Recovering the Multitrack: Semi-Supervised Separation of Polyphonic Recorded Music

Sasen Cain and Jennifer Dolson
CS229 Final Paper

December 12, 2008

1 Introduction

The goal of this project is the extraction of vocals and/or specific instruments from a polyphonic CD-quality music recording. Instrument separation is difficult because each song can be thought of as a weighted mixture of different signals, each which vary based on the note being played, the timbre (characteristic sound) of a given instrument, and the recording conditions. Before we can assign parts of the signal to a particular instrument, we must devise a generalized model of the instrument(s) of interest. Finding and using this "acoustic fingerprint" involves both a clustering problem and a recognition problem. The acoustic model that we use is called an Average Harmonic Structure, or AHS.

The approaches of previous work in this area fall into the categories of supervised and unsupervised methods. We hope to build upon the state of the art unsupervised method [2], altering a few key steps in an effort to improve the final result. Our method reduces error by prompting the user for input at crucial stages, and also by replacing some complex and error-prone procedures.

1.1 Vocal Separation

By convention in professional music recordings, vocals can be extracted by subtracting the left channel from the right channel [1]; this approach, while simple, has major shortfalls. Other tracks, such as percussion and bass, might also be removed because they share the "center" position of the mix with the vocals.[1] Any use of reverb on the vocal track introduces artifacts, since the it causes vocals to "bleed" outside of the center band. Furthermore, differential loss of quality caused by algorithms that process the channels separately (e.g., mp3 encoding) hampers perfect vocal cancellation.

1.2 Instrument Separation

There are many different ways to approach the problem of extracting a given instrument from a polyphonic audio mix. Our method builds directly on recent work by Duan, *et al.* [2] which achieves impressive results using unsupervised learning. The authors showed that their methods produce cleaner extraction results than other unsupervised methods such as pure ICA for extracting all possible parts of an audio signal. However, their algorithm cannot process chords or sources playing more than an octave above the lowest-frequency instrument.

A variety of supervised approaches have also been studied. Some methods (such as [3]) attempt to learn statistical information about different instruments, using either HMMs or other statistical models based on priors from outside training sources. In [4], the authors use a training set of multiple similar songs in order to extract vocals from a single song, and [5] takes user-labeled signatures of drums, but only classifies portions of the signal using SVM as "drums" or "not drums". In [6] and [7], models are learned based on example recordings of each separate source independently, or on a database of isolated notes. The work of Bay, *et*

[1] This centering is the reason why the subtraction trick "works" at all.

al. [8] is most similar to what we aim to do, in that it uses harmonic information about a signal, however, it requires that the fundamental frequency of each note be known. We would prefer to avoid complex instrument modelling, note transcription, or special casing of vocals because of they lessen the generality of the procedure.

2 Our Learning Algorithms

In examining the composition of the mixed input signal, we rely on the knowledge that a note is not just a single frequency, but rather excites a fundamental frequency (f_0) and its harmonics at integer multiples of f_0. For a harmonic instrument, the normalized amplitude profile of overtones is fairly stable, independent of pitch or volume. This amplitude profile, shown schematically in Figure 1b, defines the timbre, or general tonal quality of an instrument [11].

Our algorithm can be broken into two sections: AHS Estimation and Instrument Extraction. In the AHS Estimation stage, we make the underlying assumption, following [2], that each harmonic instrument present has a distinct and consistent timbre. Instrument Extraction uses these learned AHSes as feature vectors to classify and reconstruct each instrument.

2.1 AHS Estimation

2.1.1 Processing the Initial Query

In the first stage of the AHS estimation pipeline, we present the user with a small segment of the song (less than 4 seconds) and ask the user to specify the number of harmonic instruments present. We then slice the segment into smaller 4096-sample windows and use the Short Time Fourier Transform [9] to perform peak detection (as in [2], [10]) within each window (Fig. 1a). Each peak[2] is then treated as a potential f_0, and its appropriate harmonic values are found, using the amplitude of the STFT at the first 20 multiples of that peak's frequency. To ensure volume invariance, these values are normalized by the amplitude of the first peak, forming a Harmonic Structure vector, $\mathbf{B}$. The AHS defined in [2] is simply the average over all windows of $\mathbf{B}$. If our assumption about the consistency and distinctness of each instrument's Harmonic Structure holds, then clusters should form in this $\mathbb{R}^{20}$ space based around the AHS of each instrument. Noise, or peaks that were not actually f_0s should scatter in the background. We use k-means, with k as specified by the user, to find these cluster centroids and they serve as our estimates of the AHSes of the k instruments in that window.

