ICA for Musical Signal Separation

Alex Favaro Aaron Lewis Garrett Schlesinger

1 Introduction

When recording large musical groups it is often desirable to record the entire group at once with separate microphones for each instrument. This technique allows the group to record a piece as they would perform it while still producing multiple tracks for later balancing and tweaking. The mixing process is made more difficult in this scenario, however, by the sound of each instrument "bleeding in" to the other microphones such that the recorded instruments are not truly isolated. Ideally we would like to completely remove this effect by separating the signals generated by each instrument into individual tracks. In general we are unaware of the factors which contribute to the bleeding effect so the problem is an example of blind signal separation (BSS).

One of the more common solutions to BSS is Independent Component Analysis (ICA). Most ICA algorithms use a generative model that assumes that the observed signal is generated from a linear combination, i.e., instantaneous mixture, of statistically independent sources. Formally, at each time sample i we observe

$$x^{(i)} = As^{(i)} \tag{1}$$

where $s^{(i)} \in \mathbb{R}^n$ are our n source signals at time i and A is an unknown square matrix called the mixing matrix. Given this assumption, the demixing matrix $W \approx A^{-1}$ is obtained by maximizing the statistical independence of the source signals that we wish to isolate.

In practice, instantaneous mixtures of audio signals are quite rare. Microphones in a real recording scenario will pick up not only the direct sound from each source but also their reflections from walls and other objects. Even when such reflections are minimal (as might be the case in a well-equipped recording studio) the sounds will reach each microphone at different times due to propagation delay. A more accurate model describes each observed signal as a linear combination of delayed source signals. Concretely, observed signal j at time sample i is given by

$$x_j^{(i)} = \sum_{k=1}^{n} a_{jk} s_k^{(i-t_{jk})} \tag{2}$$

where n is the number of signals, a_{jk} is the j,k-th element of A, and t_{jk} is the amount of delay from source k to microphone j.

Given this formulation of the problem we attempt to extend ICA to handle the real world problem of signal separation in musical recordings. In Section 2, we discuss the data that we used to test our methods. Section 3 describes each method and its results in turn. We conclude in Section 4 with a discussion of room for improvement and future work.

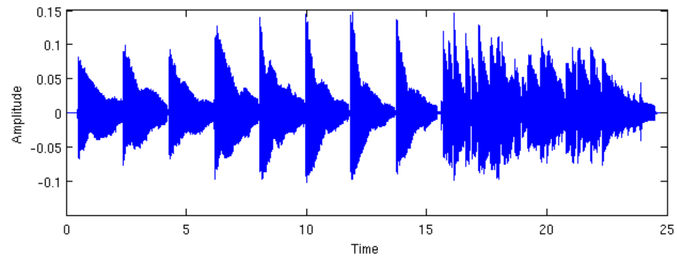
2 Data

We tested our approach on a number of different data sets. Each of our recordings includes four instruments: an electric guitar, a piano, a tenor saxophone, and a snare drum. With a separate microphone for each instrument we recorded three scenarios: each instrument playing independently (i.e., not the same piece of music), a B♭ major scale played in unison and with various rhythmic patterns, and a simplified arrangment of Tower of Power's "Ain't Nothin' Stoppin' Us Now". In each case we recorded all of the instruments together to create the bleeding effect and also separately with no bleeding. As a sanity check we also artificially mixed our separately recorded tracks to recreate the bleeding effect both with and without propagation delay.

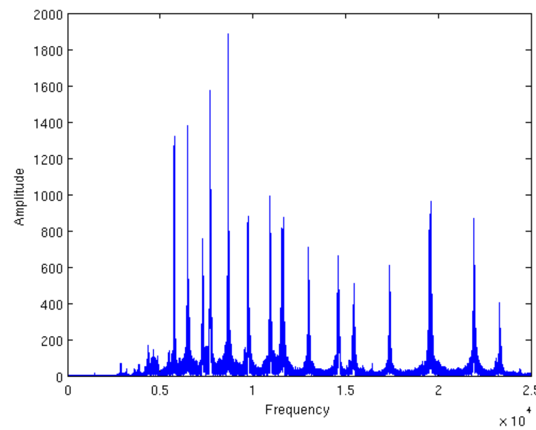
Our data is stored in a lossless audio format that allows us to easily operate on the time domain (time vs amplitude) of the signal. We also generate spectra for the signals which allow us to operate on the frequency domain (frequency vs amplitude) as well as spectrograms that represent the spectra at different time windows. Figure 1 shows the data generated from the guitar's microphone while playing a B♭ major scale.

