SOURCE SEPARATION OF A MIXED AUDIO SIGNAL

BY

VINAY MADDALI

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Bachelor of Science in Electrical and Computer Engineering
in the Undergraduate College of the
University of Illinois at Urbana-Champaign, 2013

Urbana, Illinois

Adviser:

Paris Smaragdis

ABSTRACT

The purpose of this research is to separate individual signals from a mixture of audio signals. Multiple recordings of the same signal were taken. A probabilistic algorithm was used that considers the common components that these recordings share. It considers each note played by an instrument as an individual variable and determines the probability of observing those variables in each of the recordings. These variables are then grouped according to the number of sources using a clustering algorithm. The probabilties of observing the variables and where they belong to are taken and compared with the spectrogram to obtain the sound of each source separately. Experiments were conducted on some signals in a room environment and the method has been shown to extract the signals.

To my parents, for their love and support.

ACKNOWLEDGMENTS

I would like to thank my advisor Professor Paris Smaragdis and Computer Science graduate student Minje Kim without whose help this thesis would not have been possible. It was due to the weekly research meetings that I got to learn so much in the field of machine learning and it applications in signal processing. I would also like to thank them for their extended support and advice that they gave for my graduate school applications.

My undergraduate experience in ECE at the University of Illinois has been fun and enriching. I got to learn not only from the classes but also from my friends and classmates in this department. It has been a great experience working on multiple projects with Bilal Gabula, Sartaj Grewal and Mikhail Kandel. Working on the ECE 391 operating system with Rajarshi Roy, Akshay Chanana and Gautam Vunnam was a great experience. I would also like to thank other people in ECE Illinois: Torin Kilpatrick, Gerard Mccann, Siddhartha Gupta who have helped me succeed in my class work.

TABLE OF CONTENTS

LIST OF SYMBOLS

W	Matrix representing the frequency of each component in the spectrogram
H	Matrix representing the time activation of each component in the spectrogram
$P(\boldsymbol{f} \mid z)$	Probability of observing the frequency for the given latent variable
$P(t \mid z)$	Probability of observing the time for the given latent variable
$P(z)$	Weight specified by each latent variable
$P(z \mid \boldsymbol{f}, t)$	Posterior probability of observing the specific latent variable given the frequency and time
C	A single cluster of data
μ_j	Mean of a gaussian distribution
σ_j^2	Co-variance of a gaussian distribution
π_j	Prior probability that the data belongs to a source
γ_{nk}	Probability that a data point belongs to a Gaussian mixture

CHAPTER 1

INTRODUCTION

The emergence of computers has simplified various operations and has helped achieve what was unimaginable before. The advance of computation technology has also helped in carrying out various operations in a feasible amount of time. Music can be generated using algorithms in the computer, and synthesizers nowadays can be used to generate various sounds according to the user's needs. The ability to convert signals from real world to digital has given birth to the topic of digital signal processing. The ability to process a signal in the computer has greatly relieved the stress on hardware designers and has even led to more efficiency. Digital signal processing has had a major effect on the audio and music industry. Development of robust hearing aid technologies, music recognition software and speech recognition software has been possible because of digital signal processing. The application of the principles of machine learning in DSP has boosted the variety of uses that it can be put to.

Musicians playing in a quartet or even a band will find it more convenient to hear the sound of their own instrument while practicing. This need can be met through audio source separation where the signal of the drummer or guitarist alone can be separated from a recording of a practice session or live performance of the band. In this way, the musician has an idea as to how well he/she is playing while playing with the band as such.

Techniques combining machine learning and signal processing can be used for the purpose of audio source separation. The concept of non-negative matrix factorization (NMF) was used in order to separate the components of a spectrogram and observe their time and frequency activities [1]. But such a simple method is not sufficient when we consider a complex or real world signal. A more statistical method of probabilistic latent component analysis (PLCA) can be used in order to do this source separation. We need multiple instances of the same mixed signal in order to identify the common

components among them and also the magnitude of that component in each instance.

But we now have a problem as to which origin or source each component belongs to. This is solved by regarding the whole posterior matrix obtained for each component as a Gaussian mixture model (GMM) and dividing the data into different classes accroding to the number of sources. In this way, we can almost accurately put together the correct components for each source in order to reconstruct it.

CHAPTER 2

PRIOR WORK

The task of source separation is challenging and has set many limitations in the work done previously. There is always the problem as to how we can identify the different sources in a signal. This problem has been tackled in various ways by researchers in the past.

One specific work on source separation was the non-negative matrix partial co-factorization(NMPCF) used for drum source separation. This method used a prior knowledge of the drums by having a spectrogram of the drum only signal [2]. Non-negative matrix factorization(NMF) alone just decomposes the spectrogram of the whole signal into its frequency components and their time activations. But there is no way to identify the components that belong to the drums. So a drum-only signal can be used so that we can co-factorize the spectrogram to identify the frequency and time components of the drum signals. This method works by first taking the spectrogram of the drum-only signal and dividing it into its frequency bases and time activation matrices. After we obtain the frequency and time activations of our mixed signal with drums, we identify the common components between the frequency bases of the drum-only signal and the actual signal by a simple method of co-factorization. This way we can accurately separate out the drum frequencies from the rest of the instruments. Such a method can be used to extract the sound of drums from any signal with some prior knowledge of the drums.