2.1.2 Subsequent Queries

The user is asked to report the number of instruments in various randomly selected regions of the song. In the first query, all of the harmonic structures found by k-means can be added to the list of unique instruments, but in subsequent queries, fewer of the instruments will be novel. To avoid replicating instruments, we check whether a set of AHSes is linearly independent using SVD. Each AHS returned by k-means is added to the set of previous (unique) AHSes, and the condition number, $\kappa = \sigma_H/\sigma_L$ is calculated. If the lowest singular value is approximately 2 orders of magnitude smaller than the largest singular value, then this new AHS is discarded because it represents an instrument whose AHS has already been found.

This procedure additionally serves to validate the first pipeline's results: if repeated instruments are not removed, then there is a failure within peak detection or k-means. We find that, on average, instruments repeated in subsequent queries are not added to the list of unique AHSes, adding credence to the functionality of the system we've described thus far.

[2] typically, 10–20 peaks are identified in each window

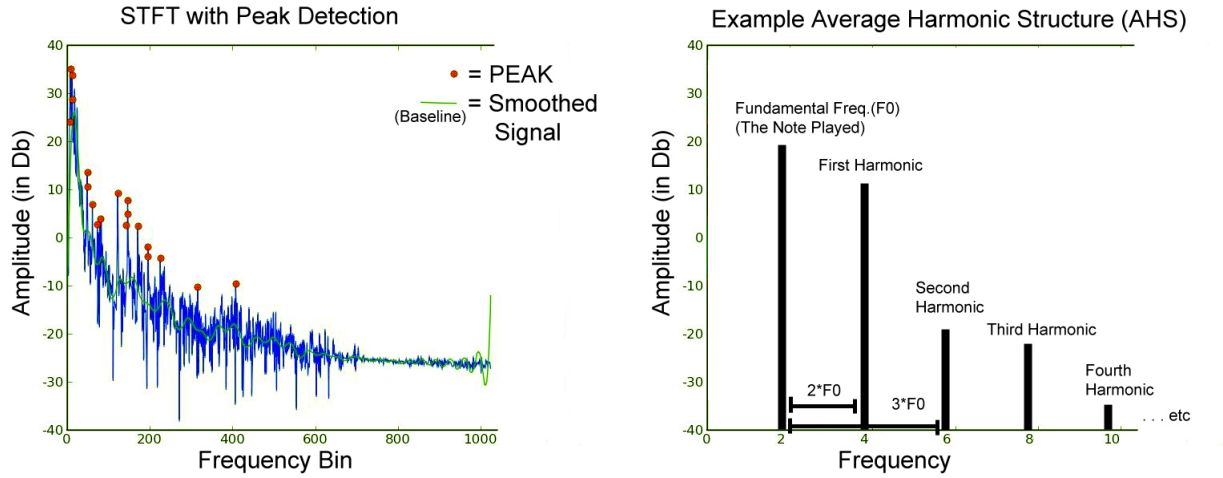

Figure 1: Feature extraction from the frequency domain. (a) The STFT contains many peaks, some of which are f_0s. (b) An AHS discards its frequency information, preserving only a ratio of amplitudes for the first 20 harmonics.

2.2 Instrument Extraction

Even given accurate AHSes for each instrument, the problem of creating a signal containing only the extracted instrument is still formidable. For each time window, we must decide which instruments are present and at what frequences (i.e., what notes are being played), and finally reconstruct the desired signal.

2.2.1 Peak Matching

In any time window, we assume that the STFT is comprised of a linear combination of the AHSes, plus noise from non-harmonic sources. Intuitively, this means that some instruments are playing at particular volumes, and some are silent. If we knew the f_0s played by each instrument in that slice, then we could easily recover the mixing volume levels using the standard least squares formulation, $y = Ax$. $y \in \mathbb{R}^{2048}$ is the observed vector of amplitude peaks. For n source instruments, $A \in \mathbb{R}^{2048 \times n}$ would contain each AHS, shifted to frequency-space by its associated f_0, arranged in columns. It is known to be full rank because of the conditioning check performed on the related AHS matrix $\in \mathbb{R}^{20 \times n}$, at the end of the first pipeline. $x \in \mathbb{R}^n$ contains the volume level of each source.

