Extracting vocal sources from master audio recordings

Derek Mendez, Tarun Pondicherry, Chris Young

Abstract

Our goal is to separate vocals from background music in single-source songs. We examined prior work ([1][2][3][5][6][7]) in the area and decided to experiment with 2D Adaptive Probabilistic Latent Component Analysis (PLCA) [1][5] because it is a good trade-off between implementation difficulty and resulting accuracy. The training data for the 2D Adaptive PLCA algorithm is a set of song segments containing only background music. The adaptive algorithm is more practical than PLCA because typically, a song will have background music only segments, but not vocal only segments. This prevents training directly on the vocal segments. Thus, the adaptive version learns about vocal segments dynamically by adding spectral basis vectors to represent voice as it's performing the separation [1][5]. We also introduce an SVM at the input stage to create the training samples for the PLCA algorithm to improve automation of the process. Perceptually, we are able to separate background music and vocals in a very noticeable way, and make use of an established method to quantify our results.

Introduction

Semi-blind source separation given a single observation containing multiple sources is a popular and difficult problem. In particular, there has been much work in the area of audio source separation, of in this work's case, vocals and background music [1][2][3][6][7]; however, there are few actual products on the market. Motivating examples range from editing previously mixed songs, creating karaoke tracks, and removing undesirable noise.

Figure 1 gives an overview of our proposed algorithm. First, a support vector machine (SVM) is used to label sections of songs that contain vocals and sections with just background music. This improves upon prior work by allowing for the separation of a large class of songs after training the SVM to classify vocal containing segments in a particular band or genre. The labeled song is then passed to the PCLA algorithm which after taking the Short Time Fourier Transform (STFT), also known as the spectrogram, uses Expectation Maximization (EM) to learn the spectral signature of the background. This spectral signature is then used with the same PCLA algorithm to estimate the spectral signature of the vocals (along with a time signature for the whole mixture). We then can extract both background and vocals by projecting the spectrogram on the learned basis. The following sections will describe the SVM and the PCLA algorithm in detail as well as summarize our results.

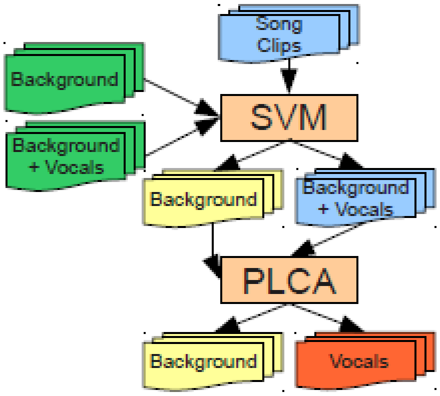


Figure 1: Overview of dataflow in our implementation

SVM

Creating training data for PLCA is tedious since it involves manually classifying song segments in every song that's being separated. To improve the scalability of this process, we experimented with training an SVM to automate this step. It is possible to simply train PLCA directly on the training data used for the SVM; however, PLCA doesn't perform well in this case because the training data isn't local enough to accurately represent the background in the section being separated. By using an SVM on the input phase, we guarantee that the training data is localized to the section that's being separated.

Feature Representation

We experimented with two potential input feature vectors. Based on prior work in the field of audio source separation, we concluded that spectrogram based approaches are most promising and focused our attention on generating an input vector from spectrograms of song segments which have vocals and song segments which don't have vocals. In our first approach, we reshaped the spectrogram matrix into a vector and used that directly as input to the SVM. We were surprised that such a simple approach yielded 85% accuracy with certain parameters.

We also experimented with re-scaling the power spectrum according to the mel frequency scale. The mel scale is derived to be a natural scale with regard to human auditory and speech processes. It is therefore a logarithmic representation of the original power spectrum. We chose to use the mel basis because many speech-from-noise separation algorithms also make use of it. We can also achieve significant reduction in training example dimensionality: while a typical audio power spectrum contains frequency values between 20 Hz and 20,000 Hz, the corresponding mel-representation contains ~50 numbers (Figure 2). With this we were able to achieve nearly 90% accuracy (Figure 3).

