2013-10

Non-Negative Matrix Factorization Based Algorithms to Cluster Frequency Basis Functions for Monaural Sound Source Separation.

Rajesh Jaiswal
Technological University Dublin, rajesh.jaiswal@tudublin.ie

Non-negative Matrix Factorization based Algorithms to cluster Frequency Basis Functions for Monaural Sound Source Separation

PhD Thesis

Rajesh Jaiswal

Audio Research Group

School of Electrical Engineering Systems

Dublin Institute of Technology

Dublin, Ireland

October 2013

Supervisors: Dr. Derry FitzGerald, Mr. Dan Barry, Dr. Scott Rickard and Prof. Eugene Coyle

Abstract

Monophonic sound source separation (SSS) refers to a process that separates out audio signals produced from the individual sound sources in a given acoustic mixture, when the mixture signal is recorded using one microphone or is directly recorded onto one reproduction channel. Many audio applications such as pitch modification and automatic music transcription would benefit from the availability of segregated sound sources from the mixture of audio signals for further processing.

Recently, Non-negative matrix factorization (NMF) has found application in monaural audio source separation due to its ability to factorize audio spectrograms into additive part-based basis functions, where the parts typically correspond to individual notes or chords in music. An advantage of NMF is that there can be a single basis function for each note played by a given instrument, thereby capturing changes in timbre with pitch for each instrument or source. However, these basis functions need to be clustered to their respective sources for the reconstruction of the individual source signals.

Many clustering methods have been proposed to map the separated signals

into sources with considerable success. Recently, to avoid the need of clustering, Shifted NMF (SNMF) was proposed, which assumes that the timbre of a note is constant for all the pitches produced by an instrument. SNMF has two drawbacks. Firstly, the assumption that the timbre of the notes played by an instrument remains constant, is not true in general. Secondly, the SNMF method uses the Constant Q transform (CQT) and the lack of a true inverse of the CQT results in compromising on separation quality of the reconstructed signal.

The principal aim of this thesis is to attempt to solve the problem of clustering NMF basis functions. Our first major contribution is the use of SNMF as a method of clustering the basis functions obtained via standard NMF. The proposed SNMF clustering method aims to cluster the frequency basis functions obtained via standard NMF to their respective sources by making use of shift invariance in a log-frequency domain.

Further, a minor contribution is made by improving the separation performance of the standard SNMF algorithm (here used directly to separate sources) obtained through the use of an improved inverse CQT. Here, the standard SNMF algorithm finds shift-invariance in a CQ spectrogram, that contain the frequency basis functions, obtained directly from the spectrogram of the audio mixture.

Our next contribution is an improvement in the SNMF clustering algorithm through the incorporation of the CQT matrix inside the SNMF model in order to avoid the need of an inverse CQT to reconstruct the clustered NMF basis

functions.

Another major contribution deals with the incorporation of a constraint called group sparsity (GS) into the SNMF clustering algorithm at two stages to improve clustering. The effect of the GS is evaluated on various SNMF clustering algorithms proposed in this thesis.

Finally, we have introduced a new family of masks to reconstruct the original signal from the clustered basis functions and compared their performance to the generalized Wiener filter masks using three different factorisation-based separation algorithms. We show that better separation performance can be achieved by using the proposed family of masks.

Declaration

I, Rajesh Jaiswal, certify that this thesis which I now submit for examination for the award of PhD, is entirely my own work and has not been taken from the work of others, save and to extent that such work has been cited and acknowledged within the text of my work.

This thesis was prepared according to the regulations for postgraduate study by research of the Dublin Institute of Technology and has not been submitted in whole or in part for another award in any other third level institution.

The work reported on in this thesis conforms to the principles and requirements of the DIT's guidelines for ethics in research.

DIT has permission to keep, lend or copy this thesis in whole or in part, on condition that any such use of the material of the thesis be duly acknowledged.

Signature _____ Date _____

Candidate

Acknowledgement

This work has been carried out in Audio research Group Lab at Dublin Institute of Technology, Ireland during 2009-2013.

The work presented in this thesis would not have been possible without the help of many people who were always there when I needed them the most. I take this opportunity to acknowledge them and extend my sincere gratitude for helping me make this Ph.D. thesis a possibility.

First and foremost I offer my deepest gratitude to my principal supervisor, Dr. Derry Fitzgerald for all his support, guidance and encouragement to continue this work. This thesis would not exist without him being always around to provide all his expertise and knowledge. I would like to specially thank him for always believing in my ability to complete the work in time especially for encouraging the final effort to complete the thesis.

I would like to thank Dr. Eugene Coyle for his constant support and encouragement for my work in Dublin Institute of Technology. I express my sincere gratitude to the members of Audio Research Group, Dan Barry, Dr. Mikel Gainza, Dr. Alan O Cinneide, Martin Gallagher and Dr. Cillian Kelly for

their involvement in making the environment in lab more enjoyable and relaxing to work in.

I would like to thank Dan Barry for proof-reading this thesis and for helpful discussions on audio signal processing.

This work is funded by ABBEST PhD Research Scholarship at Dublin Institute of Technology, Dublin, Ireland. This financial support is greatly acknowledged.

I would like to thank all the staff of the graduate research school especially Gerolmina Di Nardo, Raffaella Salvante and Mary O'Toole for their support and advices at all the stages of my research studies. My especial thank to them for solving problems related to immigration status.

Thanks also to Danute Zilbere and John Russell for helping me in hard times throughout the research work. I greatly acknowledge their encouragement and support.

I wish to thank my family members my sister, Dr. Swarna Jaiswal, my brother, Rakesh Jaiswal and my brother-in-law, Dr. Amit Jaiswal for their support in all my efforts. This thesis is dedicated to my mother, Sita Jaiswal for her encouragement, love and support to my education for all these years so that today I can stand proud with my head held high.

Finally, I thank the Almighty, for blessing me with the strength to work through all these years and teaching me through many failures to achieve any success in my life.

Abbreviations and Notations

Algorithms

- NMF → Non-negative Matrix Factorisation 1.5.4.
- SNMF_{cqt} → Standard Shifted NMF 1.5.6. The standard SNMF algorithm finds shift invariance in the constant Q spectrogram obtained directly from the audio spectrogram.
- SNMF_{ncqt} → Standard Shifted NMF using invertible CQT(approximate) 3.4.
- SNMF_{gncqt} → SNMF_{ncqt} with group sparsity 5.6
- SNMF_{map} → Shifted NMF clustering using one-to-one mapping 2.3.2. The Shifted NMF clustering algorithm finds shift invariance in the log-frequency domain frequency basis functions obtained via NMF.
- SNMF_{mask} → Shifted NMF clustering using spectral masking 2.3.2.

- SNMF$_{lmap}$ → Shifted NMF clustering with CQT incorporated into SNMF model and the signal is reconstructed using one-to-one mapping 4.2.
- SNMF$_{lmask}$ → Shifted NMF clustering with CQT incorporated into SNMF model and the signal is reconstructed using spectral masking.4.2.

  The SNMF clustering algorithm is considered to have two stages. Firstly, the NMF stage, where the NMF basis functions is calculated and Secondly, the clustering stage, where the basis functions is clustered using SNMF.

  1^{st} stage 5.3
- NMF$_{kl}$ → NMF using KL divergence
- NMF$_{gkl}$ → NMF using KL divergence with group sparsity
- NMF$_{is}$ → NMF using IS divergence
- NMF$_{gis}$ → NMF using IS divergence with group sparsity

  2^{nd} stage 5.4
- SNMF$_{kl}$ → SNMF clustering using KL divergence
- SNMF$_{gkl}$ → SNMF clustering using KL divergence with group sparsity
- SNMF$_{is}$ → SNMF clustering using IS divergence
- SNMF$_{gis}$ → SNMF clustering using IS divergence with group sparsity

  The combination of two stages of a SNMF clustering algorithm is denoted by SNMF$_{gkl-gkl}$ where '-' in the subscript divides the two stages where the

left side refers to the first stage and the right side represents the second stage. Hence, SNMF$_{gkl-gkl}$ represents SNMF clustering algorithm with GS at both the stages with KL divergence. Also, SNMF$_{kl-kl}$ is same as SNMF$_{mask}$.

Notations

$\mathbf{X}$ Magnitude audio spectrogram of size $m \times n$

X Complex Audio spectrogram of size $m \times n$

$\mathbf{A}$ Frequency Basis functions of size $m \times r$ obtained using NMF

$\mathbf{B}$ Time activations functions of size $r \times n$ obtained using NMF

$\mathbf{C}$ Frequency basic functions in log-frequency domain (Matrix notation)

$\mathcal{C}$ Frequency basic functions in log-frequency domain (Tensor notation)

$\mathcal{R}$ Translation tensor

$\mathcal{D}$ Translated frequency basis functions

$\mathcal{H}$ Activations functions corresponding to $\mathcal{D}$

$\mathbf{A} \cdot \mathbf{B}$ indicates elementwise multiplication

$\frac{\mathbf{A}}{\mathbf{B}}$ indicates elementwise division

r number of basis functions

P number of sources in mixture

p, s indexing the source

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Basic Concepts

The human auditory system is very skilful in processing the signals in a given audio mixture where multiple sources are present. This processing of the audio signals simplifies the way we perceive the signals from the sound mixture. As a result, the human auditory system can hear out certain sounds such as a conversation that takes place in a noisy environment such as a bus stand or a crowded wedding party. This ability of focusing on a particular auditory stimulus in a noisy environment is known as the cocktail party problem.

Human hearing and other senses like lip movement of the source speaker and spatial location of a source operate quite well in a relative sense to help in focusing on separating individual sources from complex mixtures even in noisy conditions [21]. The psychoacoustic cues such as binaural masking, source

classification and sound localization also help in filtering out the separate sounds from a sound mixture.

The field of study that deals with the ability to organize sounds from a mixture into perceptually meaningful sources is called Auditory Scene Analysis (ASA). With the recent development and growth of digital audio technology, much research has been carried out to design systems that can replicate the human auditory system for ASA. The Computational modelling of the human auditory system to process real world sound signals is called Computational Auditory Scene Analysis (CASA) [3], [4]. CASA systems aim to computationally implement the rules derived from psychoacoustics to segregate or stream the components of sounds in a similar way as human hearing. CASA systems aim to be able to perform similar functions to accurately characterize and group complex components of sound mixtures into their respective sources, based on the cues such as pitch, onset/offset time, spatial location, notes and harmonicity. This influenced and partially gave rise to an area of research called Blind Sound Source Separation (SSS). SSS is the process of estimation of individual sources from the mixture signal.

The attempt to replicate ASA is further complicated by the properties of the sound mixtures that need to be separated. Newer complications emerge when exploring the methods and acoustical conditions under which the mixing of sound is done. Knowledge of the nature of the sources and the recording conditions can provide vital information to the design of a separation algorithm. Some of the notable factors that determine the recording conditions are; the

number of microphones used, the distance between microphones, the number of sound sources, room reverberation and the size of room. Though absolute knowledge of these factors cannot guarantee an ideal solution, they certainly help in defining the separation problem at hand and in building the solution framework for that particular application. There are a large number of applications, such as automatic music transcription, remixing, chord estimation and pitch modification, for which source separation algorithms would be of benefit. Though all the SSS algorithms attempt to tackle the same problem, the approach and methodology employed in each case is usually different.

In the context of this thesis, the term *source* is used to refer to the audio signals that need to be separated by the algorithm. Though *source* may not be the right term conveying the correct implication, it has been used over the years consistently in Audio Source Separation. But to make it clearer, by *source* one actually means Auditory Streams which is to be understood the same way as Bregman used it decades ago [61]. An auditory stream is produced by a continuous activity of a physical source in the form of waves by interaction with the environment. For example, in the case of a piano played in a closed reverberant room, the sound waves that are produced are not due to the instrument alone, but due to each of the keys that are played along with the reverberation that is produced due to reflection of the waves from the walls and so on. In this case, though the sounds are produced by one musical instrument, logically we would have n sources, if each key played is considered a source, in addition to the reverberant room itself. In such a case, the use of the term *source*

to include all the factors producing the waves is inappropriate. The term *source* is often also misunderstood to be a single physical audio source, which is clearly not the case. It is more correct to use the term Auditory Stream to denote the continuous activity produced by the piano in conjunction with its immediate surrounding. The focus of the thesis is on sound source separation algorithms (in the context of music) that deal with the auditory streams produced by different musical instruments. As these auditory streams are perceived to be single entities, the term *source* is used to denote them.

The next term that is commonly used in audio source separation is *sensors*. This is a relatively simpler concept to understand than sources. Sensor is used to denote the physical entities that are used to detect the audio signals or sources. In real world terminology, sensors could be microphones used to record the audio signals or the channels of an audio mixture. For a stereo mixture, there would effectively be two sensors, since there are two channels, left and right. It can also be understood that the sensors form what is known as the mixing system or the mixing matrix in a source separation problem. The relationship between this mixing system and the sources forms the observation mixture or the output mixture. In the following section, classification of sound mixtures is explained in order to give a better understanding of our research goal.

1.1.1 Classification of Sound Mixtures

Before learning about the separation problem itself, it is important to give a description of the classification of the sources that the algorithms have to deal

with. In audio signal processing, sound mixtures can be roughly classified on the basis of:

- Number of sources(P) and mixtures (X)
  - *Under-determined system* where $P > X$.
  - *Determined system* where $P = X$
  - *Over-determined system* where $P < X$.
- Instantaneous and Convolutive mixing
  - *Instantaneous mixing*: In this model it is assumed the time delay describing the arrival of the sound signals at the sensors is the same for all sources and sensors or is zero.
  - *Convolutive mixing*: This model accounts for time lag between the arrival of signals at the sensors. The signal also may arrive from multiple paths through reflection for example, in a room surrounded by walls. Based on these assumption the convolutive mixing model can be further classified as *anechoic* and *echoic*. The anechoic mixing model assumes no degree of reverberation and is considered echo free while echoic mixing model assumes each reflection in the given acoustic environment is modelled as an individual source.
- Time dependence
  - *Time-invariant mixing* where mixing filters remain constant over time.

– *Time-varying mixing* where mixing filters vary with time.

Having said that, we will define the problem we attempt to solve in the course of the thesis, in the next section.

1.2 Blind Sound Source Separation Problem

In our research, we focus on separating musical signals produced by the individual sound sources (instruments) in a single channel under-determined instantaneous audio mixture. This is equivalent to a mono recording and is something of a worst case scenario for the under-determined mixture model, where the number of mixture present is equal to 1. This Blind source separation problem can be typically formulated by the equation:

$$X = AS \tag{1.1}$$

where S is a set of unknown source signal vectors denoted by $s_1, s_2,..., s_p$. Here, P is the number of sources present in the mixture such that $p \in P$. A contains the mixing matrix that are linearly mixed with the sources signals in S to give the audio mixture X. Also, we are only principally dealing with mixtures of signals produced by the pitched instruments.

1.2.1 Applications for single channel SSS

Many audio applications which involve editing, analysis and manipulation of audio data would benefit from the availability of segregated sound sources from the mixture of audio signals for further processing.

SSS can be used as a pre-processing step in automatic music transcription. It is comparatively easier to estimate the fundamental frequencies corresponding to individual notes for a given instrument rather than a mixture of instruments [5].

SSS can be used for automatic speech recognition. When, speaking in a microphone, such as a mobile phone, there may be sources of interference like background noise, that can deteriorate the target speech signal. Here, source separation can be used to separate out the noise from the target speech signal [6].

Separation of source signals can be used to remove or change temporal properties (move or extend in time) of certain instruments or vocals to create remixes or karaoke applications. Further, these SSS methodology once implemented on single channel music recordings can be extended to the up-mixing from mono to stereo or 5.1 surround sound recordings. Recently, Fitzgerald has utilised his sound source separation technologies to create the first ever officially released stereo mixes of several songs of the Beach Boys, including Good Vibrations [7].

1.2.2 The Clustering Problem

In general, the separation of the individual sound sources from a given audio mixture is done using a time-frequency representation such as a spectrogram. A detailed description of time-frequency representation is given in section 1.4. In recent years, many factorisation techniques, such as Non-negative Matrix Factorisation (NMF) [23] of magnitude spectrograms have been proposed to separate out sources from spectrograms [24, 28, 25]. NMF decomposes a spectrogram into frequency basis functions which typically corresponds to the notes and the chords in the given mixture. It is important to note that the number of notes present in a music mixture is typically more than the number of sources. Hence, the clustering of these notes to their corresponding sources is required to achieve source separation. Clustering of these basis functions is at present an open issue and is an important area of research to ensure the quality of the separated sound sources.

Many clustering algorithms have been proposed to cluster the basis functions obtained from factorisation techniques. Supervised clustering methods have been discussed in [28] and [29] to map the separated signals to their sources. Spiertz and Gnann [41] have used a source-filter model to cluster the separated frequency basis functions by mapping the basis functions to the Mel frequency cepstral domain where clustering is performed. While these methods represent a considerable improvements over previous methods, there is still room for improvement in clustering the basis functions to sources.

Recently, Shifted NMF (SNMF) was proposed in order to avoid the need of

clustering of the frequency basis functions [44]. The SNMF algorithm assumes that the timbre of the notes played by an instrument remains constant. However, this assumption is not true in general. Another drawback of using the SNMF algorithm is that it uses a log-frequency spectrogram (see section 1.4.2) and the lack of a true inverse for log-frequency spectrogram results in a deterioration of the sound quality of the reconstructed signal.

To this end, we intend to develop improved NMF based techniques for the clustering of basis functions. Also, we aim to develop an improved version of SNMF model that would assist in segregating the frequency basis functions corresponding to their sources. Finally, we propose to investigate and introduce a new family of masks to enhance the performance of the separation algorithms.

This research work have led to the following publications:

R. Jaiswal, D. FitzGerald, E. Coyle and S. Rickard, "Clustering NMF basis functions using Shifted NMF for Monaural Sound Source Separation," *in Proceedings IEEE International Conference on Acoustic Speech and Signal Processing ICASSP*, May, 2011.

R. Jaiswal, D. FitzGerald, E. Coyle and S. Rickard, "Shifted NMF using an Efficient Constant Q Transform for Monaural Sound Source Separation," *22nd IET Irish Signals and Systems Conference*, 23-24 June, 2011.

R. Jaiswal, D. Fitzgerald, E. Coyle, and S. Rickard, "Shifted NMF with Group Sparsity for clustering NMF basis functions," *Proceedings at 15th International Conference on Digital Audio Effects*, DAFx-12 September 17-21, York, UK, 2012.

D. Fitzgerald, R. Jaiswal, "On the use of Masking Filters in Sound Source Separation," *15th International Conference on Digital Audio Effects DAFx 2012, York, England,* 2012.

R. Jaiswal, D. Fitzgerald, E. Coyle, and S. Rickard, "Towards Shifted NMF for improved Monaural Separation," *Proceedings at 23rd IET Irish Signals and Systems Conference,* 20-21 June, LYIT Letterkenny, Ireland, 2013.

In light of the fact that we are dealing with audio signals, we will briefly cover the basics of sound and how it is produced. The next section focuses on fundamentals of music and musical instruments that we would require to aid the development of the digital signal processing (DSP) methods for sound source separation.

1.3 Fundamentals of Music

As we are focusing on the separation of musical sound sources, in particular pitched instruments, it is necessary to discuss briefly some of the characteristics of sound and musical instruments and the properties of music in general. Initially, we discuss the properties of sound followed by a classification of musical instruments on the basis of how the sound is produced and finally, we discuss some relevant features of music.

Sound is the audible effect of air pressure variations caused by the vibrations, movement, friction or collision of objects. Here, we review the basic physics, properties and propagation of sound waves.

1.3.1 Sound Pressure Level

Sound Pressure Level. The minimum audible air pressure variations (i.e. the threshold of hearing) p_0 is only 10^{-9} of the atmospheric pressure or 2×10^{-5} N/m^2. Sound pressure is measured relative to p_0 in decibels as

$$P(dB) = 20log_{10}(\frac{p}{p_0}) \tag{1.2}$$

From equation 1.2 the threshold of hearing is 0 dB. The maximum sound pressure level (the threshold of pain) is $10^6 p_0$ (10^{-3} the atmospheric pressure) or 120 dB. Hence the dynamic range of the hearing system of hearing is about 120 dB, although the range of comfortable and safe hearing is less than 120 dB.