The optimal f_0^i must be found by matching the set of observed peaks, P, to the AHS of the ith instrument. The optimization in equation 1 minimizes the resulting error between the observed and reconstructed peaks. Due to the combinatorial possibilities of the problem, it would take exponential time to exhaustively compute f_0^*, so a heuristic is needed.

$$f_0^* = \min_{f_0 \in P} ||y - A(f_0)x(f_0)||^2 \tag{1}$$

Gradient Descent: A Dead End

We considered using gradient descent to minimze the error function. Unfortunately, this results in an update rule for f_0^t that doesn't make any sense because there is no constraint that the computed $f_0^{t+1} \in P$. In fact, this points out that the optimization isn't even convex since P is a discontinuous set of empirically-derived points.

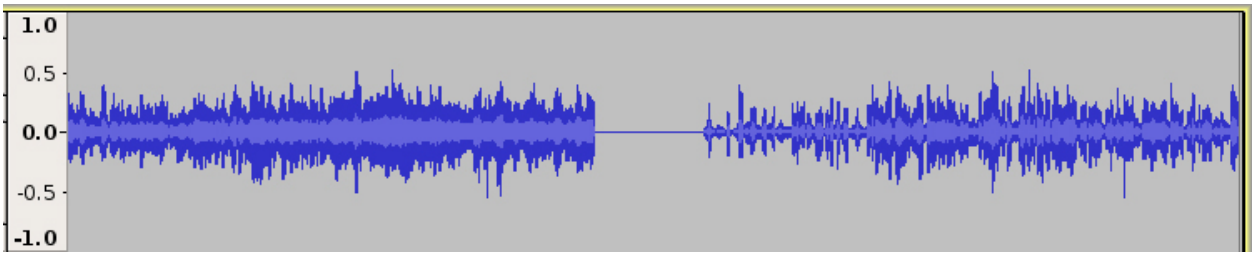

Figure 2: Time domain results. The first portion of the signal contains all instruments, while the second has excluded the guitars.

Ideas From Reinforcement Learning

To accelerate peak matching, we dropped the requirement that $y = Ax$ and chose instead to search through the song, looking for time frames whose STFT peaks were most consistent with the AHS of a particular instrument. This greedy approach uses a reward function to determine consistency based on the STFT amplitudes at peak positions. For $p \in P; i = 1, \ldots, |P|; r = 1, \ldots, 20$,

$$R = \begin{cases} 1 & \text{if } |\text{STFT}(p_i)| \geq \mathbf{B}_r |f_0 \\ 0.5 & \text{if } |\text{STFT}(p_i)| < \mathbf{B}_r |f_0 - \text{tol} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Simply put, if a frame has peaks of the right amplitude (or greater, due to contructive interference) at the right places for a particular AHS given some f_0, then the value function is highly rewarded. If the peaks are present but lower than expected, within some ratiometric tolerance, a moderate reward is received. If the peaks are absent or too low amplitude, there is no reward. This idea, borrowed from reinforcement learning, allows us to quickly estimate the f_0 of an instrument throughout the song by maximizing a value function over possible f_0s. Frames with sufficiently high value are said to contain the instrument, and if the instrument is not present during a particular segment, it should receive very little reward. This procedure yields a list of frames that contain an instrument and the associated $f_0 \in P$.

2.2.2 Final Instrument Extraction

The extraction algorithm was fairly simple and imperfect, but a quick-and-dirty method was needed so that we could focus on the learning steps. For each instrument, the frame numbers that contained it were assigned whatever values were at the right harmonics in the full-signal STFT $\in \mathbb{C}$ (i.e., not $|\text{STFT}| \in \mathbf{R}$), and the rest was set to zero. The inverse STFT of this set of frames is the "extracted" signal. The opposite actions were applied for the "removed" signal: the relevant harmonics in the right frames were set to zero while the rest of the original STFT was preserved. The inverse STFT of the signal with all harmonic sources removed yields the non-harmonic sources, including the vocals.

