Sound Source Separation and Transcription

Xuhui Wang

November 2020

Introduction

In this project, firstly, I introduce the cross-similarity matrix and self-similarity matrix for repetition detection and make the evaluation in the regular case and time stretching case. For harmonic/percussive sound source separation, I introduce the MFCCs and Chromagrams for timbre and harmony analysis respectively and give the evaluation using cross-similarity matrix and self-similarity matrix. Dynamic time warping is also discussed for measuring similarity between the different two temporal sequences. Then I use a deep learning method, spleeter, to perform the sound source separation and evaluate the performance in the cases of pitch shifting, monophonic pitch detection and kick and snare detection.

Repetition Detection

Cross-similarity and Self-similarity Matrices

(Note that, in this report, the wordings that describe the questions, algorithms and dataset are all from the course material by Dr. Shier and I will give the appropriate citation at the end of each corresponding paragraph. On the other hand, all figures, methods implementation, statistics, analysis, explanations and conclusions are all from my personal work.) In this part, as an example of the original audio clip, we use a 30 second clip of the famous song, "Smooth Criminal", by Michael Jackson. This clip has a repeating rhythm, phrase, or harmonic structure that we can identify in a self-similarity matrix.

Firstly, we repeat the 30 second recording to create a one minute recording and we plot the cross-similarity matrix for this new one minute long clip with the original 30 second recording. Then we describe how the repetition can be seen in a plot of the cross-similarity matrix. Specifically, we use MFCC features and the affinity mode.[1]

We show the cross-similarity matrix. The two most prominent structures in SSMs are block and paths.

- blocks represent homogeneous sections of music. These are darker sections, indicating higher similarity over a duration.
- paths represent repetition. This occurs when when a subsequence (melody, rhythmic pattern) is repeated, which shows up as a similarity.

From Figure 1, we can observe several long paths within some blocks. The dark block indicates a homogeneous section, and the dark diagonal line indicates that this section is pretty much a repetition. These dark lines correspond to the repeating section of the original audio in the refrain. I have plotted the vertical lines to indicate those sections.

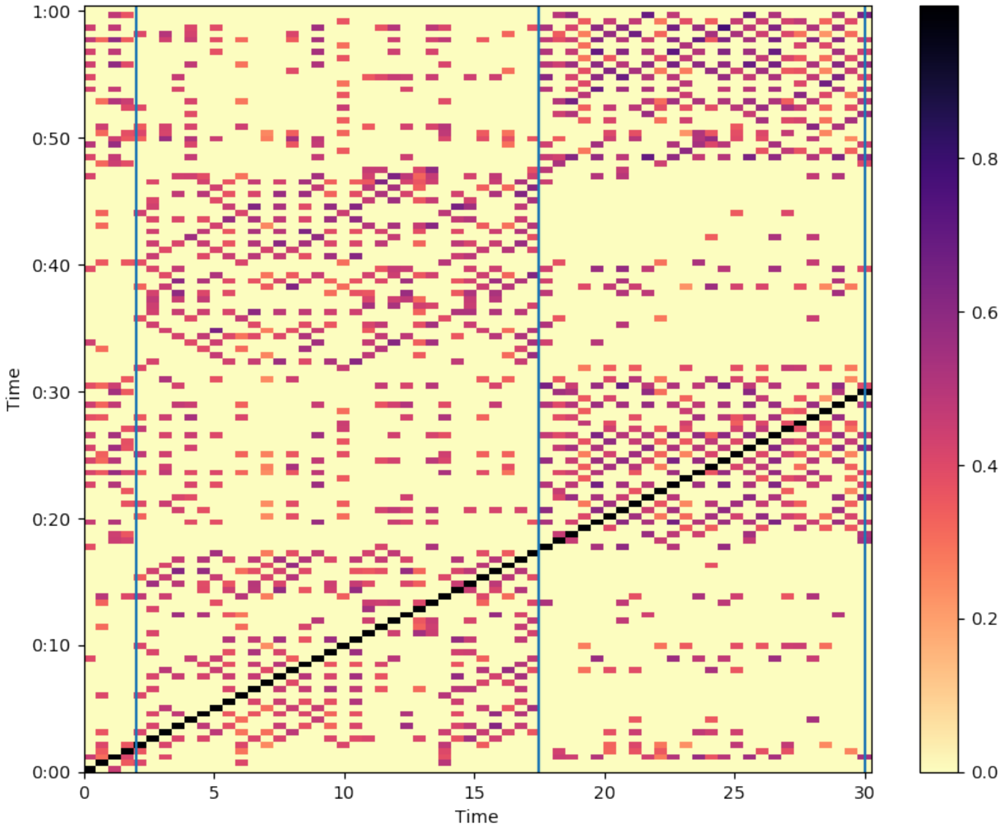


Figure 1: cross-similarity matrix

We also plot the self-similarity matrix for the original 30 second clip (i.e the cross-similarity to itself), visually identify a repeating structure (it could be a bar, a phrase, a segment) on the self-similarity matrix, describe it, and generate two audio fragments that demonstrate this repetition. Note that repetition shows as block structure, so we will need to map the dimensions of the repeating block to time to select the audio fragments.[1]

We show the self-similarity matrix. The two most prominent structures in SSMs are block and paths.

- blocks represent homogeneous sections of music. These are darker sections, indicating higher similarity over a duration.
- paths represent repetition. This occurs when when a subsequence (melody, rhythmic pattern) is repeated, which shows up as a similarity.