1.3.2 Sound Power Level

Sound power level. For a tone with a power of w watts this is defined in decibels relative to a reference power of $w_0 = 10^{-12}$ watts (or 1 pico watts) as

$$PL(dB) = 10log_{10}(\frac{w}{w_0}) \tag{1.3}$$

1.3.3 Sound Intensity Level

Sound intensity level. This is defined as the rate of energy flow across a unit area as:

$$I(dB) = 10log_{10}(\frac{I}{I_0}) \tag{1.4}$$

where $I = 10^{-12}$ watts/m^2.

1.3.4 Pitch, Notes, Timbre and Harmonics

The frequency of a sound wave is defined as the number of oscillations per second and is measured in Hertz (Hz). Being a pressure wave, the frequency of the wave is the number of oscillations per second from a high pressure (compression) to a low pressure (rarefaction) and back to a high pressure. The human ear is capable of hearing sound waves in a range of about 20 Hz to 20 kHz.

Pitch is a subjective quantity which is defined as the perceived fundamental frequency of a sound wave. The actual measured fundamental frequency may differ from the perceived fundamental frequency because of overtones and

harmonics, which are explained later in this section. However, a high pitch sound usually corresponds to a high fundamental frequency and a low pitch sound typically corresponds to a low fundamental frequency.

In music, a *note* is a pitched sound. Notes with fundamental frequencies in the following ratios, $1 : 2^n$ where n is 1,2,3 and so on, are perceived very similar and can be grouped under the same pitch class. In western music theory, pitch classes are represented by first seven letters of the Latin alphabet (A, B, C, D, E, F and G), however different countries have their own ways of representing them, for example India uses Sa, Re, Ga, Ma, Pa, Dha, Ni. The eighth note, or octave is given the same name as the first, but has double its frequency, that is the frequency ratio is 1:2.

A *harmonic* is defined as a frequency component of a sound wave and is measured as an integer multiple of the fundamental frequency. For example, if the fundamental frequency is F_o then its harmonics will have frequencies kF_o, i.e $1F_o$, $2F_o$, $3F_o$, ...

The human ear is sensitive to the frequency ratios of the notes rather than the differences between them. The notes which when played simultaneously produce a pleasant sensation are said to be consonant and the combination of notes that are not pleasing to the ear are called dissonant. This phenomenon forms the basis of the intervals in music. The intervals which are perceived to be most consonant are composed of small integer ratios of frequency such as the octave which has a frequency ratio of 2:1. This is because small integers in the ratio ensures that the repetitive pattern in sound waves is achieved in a

Interval	Frequency Ratios
Octave	2:1
Third	5:4
Fourth	4:3
Fifth	3:2

Table 1.1: Music interval and their ratios

small interval of time. As a result, the two notes played simultaneously do not sound harsh as their upper harmonics will overlap with each other. However, the same cannot be said for the notes played simultaneously whose frequency ratio is 15:16. Some common musical intervals and their ratios are listed in Table 1.1. These musical intervals are considered as universally consonant because the musical compositions built around these tone combinations are pleasing to most people in many cultures.

Most natural sounds, such as the human voice, musical instruments, or bird chirping, are made up of many frequencies, which contribute to the perceived quality (or timbre) of the sounds. Consider two instruments playing musical notes at the same pitch and loudness. The sound produced by those two instruments does not sound the same to the ear. Thus, the sound quality of a note played by two different instrument differs with the way it is produced. The tone quality of a musical note that distinguishes between the different kinds of sound production is called the *timbre* of the note. Thus, the timbre can be used to distinguish between musical notes played by two or more instruments in a music mixture of same pitch and loudness. The timbre of a note is mainly characterised by the harmonic content and the dynamic characteristics of the

note such as vibrato and the attack-decay envelope of the note in consideration. In simple words, a timbre is generally described as everything else about a sound which is not described by its pitch and loudness.

1.3.5 Frequencies of Musical Notes

There are two musical pitch standards which are widely accepted, the American pitch standard which takes A in the fourth piano octave (A4) to have a frequency of 440 Hz and the International pitch standard (A4 = 435 Hz). Both of these pitch standards define equal tempered chromatic scales, which means that each successive pitch is related to the previous pitch by a factor of the twelfth root of 2 ($\sqrt[12]{2}$ = 1.05946309436) known as a half-tone. Hence there are twelve half-tones, or steps, in an octave which corresponds to a doubling of pitch. We are assuming the use of the American pitch standard for this research.

The frequency of the intermediate notes, or pitches, can be found by multiplying (or dividing) a given starting pitch by as many factors of the twelfth root of 2 as there are steps up to (or down to) the desired pitch. For example, the G above A4 (that is, G5) in the American Standard has a frequency of 440 $(\sqrt[12]{2})^{10}$ =783.99 Hz. Likewise, in the International standard, G5 has a frequency of 775.08 Hz. G♯5 is another factor of the 12th root of 2 above these, or 830.61 and 821.17 Hz, respectively. Note when counting the steps that there is a single half-tone (step) between B and C, and E and F.

These pitch scales are referred to as 'well tempered'. This refers to a compromise built into the use of the 12th root of 2 as the factor separating

each successive pitch. For example, G and C are a fifth apart. The frequencies of notes that are a perfect fifth apart are exactly in the ratio of 1.5. G is seven chromatic steps above C, so, using the 12th root of 2, the ratio between G and C on either standard scale is $(\sqrt[12]{2})^7$ =1.49830707688, which is slightly less than the 1.5 required for a perfect fifth. For instruments such as piano it is impossible to tune all $3^{rd}s$, $5^{th}s$, etc to their exact ratios such as 1.5 for fifths and simultaneously have all octaves come out exactly in the ratio of 2. As a result, a slight reduction in frequency is required for the complete tuning of the instrument. This slight reduction in frequency is referred to as tempering.

1.3.6 Bandwidths of Music and Voice

The bandwidth of unimpaired hearing is normally between 20 Hz to 20 kHz, although some individuals may have a hearing ability beyond this range of frequencies. Sounds below 20 Hz are called *infra-sounds* and above 20 kHz are called *ultra-sounds*. The information in speech (i.e. words, speaker identity, accent, intonation, emotional signals etc.) is mainly in the traditional telephony bandwidth of 300 Hz to 3.5 kHz.

The sound energy above 3.5 kHz mostly conveys the quality and sensation essential for high quality applications such as broadcast radio/tv, music and film sound tracks. Singing voice has a wider dynamic range and a wider bandwidth than speech and can have significant energy in the frequencies well above that of normal speech. For music the bandwidth is from 20 Hz to 20 kHz. Standard CD music is sampled at 44.1 kHz and quantized with the equivalent of 16 bits

of uniform quantization which gives a signal to quantization noise ratio of about 100 dB at which the quantization noise is inaudible and the signal is transparent.

The frequency and temporal content is usually analysed with the help of time frequency representation (TFR) of a given sound mixture. We will give a brief overview of the various kinds of TFR used in the context of this thesis.

1.4 Time Frequency Representations (TFR)

Here, we discuss the concept and objective of time-frequency representations. In general, the time resolution of the time domain representation of an audio signal is dependent on the sampling frequency. Here, the time resolution is determined by the sampling rate. However, this representation has no information on the frequency content of the mixture signal. In contrast, the absolute value of the Fourier transform of the audio signal gives a magnitude spectrum, that has a very high frequency resolution. However, this representation contains the frequency components of the audio signal but fails to give any temporal information that when a particular note corresponding to a frequency is played within the audio mixture. This situation is not ideal for the analysis of audio signals where the frequency content of the signal changes with time. Hence, we need a time-frequency representation (TFR) that can bridge the gap between the two (time and frequency) representations and provide some temporal and some spectral information simultaneously. This leads us to the TFRs that are useful for the representation and analysis of the audio signals that contain multiple

time-varying frequencies.

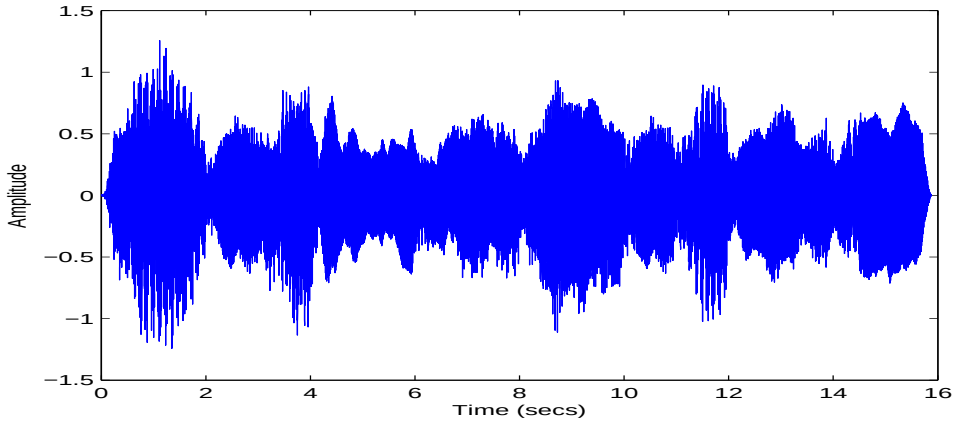

Figure 1.1: Time representation of an audio signal

First, we discuss the most widely used frequency representation obtained using the Fourier transform. Mathematically, it can be calculated as follows:

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-2\pi f t} dt \tag{1.5}$$

where, $X(f)$ gives the frequency spectrum of the time signal $x(t)$ which is continuous in time. However, in our case, the music signals are discretely sampled in time domain and are of finite length. Therefore, for the discretely sampled signals, the Fourier transform can be found using the following equation.

$$X(f) = \left(\frac{1}{N}\right) \sum_{n=0}^{N-1} x[n] e^{-\frac{j2\pi f n}{N}} \tag{1.6}$$

The above transform is referred as Discrete Fourier transform (DFT). As

noted previously, the $X(f)$ in equation 1.6 gives the frequency spectrum that is an average over the entire duration of signal. The magnitude spectrum using DFT exhibits frequency peaks corresponding to the notes in music. The relative heights of the detected peaks may tell us something about the tonality of the given music mixture but the relative timing of the notes present in the magnitude is missing. In case of music signals, the time information of the notes along with its frequency is essential to understand or identify the melodies played by a particular instrument that may help in separating the corresponding notes to their respective sources. Therefore, it can be seen that being able to see how frequency content changes with time would be advantageous in analysing signals with time varying frequency content such as musical signals.

A time-frequency representation (TFR) provides a bridge between the time domain and the frequency domain representation of the signal. A TFR provides both temporal information and spectral information simultaneously where the time and frequency resolution of the signal is determined by certain parameters. A TFR typically uses two orthogonal axes, where one axis corresponds to time and other axis represents frequency. A time domain signal $x[n]$ can then be represented over a two dimensional space of time and frequency. Here we first discuss the most commonly used TFR, the Short-time Fourier Transform (STFT) followed by the Constant Q transform (CQT).

1.4.1 Short-time Fourier Transform (STFT)

The Short-Time Fourier Transform (STFT) is a powerful general-purpose tool for obtaining a TFR. The STFT was first proposed in [9]. The STFT is used for analysing non-stationary signals, whose frequency characteristics vary with time. In essence, the STFT extracts several frames of the signal to be analysed with a window that moves with time. If the time window is sufficiently narrow, each frame extracted can be viewed as stationary so that the FT can be used in each window. With the window moving along the time axis, the evolution of the frequency content of the signal can then be analysed. This TFR maps the signal into a two-dimensional function of time and frequency. A moving overlapped window, for example, the Hanning window, is applied to the signal to divide the signal into frames. An advantage of using the overlap of the windowing functions is that it reduces the artefacts due to the edges of the windows used. Thereafter, the Fourier transform is used to obtain the complex-valued spectrogram from each divided frame. The STFT is widely used as a first processing step for most types of data analysis in audio processing.

The STFT can be summarised by equation 1.7

$$\mathcal{STFT}(x[n]) \equiv X(l,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-l]e^{-j\omega l} \tag{1.7}$$

where x[n] is a signal and w[n] is the chosen window. The magnitudes of the STFT give a spectrogram (equation 1.8) that can show the spectral content of a signal versus time. A spectrogram of a time signal is a two-dimensional

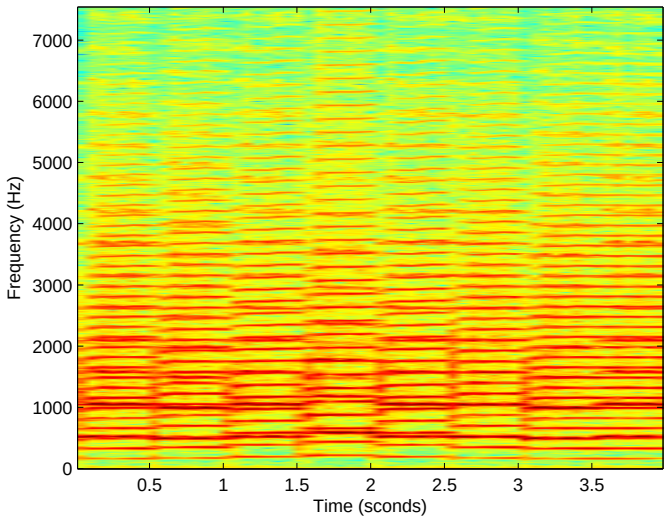

Figure 1.2: Narrow-band spectrogram

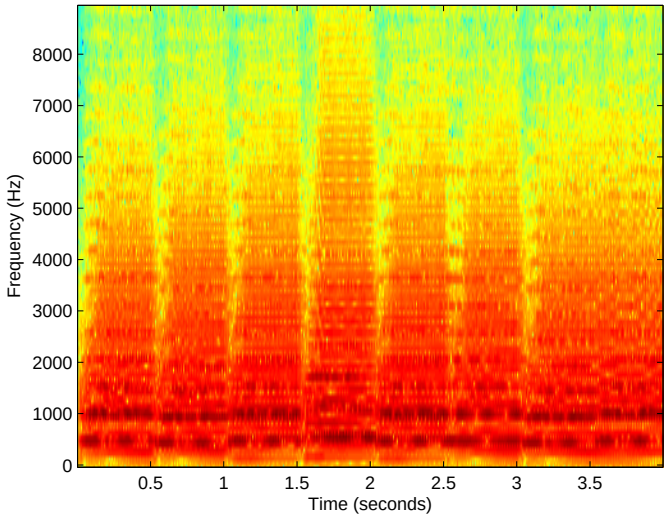

Figure 1.3: Wide-band spectrogram

representation that displays time in its horizontal axis and frequency in its vertical axis.

$$Spectrogram(x[n]) = |X(\tau, \omega)| \tag{1.8}$$

A limitation of using the STFT is that it uses a constant resolution in both frequency and time. The width of the windowing function determines how the signal is represented. In general, the product of time resolution and frequency resolution remains constant. Thus, a better time resolution is obtained at a price of poorer resolution in frequency and vice versa. In other words, to analyse the frequency content accurately, we need more samples (larger window in time) in each frame. However, the larger window makes it more difficult to identify precisely when an event occurs.

Figure 1.2 shows the STFT representation of a music signal with better frequency resolution. A TFR using STFT with better time resolution can be seen in the figure 1.3. The STFT is invertible and the original signal can be recovered by using the inverse STFT on the transform.

Another way of representing a signal in time-frequency domain is the Constant Q Transform(CQT). The CQT is a better suited representation for musical signals due to its log-frequency spectral resolution. We will now discuss how a log-frequency resolution of the spectrogram is better suited for representing music signals.

1.4.2 The Constant Q Spectrogram

As discussed in section 1.3.4, sounds are comprised of harmonic frequency components. The positions of these frequency components in the spectral domain play an important role in analysis of a given piece of music. Consider the following harmonics kF_o, i.e $1F_o$, $2F_o$, $3F_o$, ... for a fundamental frequency F_o. The absolute positions of the harmonics are dependent on the position of the fundamental frequency, F_o. However, the relative position of the harmonics are independent of the fundamental frequency if plotted against a logarithmic scale. This can be summarised by the following equation.

$$\begin{aligned} D_{nm} &= log(nF_o) - log(mF_o) \\ &= \log\left(\frac{nF_o}{mF_o}\right) \\ &= \log\left(\frac{n}{m}\right) \\ &= \text{constant} \end{aligned} \tag{1.9}$$

where, F_o denotes a fundamental frequency and D_{nm} gives the logarithmic distance between n^{th} and m^{th} harmonics. nF_o and mF_o represents n^{th} and m^{th} harmonics of the fundamental frequency, F_o, respectively. It can be seen from equation 1.9 that the logarithmic difference between the corresponding harmonics is independent of the fundamental frequency. Thus, these harmonics in sound or specifically in music contain a pattern that can be investigated using frequency analysis.

However, the conventional linear and uniform frequency separation in the DFT does not show clearly the shift-invariant property of harmonics. This can be explained as follows. Let a constant frequency resolution of 21.5 Hz i.e. sampling frequency 44.1 kHz and window size of 2048 samples is used to calculate the DFT. In the calculation of frequency component with a frequency spacing of 21.5 Hz, we will lose many notes belonging to the lower frequencies i.e. in the range of 150Hz. On the other hand, if we consider the notes containing frequencies in the range of 3kHz, we are evaluating far more frequency components to represent notes than desired. Thus, for musical analysis, a time-frequency representation using DFT or STFT is not always a suitable representation. Therefore, we need a TFR, where the resolution of the frequency bins should be geometrically related to the frequency. Also, with respect to notes the TFR should give a constant pattern of the frequency components (harmonics) for analysis and musical signal processing. This can be achieved by maintaining a constant ratio (Q) of the fundamental frequency to the frequency resolution.

$$\frac{f}{\delta f} = Q \tag{1.10}$$

where, δf denotes the frequency resolution or the bandwidth of the frequency bin and f represents the corresponding fundamental frequency.

To obtain this logarithmic resolution in TFR, a Constant Q transform (CQT) is typically used. The constant Q transform of a discrete-time signal $x[n]$ can be calculated by using the following equation:

$$\mathbf{X}_{cq}[k] = \sum_{n=0}^{N[k]-1} W[n,k]x[n]e^{-j\omega kn} \tag{1.11}$$

where $\mathbf{X}_{cq}[k]$ is the k^{th} component of the Constant Q transform of the input signal $x[n]$. $W[n,k]$ is a window function of length $N[k]$ for each value of k and k varies from $1, 2, \ldots K$ which indexes the frequency bins in the Constant Q domain. The CQT was first proposed by JC Brown [45] inspired by many earlier works including [10, 11, 12].

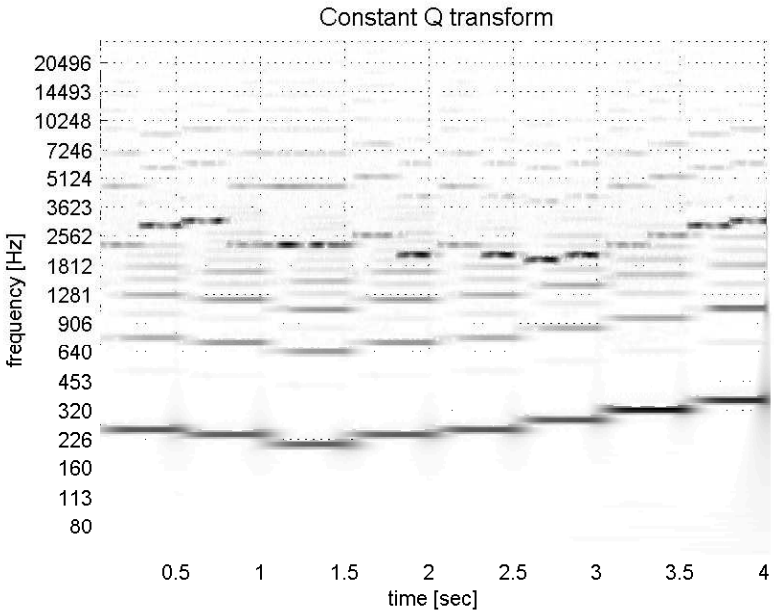

Figure 1.4: Constant Q Spectrogram of an audio mixture signal

Figure 1.4 shows the constant Q magnitude spectrogram of a test signal containing music signals of two pitched instruments.