3 Results and Complicating Factors

Using k-means in the clustering step makes our algorithm succeptible to local minima and maxima. Therfore, good results were only found after multiple trials of our algorithm. In our best result, the clustering and cluster initialization worked such that we were able to estimate the mean of the undistorted guitar in one of our test songs, and extract that instrument from the mix (Fig. 2). Given the simplifications made for the extraction steps, it was difficult not to extract other instruments or vocals that coincide with the frequencies of the guitar, so our extraction was not as clean as we would have liked. However, our ability to model and extract a multi-string instrument playing chords without any specific training or priors represents an improvement over previous work.

4 Future Work

A persistent complication in our work has been that various forms of learning and estimation are applied at each step of the process. Any errors in early stages, particularly peak detection or clustering, will be compounded by later stages, making it difficult to track down errors or determine how best to modify our approach. We are also unsure of how to quantify success rates, which would be helpful to know given that different kinds of instruments have very different structures. One solution is to obtain multitrack recordings, mix them into a single track using arbitrary parameters, and then quantify our ability to recover those parameters. Another possibility is to better label the presence or absence of an instrument throughout the song based on user response, and to then ask the user to confirm correct labelling before attempting extraction. Furthermore, some sort of smoothing or Kalman filtering technique that takes time-dependence into account would hugely improve the playback quality of the separated tracks.

5 Conclusions

The applications of an instrument/vocal extraction system could range from allowing musicians to better hear the different parts of the music they must learn to creating tracks for karaoke or rhythm-based video games (e.g., Rock Band). Furthermore, artists would be able to remix, sample, or mash-up songs without having access to the original recording tracks. We recognize that, due to the supervision required of the user, our system will not be useful for realtime, automatic instrument separation, nor will it be practical for processing a library of thousands of songs. However, this is not a major limitation: in the context of these applications, it is not too tedious for a user to have to answer questions about a target song because he or she will be concerned with the quality of separation over the quantity of audio processed. Also, our method removes the need for having a large training database of individual instruments or similar songs.

A shortcoming of our method is that it requires the user to be sufficiently knowledgeable about the music he or she is working on to correctly estimate the number of instruments present in each listening segment. Though our method is not perfect, we believe its ability to deal with string instruments represents an improvement over previous work; its failings are reasonable, and may be overcome by pursuing some of the avenues described in section 4.

References

[1] Winer, E. "The Truth About Vocal Eliminators," ProRec, The Online Audio Magazine. August 1999. (available online: `http://www.ethanwiner.com/novocals.html`)

[2] Zhiyao Duan; Yungang Zhang; Changshui Zhang; Zhenwei Shi, "Unsupervised Single-Channel Music Source Separation by Average Harmonic Structure Modeling," Audio, Speech, and Language Processing, IEEE Transactions on , vol.16, no.4, pp.766–778, May 2008.

[3] J. Hersheya and M. Casey, "Audio-visual sound separation via Hidden Markov models," in Proc. NIPS, 2002, pp. 1173–1180.

[4] S. Vembu and S. Baumann, "Separation of vocal from polyphonic audio recordings," in Proc. ISMIR, 2005, pp. 337–344.

[5] M. Hélen and T. Virtanen, "Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine," in Proc. EUSIPCO, Istanbul, Turkey, 2005, CD-ROM.

[6] L. Benaroya, F. Bimbot, and R. Gribonval, "Audio source separation with a single sensor," IEEE Trans. Audio, Speech, Lang. Process., vol. 14, no. 1, pp. 191–199, Jan. 2006.

[7] E. Vincent, "Musical source separation using time-frequency source priors," IEEE Trans. Audio, Speech, Lang., Process., vol. 14, no. 1, pp. 91–98, Jan. 2006.

[8] M. Bay and J. W. Beauchamp, "Harmonic source separation using pre-stored spectra," in Proc. ICA, 2006, pp. 561–568.

[9] J.O. Smith, *Spectral Audio Signal Processing*, October 2008 Draft, `http://ccrma.stanford.edu/~jos/sasp/`, online book, accessed November, 2008.

[10] J.O. Smith and X. Serra, "PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation," in Proc. ICMC, 1987, pp. 290–297.

[11] D.J. Levitin. *This is Your Brain on Music: The Science of a Human Obsession.* Penguin Books Ltd., 2006.