From Figure 2, we can observe several long paths within some blocks. The dark block indicates a homogeneous section, and the dark diagonal line indicates that this section is pretty much a repetition. These dark lines correspond to the repeating section of the original audio in the refrain. We can observe a big dark block at the left-down corner which clearly is a repetition section. Then we generate two audio fragments that demonstrate this repetition. I have plotted the vertical lines to indicate those sections.

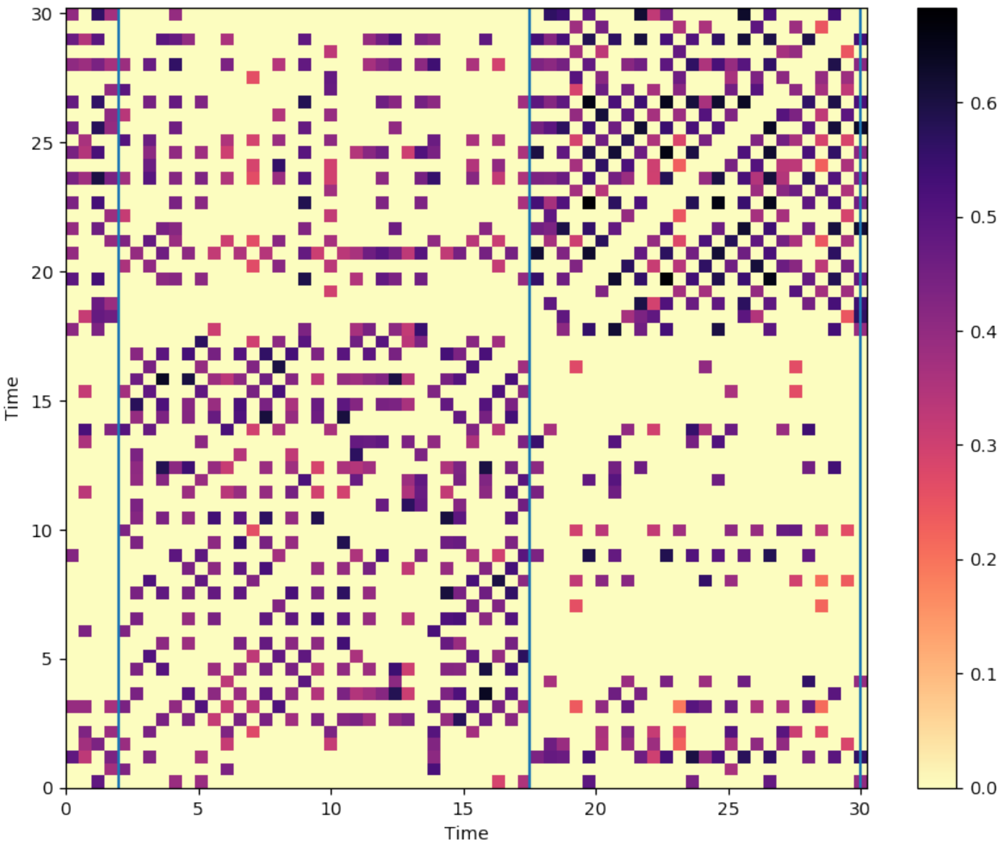


Figure 2: self-similarity matrix

By generating two audio fragments that demonstrate this repetition and listening test, these two audio fragments clearly correspond to the two famous sentences in this song: ”Annie, are you okay? Will you tell us that you’re okay?” and ”There’s a sound at the window, then he struck you — a crescendo, Annie.”

Time Stretching

We use time stretching on the audio recording to create the following modified signal: the first 10 seconds should be slowed down (rate 0.75), the middle 10 seconds should remain the same, and the last 10 seconds should be sped up (rate 1.25). Then we plot the cross-similarity matrix between the original and modified recording (use MFCC features and the affinity mode) and describe how the time-stretching can be observed visually.[1]

We show the cross-similarity matrix in Figure 3. For the matrix below, we can observe several long paths within some blocks. The dark block indicates a homogeneous section, and the dark diagonal line indicates that this section is pretty much a repetition. These dark lines correspond to the repeating section of the original audio. I have plotted the vertical lines to indicate those sections.

From the first dark block at the left-down corner, we can observe that the slowly-stretched part(2s-13s) is not as dark as the corresponding part of the above matrix, due to the stretching operation, but the unstretched part is as dark as usual, so we can say that this whole section, despite having swelled up, is still a repetition.

From the second dark block at the right-up corner, similarly, we can say that this whole section, despite having shrinked, is still a repetition. We can observe that fastly-stretched part(23s-30s) is not as dark as the corresponding part of the above matrix, due to the stretching operation, but the unstretched part is still as dark as usual.

Compared to the above matrix, we can observe time-stretching visually this way.