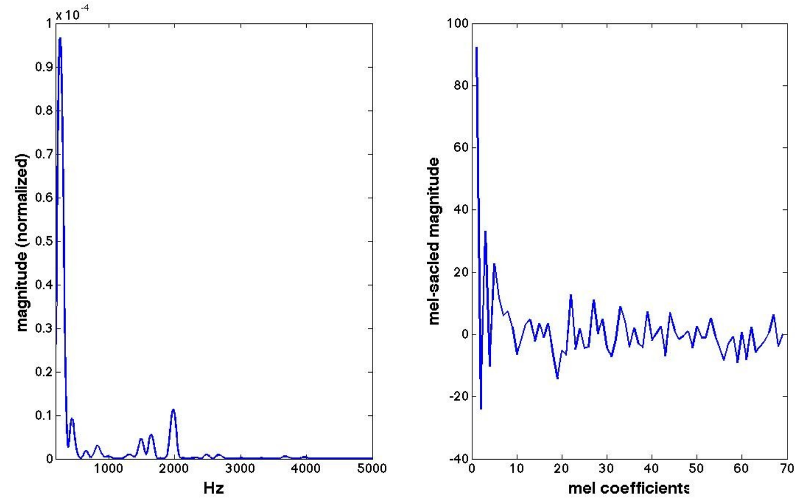


Figure 2: A comparison between the standard time-frequency scale and the corresponding mel-frequency scale. We were able to reduce the dimensionality of our problem significantly by using the mel components.

In addition to using the mel spectrum as feature variables, we also computed spectral properties such as the flatness, brightness, and root-mean squared amplitude. In doing so we were able to gather a wealth of information about a power spectrum and work with it in a relatively low dimensional space. This significantly improves the SVM training efficiency.

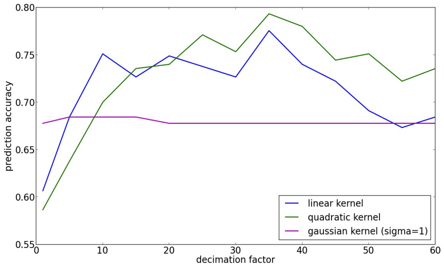
To get the training data, we acquire a library of songs, randomly select a song, and randomly select a corresponding 1-second segment. We then manually determine whether the segment has vocals or not. These segments (with labels) are the inputs to the SVM. This process is mostly automated and very rapid; hundreds of training examples can be gathered in minutes.

Accuracy Evaluation

For estimating prediction accuracy, we used 10-fold cross validation. Our preliminary training set library is the first 10 songs from the Beatles album "Rubber Soul." We observed the effects of using different kernels in the SVM as well as the effects of down sampling the audio segments. We also experimented with changing the number of FFT points and using various windows, but found no significant differences in the results by varying these parameters.

Decimation

We observed the effects of down sampling the audio segments that are chosen as training examples, and fed into the training algorithm. The audio is sampled at 44.1 kHz. By decimating the original audio we were able to achieve a higher SVM prediction accuracy until the decimation factors get large to the point where important information is lost (Figure 2).



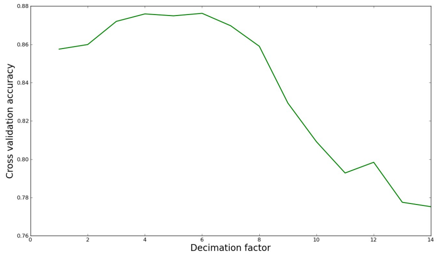


Figure 2 (Left): Effect of decimation factor on accuracy with different kernels using 1024 FFT points, a hanning window of 1024 points, ½ overlap, training on "Closer to You" and using 10-fold cross validation to determine prediction accuracy.
Figure 3 (Right): Effect of decimation factor on accuracy for a quadratic kernel using a vector of the mel components, amplitude, flatness and brightness on a spectrogram with 4096 FFT points, a hanning window of 4096 points, ⅔ overlap, training on a collection of Beatles songs and using 10-fold cross validation to determine prediction accuracy.

Kernels

We experimented with linear, quadratic and gaussian kernels (with varying width). We found that the width of the Gaussian kernel doesn't play a large role in determining the accuracy of the SVM algorithm and performs poorly overall. We achieve the best accuracy with a quadratic kernel (Figure 2).

2D Adaptive PLCA

We implemented 2D PLCA in Matlab based on [1][5]. The algorithm is an EM algorithm that operates on the magnitudes of the spectrograms, S(t,f) where t is time and f is frequency. The spectrogram is normalized and treated as a distribution such that the sum of all (t,f) pairs (quanta) is one. The equations in figure 4 are iterated on until convergence (typically 100 iterations). Refer to [1] and [5] for a detailed explanation of the theory. We later obtained and experimented with a more performance optimized, but functionally identical implementation of 2D PLCA from the authors of [1].

