Voice Removal of EDM

Kuo-Wei Lee
Electrical and Computer Engineering
Carnegie Mellon University
kuoweil@andrew.cmu.edu

Chen-Yu Tsai
Electrical and Computer Engineering
Carnegie Mellon University
chenyut@andrew.cmu.edu

1 Introduction

At the beginning of this project, we aimed to create a system that mimics DJ operations so as to remix different tracks of electronic dance music(EDM). After surveying current solutions, we found that there are some defects that would hurt the performance of the outcome such that it sounds not satisfactory. To further enhance the performance and quality, we decided to switch to finding a way of removing vocal in the tracks. Vocal separation and removal is a well-known problem that people are eager to solve. A proper design of the algorithm should separate a given sound track as foreground and background such that the foreground only contains vocal of the song while other musical components are extracted as the background.

In this work, we implemented an algorithm that relies on the underlying structure of EDM in nature to extract foreground and background from the mixture. We showed that the assumption we used to derive the algorithm is better established on music that vocal is more natural and not tuned while EDM creators tend to apply sound effects that would potentially weaken the assumption. In the end, we proposed and briefly described different methods that could avoid the hand-made assumption such that the algorithm would be more reliable and robust when applying on different genre of music.

2 Related Work

There have been a number of approaches applied to the problem of separating the foreground(vocal) from the background.

A large portion of the research works focus on applying supervised learning techniques to this problem. In this fashion, the researchers would need to collect vocal tracks and instrumental tracks of the same song and label them. A well-known dataset is MUSDB[1] that consists of 150 full-length songs of different genres along with their isolated drums, bass, vocals and others stems. Ozerov et al[2] proposed the use of Bayesian methods to model both the vocal and instrumental part of the songs and apply Wiener filters to separate them. Vembu et al[3] proposed a singing voice separation system based on non-negative matrix factorisation (NMF). They first created classifiers to discriminate automatically between sections of the music where there are no vocals then used NMF to cluster the basis functions into vocal and nonvocal basis functions. State-of-the-art researches apply deep learning techniques on this problem and the performance surpasses non-deep learning methods. Hennequin et al[4] created a tool called Spleeter which can be easily accessed using Python PIP and serves as a part of many professional and commercial music editor such as iZotope RX9[5].

In this paper, we focused on the method that relies on an observation of common music and doesn't need to collect training dataset. Z. Rafii et al.[6] proposed a series of methods that are based on the repetition nature of music. In the assumption, the background of the track(instrumental) are more repetitive than foreground(vocal). We think that this is a more obvious phenomenon in EDM. Therefore, we were curious about how far this method could go without applying supervised learning and collecting paralleled training data.

3 Dataset

We collect 30+ EDM Music that are suitable for our task from Youtube. The selected songs mainly consist of Drum and Bass. More specifically, we handpicked some famous remix with strong beat, such as Alan Walker and MARSHMELLO. Also, as a reference, we also collected some non-EDM music that also include vocal parts.

4 Method

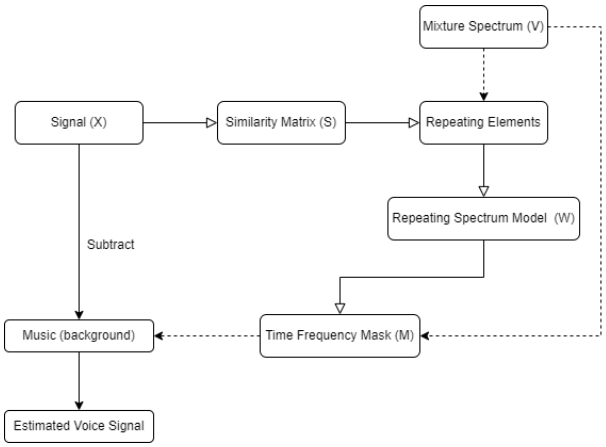

Figure 1: Flow chart of the algorithm

Given a mixture signal x, we use *librosa[7]* to obtain its Short-Time Fourier Transform (STFT) X and then take the magnitude spectrum V. To determine the repetition(similarity) of any segments in the track, we introduced the similarity matrix S. S can be calculated as below:

$$S(j_a, j_b) = \frac{\Sigma_{i=1}^{n} V(i, j_a)(i, j_b)}{\sqrt{\Sigma_{i=1}^{n} V(i, j_a)^2}\sqrt{\Sigma_{i=1}^{n} V(i, j_b)^2}}$$