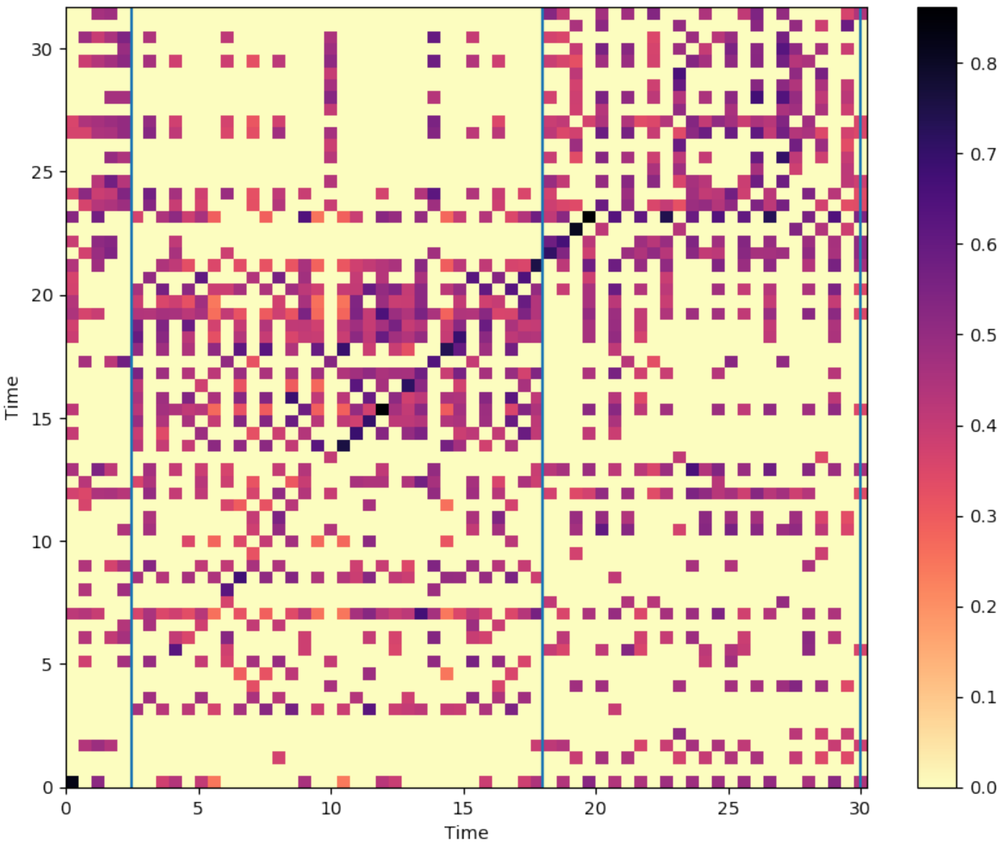


Figure 3: cross-similarity matrix

Harmonic/Percussive Sound Source Separation

We use harmonic/percussive sound source separation functions in Librosa package to generate a percussive track and a harmonic track from the 30 second example, and plot the self-similarity matrices using affinity for the percussive and harmonic versions using MFCCs as well as Chroma. Based on the resulting four plots, we discuss feature set works better for each configuration (harmonic/percussive).[1]

We show the self-similarity matrices for the percussive version in Figure 4 and Figure 5.

We can represent some different musical aspects to start to perform structure analysis. Like, for timbre analysis we can depend more on MFCCs, while for harmony analysis we can depend more on Chromagrams.

For these two matrices below, we can observe several long paths within some blocks. The dark block indicates a homogeneous section, and the dark diagonal line indicates that this section is pretty much a repetition. These dark lines correspond to the repeating section of the original audio. I have plotted the vertical lines to indicate those sections.

In this case, for the MFCC-matrix, the dark blocks and their diagonal lines of two sections that represent different repetition respectively are not obvious enough. By looking at the Chroma-matrix, through observing the several diagonal lines within the dark blocks, we can clearly identify the repetition sections. So, feature set works better for the percussive version.

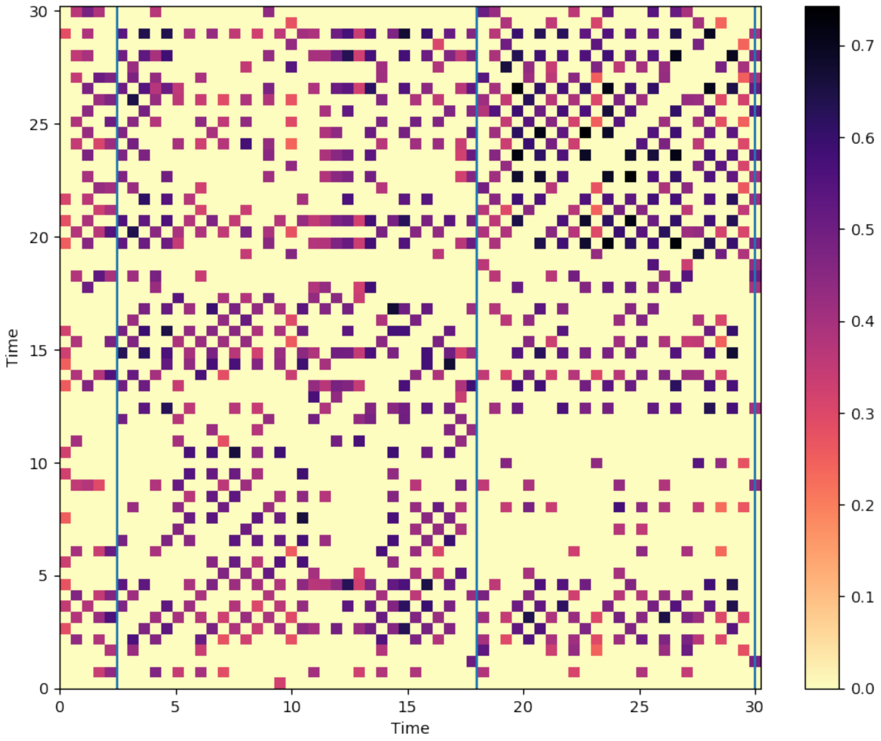


Figure 4: self-similarity matrix using affinity for the percussive version using MFCCs