Expectation:

$$P := FZT$$
$$R(f,t) := S(f,t) / P(f,t)$$

Maximization:

$$F := F \circ RT^T$$
$$T := T \circ T^T R$$
$$Z_i := \sum_j F_{ij}$$

Figure 4: Equations used for 2D PLCA on the spectrogram

F is composed of Nz (number of latent variables) columns that represent the probabilities P(f|z), that is some distribution over all frequencies for a given latent variable. Similarly T is composed of rows of temporal distributions P(t|z) and Z is a diagonal matrix of the probabilities P(z) of the latent variables. In such a way, we are able to represent the whole spectrogram as a distribution P = FZT, composed of frequency and time signatures.

Separation performs two passes of PLCA. In the first pass, PLCA is used to find a spectral basis (F_{music}) for the background. In the second pass, PLCA is used to find both a spectral (F_{mixed}) and temporal basis (T) for the mixed signal. However, during this phase, the spectral basis learned for background is kept constant and used as a component of F_{mixed}. Thus, any spectral vectors found during this phase are mainly important in expressing the vocal component, and we have $F_{mixed} = [F_{music} \mid F_{vocals}]$. To extract voice, we project S onto the basis spanned by F and T, then set the magnitudes of the basis vectors of F_{music} to zero to remove the background component. The scaled magnitude spectrogram corresponding to the background is given by $P_{music} = F_{music}ZT$ and the scaled vocals by $P_{vocals} = F_{vocals}ZT$. The signal is then reconstructed by combining these magnitudes with the phase data from the original spectrogram and taking inverse FFTs.

Toy Example

A toy example of separating two sine waves shows the algorithm in action. The background source is considered to be a sine wave at 200 Hz. The vocal source is a sine wave at 500 Hz. The mixed source consists of a linear combination of the background source and the vocal source, shown in figure 5. As parameters, we choose to consider 10 latent variables such that five P(f|z)'s are initially learned for the background and five P(f|z)'s are learned for the vocals given the mixture and holding the background P(f|z)'s constant. The learned P(f|z)'s are shown in figure 6.

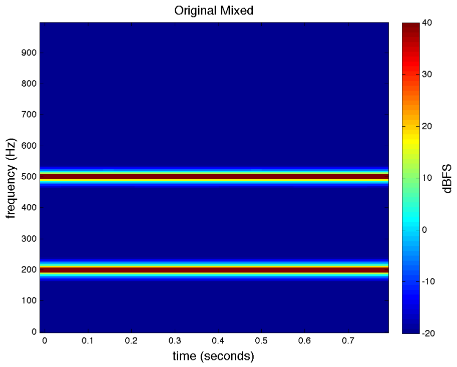


Figure 5: Spectrogram of mixed source

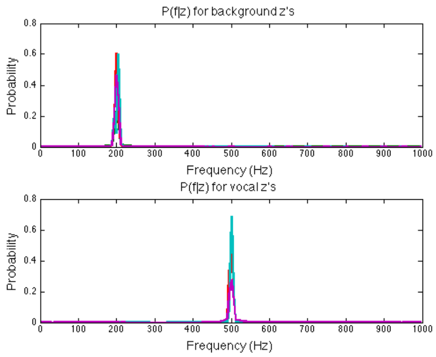


Figure 6: Learned f(f|z) during first PLCA pass

As can be seen, the P(f|z)'s contain the distributions of the frequencies estimated to be the highest likelihood basis for the background and vocals (200 Hz and 500 Hz). Figure 7 then shows the result of separation. Because we have initially trained perfectly on the 200 Hz sine wave, we see a near perfect extraction, and in the second PLCA is able to suppress the 200 Hz wave by more than 30 dB.

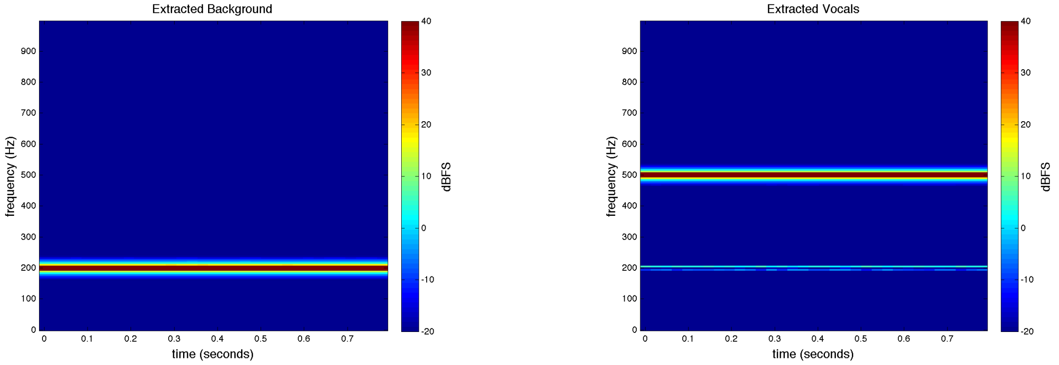


Figure 7: Result of separation after using projecting the spectrogram on the background P(f|z) basis and vocal P(f|z) basis

<u>Separating "Closer To You"</u>

The audio files of the results of the separation of several popular songs can be found at cs229.weebly.com. Figure 8 shows the original spectrogram of a nine second clip from Brandi Carlile's "Closer to You" and the resulting spectrograms of the extracted background and vocals.