We will first discuss the calculation of CQT detailed in [84]. In western music,

according to even tempered chromatic scale [52], the fundamental frequencies of the adjacent notes are geometrically spaced by a factor of $\sqrt[12]{2}$. Thus, a frequency spacing of $\sqrt[12]{2}f$ would cover all the notes for musical analysis. Therefore, the frequency of k^{th} spectral component can be calculated using

$$f_k = (\sqrt[12]{2})^k f_{min} \tag{1.12}$$

where f_{min} is the lowest frequency chosen manually. For our research, we have chosen f_{min} to be $55Hz$. The Q factor of a filter is calculated by using equation 1.10. For semi-tone spacing the Q factor can be evaluated to 17 as done in [84]. The direct evaluation of equation 1.11 is computationally inefficient as detailed in [84]. Here, we will make use of Parseval's equation to calculate the CQT coefficients.

Let $x[n]$ and $w[n]$ are discrete time function and $X(f)$ and $W(f)$ represents DFT of the discrete signals $x[n]$ and $w[n]$ respectively. Then according to Parseval's theorem,

$$\sum_{0}^{N-1} x[n]w^*[n] = \frac{1}{N}\sum_{0}^{N-1} X(f)W^*(f) \tag{1.13}$$

where, $W^*(f)$ denotes the complex conjugate of $W(f)$. Thus, the CQT can be efficiently calculated in the Fourier domain by using Parseval's equation and using the DFT coefficients in $X(f)$ and the spectral kernels (as denoted in [84]) in $Y(f)$. Here, $Y(f)$ contains the coefficients of the DFT of the time domain complex exponentials $y[n]$ corresponding to the fundamental frequencies of the

notes (geometrically spaced) present in music. These complex exponentials are used to modulate the time domain signal to obtain the logarithmically scaled frequency basis functions. The CQT can then be obtained by using the following equation:

$$\mathbf{X}_{cq}[k] = \sum_{0}^{N} X(f)Y^{*}(f) \tag{1.14}$$

where, $Y^{*}(f)$ is the complex conjugate of $Y(f)$. For simplicity, we will denote the spectral kernels in $Y(f)$ as transform matrix $\mathbf{Y}$ and the linear spectral coefficients in $X(f)$ as $\mathbf{X}$, then the constant Q transform can be formulated as

$$\mathbf{X}_{cq}[k] = \mathbf{Y}^{*}\mathbf{X} \tag{1.15}$$

where, $\mathbf{Y}^{*}$ is the complex conjugate of $\mathbf{Y}$. However, a drawback of using the CQT is that no true inverse of the CQT is possible. Therefore, it is typically impossible to get a perfect reconstruction of the original signal. Another drawback of using the Constant Q transform is that it is computationally more intensive and complex than the simple DFT or the STFT. Despite these limitations, the time-frequency representations using CQT give a far better understanding of the musical signals and can be potentially used for the musical signal processing.

An approximate inverse transform was proposed by Fitzgerald [88] with the assumption that the music signals can be sparsely represented in the linear frequency domain. However, the assumption does not hold good for all audio signals and the algorithm was extremely slow in calculating the inverse CQT

transform. Recently, Schörkhuber and Klapuri [85] has proposed an extension to the method discussed in [45, 84] to calculate the CQT in a manner which allows a high quality inverse CQT to be calculated. The algorithm processes each octave in the signal one by one starting from highest to lowest to calculate the CQT coefficients of a given spectrogram. In [85], the algorithm basically tries to improve the computational efficiency by addressing two problems. Firstly, when a wide range of frequencies is considered, the DFT blocks are very wide in length, hence the transform matrix is no longer very sparse i.e. for frequency range of $60Hz$ to $16kHz$. Secondly, when calculating the CQT coefficients of the highest frequency bins, the width between the frequency bins should be atleast $\frac{N}{2}$, where N the window length of highest CQT bin. These two problems were addressed to reduce computational efficiency.

The computational efficiency improvement is obtained as follows. Firstly, the transform matrix matrix $\mathbf{Y}$, which contains the CQT coefficients for the highest octave remains same for all the octaves. Then, the entire length of audio input signal is passed through a lowpass filter and downsampled by factor two. Thereafter, the CQT coeffients are calculated using the same transform matrix. The process is repeated until the desired lowest octave is processed. Since, the transform matrix $\mathbf{Y}$ represents the frequency bins that are separated by a maximum of one octave, the matrix $\mathbf{Y}$ remains sparse for highest frequency bins.

Secondly, many of the translated versions of $y[n]$ within the transform matrix $\mathbf{Y}$ are shifted temporally to different positions. This reduces the number of

DFTs calculations for $x[n]$ in equation 1.14. The use of this algorithm and its effect on the separation of sound sources is detailed in chapter 3. In the following section, we give a brief overview of previous techniques used for the separation of the sound sources from a given mixture.

1.5 DSP methods for Source Separation

Independent Component Analysis (ICA) was developed for the estimation of sound signals (independent components) from a given mixture [66, 57, 58] in case of determined systems. Another method Degenerate Unmixing Estimation Technique (DUET) [47] was proposed to separate a given source from an audio mixture using a time-frequency mask corresponding to that source. Barry et al developed a source separation algorithm known as ADRess that uses the pan positions of the instruments to estimate the sources in stereo recordings. Recently, NMF [23] based techniques were successfully used to separate sound sources from a monaural mixture. We will discuss these techniques in the following sections.

1.5.1 Independent Component Analysis

ICA has been successfully used to solve blind source separation problems in several application areas [64, 67]. A survey of ICA based algorithms is done in [63]. ICA separates an observation vector by finding a de-mixing matrix, so that the estimated variables, the elements of vector, are statistically independent

from each other. Consider the *cocktail party problem.* Here, n speakers are speaking simultaneously at a party, and any microphone placed in the room records only an overlapping combination of the n speakers' voices. For example, we have n different microphones placed in the room, and because each microphone is a different distance from each of the speakers, it records a different combination of the speakers' voices.

To formalize this problem, we imagine that there is some data $s \in \mathbb{R}^n$ that is generated via n independent sources. What we observe is

$$x = As \tag{1.16}$$

where A is an unknown square matrix called the mixing matrix. Repeated observations gives us a dataset $[x^{(i)}; i = 1, \ldots m]$, and our goal is to recover the sources $s^{(i)}$ that had generated our data ($x^{(i)} = As^{(i)}$).

In our cocktail party problem, $s^{(i)}$ is an n-dimensional vector, and $s_j^{(i)}$ is the sound that speaker j was uttering at time i. Also, $x^{(i)}$ in an n-dimensional vector, and $x_j^{(i)}$ is the acoustic reading recorded by microphone j at time i. Let $W = A^{-1}$ be the *unmixing matrix.* Our goal is to find W, so that given our microphone recordings $x^{(i)}$, we can recover the sources by computing $s^{(i)} = Wx^{(i)}$. For notational convenience, we also let w_i^T denote the i^{th} row of W, so that

$$W = \begin{bmatrix} w_1^T \\ w_2^T \\ \vdots \\ w_n^T \end{bmatrix} \tag{1.17}$$

Thus, $w_i \in \mathbb{R}^n$, and the j^{th} source can be recovered by computing $s_j^{(i)} = w_j^T x^{(i)}$.

ICA Algorithm

While there are many ICA algorithms, here we present a derivation of a method for Maximum likelihood estimation to find independent sources as detailed in [59]. We suppose that the distribution of each source s_i is given by a density $p(s)$, and that the joint distribution of the sources s is given by

$$p(s) = \prod_{i=1}^{n} p_s(s_i) \tag{1.18}$$

It can be noted that preprocessing of data is required to make the sources uncorrelated by centering and whitening of data. Therefore, by modelling the joint distribution as a product of the marginal distributions, we capture the assumption that the sources are independent. Further, using $x = As = W^{-1}s$, p(x) can be written as

$$p(x) = \prod_{i=1}^{n} p_s(w_i^T x)|W| \tag{1.19}$$

To this end, a density function for the individual sources p_s is needed. For the

reasons explained in [59], a cumulative density function (cdf) is better suited over probability density function. An appropriate sigmoid function can be chosen for cdf that slowly increases from 0 to 1, is the sigmoid function $g(s) = 1/(1+e^{-s})$. Most audio signals are super Gaussian, therefore a super Gaussian cdf can be ideal for audio data. Hence, given the parametric model the log likelihood of square matrix W for a training set $x^{(i)}$; i= 1, 2, ..., m can be expressed as

$$\mathcal{L}(W) = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} \log g^{'}((w_j^T x^{(i)}) + \log |W| \right) \tag{1.20}$$

$\mathcal{L}(W)$ is then maximised with respect to W. By taking derivatives and properly setting the step size, the unmixing matrix $x^{(i)}$ can be updated by gradient ascent learning rule which is defined by equation:

$$W \equiv W + \alpha \left(\begin{bmatrix} 1 - 2g(w_1^T x^{(i)}) \\ 1 - 2g(w_2^T x^{(i)}) \\ \vdots \\ 1 - 2g(w_n^T x^{(i)}) \end{bmatrix} (x^{(i)})^T + (W^T)^{-1} \right) \tag{1.21}$$

The ICA algorithm has three notable ambiguities. Firstly, the ICA algorithm can be used to recover the independent sources but the order of the sources can not defined. Fortunately, this does not matter for most audio applications.

Secondly, there is no way to recover the correct scaling of the w_i's. For instance, if A were replaced with $2A$, and every $s^{(i)}$ were replaced with $0.5s^{(i)}$, then our observed $x^{(i)} = 2A(0.5s^{(i)})$ would still be the same. Thus, it is impossible to recover the exact scaling of the sources. However, for the applications that we are

concerned with, including the cocktail party problem, this ambiguity also does not matter. Specifically, scaling a speaker's speech signal $s_j^{(i)}$ by some positive (or negative) factor α affects only the volume of that speaker's speech. Also, the signals obtained can be rescaled to the desired amplitude as required. Therefore, we will not address the above mentioned ambiguities for the algorithms discussed in the remainder of the thesis.

Finally, for P number of sources, ICA requires P number of channel informations to separate the sound source in the music mixture. Therefore, ICA is suitable for determined systems where number of sources is equal to the number of mixtures. However, this is not the case in commercially available audio recordings. Therefore, we will now look at the techniques which can handle such cases, such as DUET and ADRess.

1.5.2 Degenerate Unmixing Estimation Technique

The Degenerate Unmixing Estimation Technique (DUET) algorithm is based on the fact that a perfect reconstruction of the sound sources from the audio mixture can be obtained using binary time-frequency masks provided the TFRs of the individual sources present do not overlap with each other [47]. This phenomenon is known as W-disjoint orthogonality (WDO) [50]. Let $x(t)$ denote a mixture signal containing p number of sources such that

$$x(t) = \sum_{i=1}^{P} s_i(t) \tag{1.22}$$

where $s_i(t)$ represents signals produced by the i^{th} source. Further, the time-frequency representation of the signals can be formulated as follows

$$X(\tau, \omega) = \sum_{i=1}^{P} S_i(\tau, \omega) \tag{1.23}$$

where $S_i(\tau, \omega)$ is the TFRs of the sources $s_i(t)$ respectively in a given mixture. Here, τ indicates time and ω signifies frequency. The criteria that the sources are pairwise WDO can be expressed as follows.

$$S_i(\tau, \omega).S_j(\tau, \omega) = 0 \, \forall i, j \in p : i \neq j \tag{1.24}$$

Based on the assumption that the sources are pairwise WDO, we can say that only one source will be active within in the mixture for a given τ and ω. Therefore for a particular choice of τ and ω, $X(\tau, \omega)$ becomes

$$X(\tau, \omega) = S_{\tau,\omega}(\tau, \omega) \tag{1.25}$$

where $S_{\tau,\omega}(\tau, \omega)$ represents the source at the given (τ, ω). Thereafter, the time-frequency masks for each source can be calculated in the manner shown below.

$$M_j(\tau, \omega) = \begin{cases} 1 & S_j(\tau, \omega) \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{1.26}$$

These masks are then applied to the original time frequency representations of the mixture signal to obtain the sources.

It is assumed that the mixture signals are obtained from a linear mixture of P sources as stated in equation 1.22. The assumption of linear mixing contains the underlying assumption that the mixtures have been obtained under anechoic conditions. Therefore, for p mixture in $x(t)$ $[s_1(t), s_2(t), ..., s_{(p)}(t)]$, the mixing model can described as:

$$X(t) = \sum_{i=1}^{N} A_i s_i(t - D_i) \tag{1.27}$$

where A_i and D_i are the elements of the attenuation coefficients vector and time delays vector associated with the path of i^{th} source. Considering the case of two-microphone setup, the mixing model in the time-frequency domain can be written as:

$$\begin{bmatrix} X_1(\tau,\omega) \\ X_2(\tau,\omega) \end{bmatrix} = \begin{bmatrix} 1 \\ A_i e^{-j\omega D_i} \end{bmatrix} \begin{bmatrix} S_1(\tau,\omega) \\ S_2(\tau,\omega) \end{bmatrix} \tag{1.28}$$

Let the element-wise ratio, $R_{21}(\tau,\omega)$, of STFTs of each channel be defined as :

$$R_{21}(\tau,\omega) = \frac{X_2(\tau,\omega)}{X_1(\tau,\omega)} \tag{1.29}$$

The level ratio $R_{21}(\tau,\omega)$ uses the relative difference of attenuation from one microphone to the another to calculate the masks needed for the reconstruction of sources. Assuming all the sources are pairwise WDO, for an active j^{th} source at (τ,ω), the relative difference can be calculated using the following equation.

$$R_{21}(\tau, \omega) = A_j e^{-i\omega D_j} \tag{1.30}$$

Thereafter, the magnitude and phase information of the element-wise ratio $R_{21}(\tau, \omega)$ can be derived from equation 1.30 in terms of the parameters A_i and D_i. The phase of $R_{21}(\tau, \omega)$ is constrained between -π and π. A time-frequency mask can be obtained for each source by determining the mixing parameters A_i and D_i. The generated mask can then be used on either of the original mixture signals to obtain the separated sources.

A notable drawback of using the DUET is that the time delay between the two receivers (two microphones) used is constrained to the following condition

$$\omega_{max} D_{jmax} < \pi \tag{1.31}$$

if $\omega_{max} = \frac{2\pi f_s}{2}$, where f_s is the sampling frequency then the maximum time delay and the maximum distance between the two microphones (for a two microphone setup) is limited to

$$\begin{aligned} D_{jmax} &= \frac{1}{f_s} \\ d_{max} &= D_{jmax} c \end{aligned} \tag{1.32}$$

where d_{max} is the distance between the two microphones, and c is the speed of sound. This means that the distance d_{max} is of the order of few centimetres which is quite small in general.

Another drawback of using DUET is that the assumption that all the sources are independent to each other and and strictly W-DO is not true in general [47]. As a result, the sounds will interfere with each other in the mixture and the estimated parameters for each source A_i and D_i will deviate from its actual value.

An algorithm was proposed to solve this problem in [49]. The algorithm was based on the fact that the estimated attenuation and time delay parameters (A_i, D_i) for each source will still contain values within the close range of the actual parameter value. Therefore, smoothed 2-D weighted histograms can be constructed using the estimated mixing parameters as detailed in [48]. A resolution width for each parameter is chosen for the estimation of the histograms. This defines the window for which the histogram is constructed for each position in time and frequency. Thereafter, the peaks and the location of the peaks is determined corresponding to the particular source. A time-frequency binary mask is then constructed for each peak and are grouped together to their source. This grouping of time-frequency points can be done by using maximum likelihood function as explained in [48]. Finally, these individual source masks can be applied to the original STFT on either of the mixture signal to recover the source spectrogram

The DUET algorithm was found to give good results on anechoic mixtures of speech signals. However, the performance of the DUET algorithm degraded considerably for echoic mixtures where the the histogram peak regions were not distinct and were overlapping with each other. Also, the algorithm required at

least two microphones, hence it will not work in the case of mono signals.

1.5.3 Azimuth Discrimination and Resynthesis (ADRess)

ADRess is an efficient source separation algorithm developed by Barry et al [62]. It is based on azimuth discrimination of sources within the stereo field. It uses the pan positions to estimate the sources in stereo recordings.

The algorithm is designed for stereo recordings made in the fashion where n sources are first recorded individually as mono tracks, and then summed and spread across the two channels, left and right, using a mixing console. In the mixing process, a panoramic potentiometer is used to achieve localization of the various sources by dividing them into the two channels with different intensity ratios that are continuously variable. A source can be positioned at any location between two speakers by creating an inter-aural intensity difference between the two channels. This is done by attenuating a source signal in one of the channels which causes it to be localized in the other channel thereby causing the source to come from a particular location in the azimuth plane. Most stereo recordings have an inter-aural intensity difference (IID) between the left and right channels as the different instruments are panned to various degrees in the azimuth plane. In commercial stereophonic recordings, as a first approximation, only the intensity of the sources between the two channels differs but the phase information is exactly the same. The ADRess algorithm is created for recordings made using this methodology. The algorithm also exhibits limited

success with stereo-pair, mid-side and binaural type recordings. This algorithm is found to be effective in various audio applications such as vocal removal. Also, unlike many other algorithms, prior knowledge of the sources, sensors or the recording conditions is not required to perform the separation task. The ADRess algorithm succeeds fairly well in separating sources from commercial recordings. The degree of separation largely depends on the number of sources present, the proximity of the sources in the azimuth plane and the intensity level of the sources. Experiments reveal that a low number of sources with unique pan positions results in a low signal to noise ratio whereas a high number of sources results in missing overlapping partials. However, ADRess cannot work with mono recordings or where multiple sources are positioned at the same point in the stereo field.

1.5.4 Non negative Matrix Factorisation

For musical analysis, non-negative matrix factorization (NMF) has been [23] shown to be a useful decomposition of audio spectrograms. This is due to the fact that it gives additive parts-based decompositions, where the parts typically corresponds to the notes or the chords in the music. NMF can be defined as follows. Given a non-negative matrix, such as magnitude spectrogram $\mathbf{X}$, NMF attempts to approximate $\mathbf{X}$ by decomposing into factors $\mathbf{A}$ and $\mathbf{B}$. The equation can be expressed as:

$$\mathbf{X} \approx \hat{\mathbf{X}} = \mathbf{AB} \tag{1.33}$$

where the matrix $\mathbf{A}$ is of size $n \times r$ and the matrix $\mathbf{B}$ is of size $r \times m$, with $r < n,\ m$. Also all the elements of factors $\mathbf{A}$ and $\mathbf{B}$ are constrained to be non-negative. In equation 1.33 the input magnitude spectrogram $\mathbf{X}$ is approximated by $\hat{\mathbf{X}}$. Here, $\hat{\mathbf{X}}$ is a linear combination of the columns of matrix $\mathbf{A}$, and the corresponding rows of matrix $\mathbf{B}$. Furthermore, the use of non-negativity constraint in NMF ensures an additive parts-based decomposition of the magnitude spectrogram into basis functions.