Another useful work on source separation is probabilistic latent component sharing used in audio enhancement. This concept is a modified version of probabilistic latent component analysis(PLCA) and takes into consideration the common components that the recordings share. It can be used in cases of recordings that were taken in live concerts and have some disturbance in the background [3].This unwanted noise or disturbances can be eliminated by taking another cleaner recording even with a lower quality of sound or

the actual song itself. So the recordings can have some common components among them which will be identified by this algorithm. The individual components in each recording which mainly consist of unwanted sounds can be separated out so that we can get a clean sound of the high quality recording that we have.

The algorithm works by first maximizing the logarithmic probability of obtaining the spectrograms of the respective recordings. This way we can obtain an estimate of observing the probabilities of frequency, time and individual weights of the latent variables. An expectation maximization algorithm is used to calculate the posterior probabilities of obtaining the latent variables given frequency and time. So these parameters are calculated for each individual recording. The common components are then separated from the individual components, and hence, the cleaner version of the high quality recording is obtained. This method doesn't really need clustering of the data as there are only two sources we want to separate out. In each recording, we can clearly identify the common latent variables between the recordings and their individual components.

This algorithm using probabilistic latent component sharing has been used in this thesis for a slightly different purpose. Individual components do not exist for each recording, but they share some common variables between them. So in order to separate out the individual sources in a recording, we need to cluster the data according to what source they belong to.

CHAPTER 3

DERIVATION

The spectrogram of any recording can be divided into two matrices, one with same rows as the spectrogram and the other with the same number of columns. The other parameter of the size of these matrices is common and represents the individual elements in the signal [1]. R has to be chosen so that all the individual elements in the recording are represented appropriately. An R greater than the number of elements in the signal does not do any harm as its own frequency and time plot will represent no note being present there [1].

$$V \approx WH \tag{3.1}$$

W shows the frequency of the components in the recording. H gives the time activation of the respective components. The distance given below has to be minimized so that we get a close estimate of the activity in the audio signal

$$| V - WH |^2 \tag{3.2}$$

The algorithm used for this is a convergence based formula where we start with some random values of W and get H from $V = WH$ [4]. The equations for convergence are used:

$$H = H\frac{(W^T V)_{rm}}{(W^T WH)_{rm}} \tag{3.3}$$

$$W = W\frac{(VH^T)_{nr}}{(WHH^T)_{nr}} \tag{3.4}$$

These two equations are repeated until convergence is achieved. The objective of this iteration is to minimize the distance or the cost function as much as possible. This convergence formula can be proved using an auxiliary

function used in the Expectation-Maximization algorithm(EM) [4].

To obtain the common components between the recordings, a probabilistic approach of NMF is taken so that it becomes a statistical approach [5]. This method introduces a third matrix representing latent variable z specific to each component in the signal. It is a diagonal matrix that gives the individual weights for each component. The three matrices are therefore represented as a product of $P(f \mid z)$, $P(t \mid z)$ and $P(z)$ [5].

$$P(f, t) = \sum_{z} P(f \mid z)P(t \mid z)P(z) \tag{3.5}$$

$$P = \sum_{ft} V_{ft} \log P(f, t) \tag{3.6}$$

The objective is to find the appropriate values to maximize the log likelihood P of obtaining the input spectrogram with the probabilistic parameters [3]. But finding this using the maximum slope method is very inefficient and tedious as taking the derivative of log sum of $P(f \mid z)$. $P(t \mid z)$. $P(z)$ is not feasible. So a different approach is taken for this: An algorithm with expectation and maximization steps is used where we iterate between these two steps until convergence [6]. In the expectation step, we get the probability of the latent variable z given time and frequency.

$$P(z \mid f, t) = \frac{P(f \mid z)P(t \mid z)P(z)}{\sum_{z} P(f \mid z)P(t \mid z)P(z)} \tag{3.7}$$

The maximization step recomputes the initial parameters by comparing with the spectrogram:

$$P(f \mid z) = \frac{\sum_{t} V_{ft} P(z \mid f, t)}{\sum_{f} \sum_{t} V_{ft} P(z \mid f, t)} \tag{3.8}$$

$$P(t \mid z) = \frac{\sum_{f} V_{ft} P(z \mid f, t)}{\sum_{f} \sum_{t} V_{ft} P(z \mid f, t)} \tag{3.9}$$

$$P(z) = \frac{\sum_{f} \sum_{t} V_{ft} P(z \mid f, t)}{\sum_{f} \sum_{t} \sum_{z} V_{ft} P(z \mid f, t)} \tag{3.10}$$

Now we consider the case where we have multiple recordings of the same mixed signal. So we have say n spectrograms as inputs. We consider latent variables zc common among all the recordings [3]. So the probability of