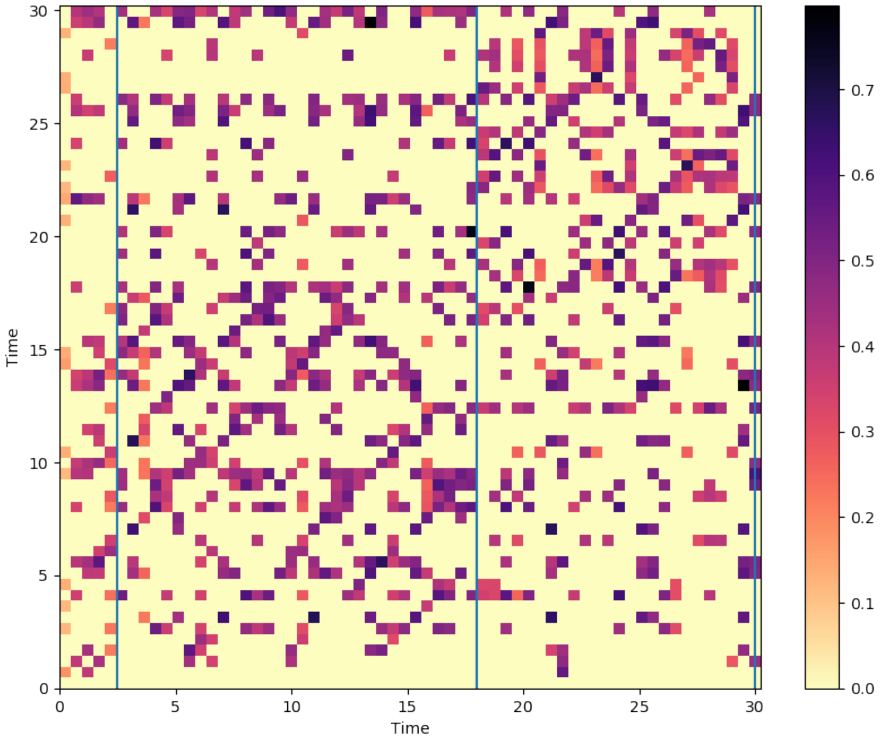


Figure 5: self-similarity matrix using affinity for the percussive version using Chroma

We also show the self-similarity matrices for the harmonic version in Figure 6 and Figure 7.

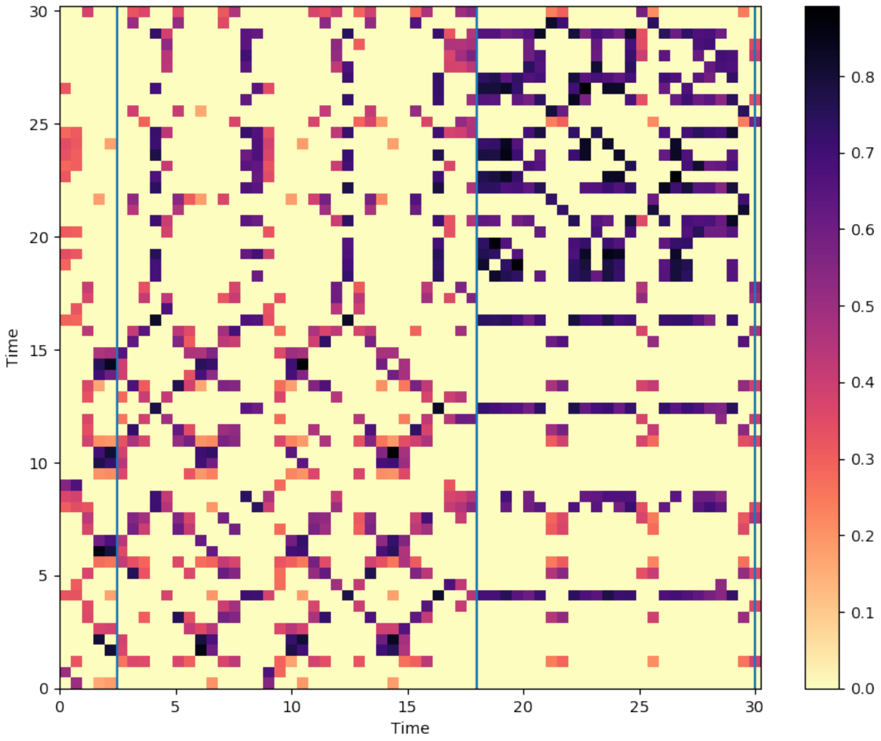


Figure 6: self-similarity matrix using affinity for the harmonic version using Chroma