Cost function

Given a magnitude spectrogram $\mathbf{X}$, there are infinite number of solutions for NMF and the NMF may be defined for a wide range of divergence measures. However, the matrix obtained should exhibit the properties discussed above. Therefore, these properties should be encapsulated in the choice of the cost function. The generalised optimisation problem for the divergence measures can be formulated as

$$\min \mathbf{D}_{fn}\left(\mathbf{X}, (\mathbf{AB})\right) \tag{1.34}$$

where $\mathbf{D}_{fn}$ represents the choice of divergence for the optimisation problem. A family of beta-divergences can be defined as

$$\mathbf{D}_\beta(x,y) = \begin{cases} \frac{1}{\beta(\beta-1)}\left(x^\beta + (\beta-1)\,y^\beta - \beta x y^{\beta-1}\right) & \text{if } \beta \in \{0,2\} \\ x\log\frac{x}{y} + y - x & \text{if } \beta = 1 \\ \frac{x}{y} - log\frac{x}{y} - 1 & \text{if } \beta = 0 \end{cases} \tag{1.35}$$

It can be noted that the $\mathbf{D}_\beta$ is continuous for β at 0 and 1. A thorough review of the beta divergences can be found in [60]. The three most commonly used divergences which are a part of the family of beta divergence are as follows:

$$\begin{array}{ll} \mathbf{D}_{EUC}(x,y) = \frac{1}{2}(x-y)^2 & \text{the Euclidean norm} \\ \mathbf{D}_{KL}(x,y) = xlog\frac{x}{y} + y - x & \text{the Kullback-Leibler (KL) divergence} \\ \mathbf{D}_{IS}(x,y) = \frac{x}{y} - log\frac{x}{y} - 1 & \text{the Itakura-Saito (IS) divergence} \end{array} \tag{1.36}$$

.

The choice of β for the decomposition of the spectrograms is an open issue. For the reasons stated in [60], the factorisation with $\mathbf{D}_\beta(x,y)$ varies with β such that

$$\mathbf{D}_\beta(\lambda x, \lambda y) = \lambda^\beta \mathbf{D}_\beta(x,y) \tag{1.37}$$

This means that the IS divergence ($\beta = 0$) is invariant to scaling i.e $\mathbf{D}_{IS}(\lambda x, \lambda y) = \mathbf{D}_{IS}(x,y)$. It also states that the factorisation obtained with $\beta > 1$ (in case of Euclidean norm or the KL divergence) will depend more on the higher data values in the given matrix compared to the lower data values

and the opposite can be expected for the divergence with $\beta < 1$. Therefore, it can be said that the β value for NMF may be chosen by keeping in mind the task it is intended for.

Multiplicative Updates

Once the cost function is defined, the problem is approached by minimizing the cost function with respect to $\mathbf{A}$ and $\mathbf{B}$. In each case, $\mathbf{A}$ and $\mathbf{B}$ can be initialised randomly, subjected to constraint $\mathbf{A}, \mathbf{B} \geq \mathbf{0}$. At the end of each iteration, a new value of $\mathbf{A}$ and $\mathbf{B}$ is found by iterative updates respectively.

Now, we will formulate the iterative multiplicative updates for each of the minimisation problems discussed for the NMF algorithm. For solving problems in equation 1.36 the multiplicative updates using the Euclidean distance cost function are as follows:

$$\mathbf{B} \leftarrow \mathbf{B} \cdot \left(\frac{\mathbf{A}^T \mathbf{X}}{\mathbf{A}^T \hat{\mathbf{X}}} \right) \tag{1.38}$$

$$\mathbf{A} \leftarrow \mathbf{A} \cdot \left(\frac{\mathbf{X}\mathbf{B}^T}{\hat{\mathbf{X}}\mathbf{B}^T} \right) \tag{1.39}$$

The multiplicative update equation for KL divergence and IS divergence can be calculated using the following equations.

$$\mathbf{B} \leftarrow \mathbf{B} \cdot \left(\frac{\mathbf{A}^T (\mathbf{X} \cdot \hat{\mathbf{X}}^{-\delta})}{\mathbf{A}^T (\hat{\mathbf{X}}^{-(\delta-1)}))} \right) \tag{1.40}$$

$$\mathbf{A} \leftarrow \mathbf{A} \cdot \left(\frac{(\mathbf{X} \cdot \hat{\mathbf{X}}^{-\delta})\mathbf{B}^T}{(\hat{\mathbf{X}}^{-(\delta-1)})\mathbf{B}^T)} \right) \tag{1.41}$$

were δ can be set to 1 and 2 for KL divergence and IS divergence respectively. The "$\cdot$" symbol indicates element-wise matrix multiplication in all equations stated above. The Euclidean distance defined by equation 1.36 is non-increasing for the update equations 1.38 and 1.39. Now we will give the proofs of convergence for the Euclidean distance and the KL divergence as detailed in [23].

Convergence proofs

The convergence proofs can be derived by making use of an auxiliary function similar to that used in the Expectation-Maximization algorithm [26], [27].

Let $g(b, b^{'})$ be an auxiliary function for function f such that it satisfies the following conditions:

$$\begin{aligned} g(b, b^{'}) &\geq f(b) \\ g(b, b) &= f(b) \end{aligned} \tag{1.42}$$

Then, function f is non-increasing under the following update rule

$$b^{(t-1)} = \arg\min_{b} g(b, b^t) \tag{1.43}$$

$\forall\, b^t$ such that $t \geq 0$. This can be explained as follows. Here, we can find a local

minimum of g by using the update rule defined in equation 1.43. For example, minimizing the auxiliary function $g(b, b^t) \geq f(b)$ ensures that $f(b^{t+1}) \leq f(b^t)$ for a defined update rule. It is important to note that when b^t corresponds to a local minimum of $g(b, b^t)$, then $f(b^{t+1}) = f(b^t)$. Thus, following the update rule repetitively will generate a sequence of $f(b^t)$ that will converge to a local minimum of $f(b)$ because $\forall\, t > 0$,

$$f(b^{t+1}) \leq g(b^{t+1}, b^t) \tag{1.44}$$

$$g(b^{t+1}, b^t) \leq g(b^t, b^t) \tag{1.45}$$

and

$$g(b^t, b^t) = f(b^t) \tag{1.46}$$

This non-increasing pattern can be illustrated using figure 1.5. This figure is taken from the presentation slides of [23].

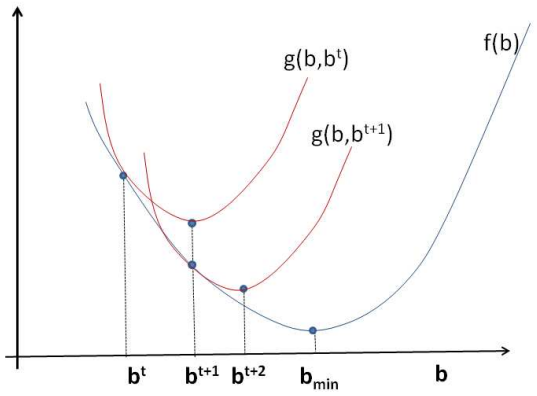

Figure 1.5: Figure showing the non-increasing pattern of $f(b^t)$

Having described the properties of the auxiliary function, we will first give the convergence proof of the update equation in 1.38 as described in [23].

Consider a function $f(b)$ such that,

$$f(b) = \frac{1}{2}\sum_{m}\left(x_m - \sum_{r}\mathbf{A}_{mr}b_r\right)^2 \tag{1.47}$$

where, x_m represents the m^{th} column of matrix $\mathbf{X}$ and $\mathbf{A}_{mr}$ denotes an element of matrix $\mathbf{A}$ corresponding to m^{th} row and r^{th} column. We will use these notations in the subsequent equation. An auxiliary function $g(b, b^t)$ for f can be defined as

$$g(b, b^t) = f(b^t) + (b - b^t)^T \nabla f(b^t) + \frac{1}{2}(b - b^t)K(b^t)(b - b^t) \tag{1.48}$$

where, $K(b^t)$ is a diagonal matrix

$$K_{rl}(b^t) = \frac{\delta_{rl}(\mathbf{A}^T\mathbf{A}b^t)_r}{b_r^t} \tag{1.49}$$

and where, δ is a Kronecker delta function

$$\delta_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \tag{1.50}$$

To prove, $g(b, b)$ is an auxiliary function of f we need to show

$$\begin{aligned} g(b, b) &= f(b) \ \& \\ g(b, b^t) &\geq f(b) \end{aligned} \tag{1.51}$$

By replacing b^t with b in equation 1.48, the last two terms becomes 0 and the equation 1.48 becomes $g(b, b) = f(b)$. It can be seen from equation 1.47 that $f(b)$ is a quadratic equation in b, therefore

$$f(b) = f(b^t) + (b - b^t)^T \nabla f(b^t) + \frac{1}{2}(b - b^t)^T(\mathbf{A}^T\mathbf{A})(b - b^t) \tag{1.52}$$

Thereafter, comparing equations 1.48 and 1.52, $g(b, b^t)$ will be $\geq f(b)$, only if

$$\begin{aligned} \frac{1}{2}(b-b^t)^T(\mathbf{A}^T\mathbf{A})(b-b^t) &\leq \frac{1}{2}(b-b^t)K(b^t)(b-b^t) \\ or, \quad 0 &\leq (b-b^t)^T\left(K(b^t)-\mathbf{A}^T\mathbf{A}\right)(b-b^t) \end{aligned} \tag{1.53}$$

Finally, to prove that the resultant matrix (right hand side of the above inequality) is positive semi-definite, let us consider a matrix M_{rl}, which is obtained by rescaling the components of $K - \mathbf{A}^T\mathbf{A}$, such that

$$M_{rl}(b^t) = b_r^t\left(K(b^t)-\mathbf{A}^T\mathbf{A}\right)_{rl} b_l^t \tag{1.54}$$

Therefore, $K-\mathbf{A}^T\mathbf{A}$ is positive semi-definite only if M satisfies the following condition.

$$x^T M x \geq 0 \tag{1.55}$$

By expanding $x^T M x$, we get

$$\begin{aligned} x^T M x &= \sum_{rl} x_r M_{rl} x_l \\ &= \sum_{rl} x_r b_r^t\left(K(b^t)-\mathbf{A}^T\mathbf{A}\right)_{rl} b_l^t x_l \end{aligned} \tag{1.56}$$

Substituting the value of $K(b^t)$ in above equation we get,

$$= \sum_{rl} x_r b_r^t \left(\frac{\delta_{rl}(\mathbf{A}^T\mathbf{A}b^t)_r}{b_r^t} - \mathbf{A}^T\mathbf{A} \right)_{rl} b_l^t x_l \tag{1.57}$$

The use of Kronecker delta zeros out non-diagonal elements in the first term and the product of the corresponding elements of x_r and x_l give x_r^2(or x_l^2)

$$= \sum_{rl} b_r^t (\mathbf{A}^T\mathbf{A})_{rl} b_r^t x_r^2 - x_r b_r^t (\mathbf{A}^T\mathbf{A}) rl b_l^t x_l \tag{1.58}$$

$$= \frac{1}{2} \sum_{rl} (\mathbf{A}^T\mathbf{A})_{rl} b_r^t b_l^t \left(x_r^2 + x_l^2 - 2x_r x_l \right) \tag{1.59}$$

$$= \frac{1}{2} \sum_{rl} (\mathbf{A}^T\mathbf{A})_{rl} b_r^t b_l^t \left(x_r - x_l \right)^2 \tag{1.60}$$

The terms outside the square in above equation are initialised with non-negative numbers and constraint to be ≥ 0, therefore, we can say that

$$x^T M x \geq 0 \tag{1.61}$$

The minimising of $g(b, b^t)$ in equation 1.48 with respect to b is done to replace $g(b, b^t)$ in equation 1.43. Thereafter, we obtain the following update rule

$$b^{t+1} = b^t - \frac{\nabla f(b_r^t)}{K(b_r^t)} \tag{1.62}$$

Further, after simplification, we get

$$b_r^{t+1} = b_r^t \frac{\mathbf{A}x_r}{(\mathbf{A}^T\mathbf{A}b^t)_r} \tag{1.63}$$

which is equivalent to

$$\mathbf{B} \leftarrow \mathbf{B} \cdot \left(\frac{\mathbf{A}^T\mathbf{X}}{\mathbf{A}^T\hat{\mathbf{X}}} \right) \tag{1.64}$$

Similarly, f can be shown to be non-increasing under the update equation for A, by simply reversing the roles of A and B. Here, f is non-increasing under the update equation of A and B because g is the auxiliary function.

To proof the convergence of KL-divergence consider the following auxiliary function

$$\begin{aligned} g(b, b^t) = &- \sum_m x_m \left(1 - \log(x_m)\right) + \sum_{mr} \mathbf{A}_{mr} b_r \\ &- \sum_{mr} x_m \frac{\mathbf{A}_{mr} b_r^t}{\sum_l \mathbf{A}_{ml} b_l^t} \left(\log(\mathbf{A}_{mr} b_r) - \log\left(\frac{\mathbf{A}_{mr} b_r^t}{\sum_l \mathbf{A}_{ml} b_l^t} \right) \right) \end{aligned} \tag{1.65}$$

for the divergence function $f(b)$ in equation 1.66

$$f(b) = \sum_m x_m \log\left(\frac{x_m}{\sum_r \mathbf{A}_{mr} b_r} \right) - x_m + \sum_r \mathbf{A}_{mr} b_r \tag{1.66}$$

Again to prove that $g(b, b^t)$ is an auxiliary function for $f(b)$, we need to prove the set of axioms defined in 1.51. For $g(b, b) = f(b)$, we replace b^t by b in equation 1.65.

$$
\begin{aligned}
g(b,b) &= f(h) + \sum_m x_m \log(\sum_r \mathbf{A}_{mr} b_r) \\
&\quad - \sum_{mr} x_m \frac{\mathbf{A}_{mr} b_r}{\sum_l \mathbf{A}_{ml} b_l} \left(\log(\mathbf{A}_{mr} b_r) - \log\left(\frac{\mathbf{A}_{mr} b_r}{\sum_l \mathbf{A}_{ml} b_l} \right) \right) \\
&= f(h) + \sum_m x_m \log(\sum_r \mathbf{A}_{mr} b_r) - \sum_m x_m \frac{\sum_r \mathbf{A}_{mr} b_r}{\sum_l \mathbf{A}_{ml} b_l} \left(- \log\left(\frac{1}{\sum_l \mathbf{A}_{ml} b_l} \right) \right) \\
&= f(h) + \sum_m x_m \log(\sum_r \mathbf{A}_{mr} b_r) - \sum_m x_m \log(\sum_l \mathbf{A}_{ml} b_l) \\
&= f(h)
\end{aligned}
\tag{1.67}
$$

Further, to show that $g(b, b^t) \geq f(b) \forall t \geq 0$, the convexity of the log function can be used, that satisfies the following inequality

$$
- \log \sum_r \mathbf{A}_{mr} b_r \leq - \sum \alpha_r \log \left(\frac{\mathbf{A}_{mr} b_r}{\alpha_r} \right) \tag{1.68}
$$

where, α_r contains all non-negative elements that sum to unity, such as

$$
\alpha_r = \frac{\mathbf{A}_{mr} b_r}{\sum_l \mathbf{A}_{ml} b_l} \tag{1.69}
$$

Substituting the value of α_r from equation 1.69 in equation 1.68 we get,

$$
- \log \left(\sum_r \mathbf{A}_{mr} b_r \right) - \sum_r \frac{\mathbf{A}_{mr} b_r^t}{\sum_l \mathbf{A}_{ml} b_l^t} \left(\log(\mathbf{A}_{mr} b_r) - \log\left(\frac{\mathbf{A}_{mr} b_r^t}{\sum_l \mathbf{A}_{ml} b_l^t} \right) \right) \tag{1.70}
$$

By eliminating common terms in $g(b, b^t)$ and $f(b)$ in equations 1.65 and 1.66 respectively and then comparing it with the inequality defined in equations 1.70, we can see that $f(b) \leq g(b, b^t)$.

Now by replacing $g(b, b^t)$ in the update rule (see equation 1.43) by equation 1.65, we get the following update equation,

$$b_r^{t+1} = \frac{b_r^t}{\sum_l \mathbf{A}_{kl}} \sum_m \frac{x_m}{\sum_l \mathbf{A}_{ml} b_l^t} \mathbf{A}_{ml} \tag{1.71}$$

This is done by minimising $g(b, b^t)$ with respect to b i.e. by equating the $\nabla g = 0$ as shown below

$$\begin{aligned} \nabla g(b, b^t) = -\sum_m x_m \frac{\mathbf{A}_{mr} b_r^t}{\sum_l \mathbf{A}_{ml} b_l^t} \frac{1}{b^r} + \sum_m \mathbf{A}_{mr} = 0 \\ \Rightarrow \sum_m \mathbf{A}_{mr} = \sum_m x_m \frac{\mathbf{A}_{mr} b_r^t}{\sum_l \mathbf{A}_{ml} b_l^t} \frac{1}{b^r} \end{aligned} \tag{1.72}$$

To avoid confusion, we can change the index on left hand side of the equation from i to k and rearranging, we will get the required update rule defined in equation 1.71. Further, it can be written in the form shown below:

$$\mathbf{B}_{rn} \leftarrow \mathbf{B}_{rn} \cdot \frac{1}{\sum_k \mathbf{A}_{kr}} \sum_m \frac{\mathbf{A}_{mr} \mathbf{X}_{mn}}{\hat{\mathbf{X}}_{mn}} \tag{1.73}$$

Again, function f is non-increasing the update rule of B because $g(b, b)$ is the auxiliary function of $f(b)$. We can prove convergence of A in a similar manner. It is important to note that the update equation in 1.73 is the same as defined

above in equation 1.41.

Here, we have showed the convergence of the cost function to a local minima with respect to either **A** or **B**. However, in general, the above discussed cost functions is not convex with respect to both **A** and **B** and it would not be feasible to find the global minima. A detailed description and derivation of IS divergence update equations and its application in musical analysis can be be found in [22].

Having discussed the convergence proofs of the cost functions, now we will explain how the decomposition of the magnitude spectrogram using NMF is of benefit in musical applications especially sound source separation.

NMF decomposition

The decomposition of the magnitude spectrogram is done using NMF that results in non-negative matrices **A** and **B**. To demonstrate the workings of a NMF of a music mixture, figure 1.6 shows a magnitude spectrogram of a toy audio mixture.