Figure 1: Guitar data

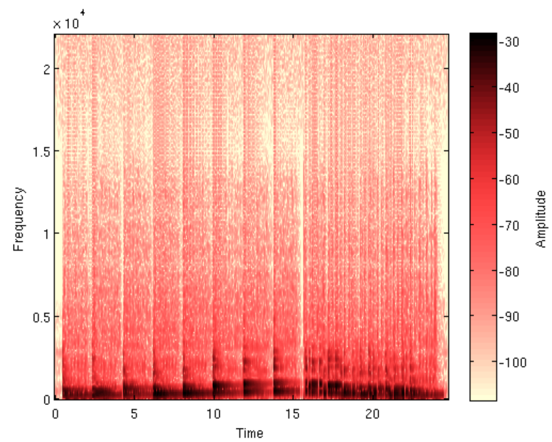
(a) Time domain



(b) Frequency domain



(c) Time-frequency domain



3 Methods and Results

Although ICA performed well in the time domain on our artificially created instantaneous mixtures, the algorithm's performance degraded rapidly when propagation delays were introduced. The recovered signals from our real world recordings were less isolated than the observed signals. To account for these propagation delays we subsequently focused our efforts on separation in the frequency domain.

3.1 ICA in the Frequency Domain

Note that after applying the Fourier transform to our signals Equation 2 becomes

$$\hat{x}_j^{(i)} = \sum_{k=1}^{n} a_{jk} \exp(-\mathbf{i} t_{jk} \omega^{(i)}) \hat{s}_k^{(i)} \tag{3}$$

where $\hat{x}_j^{(i)}$ and $\hat{s}_k^{(i)}$ are the Fourier transforms of observed signal j and source signal k, respectively, and $\omega^{(i)}$ is the frequency at sample i. Thus propagation delay in the time domain becomes complex rotation in the frequency domain so the observed signals are now instantaneous mixtures of the source signals. Our mixing matrix, however, is now a function of signal frequency.

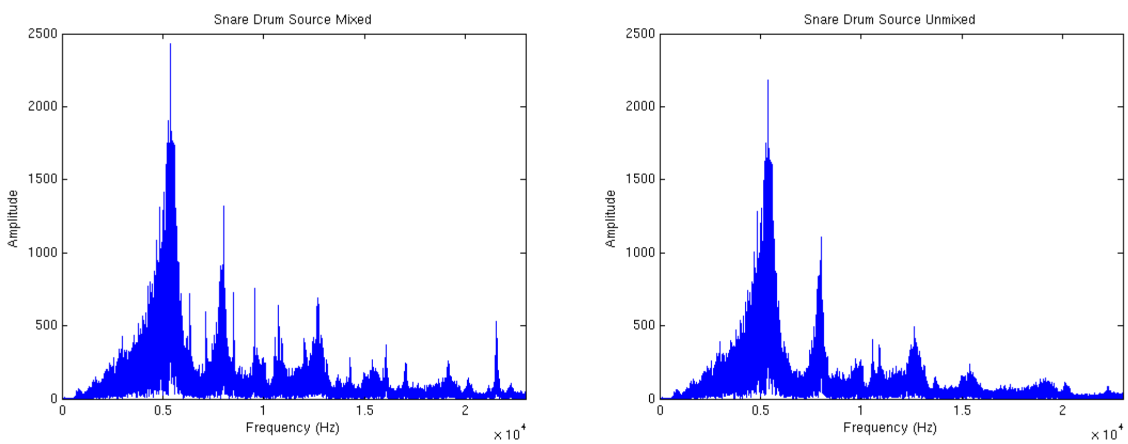
Initially we ignored the frequency dependency in the mixing matrix by running a version of FastICA for complex-valued data (CFastICA [2]) over the Fourier transforms of our observed signals. We recovered the source signals by applying the inverse Fourier transform to the resulting independent components. Our hope was that the propagation delays ($\approx$ 3ms) would be small enough that the frequency dependent components of the mixing matrix would be negligible.

We only had small success in separating tracks using CFastICA. In artificially mixed B♭ scale, the drums were entirely separated out of one track, though the melodic instruments are all mixed to a greater extent than in the source tracks. In all tracks with propagation delay, both natural and artificial, the outputted signals were more mixed than the source files. This mixture occurred because the source tracks are co-dependent in the frequency domain.

We also decided to try running FastICA on the magnitude of our frequency responses as a heuristic to generate the mixing matrix. This greatly simplifies the signal by removing the phase information, which in turn ignores any propogation delays. To recover our signals, we take the resulting demixing matrix and apply it to the frequency response of our observed signals. We then apply an inverse Fourier transform on the results to get our estimated independent components.

Figure 2: Frequency Domain Results

(a) Observed (b) Recovered



We had success in isolating artificially mixed tracks by running FastICA on the magnitude of the Fast Fourier Transform. In both the artificially mixed scale and jam tracks, the piano and snare drum separated well. The snare drum in particular isolated with effectively no audible interference from other sources. Figure 2 shows the observed and recovered frequency domain signals for the snare drum on B♭ scale. We hypothesize that the snare drum isolates particularly well in the frequency domain because its frequencies are the most independent. The guitar, piano, and saxophone play many of the same notes over the course of a track (and in the case of the B♭ scale, all of the same notes). This means that their frequencies are heavily dependent, leading ICA to perform poorly. However, the snare drum does not vary in frequency over the course of a track and is in this way the most unique and independent instrument, so ICA is able to recover the drum.