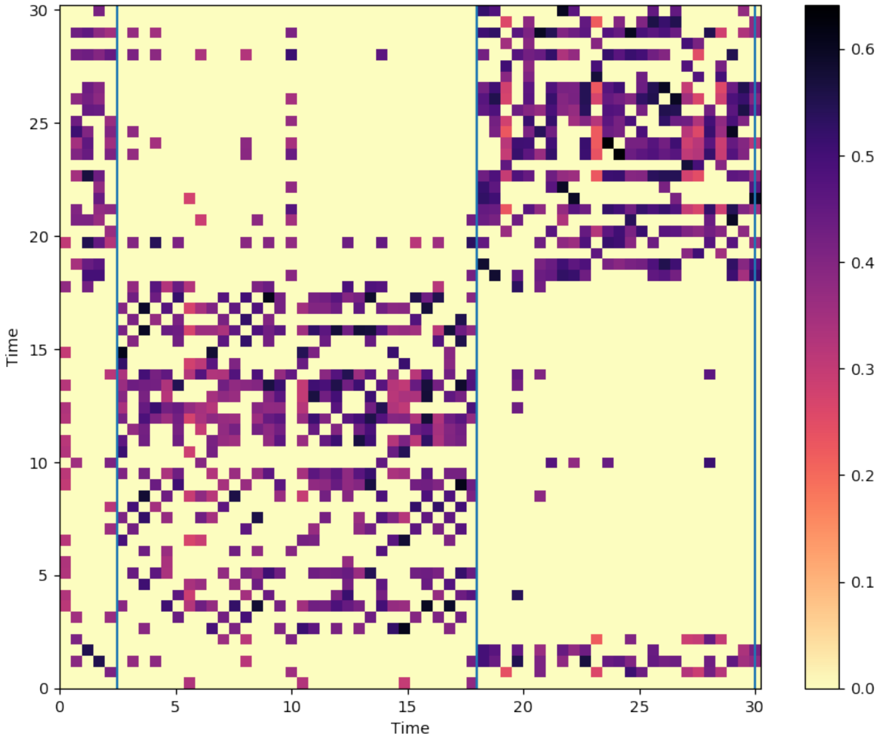


Figure 7: self-similarity matrix using affinity for the harmonic version using MFCCs

We can represent some different musical aspects to start to perform structure analysis. Like, for harmony analysis we can depend more on Chromagrams, while for timbre analysis we can depend more on MFCCs.

For these two matrices below, we can observe several long paths within some blocks. The dark block indicates a homogeneous section, and the dark diagonal line indicates that this section is pretty much a repetition. These dark lines correspond to the repeating section of the original audio. I have plotted the vertical lines to indicate those sections.

In this case, for the Chroma-matrix, the dark blocks and their diagonal lines of two sections that represent different repetition respectively are relatively obvious. Then, by looking at the MFCC-matrix, through observing the very obvious dark blocks, it's safer to say that we can clearly identify the repetition sections. So, feature set works better for the harmonic version.

Dynamic Time Warping

In this part, we use Dynamic Time Warping using the original and modified (time-stretched) recording we just created in the previous subsuestion and plot the cost matrix and associated optimal path and describe how the optimal path reflects the time stretching. Then we show how we can estimate the time-stretching rates from the optimal path. Specifically, we assume that we know that the rate is going to change every 10 seconds but we don't know what the corresponding rates are. In addition, we test the procedure with a set of different time stretching rates.[1]

In Figure 8, we plot the cost matrix and the associated optimal path for the original track and the time-stretched track in the previous part.

By looking at the graph, we can observe that the optimal path can be divided into 3 sections, each of which has different slope: 0.75, 1, 1.25. Because we know that the rate is going to change every 10 seconds, the slopes correspond to the different time-stretching parameters we used in the previous part. So by discussing the slope for each section of the optimal path, we know how the optimal path reflects the time stretching and how we can estimate the time-stretching rates from the optimal path.

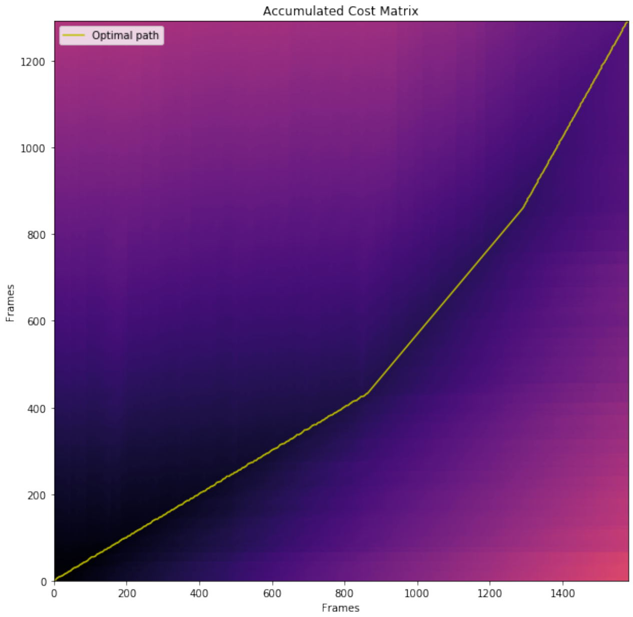


Figure 8: Accumulated Cost Matrix

In Figure 9, we plot the cost matrix and the associated optimal path for the original track and the time-stretched track with parameters 0.6,1,1.4.

By looking at the graph, we can observe that the optimal path can be divided into 3 sections, each of which has different slope: 0.6, 1, 1.4. Because we know that the rate is going to change every 10 seconds, the slopes correspond to the different time-stretching parameters we set. So by discussing the slope for each section of the optimal path, we know how the optimal path reflects the time stretching and how we can estimate the time-stretching rates from the optimal path.