where j_a and j_b are frames of the spectrum V at time a and b, n is the #of frequency channels. In other words, we are calculating the cosine similarity of any pair of j_a and j_b in V.
After getting S, for every frame j, we can produce a vector J that is the sorted indices of which its most similar frame. Here, we introduced a parameter d, the minimum allowed (time) distance between two consecutive repeating frames deemed to be similar enough to indicate a repeating element to avoid high similarity without representing new instances of the same structural elements. Once we get J, we can obtain repeating spectrum model W such that:

$$\begin{aligned} W(i,j) &= median\{V(i, J_j(l))\}, l \in [1, k] \\ J_j &= [j_1, j_2, \dots j_k] = \text{indices of repeating frames} \\ k &= \text{max number of repeating frames} \\ i \in [1, n] &= \text{frequency channel index} \\ j \in [1, m] &= \text{time frame index} \end{aligned}$$

The rationale of picking the median of all similar frames for every frame is based on our assumption that the repeating structure of music is the background. Thus, choosing the median can smooth out the vocal part(if exists).

After W is calculated, we need to get W' by taking the minimum between W and V for every time-frequency bin. This is crucial since it ensures that $W \leq V$ such that we can obtain foreground $V - W$.

As the last step, we want to computer the mask M, such that:

$$M(i,j) = \frac{W(i,j)}{V(i,j)} \in [0,1]$$

We can see that, $W(i,j) \approx 1$ if that bin is identified as background; $W(i,j) \approx 0$ if that bin is identified as foreground.

In the end, we did something different than the original paper: instead of hard-masking, we applied soft-masking of M on X to derive the results. We obtain $M' = 1 - M$ as the mask of foreground and set:

$$M_{background} = \frac{M^p}{M^p + M'^p}$$
$$M_{foreground} = \frac{M'^p}{M'^p + M^p}, \text{ where } p \text{ is the power parameter}$$

This allows us to better control the mixing ratio and the quality of output result. Finally, we apply $M_{background}$ on X to obtain background music and vice versa.

5 Results

Evaluation Metrics
We used three common metrics for our evaluation - SDR, SIR and SAR. SDR stands for Source to Deistortion Ratio, where

$$SDR = \log \frac{||s_{target}||^2}{||e_{iterf} + e_{noist} + e_{artif}||^2}$$

SIR stands for: Sources to Interfernece Ratio, where

$$SIR = \log \frac{||s_{target}||^2}{||e_{iterf}||^2}$$

SAR stands for: Sources to Artifacts Ratio, where

$$SIR = \log \frac{||s_{target} + e_{interf} + e_{noise}||^2}{||e_{artif}||^2}$$

There are many different evaluation metrics (e.g. Signal-to-Noise Ratio (SNR)), however, we decided to apply SDR in our project. [8] Since SDR is usually considered an overall measure of how good a signal sounds. Note that for all the metrics, higher scores indicate better results.

Here we do not discuss the SIR score as it usually interpreted as the amount of "other" sources can be heard in a source estimate. Our source signal and our separation did not contain source other than vocal and background music. Therefore, all of the SIR would have "inf" value. Also, the SDR actually will have the value as SAR due to the fact that we did not add any noise.

The equation acutally becomes

$$SDR = SAR = \log \frac{||s_{target}||^2}{||e_{artif}||^2}$$

	Yoasobi vocal	Yoasobi bg	Faded vocal	Faded bg
baseline	-5.68	-3.00	4.74	-1.23
proposed method	1.75	5.61	-2.30	4.59

Table 1: Source-to-Distortion Ration (SDR) results. Higher values are better. We compare the results of the baseline and the proposed method. The results showed that the using similarity matrix and repeating model (proposed method) performs better than the baseline.

	Yoasobi vocal	Yoasobi bg	Faded vocal	Faded bg	Alone vocal	Alone bg
SDR	1.75	5.61	-2.30	4.59	-5.95	9.03
SAR	1.75	5.61	-2.30	4.59	-5.95	9.03

Table 2: Source-to-Distortion Ration (SDR) results. Higher values are better. We compare the results of Non-EDM song and EDM songs. All of the background music performs well since we are able to catch the repeating pattern. However, the vocal in EDM musics have not only human voice but electrical sound effect. Therefore, it might perform slightly worse than the pure human vocal, e.g. the NON-EDM yoasobi.

Baseline & Comparison

We comparing our results to the 1) baseline which applying the baseband filter (300-3000hz) to the original song to filter out the vocal, and 2) existing vocal removal application that applied deep learning model. Since the existing vocal application can separate the vocal nearly perfect, we used the results as our ground truth (reference) and calculate the SDR.