3.2 Frequency Banded ICA

We can rewrite Equation 3 in a more familiar form as

$$\hat{x}^{(i)} = A(\omega^{(i)})\hat{s}^{(i)} \tag{4}$$

where $A(\omega^{(i)})$ is our mixing matrix as a function of frequency. Thus the problem in the frequency domain is a set of instantaneous mixtures as in Equation 1. Since the frequency dependencies in the mixing matrix are similar for close values of ω we can run ICA on a number of relatively small frequency bins. The source signals are recovered by appending the resulting independent components and applying the inverse Fourier transform.

One issue that arises with this approach is known as the permutation problem. Given only the observed signals, the permutation of the recovered sources is arbitrary. We must therefore ensure that the permutation of sources recovered by ICA is the same for each frequency bin. A number of approaches have been suggested to overcome the permutation problem [3, 4]. We implemented the simplest of these, which calculates the demixing matrix for the frequency bins one at a time using the matrix calculated for the previous bin as the initial guess for the next bin. Since the neighboring frequency values should be somewhat close to one another this helps to ensure that the permutation will not change from bin to bin.

Unfortunately this approach to the permutation problem was insufficient to overcome the complexity of our data. Although we believe that Equation 4 was a good way to view the problem (and the literature would seem to agree [3, 4]), the results we obtained from this method were unsatisfactory. Many of the recovered signals were washed out and clearly contained sounds generated by all of the sources. For our data at least, a more sophisticated solution to the permutation problem is necessary.

3.3 ICA with Linear Regression

Our third approach to the propagation delay problem was to modify how the mixing matrix is computed in ICA directly. By inverting the problem we define the j, k-th element of our demixing matrix as follows

$$w_{jk}(\omega) = c_{jk} \exp(\mathbf{i} t_{jk} \omega) \tag{5}$$

FastICA uses a deflation method that solves for the source signals one at a time [1]. In the iteration that computes source signal j, w_{jk} is updated to be the mean over all $w_{jk}^{(i)}$ where $w_{jk}^{(i)}$ is the estimate for w_{jk} computed from sample i. To remove the frequency dependency from our model, we modify this update step to instead calculate c_{jk} and t_{jk} from the $w_{jk}^{(i)}$'s.

Taking the natural logarithm of Equation 5 we obtain

$$\log w_{jk} = \log c_{jk} + \mathbf{i}t_{jk}\omega \tag{6}$$

which is linear in ω. We can therefore use linear regression on $\log w_{jk}^{(i)}$ and $\omega^{(i)}$ to obtain estimates for $\log c_{jk}$ and $\mathbf{i}t_{jk}$ from which we calculate c_{jk} and t_{jk}. Once all the c_{jk} and t_{jk}'s are computed we can use Equation 5 once more to write our demixing matrix as a function of ω and recover the source signals.

This change essentially modifies our estimate of w_{jk} by fitting a 1-deminsional polynomial to the w_{jk}'s at each frequency instead of a 0-dimensional polynomial. While this change seemed promising and logical, its results were not satisfactory. Introducing the new degrees of freedom resulted in FastICA's gradient descent failing to converge in any reasonable amount of time. The values of c_{jk} and t_{jk} produced at each iteration appeared at times to be oscillatory and at other times to randomly shift. This method may have the potential to be successful with future work and investigation but at the time of writing was not successful.

4 Conclusion

In conclusion, signal separation on real world data is difficult. We primarily focused our separation methods on accounting for the volume decay and propagation delay present in recording multiple instruments in one setting. However, solving for these variables given the different mixes and the knowledge that the sources are independent pieces of music was a tougher task then we expected. We started with an algorithm that was capable of separating the observed signals if there was no propagation delay present, and throughout our various methods the best results were to separate out one or two instruments. We attribute this result to the difficulty of simultaneously solving for volume decay and propagation delay as well as the difficulty present in musical data sets - that the sources are not entirely independent.

While our results are not indicative of finding an optimal solution to the problem, we do feel that we have made progress. We were able to separate some of the signals and succesfully isolate some of the instruments in our data set. In addition, we also investigated innovative methods to solve for both the volume decay and propogation delay which, given more time and effort, may be able to produce better results.

References

[1] Hyvärine A., "Fast and Robust Fixed-Point Algorithm for Independent Component Analysis", *IEEE Trans. on Neural Networks*, 10(3):626-634, 1999.

[2] Bingham E. and A. Hyvärine, "A fast fixed-point algorithm for independent component analysis of complex valued signals", Helsinki University of Technology, 2000.

[3] Smaragdis P., "Information Theoretic Approaches to Source Separation", MAS Department, Massachusetts Institute of Technology, 1997.

[4] Mitinoudis N. and M. Davies, "Audio Source Separation of Convolutive Mixtures", *IEEE Trans. on Speech and Audio Processing*, 11(5), 2003.