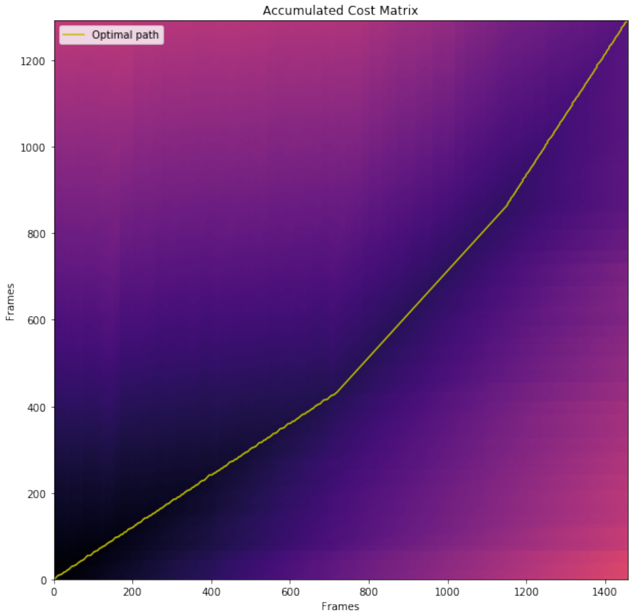


Figure 9: Accumulated Cost Matrix

Finally, in Figure 10, we plot the cost matrix and the associated optimal path for the original track and the time-stretched track with parameters 0.5,1,1.5.

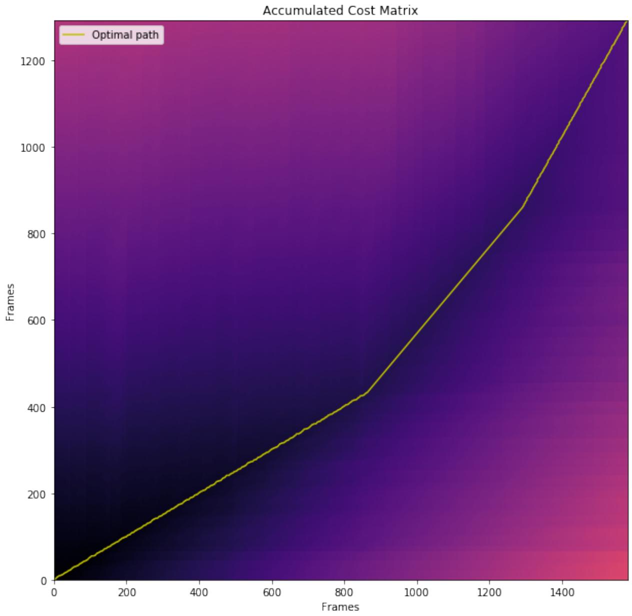


Figure 10: Accumulated Cost Matrix

By looking at the graph, we can observe that the optimal path can be divided into 3 sections, each of which has different slope: 0.5, 1, 1.5. Because we know that the rate is going to change every 10 seconds, the slopes correspond to the different time-stretching parameters we set. So by discussing the slope for each section of the optimal path, we know how the optimal path reflects the time stretching and how we can estimate the time-stretching rates from the optimal path.

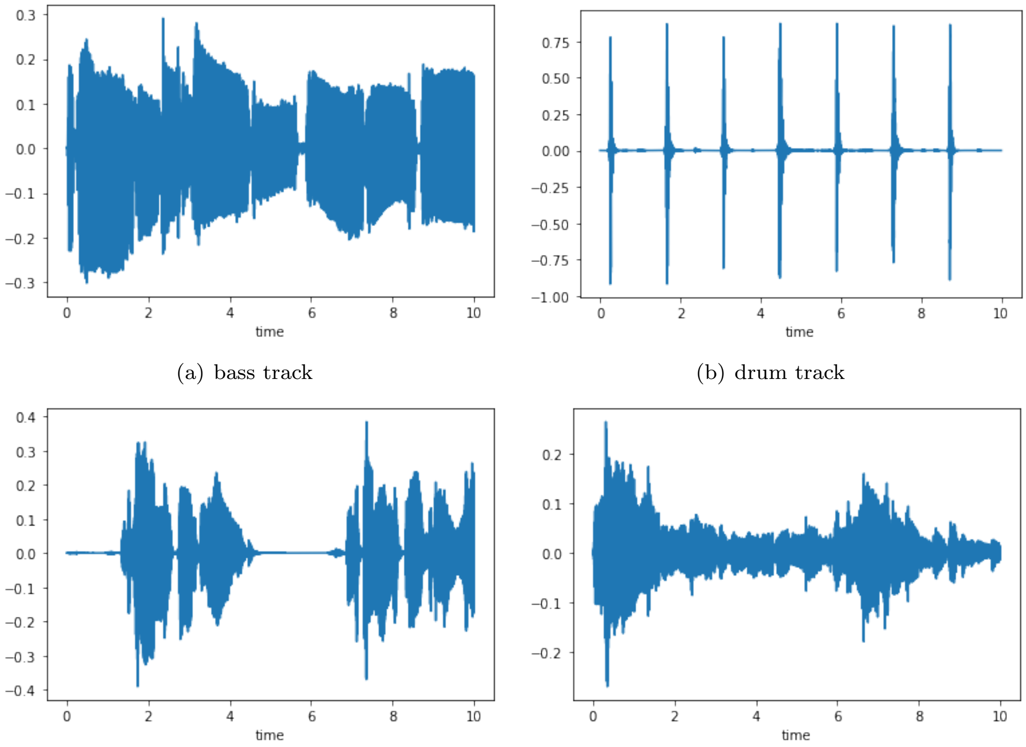
Deep Learning Method

Sound Source Separation

In this part, we will be using a state-of-the art deep learning sound source separation library, spleeter, to explore sound source separation, and transcription. We use a sound track of one sentence in ”Wonderful tonight” by Eric Clapton as the example in this part. Firstly, we run the sound source separation with the 4 stem model on the example audio recording. Then we listen and plot the time-domain waveforms of the 4 individual stems.[1]