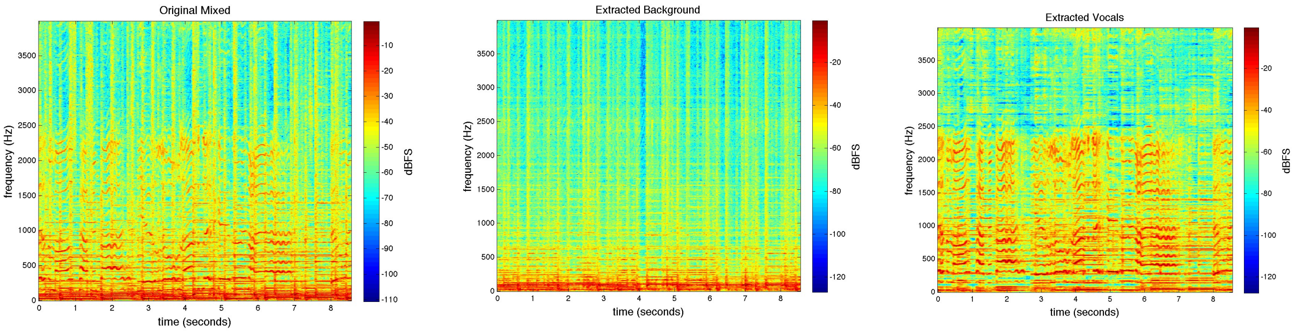


Figure 8: Resulting spectrograms of Closer to You after Separation

The vocals can be seen as the harmonics with a lot of vibrato. The background music is focused more in the lower frequencies of of the spectrum. One can visually see the success of the separation.

<u>Accuracy Evaluation</u>

How well our proposed algorithm performs is inherently a subjective measure as humans will have different perceptual tastes. However, in order to evaluate the performance of our algorithm as well as optimize its various parameters, it is desirable to have some metric that given the actual separate vocal and background tracks for a song will quantify how well the extracted track matches the actual track. This is a very involved (and open) research problem in itself. For instance, we may suppress the vocals of a song very well, however in the process also suppress a portion of the background music we are trying to keep. While the result may sound fairly pleasing, if one was to take a standard cross correlation with the actual background music track, we might find that a maximum value for the correlation would be achieved for a different set of parameters that still audibly left a significant portion of vocals. Thus, for this project we made use of the PEASS toolkit that is meant for evaluating blind audio source separation algorithms [4]. The toolkit computes a target-related perceptual score, an artifact-related perceptual score, an interference-related perceptual score, and an overall combination of these scores. Each score is between 0 and 100, with 100 being the best perceptual match. We found that the overall score generally matched our particular tastes, yet there was still slight variability in our preferences.

<u>Dimensionality of Z</u>

There are many parameters that can be varied in the proposed separation algorithm. These include the number of FFT points for the construction of the spectrogram, the type and size of the time window used in the spectrogram, and the overlap of successive time windows. Based upon trial and error, as well as suggestions in prior work [1], we set the number of FFT points as 4028, and used a hanning window of the same size with an overlap of 1024 samples. We also experimented with decimation of the original 44.1 kHz sampling rate, but ultimately decided upon no decimation.

Another parameter that had a great effect on the quality of separation was the number of latent variables, Nz, chosen for the model. In addition to Nz, the proportion of variables allocated to each source, background Nzb, and vocals Nzv (where Nz = Nzb + Nzv) also greatly affected the quality of separation. The following figures show an experiment performed on the clip from Brandi Carlile's "Closer to You." Nzb and Nzv were swept from 20 to 140 and the overall perceptual score using the PEASS toolkit was evaluated for both the background extraction and the vocal extraction (between 0 - 100).