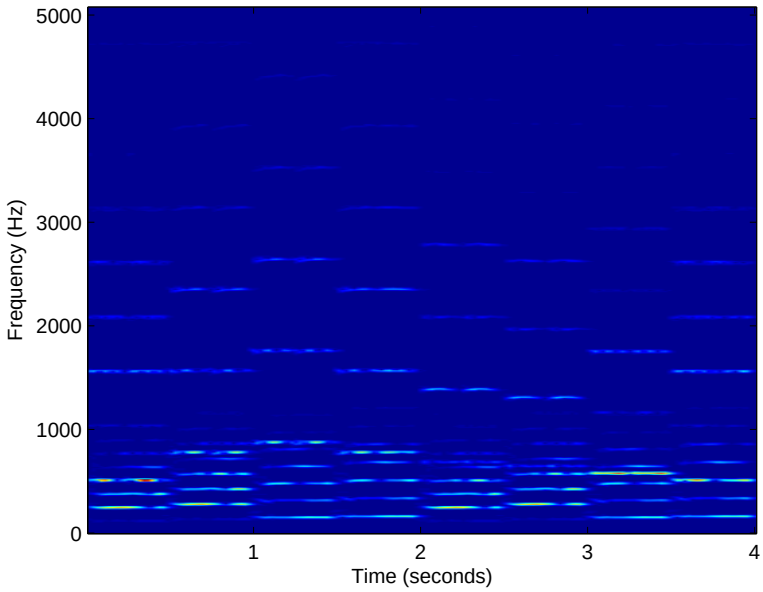

Figure 1.6: Matrix representing spectrogram of an audio signal

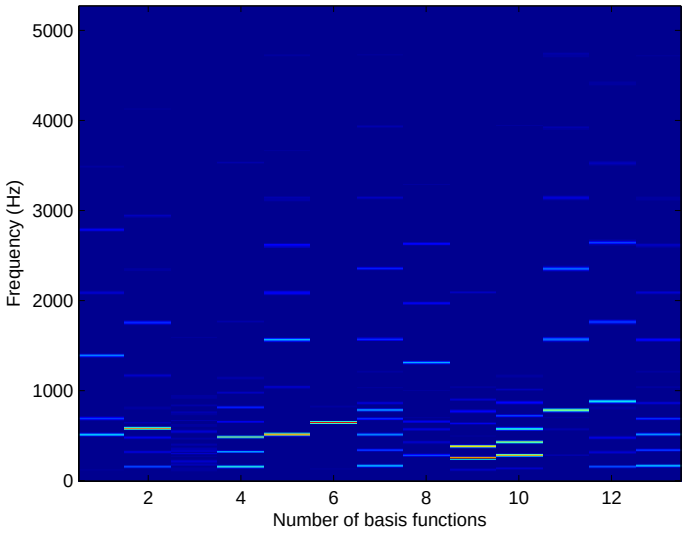

Figure 1.7: Columns of matrix **A** containing NMF frequency basis functions

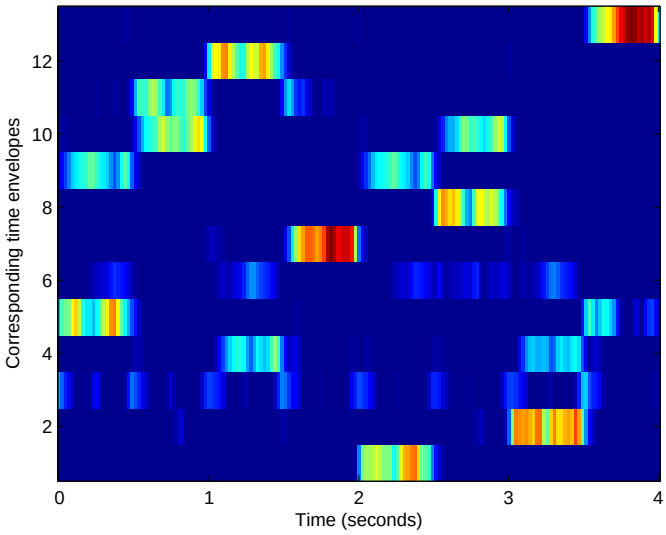

Figure 1.8: Rows of matrix **B** representing time envelopes corresponding to NMF frequency basis functions in figure 1.7

Figure 1.7 represents the matrix **A** and the figure 1.8 represents the matrix **B**. Matrix **A** contains the NMF frequency basis functions and by close inspection, it can be seen that the columns of **A** correspond to note-like entities, exhibiting harmonic structure and the matrix **B** contains the time activation function along its rows that indicates when the corresponding note-like entities in **A** is active within the mixture. This separation of basis functions approximately representing the notes can be used to determine pitches corresponding to notes and further by using the time activation functions associated with the basis functions would allow a basic transcription [25].

1.5.5 Limitations of Standard NMF

A drawback of using NMF is that it typically results in a larger number of basis functions than there are active sources in the mixture. Therefore, clustering of these basis functions is required for separating the sources. Clustering of these basis functions is at present an open issue and is an important area of research to ensure the quality of separated sound sources. Also, clustering of the basis functions is one of the principal aims of this thesis.

However, in an effort to avoid the need for clustering of basis functions, FitzGerald et al proposed an algorithm [44], which we will elaborate in the next section.

1.5.6 Shifted Non-negative Matrix Factorisation

Shifted Non negative Matrix factorisation (SNMF) was proposed as a means of avoiding the problem of clustering provided that a log frequency resolution is used for the frequency basis functions. The SNMF algorithm [44] assumes that the timbre of a note does not change for all the pitches produced by an instrument. The basic principle used in the SNMF algorithm is well motivated by the fact that, in western music the fundamental frequencies of each half tone are geometrically spaced by a factor of $\sqrt[12]{2}$. Therefore, a translated version $\mathcal{D}$ of a frequency basis function of a particular instrument can then be used to approximately cover the entire range of melodies played by the instrument in consideration. Also, if the frequency bins are a semitone apart, a shift up or down of the frequency basic function by one frequency bin can be used to

approximate the frequency basis function of another note higher or lower by half a note respectively. However, a log-frequency resolution of the frequency basis functions is required to exploit this shift invariant property. A constant Q transform can be used to obtain the log-frequency resolution.

Notations

We now define the parameters and notations used in the SNMF model. The notations for tensor parameters used to define the SNMF model [44] is as per the conventions described in [90]. Calligraphic upper-case letters ($\mathcal{R}$) are used to denote tensors of any given dimension. A contracted tensor product of two tensors of finite dimension is defined as follows. Let a tensor $\mathcal{R}$ be of dimension $I_1 \times \cdots \times I_S \times L_1 \times \cdots \times L_P$ and tensor $\mathcal{D}$ be of dimension $I_1 \times \cdots \times I_S \times J_1 \times \cdots \times J_N$ then equation 1.74 denotes the contracted tensor multiplication of $\mathcal{R}$ and $\mathcal{D}$ along the first p modes. Indexing of tensor elements is done using lower case letters, such as j and is denoted by $\mathcal{R}(i, j)$.

$$\langle \mathcal{R}\mathcal{D} \rangle_{\{l_1,\ldots,l_p,j_1,\ldots,j_p\}} = \sum_{l=l_1}^{l_p} \cdots \sum_{j=j_1}^{j_p} \mathcal{R}_l \times \mathcal{D}_j = \mathcal{Z} \tag{1.74}$$

The dimensions along which the tensors $\mathcal{R}$ and $\mathcal{D}$ are to be multiplied is specified in curly brackets. The resultant tensor $\mathcal{Z}$ will be of dimension $l_1 \times \cdots \times l_p \times j_1 \times \cdots \times j_p$.

SNMF Algorithm (SNMF$_{cqt}$)

As noted previously, a log-frequency resolution of the frequency basis function is required for the Shifted NMF. Here, the CQT is used to obtain the log-frequency resolution. A CQT spectrogram can be obtained by multiplying the transform matrix $\mathbf{Y}$ (see equation 1.14) with $\mathbf{X}$, where $\mathbf{X}$ is the linear domain magnitude spectrogram.

$$\mathbf{C} = \mathbf{YX} \tag{1.75}$$

Having obtained a Constant Q spectrogram $\mathbf{C}$ of size $n \times m$, where m is the number of time frames along the n frequency bins, SNMF can be used to separate the instrument basis functions. In practice, for a given number of p sources the spectrogram $\mathbf{C}$ can be decomposed using the SNMF model into tensors as shown in equation:

$$\mathbf{C} \approx \langle \langle \mathcal{RD} \rangle_{\{3,1\}} \mathcal{H} \rangle_{\{2:3,1:2\}} \tag{1.76}$$

where, $\mathcal{R}$ is a translation tensor of dimension $n \times k \times n$ for k possible translations. $\mathcal{R}$ translates the instrument basis functions in $\mathcal{D}$ up or down to approximate various notes played by an instrument in question. Tensor $\mathcal{D}$ is of size $n \times p$ contains a frequency or instrument basis function for each source. $\mathcal{H}$ is a tensor of size $k \times p \times m$ such that $\mathcal{H}(i\,,\, s\,,\, :)$ represents the time envelope for the i^{th} translation of the s^{th} source, which informs when a given note is played by a particular instrument.

For a given s number of sources, SNMF will decompose the constant Q spectrogram $\mathbf{C}$ into instrument basis functions and sets of associated time activations that can be used to approximately represent $\mathbf{C}$. The cost function used to approximate tensors $\mathcal{D}$ and $\mathcal{H}$ is the same as used for NMF. To approximately cover all the notes played by the instrument, the number of translation k is chosen empirically. The translated (frequency-shifted) version of an instrument basis function approximately captures all the notes played by a given instrument considered in a mixture. Thus, the need of clustering NMF basis functions is avoided, as each instrument is now represented by a single instrument basis function. The SNMF algorithm requires the use of a log-frequency spectrogram for segregating the frequency basis functions. In music processing, a CQT is typically used to achieve log-frequency resolution.

The SNMF algorithm has two notable drawbacks. Firstly, the spectral envelope of notes played by an instrument changes with the pitch, therefore, the assumption that the timbre of any note played by an instrument remains unchanged, regardless of pitch, is not true in general. However, this approximation holds reasonably well over a limited pitch range.

Secondly, the lack of an inverse CQT results in a deterioration of the separation quality of the reconstructed signal. However, the shift-invariant property of the instrument basis function can be exploited to capture all the notes played by pitched instruments in the audio mixture. We will attempt to address these limitations to develop improved SNMF algorithms for monaural sound source separation in chapters 2 and 3.

1.6 Previous Clustering Techniques for NMF basis functions

Recently, a data-adaptive method was proposed by Virtanen [29] to segregate the NMF frequency basis functions obtained from the power spectrogram of the input signal. The proposed method was based on the fact that the high-energy components of the input signal can be compressed by modelling the loudness perception of human auditory system using perceptually motivated weights for each critical band in each frame [30]. Thus, for each critical band, the perceptually significant low-energy characteristics of sources can also be estimated. The individual components corresponding to sources were estimated by minimizing the weighted divergence between the above model and the observed power spectrogram. Another method was proposed in [28] that uses sparse coding [17] with some modifications and as well as a temporal continuity constraint [31]. A cost term, comprised of the sum of squared differences between the gains in the adjacent frames of the activation function in $\mathbf{B}$, was used to impart the temporal continuity and sparseness was favoured by penalizing non-zero gains in $\mathbf{B}$. In [82], the clustering was done manually. A non-negative sparse coding algorithm was suggested by Abdallah and Plumbley in [24] that assumes that the sources sum in the power spectral domain, so that the observation vector and basis functions are power spectra. Despite these improvements to group the NMF basis function for sound source separation, all the previous proposed clustering algorithms were unable to separate robustly

the notes corresponding to a given set of pitched instruments overlapping in frequency and time in a given mono mixture. Hence, there is a room for much improvement. The first systematic attempt at unsupervised clustering of the NMF basis functions was done by Spiertz and Gnann in [41] who used a source-filter model to generate MFCC coefficients for NMF basis functions. This method of unsupervised clustering is explained in the following section.

1.6.1 Source-Filter Based Clustering for Monaural BSS Separation

According to the source-filter model in [20] and [79], each frequency basis vector in $\mathbf{A}$ is a product of an excitation or source signal E and an instrument-specific resonance filter R. These filters are mainly responsible for the formants in the mixture. The MFCC-based source separation method exploits this instrument-specific information to filter out the resonance effect in the mixture. Here, we will briefly cover the calculation of MFCC coefficients. The Mel scale [72] is defined as a perceptual scale of any two consecutive pitches perceived by listeners to be equidistant from one another. The frequency f in mels m is given by:

$$R = 2595 log_{10}\left(\frac{f}{700} + 1\right) \tag{1.77}$$

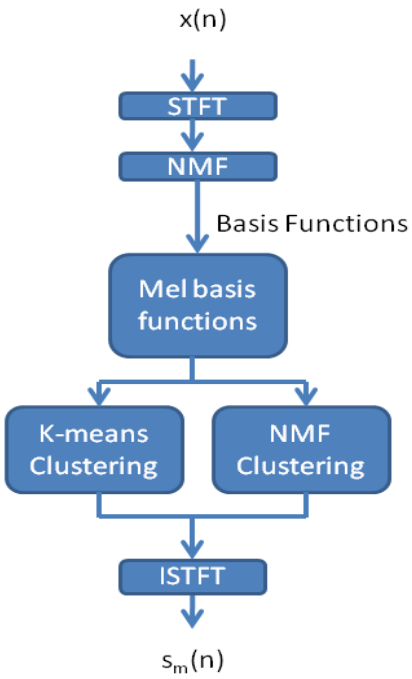

Figure 1.9: Signal flowchart for the clustering using source-filter model

Flowchart 1.9 shows the signal in the clustering algorithm using the source-filter model. In the following section the MFCC and NMF clustering method is discussed.

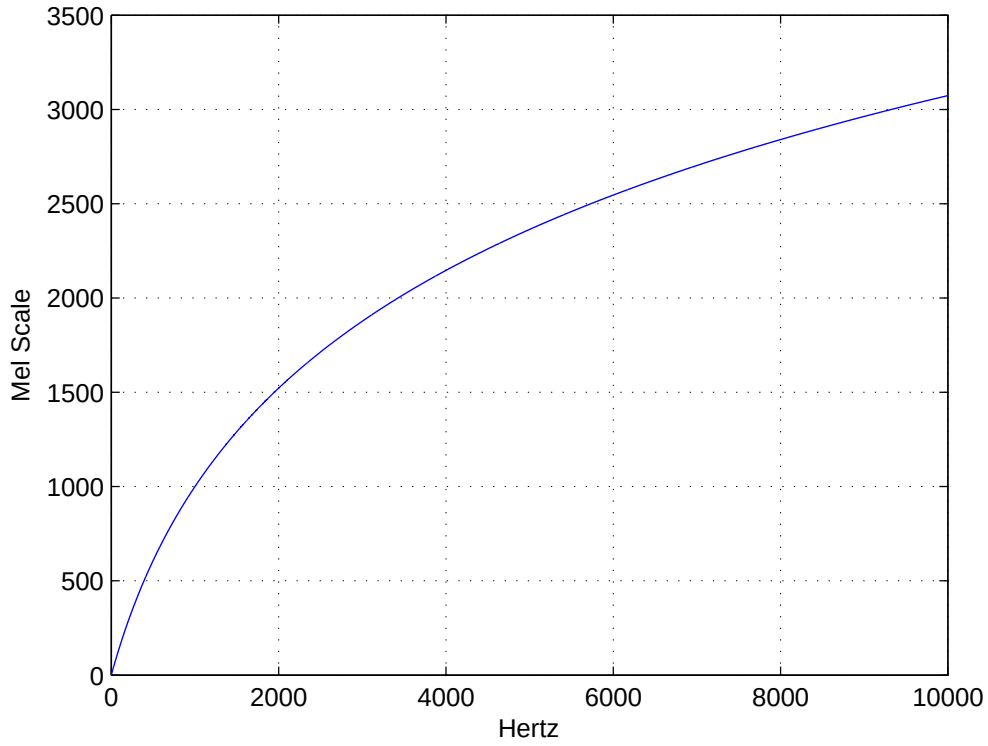

Figure 1.10: Pitch mels Vs frequency f (Hz)

MFCC Clustering

Let N denote a Mel filterbank with N_{mel} filters where each row in N represents a filter with a triangular shaped weighing function. The distance between the centre frequencies f for all the filters in the Mel filterbank is chosen as per the Mel scale defined by equation 1.77. Having obtained the filter bank of N_{mel} filters the MFCC coefficients for NMF basis functions is calculated as follows. The frequency basis functions in $\mathbf{A}$ is obtained in a similar manner as shown in

equation 1.33. The inner dot product of input vectors in $\mathbf{A}$ (columns of matrix $\mathbf{A}$), representing NMF basis functions, is filtered by N_{mel} filters in N to obtain basis vectors $\mathbf{M}$ in the Mel frequency domain.

$$\mathbf{M} = N\mathbf{A}^2 \approx N[E^2 * R^2] \tag{1.78}$$

where $\mathbf{M}$ contains the MFCC coefficients corresponding to the NMF basis functions and each column of $\mathbf{M}$ is denoted by $Mf_r(n)$, such that $1 \leq n \leq N_{mel}$ and $1 \leq r \leq R$. To separate out the excitation component, the logarithm is used as it converts the multiplication operation into simple addition and thus the equation becomes

$$\log(cMf_r + 1) \approx \log(NE_r^2) + \log(NR_m^2) \quad \textit{where} \quad c = \frac{a_t}{max(Mf_r(n))} \tag{1.79}$$

Here an offset of +1 is added to counter any negative logarithmic value and the constant c is used to normalised the basis vectors Mf_r by the maximum amplitude, where $max(Mf_r(n))$ and a tuning element a_t (usually between range 0.1 to 0.01) is also used to make the model linear.

$$\log(cMf_r + 1) = \log(c) + \log(Mf_r) + \log\left(1 + \frac{1}{c(Mf_r)}\right) \tag{1.80}$$

As the tuning element a_t is increased, the linearity is increased by making the last term in equation 1.80 smaller but it also increases the offset value $\log(c)$. Therefore, there is a trade off between these two and the value of a_t was

determined through experiments.

Then, the Discrete Cosine Transform (DCT) [70] is used to separate out or decorrelate the source component and spectral component by dropping out eigenvalues corresponding to signal energy (first coefficient) and higher frequency components. In the last step, a simple clustering (of sources) method like k-means [46] is applied on k mfcc components to find a permutation matrix that indicates which basis function belongs to which source in the given mixture.

NMF Clustering

As discussed in section 1.6.1, the DCT helps in decorrelating the two signals $\log(NE_r^2)$ and $\log(NR_m^2)$ corresponding to source and resonance filter respectively. However, this decorrelation of the spectral components for a given channel is performed without any information of the other channels. On the other hand, NMF can be used instead of the DCT to extract, R_m which contains the activation functions corresponding to the resonance filters. This is done as stated in [41]. The input log signal for the decorrelation in 1.80 can be rearranged as follows

$$\mathbf{L}(n, i) = \log(cMf_r + 1) \tag{1.81}$$

where each column of $\mathbf{L}$ is approximately equal to the summation of corresponding columns of $\log(NE_r^2)$ and $\log(NR_m^2)$ (see equation 2.15), n and i are used to index the channels and sources respectively. Thereafter, two matrices $\mathbf{T}$ of size $N_{mel} \times M$ and $\mathbf{T}$ of size $M \times I$ are initialised with positive random numbers.

$$\mathbf{L} = \mathbf{TZ} \tag{1.82}$$

Then, the NMF factors $\mathbf{T}$ and $\mathbf{Z}$ are obtained by minimising the KL divergence cost function described in equation 1.36. Furthermore, this method does not use k-means clustering as the clustering is carried out by comparing and finding the dominant of the source components for each of the channel in M corresponding to the rows in matrix $\mathbf{Z}$ and thus obtaining a permutation matrix $g(i)$ as shown in equation 1.83:

$$g(i) = arg \max_{m} \mathbf{Z}(m, i). \tag{1.83}$$

1.6.2 Incorporation of group sparsity in NMF with IS divergence

The term sparse refers to a signal model, where only a few units of data out of a large population can be used to efficiently represent a typical data vector [14]. A property of NMF is that it typically generates a sparse representation of the given audio data. This makes the frequency basis function sparse in nature. However, NMF does not impose any quantitative constraint on the nature of sparsity. Also, the level of sparseness in NMF representation varies depending on the signal. Therefore, it is hard to set the optimal level of sparsity automatically. Nevertheless, there are cases in which additional constraints may be imposed to control the degree of sparseness to identify components in mixtures [16, 28].

Such a constraint has been proposed by [53] that generates a set of NMF basis functions which benefits from sparsity at a group level.

Given a magnitude spectrogram, $\mathbf{X}$ of size $m \times n$, the power spectrogram can be calculated by

$$\mathbf{V} = |\mathbf{X}|^2 \tag{1.84}$$

Then, the frequency basis functions can be obtained by optimising the following equation

$$\mathbf{V} \approx \hat{\mathbf{V}} = \mathbf{WH} \tag{1.85}$$

where, the frequency basis functions are contained in $\mathbf{W}$ and the local amplitudes corresponding to the frequency basis functions are stored in $\mathbf{H}$. The incorporation of group sparsity in NMF is due the fact that the activation of the NMF results in the frequency basis functions that corresponds to the instruments (groups) present in the music mixture. Therefore, we want the NMF frequency basis functions to be sparse in a group sense. GS assumes that each instrument is turned on (played) for as little a time as possible and that an individual instrument activation is much sparser than that of a mixture of instruments. It is hoped that this prior knowledge of GS may reduce the time-frequency overlapping of the frequency basis functions, hence give improved clustering of the basis functions.

In [53], GS is incorporated in NMF with the hypothesis that the local amplitudes of the sources are independent and may be derived as a marginal distribution for the activation function $\mathbf{H}$. Further, they used Itakura-Saito (IS)

divergence as the cost function. This is done to exploit the equivalence between IS-NMF method and maximum-likelihood estimation of $(\mathbf{W}, \mathbf{H})$ when power spectrum density (PSD) of the input signal is used to calculate the frequency basis functions.

