SEPARATING MIXED AUDIO FILES USING FASTICA

CMSC 191 - MACHINE LEARNING

Harold R. Mansilla
Department of Physical Sciences and Mathematics
College of Arts and Sciences
University of the Philippines Manila
hrmansilla@up.edu.ph

Virgilio M. Mendoza III
Department of Physical Sciences and Mathematics
College of Arts and Sciences
University of the Philippines Manila
vmmendoza1@up.edu.ph

March 18, 2019

1 Introduction

1.1 Noise Reduction

Noise Reduction, or denoising, is defined as the attempt the noise present in the signal. Noise refers to any kind of random disturbance of the signal [**?**]. The noise reduction is a technique applied to sound and visual media where clarity is of importance. One of the techniques to implement noise reduction is known as the Independent Component Analysis (ICA) which will be discussed later on. Using the said technique, the original form of sound or image can be retrieved albeit with in a different scale.

1.2 Independent Component Analysis (ICA)

Suppose in a noisy room, there are two microphones placed such that one person's voice is recorded. The microphones record both voices along with the noise in the background. The problem now is to distinguish the two voices in the recording while eliminating the noise. This is what is known as the *cock-tail-party problem* [**?**].

The Independent Component Analysis (ICA) is a statistical and computational technique for revealing hidden factors in that underlie sets of random variables, measurements, or signals [**?**]. This technique was originally developed to tackle with the *cocktail-party problem* and cases similar to it such as removing noise from images and analyzing electroencephalograms (EEGs). It has also been used in feature extraction to find suitable representations for images and sounds [**?**].

One thing to note about the ICA is that the data to be processed must be non-Gaussian, that is, the data points are independent from each other [**?**]. It must also be noted that the technique is unable to properly determine the scales(variances) of the individual components [**?**].

2 The Audio Tracks

Two audio tracks were used for this paper, namely "Bboom Bboom" and "Baam" by Korean Pop girl group "MOMOLAND". Both songs were individually downloaded in `mp3` format. To reduce possible errors in running ICA in the mixed audio file, both songs were shortened to one minute using "Audacity", a software for working on audio. The files were then exported to the `.wav` format.

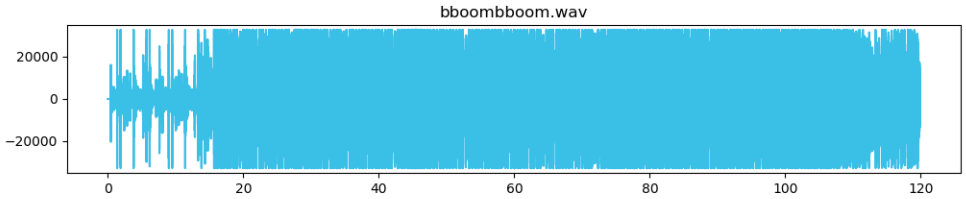

Figure 1: Waveform plot of "Bboom Bboom"

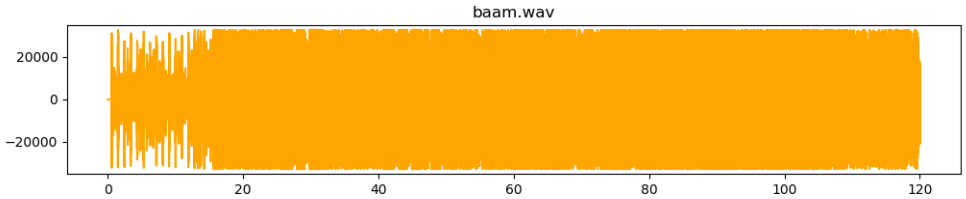

Figure 2: Waveform plot of "Baam"

To create the mixed audio file, both files were read using the Python library `wave`. Each song's signal, rate and time length were gathered. The final, mixed audio was created using the `zip` function.

3 FastICA Process

FastICA was implemented using `scikit-learn`'s FastICA library. The number of components was set to 2 (for two songs to be separated). The transformed waveform was returned after calling the `fit_transform` function. This file was then saved as another `.wav` file.

4 Results

The transformed file from the FastICA output had a shape of (5287434, 2).

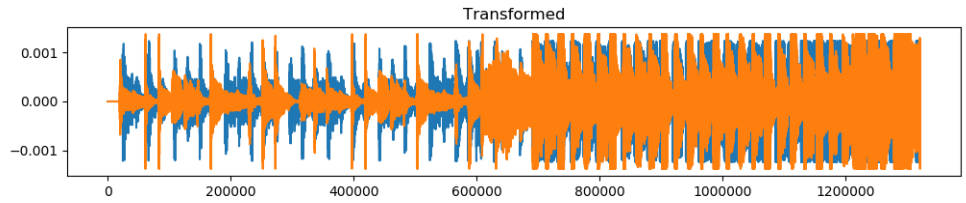

Figure 3: Waveform plot of the transformed file (One-fourth of total length)

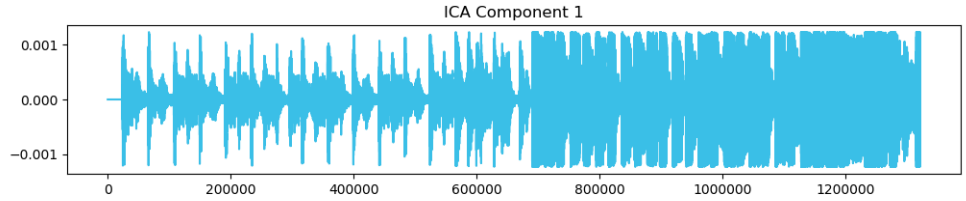

Figure 4: Waveform plot of the first ICA component (One-fourth of total length)

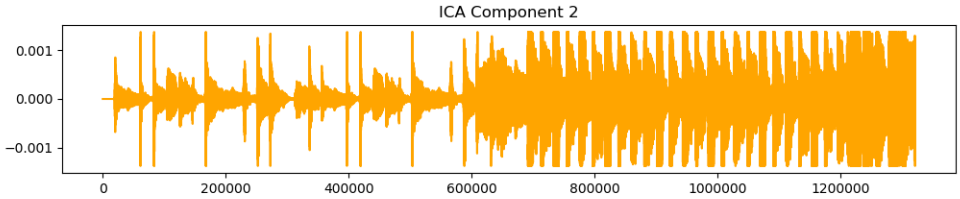

Figure 5: Waveform plot of the second ICA component (One-fourth of total length)

From the figures above, it can be noticed that the scaling is significantly smaller compared to the original audio files. The `.wav` file was then opened in Audacity to be listened to. Playing the unaltered file results into nothing in the playback (due to the scaling). Furthermore, the playback rate was noticeably half of the original rate. This was promptly tweaked in the code for writing the output file (doubling the rate and multipyling the output matrix by 3500000 . The two songs were each in their own channel ("Baam" in the left channel and "Bboom Bboom" in the right).

5 Conclusion

Using `scikit-learn`'s implementation of FastICA, separating two audio files mixed together was possible. For the selected audio tracks to be mixed, the estimated components were successfully separated from one another, each in their own stereo channel. The only noticeable difference was that the separated audio tracks have been scaled down significantly and after using FastICA on the mixed track.