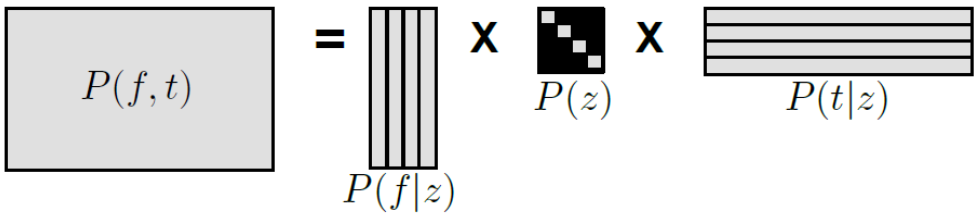

Figure 3.1: Representation of how the matrix is divided according to PLCA [3]

getting the input matrix that we are looking to maximize is:

$$P = \sum_{n} \sum_{f,t} V_{ft} log \sum_{z} P(f \mid z) P(t \mid z) P^{(n)}(z) \tag{3.11}$$

This includes the summation over all the recordings that we have

The E-step of the PLCS now involves the probability of the latent variable given the time and frequency specific to each recording

$$P(z^{(n)} \mid f, t) = \frac{P(f \mid z) P(t \mid z) P^{(n)}(z)}{\sum_{z} P(f \mid z) P(t \mid z) P^{(n)}(z)} \tag{3.12}$$

The M-step now requires the computation of $P(f \mid z)$ and $P(t \mid z)$ which will be common across all the recordings:

$$P(f \mid z) = \frac{\sum_{n} \sum_{t} V_{ft}^{(n)} P^{(n)}(z \mid f, t)}{\sum_{n} \sum_{f} \sum_{t} V_{ft}^{(n)} P^{(n)}(z \mid f, t)} \tag{3.13}$$

$$P(t \mid z) = \frac{\sum_{n} \sum_{f} V_{ft}^{(n)} P^{(n)}(z \mid f, t)}{\sum_{n} \sum_{f} \sum_{t} V_{ft}^{(n)} P^{(n)}(z \mid f, t)} \tag{3.14}$$

recomputation of P(z):

$$P^{(n)}(z) = \frac{\sum_{f} \sum_{t} V_{ft}^{(n)} P^{(n)}(z \mid f, t)}{\sum_{f} \sum_{t} \sum_{z} V_{ft}^{(n)} P^{(n)}(z \mid f, t)} \tag{3.15}$$

To recover the activity of the specific latent variable from each recording, we have to multiply the posterior probability of that variable $P(z \mid f, t)$ to the input spectrogram for that specific recording.

$$S^{(n)} = \sum_{z} V_{ft} P^{(n)}(z \mid f, t) \tag{3.16}$$

CHAPTER 4

CLUSTERING

We have divided the recording into different latent variables and also found the activity of each variable individually. We can reconstruct the sound of each individual variable with equation (2.16). But now we have to group these variables accordingly so that the whole sound of the source is represented accurately.

So first we have to know the number of sources which is taken as a given parameter. This is important as we need to know how many classes or groups exist in order to divide the variables. We have to divide the data into clusters that appropriately represent the two sources.

There a lot of methods to do this. One is the agglomerative algorithm which basically involves grouping the data into groups and then merging those groups into bigger groups [7]. Two groups with the smallest eucledian distance or any other criteria taken are merged in every step and this is repeated in each step until there is only one cluster representing the whole data. For our purpose, we can stop this merging of clusters when there is only the source number of cluster groups left.

If we represent our data as a single row vector:

$$X = \{x_1, x_2, x_3, ..., x_n\} \tag{4.1}$$

We represent each point as a single cluster C. At the first step we have:

$$R = \{C_1, C_2, C_3, ..., C_n\} \tag{4.2}$$

We then merge clusters with the minimum distance between each other at every step and then get the new R with the new clusters:

$$mindist\{C_a, C_b\} \tag{4.3}$$

The new R is:

$$R = (R_{prev} - \{C_a, C_b\}) \cup \{C_a \cup C_b\} \tag{4.4}$$

Equations (3.3) and (3.4) are repeated until we have only source number of clusters in R. This should give an accurate representation of the data belonging to both the sources.

Another approach is the divisive algorithm which is the inverse of the agglomerative algorithm [7]. In this case, we take all the data points together as a single cluster and then divide clusters in each step with the criteria being the maximum distance.

We begin with:

$$R = \{X\} \tag{4.5}$$

At every step, we find sub-clusters in each cluster that have the maximum distance between each other and separate them accordingly:

$$maxdist\{C_{a,x}, C_{a,y}\} \tag{4.6}$$

The new R is then:

$$R = (R_{prev} - \{C_a\}) \cup \{C_{a,x}, C_{a,y}\} \tag{4.7}$$

The problem with both these approaches is that they are very inefficient when we have a large set of data points. This is very likely in our case as we have real world recordings with a significant length and a very high sampling rate. So using these algorithms will not be feasible. For the agglomerative algorithm, the total number of comparisons we have are:

$$\sum_{t=0}^{N-1} \binom{N-t}{2} \tag{4.8}$$

where N is the number of data points we have. We have to choose from $N - t$ number of clusters at a given step t to merge. This is very big and similarily for the divisive algorithm the comparison step is very computation intensive.

A more sensible approach to clustering would be to use an iterative method like expectation-maximization algorithm in the previous chapter. Each cluster can be defined as a gaussian distribution and we can use a probabilistic

approach to determine where the centers are located and as to which data points belong to which center [7].