Here, the power spectrum matrix is used for the calculation of the frequency basis functions. This is motivated by the fact that the components in the power spectrum on average sum linearly. This is analogous to the fact that the time domain signals obtained from the concurrent sound sources and their complex spectra sum linearly, which is not the case in magnitude spectra. However, in many audio applications the linear summation of magnitude spectra has produced better results.

With the assumption that the local amplitudes of the sources are independent from each other and for the reasons stated in [51] [22], the minimisation of the IS divergence cost function, $\mathbf{D}_{IS}(\mathbf{V}||\hat{\mathbf{V}})$ (see equation 1.36) is equivalent to the maximum likelihood problem of estimating ($\mathbf{W}$, $\mathbf{H}$) in sum of Gaussian components. This is based on the assumption that the components in spectrogram $\mathbf{X}$ is a linear instantaneous mixture of i.i.d Gaussian signals. Then, ML estimation of $\mathbf{W}$ and $\mathbf{H}$ from $\mathbf{X}$ is equivalent to estimating ($\mathbf{W}$, $\mathbf{H}$) from power spectrogram $\mathbf{V}$ using NMF where IS divergence is used [22]. Hence, $\mathbf{V}$ has the following distribution :

$$p(\mathbf{V}|\hat{\mathbf{V}}) = \prod_{m,n} \frac{1}{\hat{V_{mn}}} exp\left(-\frac{V_{mn}}{\hat{V_{mn}}}\right) \tag{1.86}$$

It is also assumed that a source can be characterised by a subset of

components g. Therefore, the source spectrogram corresponding to a group, $\mathbf{X}^g$, where

$$\mathbf{X}^g = \sum_{r \in g} \mathbf{X}^r, \tag{1.87}$$

can be estimated by the following Wiener filter estimator [53]:

$$E\left(\mathbf{X}^g | \mathbf{X}, \mathbf{W}, \mathbf{H}\right) = \mathbf{X} \cdot \left(\frac{\mathbf{W}_g \mathbf{H}_g}{\mathbf{W}\mathbf{H}}\right). \tag{1.88}$$

Maximum Likelihood with group sparsity

The r number of basis functions in $\mathbf{W}$ needs to be divided into g non-overlapping groups, where each group contains the frequency basis functions corresponding to a given source. Following the conventions used in [53], for a given time-frequency frame n, if a source (group) is not active, then the corresponding activation gain $\mathbf{H}_{gn}$ is made equal to zero. Let $\mathbf{H}_{gn}$ is a vector of basis functions r_i such that r_i is a member of a given group g ($r_i \in g$ where $1 \leq i \leq m$). Let H_n^g be defined as a time envelop of the given source for a given time frame n such as

$$H_n^g = ||\mathbf{H}_{gn}||_1 \tag{1.89}$$

where $||.||_1$ represents the L1 norm function. Furthermore, it is assumed that the activation gain H_n^g for all the individual sources are mutually independent inverse gamma random variables. Thereafter, by using the conditional probability on the activation function $\mathbf{H}$ at frame n for r basis functions, the activation gains can be factorized into groups to determine respective sources ;

and H_{rn} are exponentially distributed conditionally on H_n^g, with mean H_n^g. This can be denoted as:

$$p(\mathbf{H}_n|H_n^g) = \prod_g \prod_{r \in g} p(H_{rn}|H_n^g) \tag{1.90}$$

The prior of the activation functions $\mathbf{H}_n$ can be calculated using the marginal distribution as follows:

$$p(\mathbf{H}_n) = \prod_g \frac{\Gamma(g+\eta)}{\Gamma(\eta)} \frac{\alpha^\eta}{(\alpha + H_n^g)^{(\eta+g)}} \tag{1.91}$$

where the parameters α and η define the shape of the inverse gamma distribution. By providing this prior information, the ML estimation with group sparsity can be defined as follows:

$$(\mathbf{W}, \mathbf{H}) = \min_{\mathbf{W},\mathbf{H} \geq 0} \mathbf{D}_{IS}(\mathbf{V}||\mathbf{WH}) + \lambda \Phi(\mathbf{H}) \tag{1.92}$$

where $\lambda \in [0, 1)$ is a scaling factor that regularises the optimisation term $\Phi(\mathbf{H})$. $\Phi(\mathbf{H})$ defines the the grouping pattern.

Equation 1.93 and 1.95 show the multiplicative updates for $\mathbf{H}$ and $\mathbf{W}$ respectively.

$$\mathbf{H} \leftarrow \mathbf{H} \cdot \left(\frac{\mathbf{W}^T(\mathbf{V} \cdot \hat{\mathbf{V}}^{-2})}{\mathbf{W}^T(\hat{\mathbf{V}}^{-(\delta-1)}) + \lambda \Phi'(||\mathbf{H}_{gn}||_1)} \right) \tag{1.93}$$

where

$$\Phi(z) = \log(\alpha + z) \tag{1.94}$$

$$\mathbf{W} \leftarrow \mathbf{W} \cdot \left(\frac{(\mathbf{V} \cdot \hat{\mathbf{V}}^{-\delta})\mathbf{H}^T}{(\hat{\mathbf{X}}^{-1})\mathbf{H}^T + \lambda \sum_n H_{rn} \Phi'(||\mathbf{H}_{gn}||_1)} \right) \tag{1.95}$$

The derivation of the update equations can be found in [53]. After the convergence, the grouped frequency basis functions can be used to re-synthesize the individual sources.

The clustering of the frequency basis functions using group sparsity were found to achieve a good separation for temporally overlapping of sources up to 66%, where the sources where not active constantly with time. However, in many cases the sources will have considerably more overlap than that, and so the clustering based on GS may fail in these cases. Nevertheless, the group sparsity in NMF was found to reduce the amount of overlapping of the sources in mixture. Thus, it can potentially be used to improve the clustering of the NMF basis functions. We will exploit and discuss this idea in chapter 5.

1.7 Conclusions

In this introductory chapter, we have attempted to provide the motivation for blind sound source separation and explained its importance in the field of various audio applications. We considered the basic assumptions such as number of sources, number of sensors, time-invariance and so on, under which sound mixtures can be classified. We also presented the fundamentals of music and musical instruments that we require to aid the development of sound source separation methods.

We have described a standard ICA SSS technique for instantaneous

determined sound mixture model. Then, we discussed the limitation of using standard ICA, in particular that it will fail to work if the number of sources present is more than the given number of mixtures or over-determined mixtures. Further, we introduced other sound source separation algorithms such as DUET, ADRess for audio mixture model. Although, these techniques could handle cases where there was more sources than sensors, but still required at least stereo signals and so will not work for mono signals. Then, a widely used factorisation technique, NMF, for sound source separation was discussed. We showed that how NMF attempts to give a part-based decomposition of the audio spectrogram where the individual parts (basis functions) correspond to the notes in the given mixture. However, these basis functions are usually greater in number than the active sources present in the mixture.

As a consequence, we mentioned the need for clustering of these basis functions into their respective sources to achieve source separation. Thereafter, we discussed the SNMF model that attempts to use a single instrument basis function per source to avoid the need for clustering of the basis functions. We then followed up with an overview of previous techniques for unsupervised clustering of the frequency basis functions, including the MFCC based source filter method [41]. Finally, we discussed a recent approach of clustering basis functions using a technique called group sparsity.

Following from this, we now give an outline of the chapters in the remainder of the thesis. In chapter 2, we introduce two novel methods for clustering the basis functions. The first of the two methods is the locally-linear embedding

method that uses the source-filter model detailed in section 1.6.1 to group the Mel scale basis functions. The second method uses the shift invariant property of the SNMF model in an attempt to improve the clustering of the basis functions. Chapter 3 deals with an improvement to the standard SNMF algorithm (see section 1.5.6) obtained using a recently proposed CQT method [85]. Further, in chapter 4, a novel attempt is made to incorporate the CQT matrix inside the SNMF model in order to improve the working of the SNMF clustering algorithm discussed in 2. Chapter 5 deals with the idea of incorporating GS in the SNMF clustering algorithm such that the sparsity in NMF favours at group level. Chapter 5 also gives an overview of the effects of the GS in the various proposed SNMF clustering algorithms in the context of separation of sound signals from a mono mixture. Chapter 6 deals with a new family of masks to reconstruct the original signal from the clustered basis functions and how the proposed family of masks are better performing masks as compared to the generalized Wiener filter masks. Finally, Chapter 7 outlines the techniques discussed in this thesis and focuses on the possible area of future work that would improve source separation algorithms.

Chapter 2

Shifted NMF Sound Source Separation

2.1 Introduction

NMF has found use in single channel separation of audio signals, as it gives a parts-based decomposition of audio spectrograms where the parts typically correspond to individual notes or chords. However, a notable shortcoming of NMF is the need to cluster the basis functions to their respective sources after decomposition. Despite recent improvements in algorithms for clustering the basis functions to sources, much work still remains to further improve these algorithms. In this chapter we will introduced two new methods for clustering of NMF basis functions. Firstly, we will use a dimension reduction method called locally-linear embedding (LLE) with limited success. For LLE we have used the

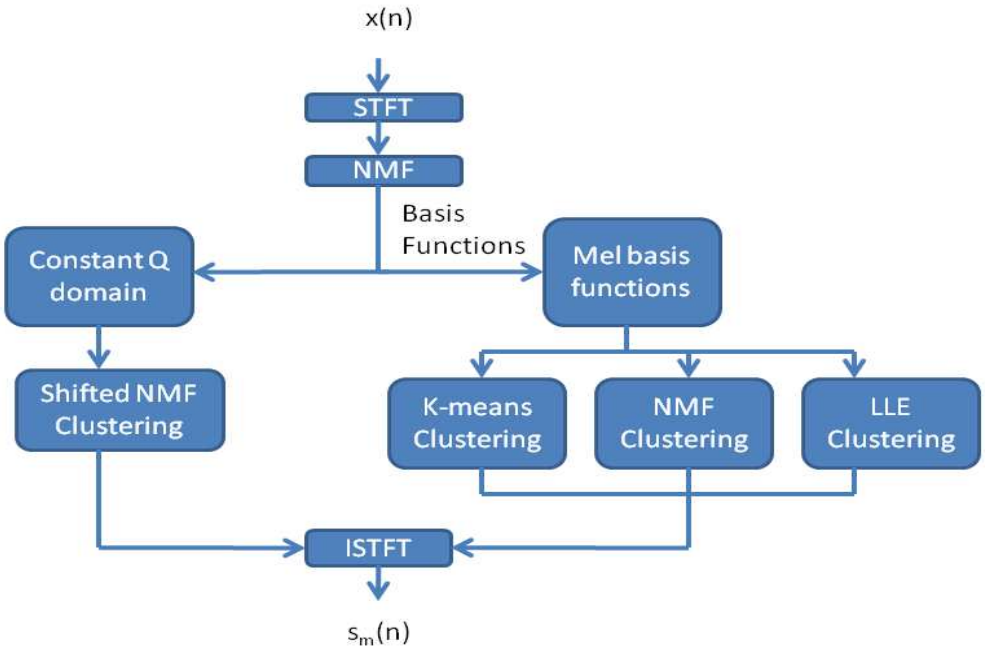

Figure 2.1: Signal flowchart of the System model

source-filter model discussed in section 1.6.1. Also, we present a novel clustering algorithm (a major contribution to this research work) which overcomes some of the limitations of previous clustering methods. This involves the use of SNMF as a means of clustering the frequency basis functions obtained from NMF. Finally, we will test the proposed algorithms to evaluate their performance using a testset of mono mixtures.

The block diagram in the figure 2.1 shows how the various clustering techniques we are comparing are related. Earlier, in chapter 1, we have discussed the source-filter based NMF clustering and the k-means clustering methods as proposed in [41] that uses the frequency basis functions in mel-scale for grouping them to their sources. The first of the two techniques presented in this chapter makes an attempt to group the Mel-scale frequency basis functions, is based

on the simple modification of the clustering stage of the source-filter model as shown above in the block diagram. Here, we use LLE, a dimension reduction technique proposed by Saul et al in [74]. The Mel-scale basis functions are obtained from NMF frequency basis function in a similar manner as detailed in section 1.6.1. Then, an attempt is made to group the frequency basis functions in mel domain using LLE.

The second algorithm proposed here is designed by combining two techniques, the NMF and SNMF, described in chapter 1. Again, the frequency basis functions are obtained by using NMF on the magnitude spectrogram. Then, the Constant Q mapping is used to convert the frequency basis functions from linear domain to log-frequency domain to impart frequency shift-invariability. Thereafter, the Shifted NMF method is used to cluster the frequency basis functions. The SNMF clustering algorithm is detailed in section 2.3.

It is important to note that the SNMF clustering algorithms (SNMF_{map} and SNMF_{mask}) described here is different from the standard SNMF algorithm (SNMF_{cqt}). The difference is in the way the frequency basis functions and the inputs to the SNMF models are determined. For the SNMF clustering algorithm the frequency basis functions are obtained by factorising the magnitude spectrogram (obtained using STFT) using NMF and the SNMF model finds shift invariance in sets of CQ domain frequency basis functions. However, the standard SNMF algorithm finds shift invariance in a CQ spectrogram of the full audio mixture.

Having said that, we move our focus to the LLE algorithm which is described in the following section.

2.2 Locally Linear Embedding

The LLE algorithm is a non-linear unsupervised learning algorithm for dimension reduction for a given set of sampled data obtained from an underlying manifold [74]. The algorithm is based on simple geometric intuitions. This algorithm exploits the spatial property of the data set such that the nearby data points in the high dimensional space remains in the neighbourhood with respect to each other even in the low dimensional space.

Given a sampled dataset P (which defines the underlying manifold) that contains N data points of dimension D. $P_i \in P$ can be used to represent a data vector (data point), where i varies from 1 to N. It is important that the manifold is sampled well enough to restore the neighbourhood properties of the closely located data points across the dataset. The algorithm assumes that the given set of data lies on a smooth manifold of dimension d such that $d << D$. Therefore, based on this assumption there exists an approximate linear mapping such that it maps the high dimensional coordinates of each neighbourhood to the global coordinates of the given low dimensional manifold.

Next, we will discuss the steps involved in the LLE algorithm. First, the algorithm computes the k nearest neighbours for each data point P_i. Then, the weights W_{ij} are obtained such that the it minimises the least squared cost

function in equation 2.1

$$D_{LLE1} = \sum_{i=1}^{N} |P_i - \sum_{j}^{k} W_{ij} P_j|^2 \tag{2.1}$$

where P_j represents the neighbour j of the data point P_i and W_{ij} gives the contribution of the j^{th} data point to the i^{th} reconstruction. Notably, the cost function in equation 2.1 is subjected to two constraints. Firstly, each data point P_i should be reconstructed only from its neighbours ensuring that $W_{ij} = 0$ if P_j is not a neighbour of P_i. The second constraint is that the summation of the rows of weights for a particular data point is 1 i.e. $\sum_j W_{ij} = 1$. These constraints ensures that the weights computed restores the intrinsic geometrical properties of the original data. Details on how the defined constraints help in optimising the calculation of weights can be found in [80].

Having obtained the weights W_{ij}, the algorithm maps a low dimensional data point corresponding to each of the high dimensional data points in P. This is done by randomly initialising dataset Y that contains d dimensional data points Y_i and minimising the cost function defined in equation 2.2.

$$D_{LLE2} = \sum_{i=1}^{N} |Y_i - \sum_{j}^{k} W_{ij} Y_j|^2. \tag{2.2}$$

Equation 2.2 uses the optimised weights obtained earlier when the original data set was used. Since, the weights used reflects the intrinsic geometrical properties of original data, it is expected that the constructed low-dimensional dataset should have same properties.

Having described the LLE algorithm, we will try to use the LLE algorithm to obtain the grouping of the NMF frequency basis functions. This is based on the idea that the frequency basis functions corresponding a source will fall into neighbourhood of each other in terms of euclidean distance, when in the MFCC domain and thus help in grouping them corresponding to their sources.

It can be seen from the block diagram in figure 2.1, that the LLE algorithm is implemented for clustering of the mel domain basis functions. The frequency basis functions in $\mathbf{A}$ (see equation 1.33) is converted into mel domain by using equation 1.78. Here, the mel domain basis functions in $\mathbf{M}$ contains the basis vector M_i corresponding to i^{th} basis function.

Following the steps of the LLE algorithm, for k nearest neighbours the cost function for this problem can be defined as:

$$\epsilon(O) = \sum_{i=1}^{r} |M_i - \sum_{j}^{k} O_{ij} M_j|^2 \tag{2.3}$$

where r is the number of basis functions and the neighbourhood is defined by taking the Euclidean distance between the basis functions. The optimised weights O_{ij} are obtained after the cost function in equation 2.3 is converged. These optimised weights are then used to construct instrument basis functions (frequency basis functions corresponding to a particular instrument). This is done by randomly initialising a dataset F, where each F_s in F contains frequency basis functions corresponding to source s. Here, the number of sources are set to P. The cost function here can derived from equation 2.2.

$$\eta(F) = \sum_{p=1}^{P} |F_p - \sum_{j}^{k} O_{pj} F_j|^2 \tag{2.4}$$

Here, F_j represents the neighbour j of F_p and O_{pj} gives the contribution of the j^{th} data point. Once the frequency basis functions in F is optimised, then the time domain signals are reconstructed as detailed in section 2.3.2. Performance evaluation of the LLE algorithm is discussed in section 2.5

A new method of clustering using SNMF is discussed in the following section.

2.3 SNMF Clustering Algorithm

The SNMF clustering algorithm proposed here follows the same steps to obtain the NMF frequency basis functions as discussed earlier in section 1.5.4. We will write the NMF equation again for convenience:

$$\mathbf{X} \approx \mathbf{AB} \tag{2.5}$$

where the matrix $\mathbf{A}$ is of size $n \times r$ and the matrix $\mathbf{B}$ is of size $r \times m$, with $r < n$, m. Here, the matrix $\mathbf{A}$ that contains frequency basis functions is considered as a spectrogram. The clustering of frequency basis functions is obtained using the Shifted NMF.

2.3.1 Shifted Decomposition

As noted previously in chapter 1, an advantage of NMF is that there can be a single basis function for each note played by a given instrument, thereby capturing changes in timbre with pitch for each instrument or source. Therefore, instead of using a single frequency basis function to approximate all the notes played by the instrument (as with the standard Shifted NMF), here we will use all the frequency basis functions obtained using the NMF. This may solve the problem of the change of timbre with pitch.

Having obtained a set of basis functions using NMF, SNMF attempts to cluster the frequency basis functions as follows. At first, the matrix $\mathbf{A}$ is multiplied with a the transform matrix $\mathbf{Y}$ to scale the components in $\mathbf{A}$ from linear to log-frequency domain.

$$\mathbf{C} = \mathbf{YA} \tag{2.6}$$

The transform matrix $\mathbf{Y}$ is obtained by taking the absolute value of the Fourier transform of a bank of the complex exponentials, whose centre frequencies are geometrically spaced. In effect, it is the absolute value of the CQT proposed by Brown [45]. The log frequency basis function spectrogram $\mathbf{C}$ is then passed as an input to SNMF:

$$\mathbf{C} \approx \langle \langle \mathcal{RD} \rangle_{\{3,1\}} \mathcal{H} \rangle_{\{2:3,1:2\}} \tag{2.7}$$

Given the number of sources, P, then SNMF will look for instrument basis

functions that can be used to approximate $\mathbf{C}$. Here, $\mathcal{R}$ is a constant translation tensor. The multiplicative update equations for tensors $\mathcal{D}$ and $\mathcal{H}$ that defines frequency basis functions are as follows:

$$\mathcal{H} \leftarrow \mathcal{H} \cdot \left(\frac{\langle \langle \mathcal{RD} \rangle_{\{3,1\}} \mathcal{Y} \rangle_{\{3,1\}}}{\langle \langle \mathcal{RD} \rangle_{\{3,1\}} \mathcal{O} \rangle_{\{1,1\}}} \right) \tag{2.8}$$

where

$$\mathcal{Y} = \frac{\mathcal{C}}{\langle \mathcal{PH} \rangle_{\{2:3,1:2\}}} \tag{2.9}$$

$$\mathcal{P} = \langle \mathcal{RD} \rangle_{\{3,1\}} \tag{2.10}$$

and $\mathcal{O}$ is a tensor of all ones. Tensor $\mathcal{P}$ contains the translated instrument basis functions.