The four resulted sound tracks consisting of bass track, drum track, vocal track and the other track. The bass track is a pure bass track separated from the original audio, almost without noise from other single track. The drum track is a pure drums track separated from the original audio, almost without noise from other single track. The vocal track is a pure vocals track separated from the original audio, almost without noise from other single track. The other track is a pure accompaniment and harmony track separated from the original audio, almost without noise from other single track.

We give the sinusoidal wave plots for the different four resulted tracks in Figure 11.



(a) bass track

(b) drum track

(c) vocal track

(d) other track

Figure 11: sinusoidal waves for four resulted tracks

From the above figure, it could be clearly seen that the bass track sinusoidal wave has the feature of low frequency, the drum track has the feature of periodical interval, and the vocal track and other track also correspond the vocal and harmony sound feature.

Pitch Shifting

We make the pitch shift the vocal stem by a major third (4 semitones) and mix the pitch shifted audio with the other stems.[1] Listening to the result, we can hear the mix of the third track shifted-up from the root vocal track and the original other tracks. A complete triad track is composed of root track, third track and fifth track. Triad is the most fundamental and common harmony in music composing, base on the theories of harmony wave. So, even if we have shifted the root vocal track to the third track, the mix sounds still convenient and smooth.

Monophonic Pitch Detection

Here we run monophonic pitch detection on the vocal stem, sonify it using a sinusoid and mix with the drum track.[1] Listening to the result, we can still recognize the song because the sonified audio resulted from the monophonic detection performed on the original vocals has almost the same frequency spectrum with the original vocals. Mixed with the drums track, the result sounds like the original music so much without the lyrics, bass and accompaniment.

Kick and Snare Detection

Here we inspect the magnitude spectrogram of the drum track and try to identify features that characterize the kick sound and the snare sound. Based on the visual inspection, we use our own-proposed kick and snare detection method that outputs the times where a kick or snare drum occurs. Then we create a new track by placing the kick.wav and snare.wav samples from the resources folder in the detected locations. Finally, we mix the synthetic toy drum track and sinusoid sonification from the previous part and listen to the result.[1]

By listening to the drum track and looking at the spectrogram in Figure 12, we can clearly find that kicks and snares play over the duration of the drum track uniformly. Furthermore, the kicks are shown as low frequency waves while the snares are shown as relatively higher frequency waves. Especially, they are always mixed together.

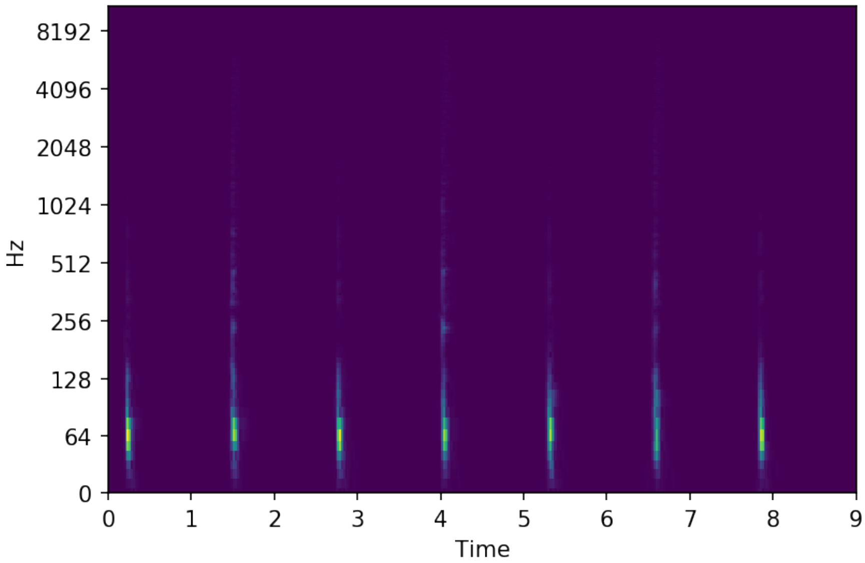


Figure 12: spectrogram

For kick and snare detection, we try to use cross similarity matrices between the drums track and the snare.wav as well as the kick.wav samples to locate every kick and snare in the drums track over the time axis, and then we create a new track by placing the kick.wav and snare.wav samples from the resources folder in the detected locations.

In Figure 13, we show the cross similarity matrix between the drums track and the snare.wav sample. We can clearly see where each snare happens over the time axis.