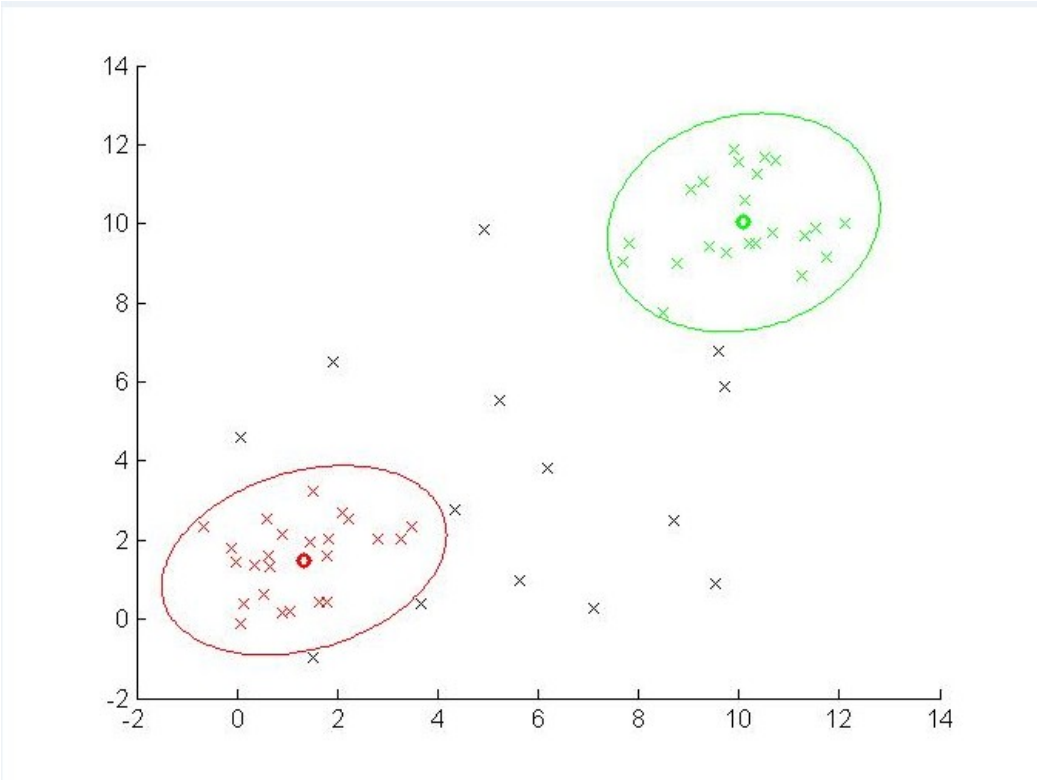

Figure 4.1: A Gaussian mixture model with two centers [8]

In order to obtain the source number of gaussian distributions, we have to get the ILD between the recordings. In our case we have two sources and two recordings, so we obtain the ILD between the recordings:

$$ILD_{1by2} = 20log_{10}\frac{abs(V_{ft}^{(1)}P(z^{(1)} \mid f,t))}{abs(V_{ft}^{(2)}P(z^{(2)} \mid f,t))} \tag{4.9}$$

We then convert this ILD into a single row vector. We have say k Gaussian distributions and with means μ_j, covariance σ_j [9] and prior probability π_j. We use the same iterative method with the expectation and maximization step so that the parmeters converge to the right values.

We first pick random values of μ_j, σ_j and π_k. The E-step then involves computation of γ_{nk} which is the probability that a data point belongs to a gaussian.

$$\gamma_{nk} = \frac{\pi_k N(x_n \mid \mu_k, \sigma_k)}{\sum_j \pi_j N(x_n \mid \mu_j, \sigma_j)} \tag{4.10}$$

The M-step then recomputes these parameters with the obtained γ_{nk}:

$$N_k = \sum_n \gamma_{nk} \tag{4.11}$$

$$\mu_k = \frac{1}{N_k} \sum_n \gamma_{nk} x_n \tag{4.12}$$

$$\sigma_k^2 = \frac{1}{N_k} \sum_n \gamma_{nk} (x_n - \mu_k)(x_n - \mu_k)^T \tag{4.13}$$

$$\pi_k = \frac{N_k}{N} \tag{4.14}$$

Once we get the accurate values of posterior probabailities for the gaussian mixtures, we multiply these values with the posterior probability obtained from equation (2.12) and with the spectrogram to obtain the individual spectrogram of each source. The gaussian mixture that we take for reconstruction is usually the one with the positive mean for the reconstruction of the source in the first recording and the one with the negative mean for the second recording. This is because we calculate ILD by subtracting second recording from the first.

$$S_I = \sum_z \gamma_{zI} P^I(z \mid \boldsymbol{ft}) V_I \tag{4.15}$$

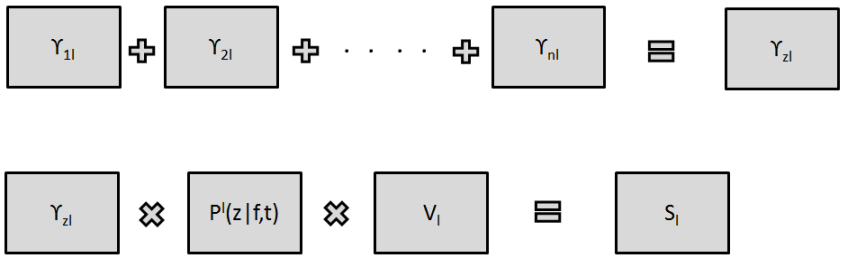

Figure 4.2: A representation of how the spectrogram of the reconstructed signal is obtained