The multiplicative updates for the tensor $\mathcal{D}$ can be calculated by using following equations:

$$\mathcal{D} \leftarrow \mathcal{D} \cdot \left(\frac{\langle \mathcal{ZH} \rangle_{\{1:3,1:3\}}}{\langle \mathcal{WH} \rangle_{\{1:3,1:3\}}} \right) \tag{2.11}$$

where,

$$\mathcal{W} = \langle \mathcal{RO} \rangle_{\{1,1\}} \tag{2.12}$$

and

$$\mathcal{Z} = \langle \mathcal{RY} \rangle_{\{1,1\}} \tag{2.13}$$

Operator $\cdot$ in all the equations indicate elementwise multiplication. All division operations in all equations are elementwise unless otherwise stated. The number of translations k of an instrument basis function is appropriately chosen

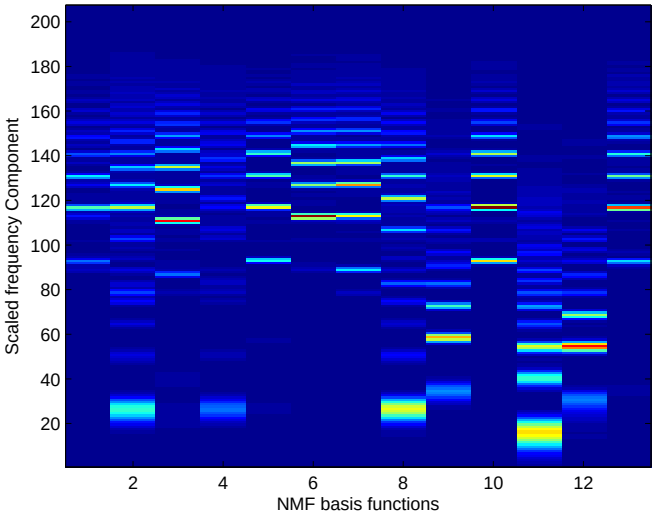

Figure 2.2: NMF basis function of input mixture in constant Q domain.

so that it covers all the notes played by a particular instrument in the music mixture. Assuming that each basis function in $\mathbf{C}$ corresponds to an individual note played by the individual instrument, then the activations of the SNMF model should indicate which basis functions in $\mathbf{C}$ are associated with which individual source, in effect clustering the basis functions.

Figure 2.2 shows the NMF basis functions in Constant Q domain of a input mixture of two sources. Figures 2.3 and 2.4 show the separated basis functions corresponding to source 1 and source 2 respectively. The x-axis shows the number of basis functions for individual notes to cover the highest pitch range played by the instrument in the test mixture. The figure shows the clear separation of basis functions associated with the different sources, hence these clustered basis function can be used to segregate the sources in question.

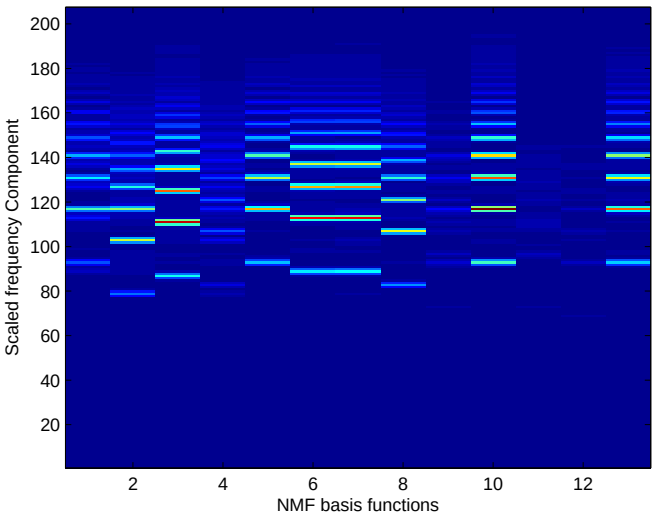

Figure 2.3: Separated Constant Q NMF basis functions for Source 1

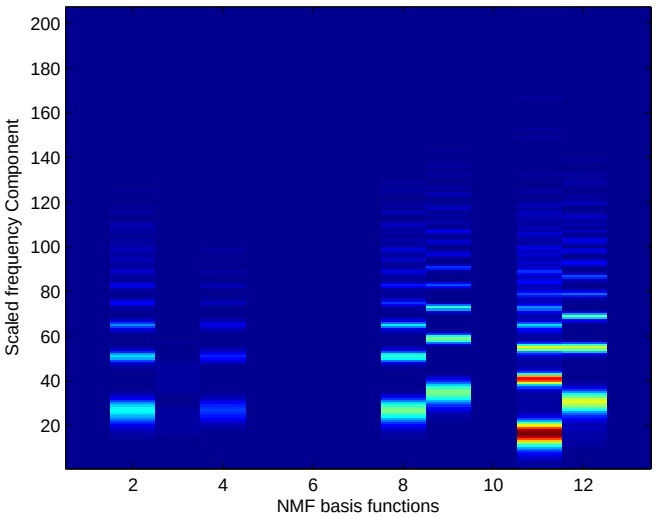

Figure 2.4: Separated Constant Q NMF basis functions for Source 2

2.3.2 Signal reconstruction

The next step of the clustering algorithm requires the identification of the clusters within the tensors obtained after the convergence and the recovery of the source spectrograms for the estimation of sources. To this end, we have introduced two different approaches to identify and recover the clusters of frequency basis functions.

One-to-one Mapping

In practice, identification of clusters is carried out by reconstructing the individual basis function spectrograms, $\mathbf{C}_s$ and comparing the energy in each source at each frame. Here $\mathbf{C}_s$ denotes the magnitude source spectrogram of the s^{th} source, where the total number of sources is equal to P. After the optimization of tensors $\mathcal{D}$ and $\mathcal{H}$, a frequency basis function spectrogram $\mathbf{C}_s$ can be represented by using the slices of tensors, $\mathcal{D}(:, s)$ and $\mathcal{H}(:, s, :)$, associated with a given source s.

$$\mathbf{C}_s = \langle\langle \mathcal{R}\mathcal{D}(:, s)\rangle\rangle_{\{3,1\}} \mathcal{H}(:, s, :)\rangle_{\{2:3,1:2\}} \tag{2.14}$$

The energy of the individual frame in each spectrogram $\mathbf{C}_s$ is compared with the corresponding frame of the other sources and the basis function in the original matrix $\mathbf{C}$ is allocated to the source which has the highest energy at that frame. This can be formulated as follows:

$$\mathbf{E}_s = \sum^{n} \mathcal{C}_s \tag{2.15}$$

where, matrix $\mathbf{E}$ of size $r \times s$ contains energy of each frame of each spectrogram of frequency basis function corresponding to the individual sources. The individual basis function is indexed by δ_s corresponding to respective sources. For a particular source s, δ_s can be defined by the following equation:

$$\delta_s(r) = \arg\max_s \left(\mathbf{E}(:, s)\right) \tag{2.16}$$

The index vector δ_s is of length r and it contains a binary mask indicating membership of a source or otherwise. The contents of δ_s are repeated along the rows and columns to match the number elements corresponding either to $\mathbf{A}$ or $\mathbf{B}$. This is done to filter out the time-frequency frames corresponding to the frequency basis functions that does not belong to the source in question. Here, we are using the matrix δ_{s1} of size $n \times r$ corresponding to $\mathbf{A}$. Thus, on the basis of energy content of individual frames in $\mathbf{C}$, the one-to-one mapping is achieved from the clusters in $\mathbf{C}$ to those in $\mathbf{A}$. As a result, for each instrument the frequency basis functions for each sound source are grouped together. The use of index matrix δ_{s1} to generate the source spectrogram $\mathbf{X}_s$ corresponding to the source s can be formulated as follows:

$$\mathbf{X}_s = (\mathbf{A} \cdot \delta_{s1})\mathbf{B} \tag{2.17}$$

Further, the X_s is used to generate a mask M_s corresponding to the source s:

$$\mathbf{M}_s = \left(\frac{\mathbf{X}_s^{\cdot 2}}{\sum_{p=1}^{P} \mathbf{X}_p^{\cdot 2}} \right) \tag{2.18}$$

where P is the number of sources. The mask M_s is then applied on the original complex-valued source spectrogram, X, to recover the phase information for the estimated magnitude spectrograms. This done as follows:

$$X_s = X \cdot \mathbf{M}_s \tag{2.19}$$

where X_s is the complex-valued source spectrogram estimated for source s. Finally, the re-synthesis of the separated sources is done by using the inverse STFT on the separated complex spectrograms.

SNMF Masking

An alternate approach to map the individual source spectrogram, $\mathbf{C}_s$ back in linear domain yielding $\mathbf{A}_s$ is also implemented. This is based on the fact that there is a one to one correspondence between the basis functions in $\mathbf{C}$ and $\mathbf{A}$. Therefore, the clustering obtained for $\mathbf{C}$ is equally valid for clustering in $\mathbf{A}$ which can be further partitioned into individual $\mathbf{A}_s$, where $\mathbf{A}_s$ contains the frequency basis functions associated with the s^{th} source. Hence, an approximate inverse CQT of the individual basis function spectrograms $\mathbf{C}_s$ may be used to obtain the frequency basis functions in $\mathbf{A}_s$. This is done as follows. The matrix $\mathbf{Y}^{'}$ (see equation 2.6) is multiplied with the basis function spectrogram $\mathbf{C}_s$ to obtain corresponding $\mathbf{A}_s$.

$$\mathbf{A}_s = \mathbf{Y}^{'}\mathbf{C}_s \tag{2.20}$$

Having obtained individual spectrogram for frequency basis functions $\mathbf{A}_s$,

the source spectrogram can be reconstructed using spectral masking.

The recovered source frequency basis functions $\mathbf{A}_s$ are used to generate a mask which is applied to $\mathbf{A}$. Then, $\mathbf{A}$ is passed through these masks to obtain the source frequency basis functions $\hat{\mathbf{A}}_s$. The frequency basis functions in $\hat{\mathbf{A}}_\mathbf{s}$ corresponding to source s is calculated using the following equation:

$$\hat{\mathbf{A}}_s = \mathbf{A} \cdot \left(\frac{\mathbf{A}_s^{\cdot 2}}{\sum_{p=1}^{P} \mathbf{A}_p^{\cdot 2}} \right) \tag{2.21}$$

As each row vector in $\mathbf{A}$ has a corresponding column vector in $\mathbf{B}$, clustering of the time activations is handled automatically. Then, the source magnitude spectrogram is obtained as follows:

$$\mathbf{X}_s = \hat{\mathbf{A}}_s \mathbf{B}_s \tag{2.22}$$

Thereafter, the generation of the complex valued spectrogram is done using the mask generated from $\mathbf{X}_s$ and the resynthesis of the individual sources is obtained as stated in equation 2.18 and 2.19.

2.4 Experiments

The algorithm was implemented in Matlab for single channel audio mixtures. The SNMF model was tested for 25 monaural input mixtures of 2 instruments from a total of 15 different orchestral instruments taken from a sample library [91] including brass, woodwind and strings. The signals in the test set varied

in duration of roughly 4 to 8 seconds with a sampling frequency of $44.1kHz$. To imitate real world melodies, the notes played by individual instruments in the input mixture were in harmony and covered pitches from as low as $87Hz$ to pitches up to $1500Hz$. The source signals were mixed with unity gain. More details on how the database was created can be found in [20].

The magnitude spectrogram of the time-domain signal were obtained using the STFT. Hann windows of 4096 samples in length of were used and there was 75% overlapping between successive Hann windows. The number of NMF basis functions for all the test signals were equal to 13. The number of frequency basis functions may vary with the length (time duration) of the test samples in the testset used. NMF was run for 300 iterations. The constant Q transform used 24 frequency bins per octave covering frequencies ranging from $55Hz$ to $22.05kHz$. Tensors $\mathcal{D}$ and $\mathcal{H}$, in equation 2.7, were randomly initialised with non negative values. As discussed in section 1.5.6 the cost function used for SNMF decomposition is the commonly used KL divergence (see equation 1.36). The multiplicative updates and positive initialization for $\mathcal{D}$ and $\mathcal{H}$ ensures the factorisation is non negative. The algorithm is set for number of sources equal to 2 and it ran for 50 iterations. Here, the number of translations can be varied in the range between 5 to 12 to check the robustness of the algorithm. However, for the given testset, the number of time shifts i.e. allowable translations, k, was set to 7.

An example of an audio mixture spectrogram is shown in figure 2.5. The audio mixture comprises of two sources. Figures 2.6 and 2.7 show the separated

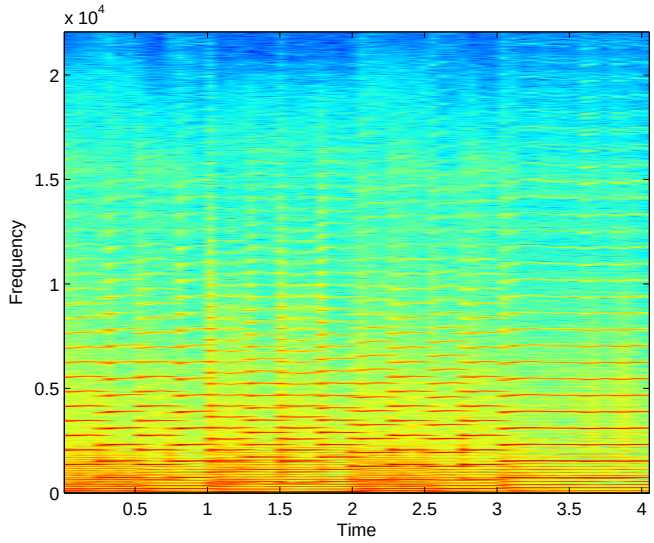

Figure 2.5: Mixture spectrogram of the two sources

signals, corresponding to source 1 and source 2 respectively, using the SNMF clustering algorithm with one-to-one mapping. Figures 2.8 and 2.9 show the separation of two sources using the SNMF clustering algorithm using mask. It can be seen from the figures that the interference due the sources in the time period between 1 and 2 seconds is considerably lower for the SNMF clustering algorithm using a mask that those for one-to-one mapping. However, both the methods can be potentially used for separating sound sources in mono mixtures. The quality of separation is evaluated in the following section. Audio examples of the testset and the separated source signals can be found at [92].

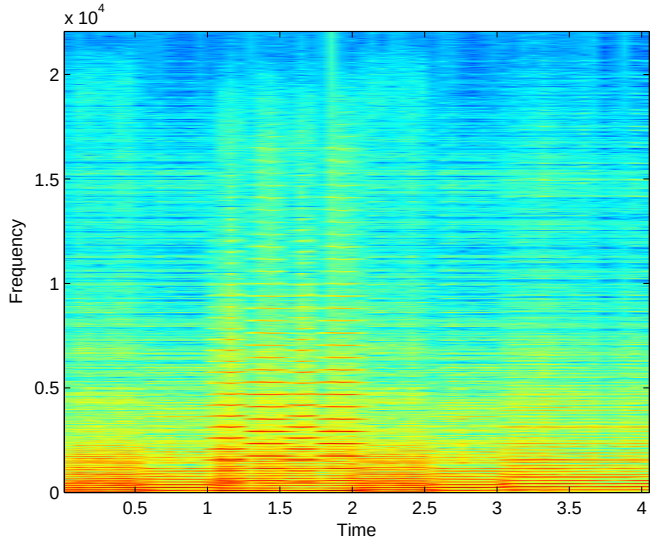

Figure 2.6: Separated source 1 using SNMF clustering with one-to-one mapping.

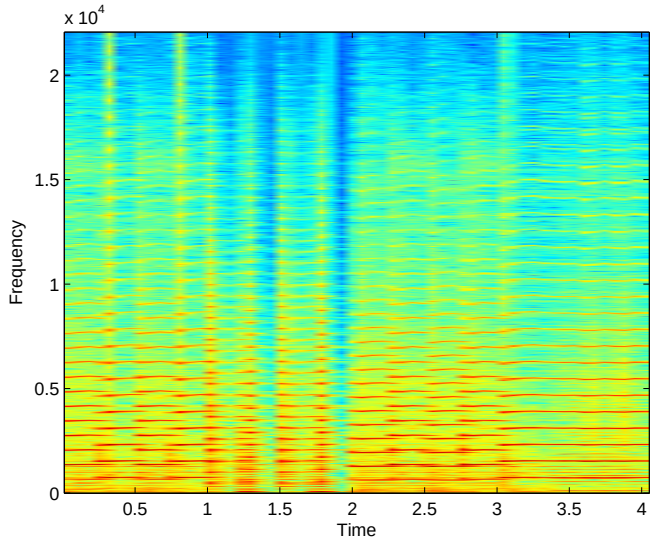

Figure 2.7: Separated source 2 using SNMF clustering with one-to-one mapping.

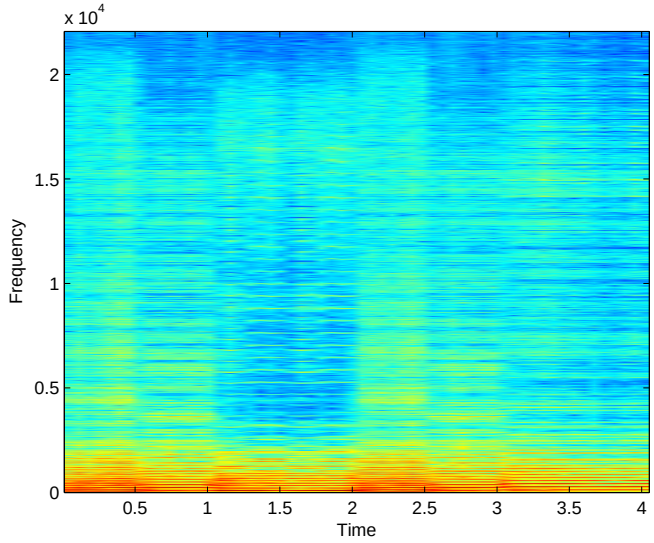

Figure 2.8: Separated source 1 using SNMF clustering algorithm using mask.

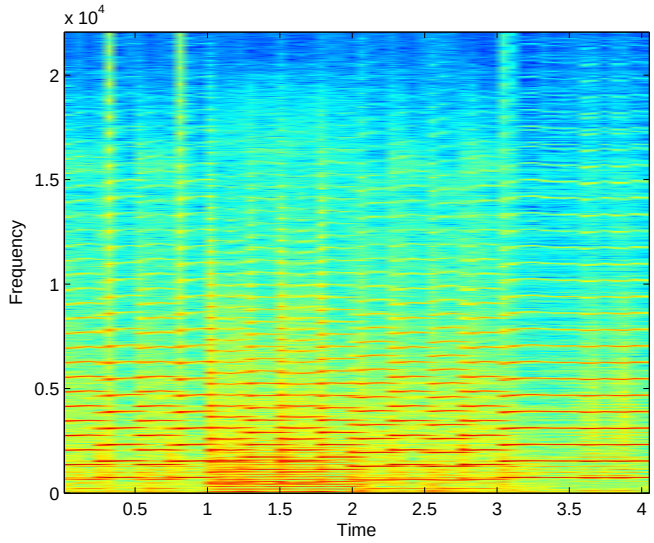

Figure 2.9: Separated source 2 using SNMF clustering algorithm using mask.

2.5 Results

The performance of the SNMF clustering algorithms were evaluated using the quality measures signal-to-distortion ratio (SDR), the signal-to-interference ratio (SIR), and the signal-to-artifacts ratio (SAR). These measures are widely used for the evaluation of separation quality and the details of these metrics can be found in [89]. SDR determines the overall sound quality of the recovered signal, SIR measures the interference of other sources in the separates sound source and SAR calculates the artefacts present in separated signal.