Table 1. showed the comparison between our baseline and the proposed method. The results showed that extracting the repeating pattern is a better to separate vocal and background music. Table 2. showed our main contribution, the pros and cons of using repeating model to separate EDM and non-EDM musics.

We've shown that the repeated background music can be separated well under our method. This applies to either EDM or non-EDM music.

Note that we are using the "mir_eval" python package for evaluating the SDR and SAR. To be a bit more specific, we used "bss_eval_sources".

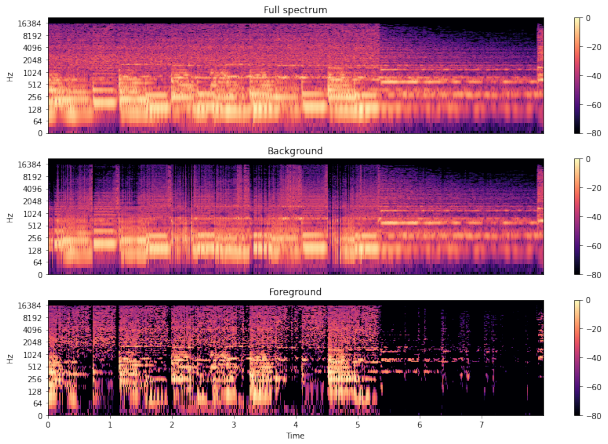

Figure 2: Alone

6 Discussion and Analysis

Our generated music clips sounds decent but perform normally according to our metrics. However, [8] some research papers suggests that the higher SDR may not imply better quality. The number of

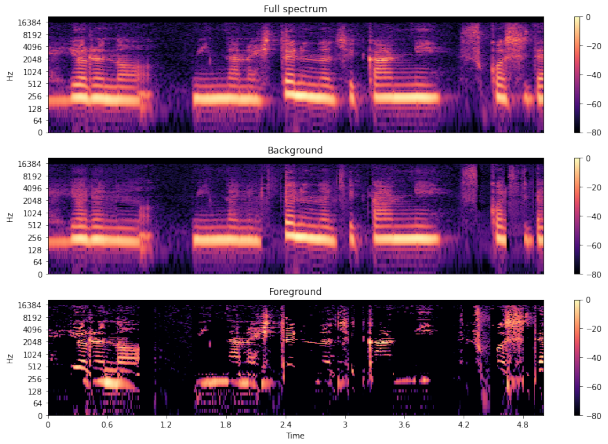

Figure 3: Yoasobi

SDR do imply the quality of the sound, but it might not apply to human's understanding of music. Figure 2. and Figure 3. shows the original Spectrum and our estimated background and foreground signal.

The vocal is not separated from the background clearly. We can still hear some background beats in the vocal part and also some vocal in the background part. We observe that the relatively digital vocal sound is harder than the human voice for our algorithm to separate. In other words, our algorithm performs better when the vocal is more natural and the background is simpler. Also, we notice that the music sounds distorted as well. But we can control it by configuring the ratio of separated sounds that form the final result yet it would take us time to fine tune the parameters.

In conclusion, the assumption that the repeating part of the music resembles background and non-repeating is the vocal might not hold well in EDM. In music that doesn't have too many instruments and sound effect, it works relatively better. A solution we have is to apply supervise learning. For such task, we can collect music that has separated vocal and instrumental tracks and use SVM or any supervised learning method to learn the representation of vocal and instrument so as to separate them. This doesn't require us to form assumptions manually since the algorithms should learn itself given the training data is sufficient.

References

[1] https://sigsep.github.io/datasets/musdb.html

[2] A. Ozerov, P. Phillipe, F. Bimbot, and R. Gribonval, Adaption of Bayesian models for single channel source separation and its application to voice/music separation in popular songs, IEEE Transactions on Audio Speech and Language Processing, 2007.

[3] S. Vembu and S. Baumann, Separation of vocals from polyphonic audio recordings, in Proc. Int. Symp. Music Inf. Retrieval (ISMIR05), 2005, pp. 337344.

[4] Hennequin et al., (2020). Spleeter: a fast and efficient music source separation tool with pre-trained models. Journal of Open Source Software,5(50), 2154. https://doi.org/10.21105/joss.02154

[5] https://www.izotope.com/en/shop/rx-9-standard.html

[6] Rafii, Zafar and Bryan Pardo. "Music/Voice Separation Using the Similarity Matrix." ISMIR (2012).

[7] https://librosa.org/doc/latest/index.html

[8] https://source-separation.github.io/tutorial/basics/evaluation.html

[9] https://www.edityouraudio.com/vocal-remover