CHAPTER 5

EXPERIMENTAL RESULTS

The probabilistic latent component sharing was tested with two violins playing one note each. This way we do not have to worry about clustering as we have only two latent variables and know what source they belong to. The spectrograms of the two recordings of two sources are shown:

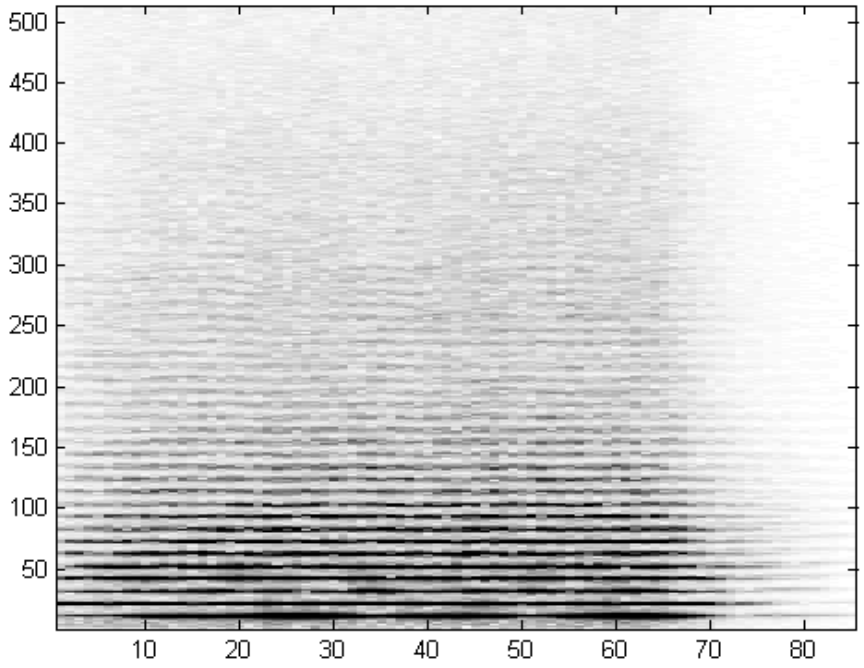

Figure 5.1: Spectrogram of first recording

The first spectrogram has a higher magnitude centered on the left and seond has a higher magnitude on the right. This is because the violins play the notes one after the other. The first sensor was placed next to the first violin and vice versa for the second.

Now the probabilistic latent component sharing algorithm was implemented

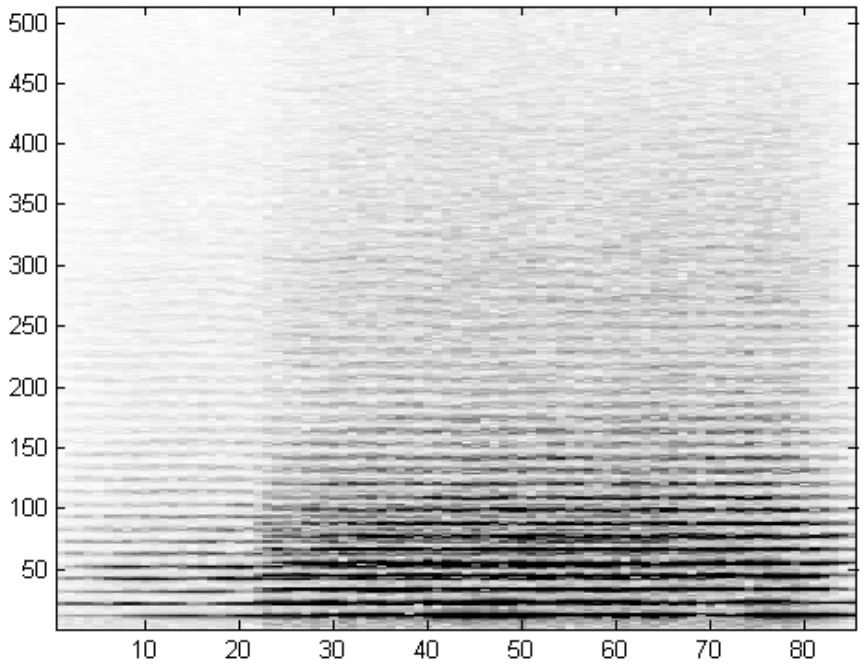

Figure 5.2: Spectrogram of second recording

on these two spectrograms. The spectrogram from the instrument placed further away for each sensor has not been shown. But a signal with lesser magnitude has been obtained for each. The extracted signals represent each source cleanly:

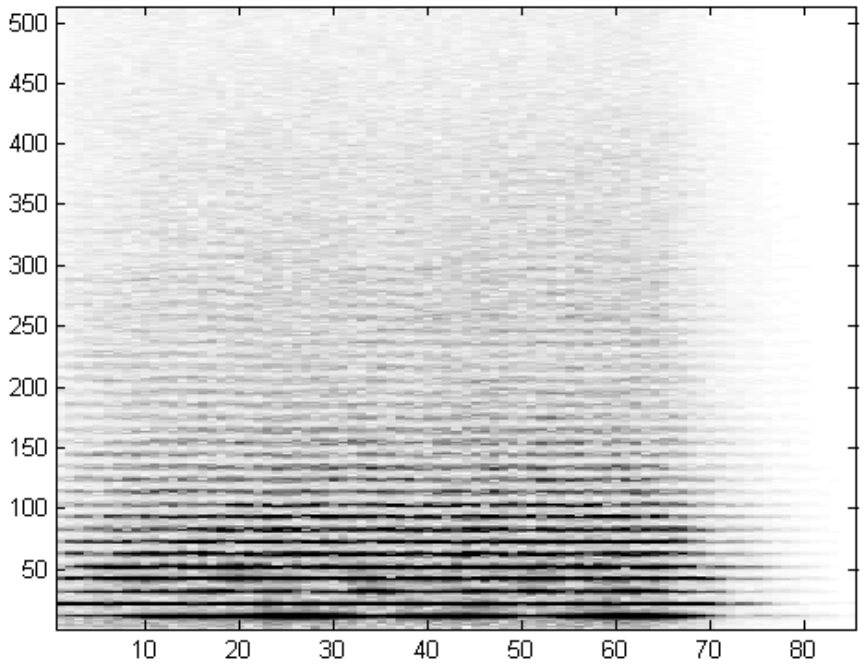

Figure 5.3: Spectrogram of first source from first recording

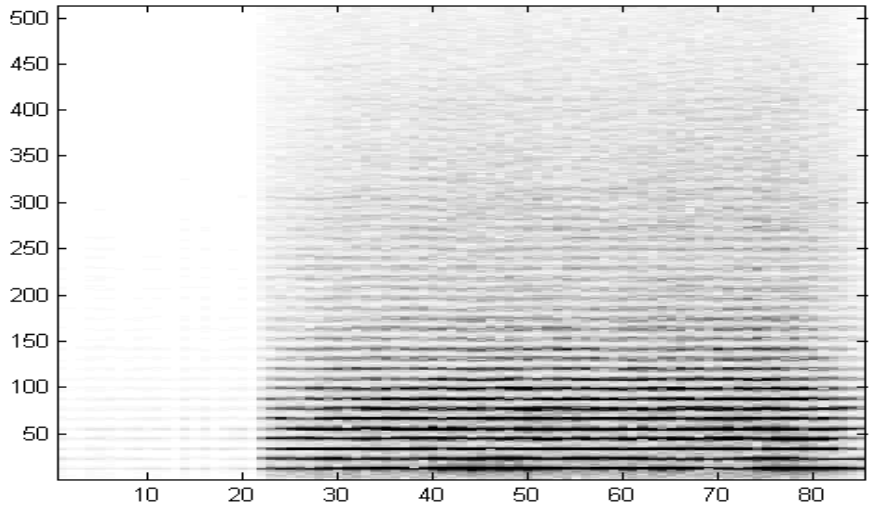

Figure 5.4: Spectrogram of second source from second recording

After this, two simulated recordings of two sources located at different locations in a room environment acting at the same time were taken for source separation. The number of latent variables assumed in this case were 50. After using the PLCA and clustering algorithms on these two recordings, the two signals were separated in a fairly clean manner.

The cases tested were that of two female voices, two male voices and a male and female voice. To determine the accuracy of the extracted signal compared to the actual original signal of that signal alone, we calculate the Signal to Distortion ratio:

$$S-D-R = 10log_{10}\frac{{}_t\{V_{orig}^{(f)}(t)\}^2}{{}_t\{V_{orig}^{(f)}(t) - V_{recons}^{(f)}(t)\}^2} \qquad (5.1)$$

For each case the gaussian curve representing the total distribution of the ILD between the two recordings is shown along with the signal to distortion ratio.

Case of both female voices:

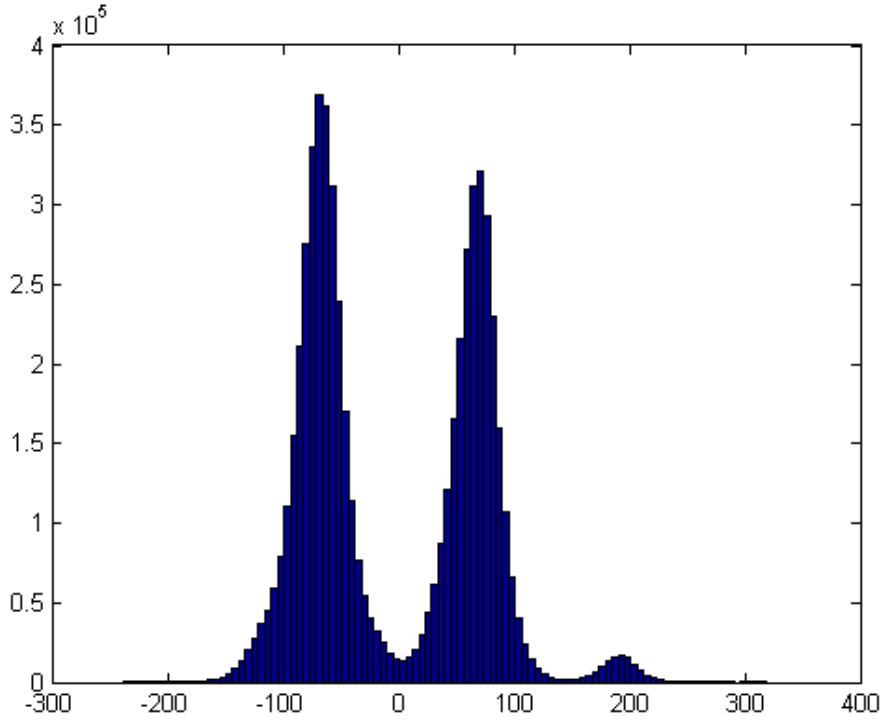

Figure 5.5: Gaussian curve of both female voices

Signal-to-distortion ratio of the first female voice = 79.84

Signal-to-distortion ratio of the second female voice = 62.1201

Case of both male voices:

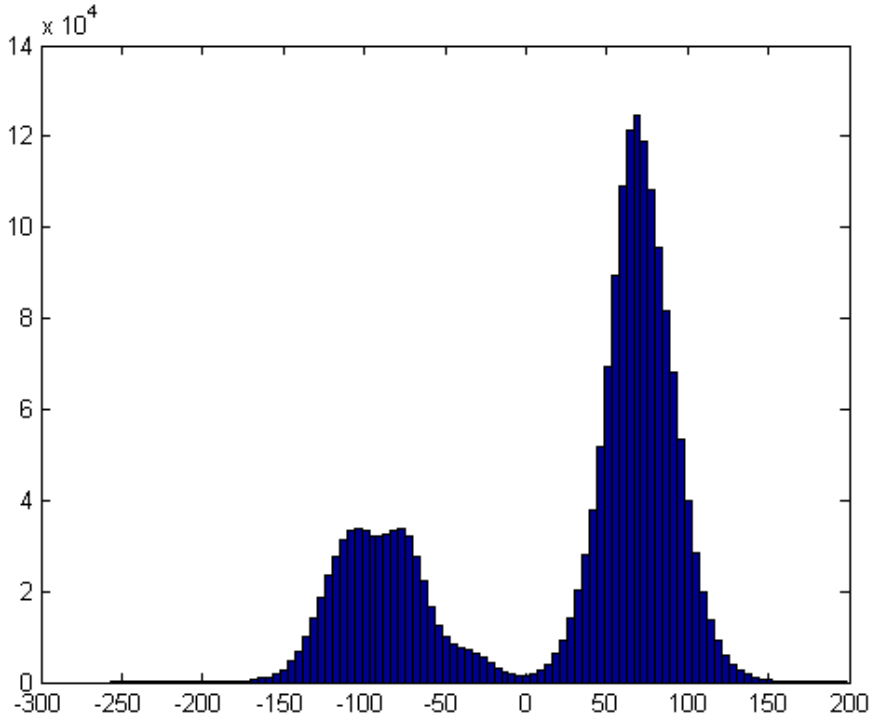

Figure 5.6: Gaussian curve of both male voices

Signal-to-distortion ratio of the first male voice = 80.91

Signal-to-distortion ratio of the second male voice = 49.82

Case of a male and female voice:

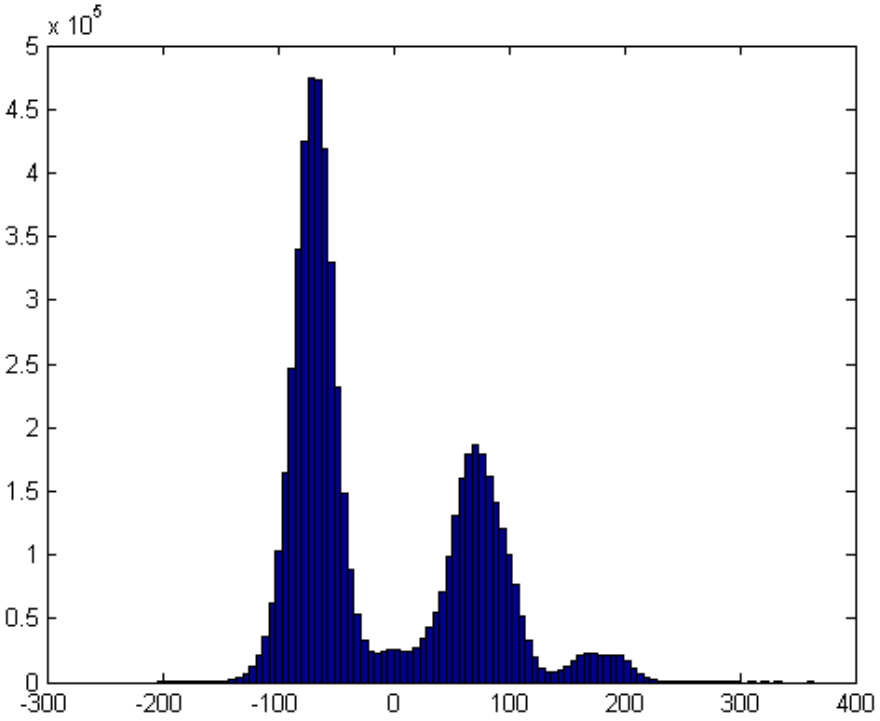

Figure 5.7: Gaussian curve of a male and female voice

Signal-to-distortion ratio of the first male voice = 75.71

Signal-to-distortion ratio of the second male voice = 60.61

As it can be seen the signal to distortion ratios are very high in magnitude for each case which says that the spectrogram of the reconstructed signal is very close to that of the original signal.