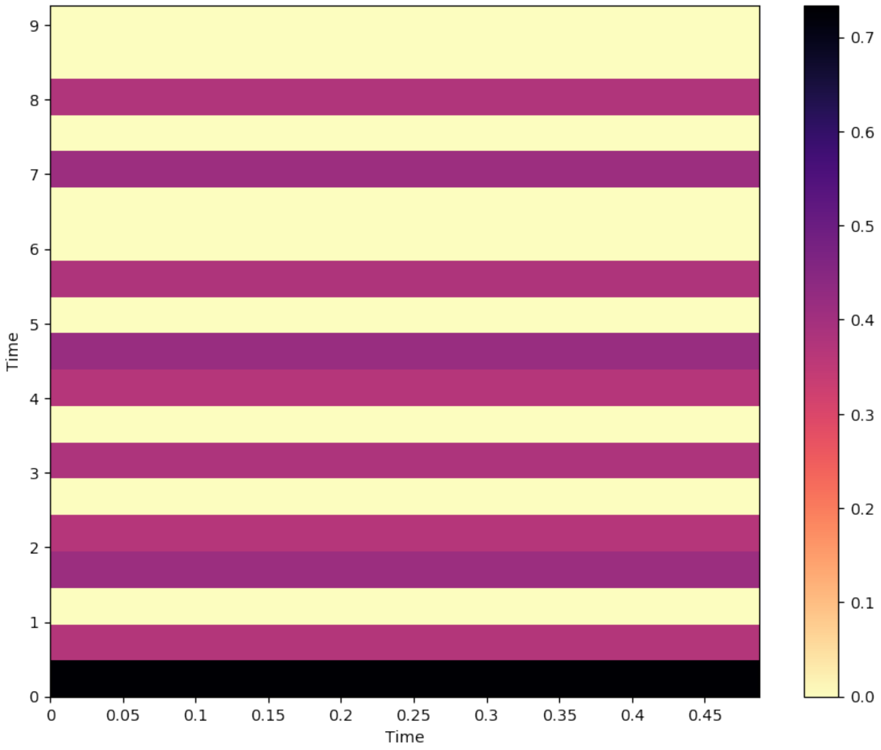


Figure 13: cross-similarity matrix using affinity between snare and drum track using MFCCs

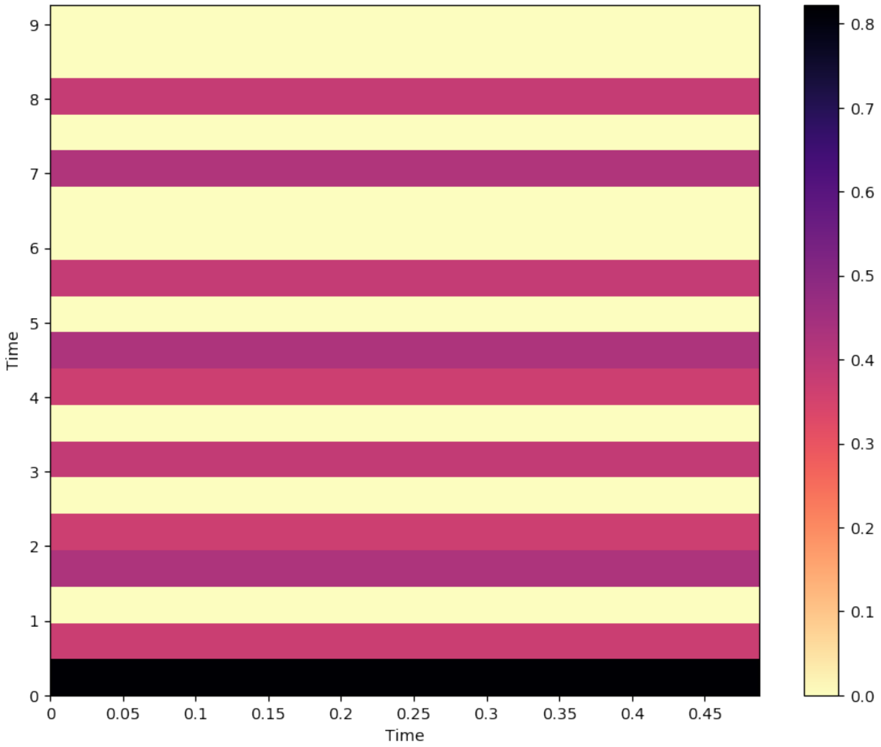


Figure 14: cross-similarity matrix using affinity between kick and drum track using MFCCs

In Figure 14, we show the cross similarity matrix between the drums track and the kick.wav sample. We can clearly see where each snare happens over the time axis.

Legevaring the analysis based on the cross-similarity matrices, we define a function to perform kick and snare detection and return the times where kicks or snares drum occur. Then we mix the synthetic toy drum track and sinusoid sonification from the previous part. By listening to the mix track, we can still identify the song 'Wonderful tonight'.

Conclusion

In this project, firstly, I introduced the cross-similarity matrix and self-similarity matrix for repetition detection and make the evaluation in the regular case and time stretching case. For harmonic/percussive sound source separation, I introduced the MFCCs and Chromagrams for timbre and harmony analysis respectively and give the evaluation using cross-similarity matrix and self-similarity matrix. Dynamic time warping is also discussed for measuring similarity between the different two temporal sequences. Then I used a deep learning method, spleeter, to perform the sound source separation and evaluate the performance in the cases of pitch shifting, monophonic pitch detection and kick and snare detection.

References

[1] J. Shier, *Csc475: Music retrieval techniques course materials.*