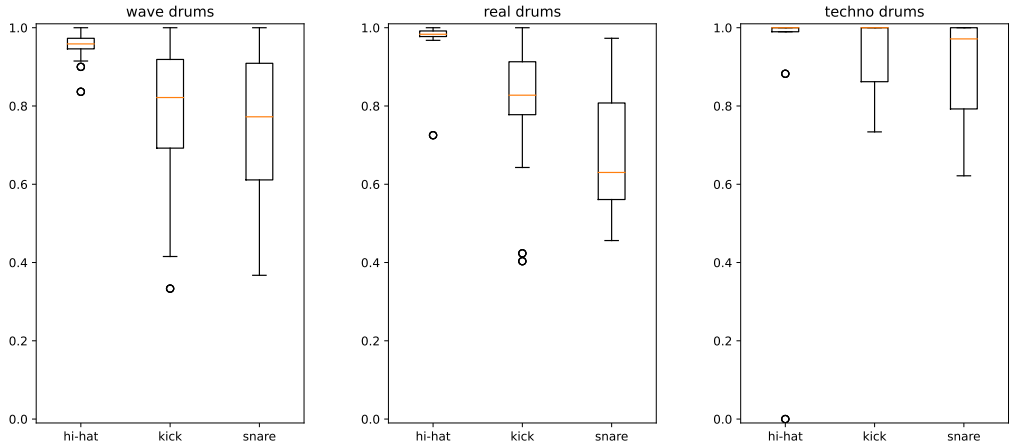
We only experimented with Zero Crossing Rate and RMS amplitude features. The sample audio consists of various drum hits in a sequence. The result of the extraction so far was a 2xN matrix, where there are N frames. The following figure shows each of these extracted features over the course of all of the frames.

We did not end up incorporating the time domain features since we were not able to revert the features back to a format compatible with the original sound after performing the NMF. Because the NMF method uses additive features to perform the separation, frequency domain features are better suited for NMF than time domain features.

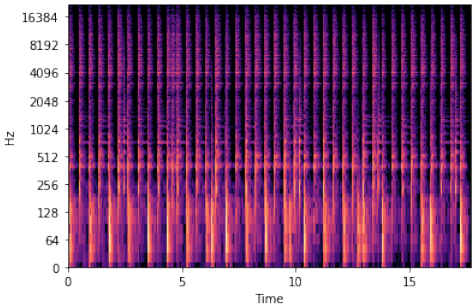


5 Results

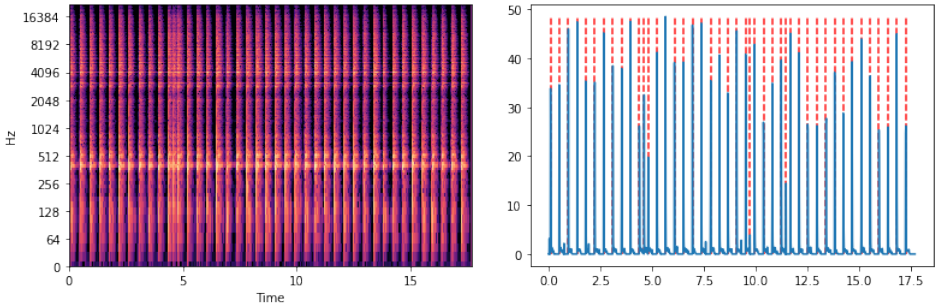
The accuracy of our method was about what we expected. The snare drum perfomed the worst overall in all three categories, likely because of its spectral collision with the other two drum types and high bandwidth. Qualitatively, all three extracted signals sound very well-isolated, with only a little bit of artifacting and noise. Our method did not perform noticably better or worse than existing methods.



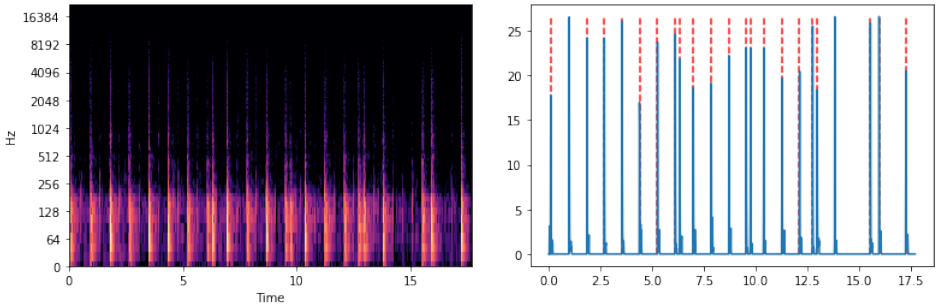
Mixed audio (live recorded drums)



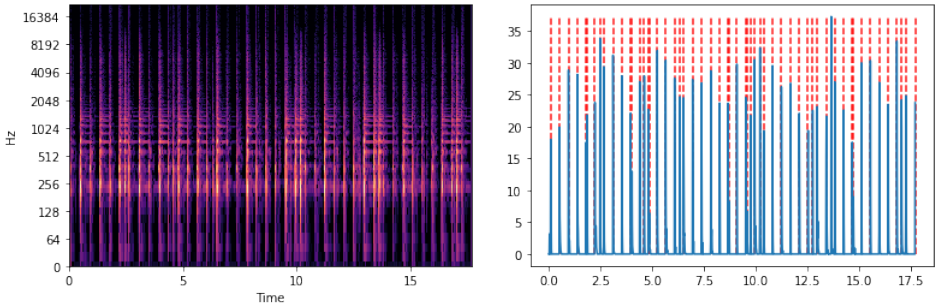
Separated hi-hat (Accuracy: 0.9677)



Separated kick drum (Accuracy: 0.7857)



Separated snare drum (Accuracy: 0.9730)



It is very possible that our results are overly optimistic. The main potential concern is that all of the input drum audio was completely isolated. There were no non-drum sounds in the inputs, and it is very rare for drums to be so isolated in real-life applications. In addition, something like an echoey environment could have a drastic impact on the accuracy of our model. To alleviate these issues, we could potentially perform an additional preprocessing step, such as a more generalized source separation, or de-reverberation.

References

[1] E. Battenberg. *Techniques for Machine Understanding of Live Drum Performances*. PhD thesis, EECS Department, University of California, Berkeley, Dec 2012. URL `http://www2.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-250.html`.

[2] C. Dittmar and D. Gärtner. Real-time transcription and separation of drum recordings based on NMF decomposition. *Proc. Intl. Conf. on Digital Audio Effects (DAFx), Erlangen, Germany*, pages 187–194, September 2014.

[3] C. Dittmar and M. Müller. Reverse engineering the amen break – score-informed separation and restoration applied to drum recordings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(9):1531–1543, 2016.

[4] A. T. et. al. Retrieval of percussion gestures using timbre classification techniques. *Proc. Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2004.

[5] F. Quiros-Corella. Automatic drum transcription (ADT) using ConvNets. *Medium*, 2021. URL `https://fabisqc0207.medium.com/automatic-drum-transcription-adt-using-convnets-175f3efe4b0c`.

[6] S. Shamma. Encoding sound timbre in the auditory system. *IETE Journal of Research*, pages 145–156, 2003.

[7] S. Town and J. Bizley. Neural and behavioral investigations into timbre perception. *Frontiers in Systems Neuroscience*, page 88, 2003.