CHAPTER 6

LIMITATIONS AND FURTHER RESEARCH

As we saw in the previous section, the individual signals in all the recordings are being separated with less distortion and attenuation and represent the original signal almost accurately. But the case that was taken is very superficial and real-world cases may be more complex than this. Musicians cannot always represent the exact environment that we set here.

In our case we took the sources to be placed very close to a microphone and away from each other. So in each recording,it was easy to distinquish between the higher magnitude and lower magnitude sources. Due to this, when we calculated the ILD between the two recordings, we got two significantly disntinct peaks with the means clearly away from each other. But in the situations of having the two microphones close to each other or having two sources close to one microphone, we'll have a histogram that will not distinquish the data of the two sources that clearly. So the clustering in this case will not give an accurate result as it is not clear as to whether the latent variable belongs to which source. A case in which the two sources were placed close to each other was taken to test. The histogram had two peaks with their gaussians overlapping with each other.

The second factor is that this research is limited to the source separation of a mixture of just two signals. We can further consider the case of having three or more sources with the same number of recordings for each as the sources. This becomes more complicated as now each recording will have the sound of four sources. There will be four gaussian mixtures that we have to differentiate from. The question arises as to which recordings the ILD has to be taken between and whether just one ILD is enough to extract a source or should we compare the ILDs between multiple recordings in order for extraction.

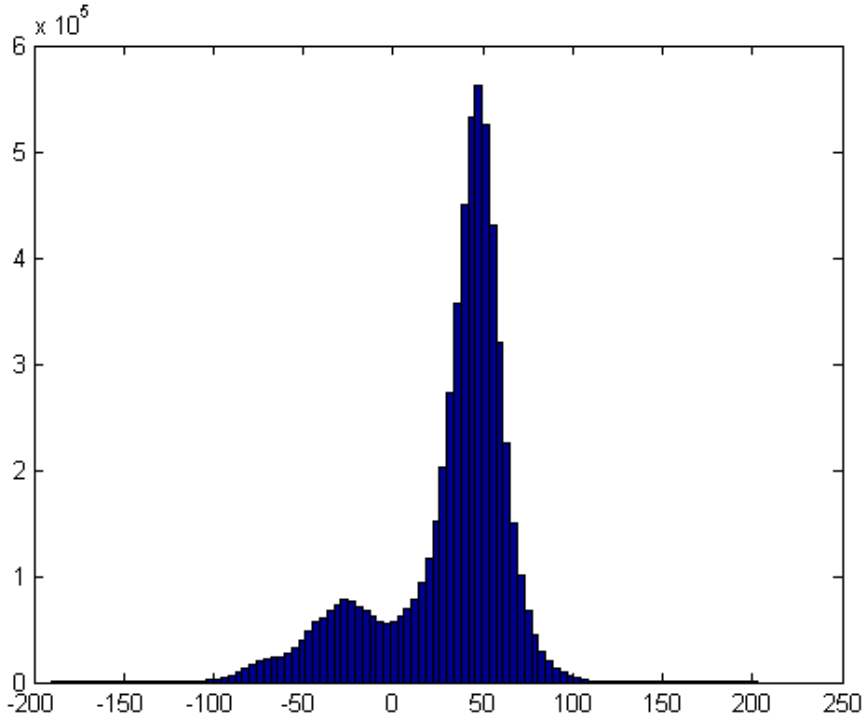

Figure 6.1: Histogram of ILD with two sources very close to one mic

REFERENCES

[1] J. C. Brown, P. Smaragdis, "Non-negative matrix factorization for polyphonic music transcription," *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 177–180, 2003.

[2] K. Kang. M. Kim, J. Yoo and S. Choi, "Nonnegative matrix partial co-factorization for spectral and temporal drum source separation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1192–1204, 2011.

[3] P. Smaragdis. M. Kim, "Collaborative audio enhancement using probabilistic latent component sharing," *IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, Canada*, 2013.

[4] H. S. Seung. D. D. Lee, "Algorithms for non-negative matrix factorization," *NIPS*, vol. 13, 2000.

[5] T. Hofmann, "Probabilistic latent semantic analysis," in *Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval (SIGIR99)*, 1999.

[6] M. Shashanka, P. Smaragdis, B. Raj, "A probabilistic latent variable model for acoustic modeling, advances in models for acoustic processing workshop," *NIPS*, 2006.

[7] P. Smaragdis, "Clustering," CS 598 Lecture Fall 2012 pp 1–77.

[8] "Em algorithm for gaussian mixture model with background noise," Mathworks. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/36721-em-algorithm-for-gaussian-mixture-model-with-background-noise/content

[9] A. Krishnamurthy, "High-dimensional clustering with sparse gaussian mixture models." *Carnegie Mellon University*, 2011.