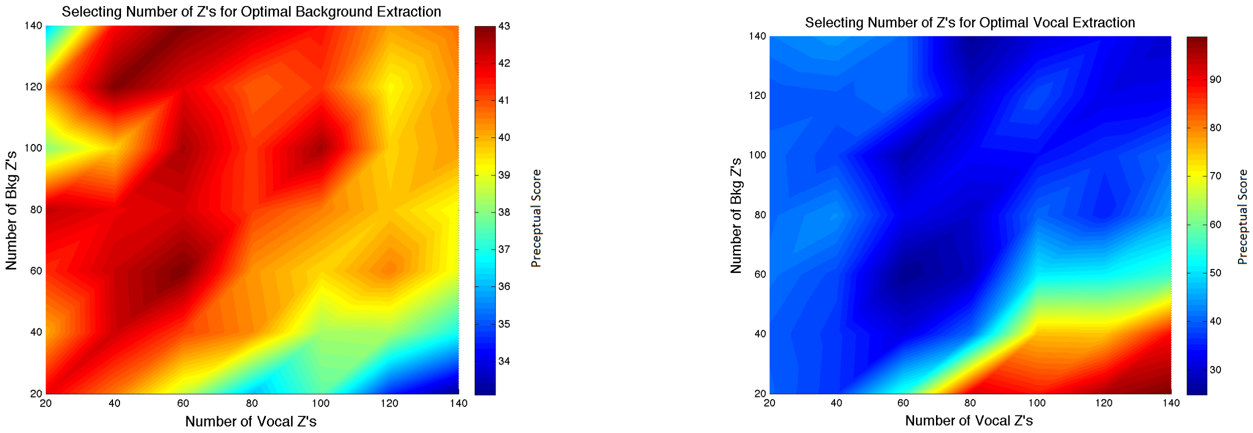


Figure 9: Error according to PEASS for various values of Nzb and Nzv

It should be noted that upon listening to the results of these experiments, we do not necessarily agree that the tracks that received the highest perceptual score were the best. However, the general trend was preserved. In agreeance with the authors of [1], we found that Nz = 200 was a good general number of the latent variables. Further, for optimal background extraction we found that if we proportioned the variables such that Nzb ~ 30 and Nzv ~ 170, we were near the best we could do for background extraction. However, this was not the case for optimal vocal extraction. While figure 9 suggests that we would achieve good results with fewer background latent variables and more vocal latent variables, we found that this ultimately depended on the song selected.

According to [5], the logic behind choosing the number of latent variables for each case is this: if we choose too many variables for either the background or the vocals, then we are prone to capturing components of the undesired track in our spectral description. Further, if we choose too few latent variables, then we are not able to fully represent the extracted source and we might notice it sounds like there are missing high or low frequency components. Unfortunately, we could not easily exploit any trends in our experiments, as the resulting function of Nzb and Nzv is not convex. Thus, we swept the entire space to determine the best separation for each song.

Conclusion

We found that PLCA is useful for separation in low fidelity environments and analyzed the error under various different parameters. We also improved upon prior work by introducing an SVM phase to automate more of the process making it more practical for real world use. Some samples of our results are on cs229.weebly.com. We found that this performs poorly on songs where much of the frequency distribution of the background is close to the vocal range (for example female vocals and guitar). In the future, one could explore using various other transforms (DCT, wavelet, continuous q) to improve accuracy of the SVM as well as separation. Using a measure of correlation between the testing and training could also be used to improve the localization of the distributions learned by PLCA and improve separation.

References

[1] B. Raj, P. Smaragdis, M. Shashanka, and R. Singh, *“Separating a foreground singer from background music,”* International Symposium on Frontiers of Research on Speech and Music, Mysore, India, (2007).
[2] Fuentes, Benoit, Roland Badeau, and Gaël Richard. "Blind Harmonic Adaptive Decomposition Applied to Supervised Source Separation." (2012).
[3] L. Benaroya, F. Bimbot, and R. Gribonval, *“Audio source separation with a single sensor,”* IEEE Transactions on audio, speech, and language processing, **14**(1), 191 (2006).
[4] PEASS Toolkit: http://bass-db.gforge.inria.fr/peass/PEASS-Software.html
[5] Smaragdis, Paris, and Bhiksha Raj. "Shift-invariant probabilistic latent component analysis." *Journal of Machine Learning Research* (2007).
[6] Virtanen, Tuomas. "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria." *Audio, Speech, and Language Processing, IEEE Transactions on* 15.3 (2007): 1066-1074.
[7] Yipeng Li; DeLiang Wang; , "Separation of Singing Voice From Music Accompaniment for Monaural Recordings," *Audio, Speech, and Language Processing, IEEE Transactions on* , vol.15, no.4, pp.1475-1487, May 2007. doi: 10.1109/TASL.2006.889789