Following the notations of [89], let s be the original input signal. Then, s can be decomposed as:

$$s = s_{target} + e_{interf} + e_{noise} + e_{artif} \tag{2.23}$$

where s_{target} is the reconstructed output and where e_{interf}, e_{noise} and e_{artif} are respectively the interferences, noise and artifacts error terms. SDR determines the overall sound quality of the recovered signal and it is fomulated as:

$$SDR_{dB} = 20log_{10}\left(\frac{s_{target}}{e_{interf} + e_{noise} + e_{artif}}\right) \tag{2.24}$$

SIR is the measure of the interference of each source on the other separated sound sources. It can be defined as:

$$SIR_{dB} = 20log_{10}\left(\frac{s_{target}}{e_{interf}}\right) \tag{2.25}$$

And SAR calculates the artifacts present in separated signal. It can be calculated by the following equation:

$$SAR_{dB} = 20log_{10}\left(\frac{s_{target} + e_{interf} + e_{noise}}{e_{artif}}\right) \tag{2.26}$$

The original source signals were used as a reference for the performance evaluation.

clustering	SDR	SIR	SAR
C_{km}	0.80	10.96	3.30
C_{nmf}	2.89	12.72	4.59
SNMF$_{map}$	5.40	15.27	6.90
SNMF$_{mask}$	8.94	23.69	9.72

Table 2.1: Mean SDR, SIR and SAR for separated sound sources using SNMF clustering

C_{nmf} and C_{km} are the two other clustering methods used for comparison [41]. The clustering algorithms C_{nmf} and C_{km} represents the NMF clustering and k-means clustering discussed earlier in chapter 1 . All the clustering algorithms were tested using the same set of input mixtures to compare the results. All the results for mean SDR, SIR and SAR are shown in dB. The performance of two proposed clustering algorithms Shifted NMF with one-to-one mapping, SNMF$_{map}$ and shifted NMF with masking, SNMF$_{mask}$ are shown in the Table 2.1. It can be seen from the data that SNMF clustering using the mask gave better results than the SNMF clustering with '*one-to-one*'. It is also evident from the Table 2.1 that both the proposed clustering algorithms SNMF$_{map}$ and SNMF$_{mask}$ outperform the other clustering techniques. This was also evident

from the informal listening test that the separation quality of the estimated sources using SNMF$_{mask}$ were better than those for other listed algorithms in table 2.1. We tested the LLE clustering algorithm detailed in section 2.2 for the same set of audio mixtures. However, we did not observe any convergence of the LLE cost function, hence the results were uncorrelated and so were not included. Also, the results obtained using the LLE method were not consistent, i.e. we were getting a random separation each time for the same test mixture. This may be due to the fact that NMF gives a sparse representation and for the reasons stated in [75, 76], LLE is quite sensitive to the sparse data sets. Processing of sparse data by LLE results in deteriorating the local geometry of the data manifolds in the embedding space. This is because the reconstruction weights of the embedding space are replaced by the reconstruction weights obtained from the original data space mainly due to the insufficient number of data points.

2.6 Conclusions

In this chapter, we have introduced two novel methods of clustering of NMF basis functions. Firstly, we implemented LLE clustering which maps a higher dimension data to a lower dimension data by learning weights. For this method we used the source-filter model discussed in section 1.6.1. The sound source separation obtained by LLE clustering did not show improvements over the previously implemented methods.

Secondly, we presented two SNMF based clustering algorithms for single

channel blind source separation which used SNMF to cluster the frequency basis functions obtained from the standard NMF. We also dealt with the change in timbre with pitch by assuming a separate basis function for each note being played by the individual instruments. For the first SNMF clustering algorithm, we used one-to-one mapping from the Constant Q domain to the linear spectrogram to eliminate the need of inverse Constant Q transform. Alternatively, we used an approximate inverse transform followed by masking of the original spectrogram (containing basis functions) with the recovered basis functions to obtain the clustered basis function in the linear domain. We tested the algorithms on various test input mixtures of two sources. The tests show a significant improvement of the sound quality as compared to the unsupervised clustering done by Spiertz and Gnann [41]. Furthermore, these clustering algorithms can be extended for input mixtures of n sources. Therefore, clustering using SNMF is an effective way to cluster pitched basis function to separate out harmonic instruments. In the next chapter, we will discuss a recently proposed method [85] to calculate CQT in order to improve the separation performance of the standard SNMF algorithm (SNMF_{cqt}).

Chapter 3

Shifted NMF algorithm using an improved Constant-Q Transform

3.1 Introduction

This chapter contains a minor contribution of the thesis. Here, we will discuss two previously proposed algorithms and will make an attempt to combine them in order to improve the performance of a separation algorithm. In chapter 1, we have discussed various techniques to cluster the NMF basis functions to separate sound sources from an single channel audio mixture. Thereafter, we proposed two methods to map the frequency basis functions to their respective sources in chapter 2. Furthermore, we noted that the Shifted NMF based algorithms use the shift invariant property of the instrument basis function to cluster the frequency basis functions obtained from a given mixture signal. Also, the Shifted

NMF requires a log-frequency spectrogram which is obtained using the constant Q transform. However, the use of CQT makes the problem difficult to solve due the fact that a true inverse of a constant Q transform does not exist. Therefore, there is no guarantee of the perfect reconstruction of the sources from the instrument basis functions (frequency basis functions corresponding to an instrument) even though we achieve a good separation of the underlying sources. We will try to address this issue in the chapter by using a recently proposed improved method [85] to calculate the CQT and the approximate inverse CQT. This work was inspired by the method developed in [84]. We argue that by using this method to calculate CQT, we may improve the separation of the sources using the standard Shifted NMF algorithm. Further, we will test whether the inverse CQT gives a better reconstruction of the separated signals.

Before we discuss the proposed SNMF algorithm in detail, it is important to note that the standard SNMF algorithm presented here is different from the SNMF clustering algorithm discussed in chapter 2. Here, the standard SNMF algorithm finds shift invariance in a CQ spectrogram obtained directly from the magnitude spectrogram of the original mixture signal.

3.2 System model

Figure 3.1 shows the signal flow in the system model. A test mixture in time domain is first converted into the constant Q domain using the CQT. Thereafter, the shift-invariant property of SNMF algorithm is used to determine

the instrument basis functions. Furthermore, spectral masking is incorporated to improve the quality of separation. Finally, the separated signal is recovered by an improved method to calculate the approximate inverse CQT. We will first explain the basic principle involved in calculating the CQT as it would assist in understanding other features in the system model.

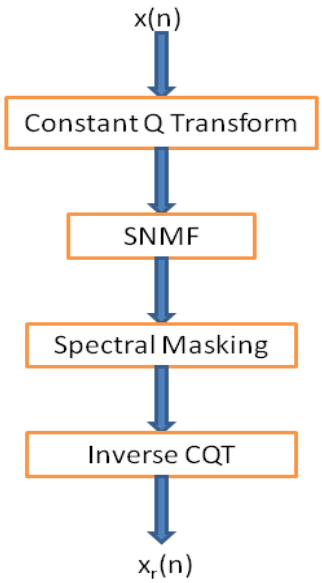

Figure 3.1: Signal flowchart of the System model

3.3 Constant Q Transform

As discussed earlier in chapter 1, the CQT can be obtained by modulating the audio signal with a bank of complex exponentials, whose centre frequencies are

geometrically spaced. The geometrical spacing was used because it matched the resolution of the most widely used tuning system. Further, we mentioned that the CQT can be efficiently calculated in the Fourier domain by taking the transform of the complex exponentials used to obtain the CQT, yielding a sparse matrix $\mathbf{Y}$. The CQT can then be obtained by multiplying a linear domain spectrogram $\mathbf{P}$ by the the conjugate transpose of the sparse matrix $\mathbf{Y}$ as shown in equation. Following the terminology of [85] we will the address the matrix $\mathbf{Y}$ as a spectral kernel.

$$\mathbf{Q} = \mathbf{Y}^*\mathbf{P} \tag{3.1}$$

The CQT method [85] used here is an extension to the method discussed above[84]. Earlier, since a wide range of frequencies was considered ($60Hz$ to $16kHz$), the spectral kernel generated was not very sparse. This is due to the reason that the frequency response of the higher bins are wider as compared to the lower frequency bins. To overcome this, here, the new CQT algorithm processes each octave in the signal one by one starting from highest to lowest to calculate the CQT coefficients of a given spectrogram. This is done as follows.

A spectral kernel matrix $\mathbf{Y}$ is used that produces the CQT for the highest octave only. Once, the highest-octave CQT bins are calculated using the spectral kernal and the DFT block corresponding to the entire signal, the input signal is lowpass filtered and downsampled by a factor two. Again, the CQT bins are calculated for the second highest octave and the process is repeated until the lowest desired octave is reached in terms of calculating the CQT bins. It is

important to note that for each subsequent octave the CQT bins are calculated using exactly the same DFT block size, $\mathbf{X}_d$ and the spectral kernel, $\mathbf{Y}$ as shown in equation 3.2, where d is an index value of the d^{th} octave.

Let $x[n]$ be the input signal for which a CQT is needed. Therefore, for the d^{th} successive octave, $x[n]$ is lowpass filtered and down-sampled by a factor of 2^d. let $x_d(n)$ represents a signal generated by decimating $x[n]$ by 2^d times. And let $\mathbf{X}_d$ contain the DFT values of $x_d(n)$. Then, the CQT coefficients $\mathbf{C}_d$ for octave d is obtained as

$$\mathbf{C}_d = \mathbf{Y} * \mathbf{X}_d \tag{3.2}$$

The above process is repeated for the subsequent octaves. More importantly, the spectral kernal remains constant for all the octaves. Also, the DFT length in samples remains constant but the effective FFT length in terms of seconds doubles after every decimation. For simplicity, let $\mathbf{C}$ denote a Constant Q spectrogram and it contains the absolute value of all the CQT coefficients obtained by processing the desired number of octaves.

In the proposed kernel structure, the number of points where $\mathbf{C}$ is evaluated is not same for all bins from the lowest frequency bin of the lowest octave to the highest frequency bin of the highest octave. It decreases by a factor two per octave as we move down from the higher to lower octave. This would give a representation of CQ spectrogram that is not useful for factorisation techniques as it does not yield a rectangular matrix. To overcome this problem a rasterised version of the CQT was then obtained. Data interpolation between the time

points in $\mathbf{C}$ was used to obtain this rasterised CQT data structure. This data structure and the user interface, it is possible to obtain the entire CQT matrix representing the input signal or the CQT coefficients corresponding to a certain time slice.

Having obtained the rasterised CQT matrix, $\mathbf{C}$, we will now use it as the input to the SNMF algorithm to perform sound source separation.

3.4 Shifted Non-negative Matrix Factorisation

As noted previously, the SNMF model exploits the shift-invariant property of the frequency basis function to cluster them to corresponding sources, provided log-frequency resolution is used. Having obtained the Constant Q spectrogram $\mathbf{C}$ (obtained from the audio spectrogram of the audio mixture) of size $n \times m$, where m are the number of time frames along the n frequency bins, SNMF is used to obtain instrument basis functions in a similar manner as detailed in section 2.3.1.

As an example of SNMF using the new CQT method, figure 3.2 gives a pictorial representation of the frequency basis functions of the input mixture in the Constant Q domain. Figures 3.3 and 3.3 show the Constant Q spectrogram of the instrument basis functions obtained after the activation of SNMF model. It can seen through visual inspection that the frequency basis functions have separated reasonably well (frequency basis functions between 1 second to 1.5 seconds is wrongly separated) and would assist in reconstruction of individual

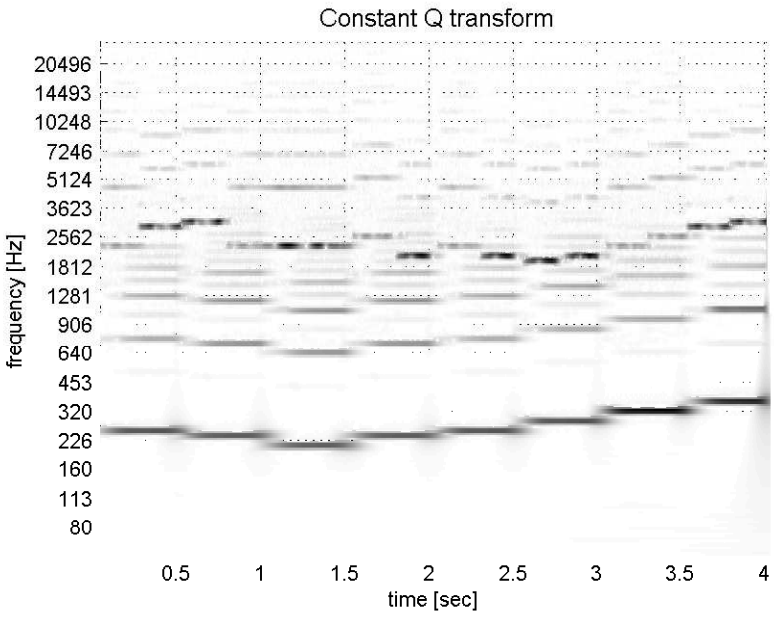

Figure 3.2: Frequency basis functions in Constant Q domain of a music mixture source signals.

3.5 Spectral masking and Signal reconstruction

As discussed in section 2.3.2 the individual source spectrogram C_s for s^{th} source can be reconstructed by using the slices of tensors by using the slices of tensors, $\mathcal{D}(:, s)$ and $\mathcal{H}(:, s, :)$.

$$\mathbf{C}_s = \langle\langle \mathcal{R}\mathcal{D}(:, s)\rangle\rangle_{\{3,1\}} \mathcal{H}(:, s, :)\rangle_{\{2:3,1:2\}} \tag{3.3}$$

The estimated source spectrogram is used to generate a mask to be applied

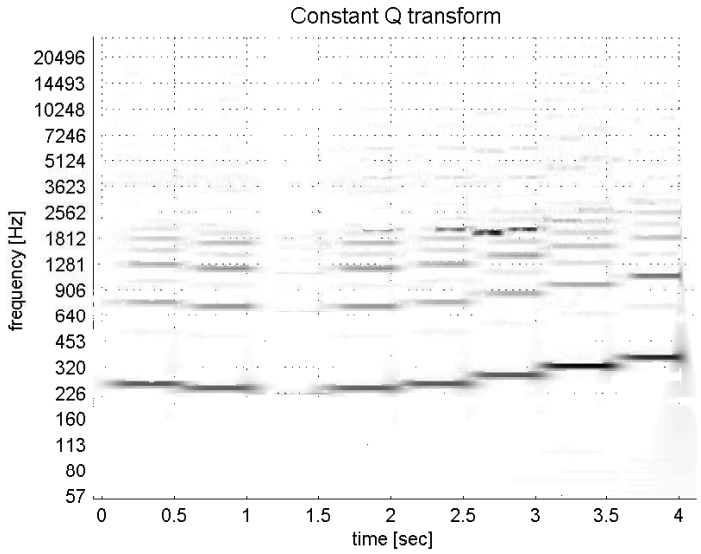

Figure 3.3: Separated Constant Q frequency basis functions of Source 1

on the original spectrogram $\mathbf{C}$. The generated masks are then applied to $\mathbf{C}$ to obtain filtered sources spectrograms $\hat{\mathbf{C}}_\mathbf{s}$. A masking filter $\hat{\mathbf{M}}_\mathbf{s}$ can be calculated as shown in equation:

$$\hat{\mathbf{M}}_s = \left(\frac{\mathbf{C}_s^2}{\sum_{p=1}^{P} \mathbf{C}_p^2} \right) \tag{3.4}$$

Furthermore, the filtered source spectrograms $\hat{\mathbf{C}}_\mathbf{s}$ in constant Q domain are obtained using the equation:

$$\hat{\mathbf{C}}_\mathbf{s} = \mathbf{C} \cdot \hat{\mathbf{M}}_s \tag{3.5}$$

where $\cdot$ indicates element-wise multiplication in equation 3.5. The recovered source spectrograms are then converted into time domain signal by using an approximate inverse CQT (ICQT) [85].

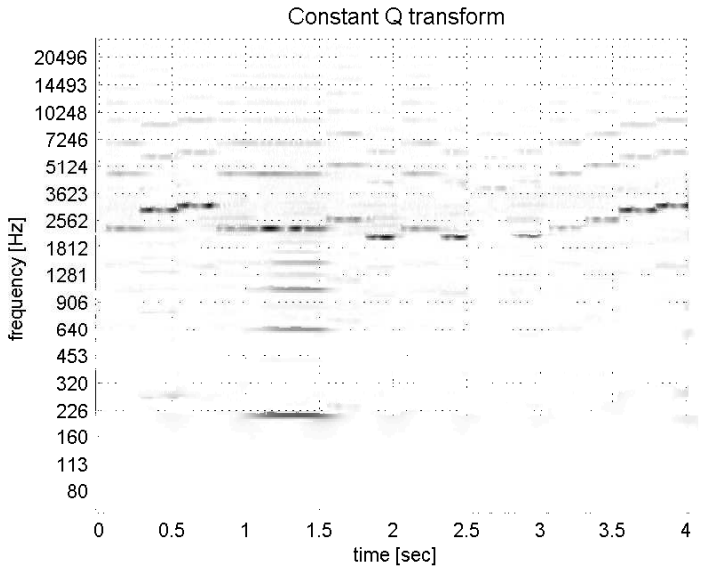

Figure 3.4: Separated Constant Q frequency basis functions of Source 2

3.6 Inverse CQT

The ICQT is used to approximately reconstruct the input signal $x[n]$. This is done by using CQT coefficients in matrix $\hat{\mathbf{C}}_\mathbf{s}$ obtained using equation 3.5. This is done octave wise in a similar manner as we calculated the forward CQT but is in a reverse order. Let $\hat{x_d}$ is the part of time-domain input signal $x[n]$ that represents one octave. The complex valued DFT coefficients are calculated using the following equation.

$$\hat{\mathbf{X}}_d = \mathbf{Y}\hat{\mathbf{C}}_\mathbf{sd} \tag{3.6}$$

Then, the inverse STFT is used to calculate the $\hat{x_d}$. Thus, processing the CQT spectrogram over all the octaves, the ICQT reconstruct $\hat{x}(n)$ which is an

approximation of the input signal $x[n]$.

This new method to calculate the CQT coefficients for each octave gives a better inverse CQT than the previously proposed methods because of the following reasons. Firstly, the method uses a spectral transform matrix $\mathbf{Y}$ for b_o number of CQT bins, the signal is repetitively analysed from higher octaves to lower octaves to obtain the CQT coefficients which increases the redundancy of the transform. However, this increase in redundancy helps in capturing most of the data features required to obtain a high quality inverse CQT. The redundancy,R_f, is directly proportional to the highest frequency analyzed i.e. the highest octave chosen. The separation quality can be further improved by increasing the number of CQT bins per octave, B_o. However, R_f and B_o are optimally chosen for computational efficiency. An analysis of quality of reconstruction as a function of R_f and B_o can be found in [85]. It is important to note that this octave by octave processing of audio signal makes it impossible to use this version of the CQT for clustering of linear domain basis functions as there is no way to directly map the linear frequency basis functions to the CQT domain using this approach.

A complete implementation of CQT can be found in [85]. In this chapter, we have used the MATLAB toolbox of the reference implementation of the above discussed method provided at [83] to obtain the Constant Q spectrogram.

3.7 Experimental Set-up

The experimental setup for this experiment is the same as described in section 2.4. The same set of input mixtures were taken for evaluating the performance of the algorithm. The parameters for CQT is described as follows.

The Fast Fourier Transform is used to calculate the successive DFT blocks for each octave. 24 frequency bins per octave, $\mathbf{B}_o$, are used for the CQT to obtain log-frequency resolution. Frequencies ranges from 55 Hz to 15 kHz are used to calculate the frequency bins. More details on the calculation of CQT coefficients for each octave in the spectrogram can be found in [85]. For the SNMF model, the number of sources is equal to 2. The cost function used for the SNMF decomposition is same as used for NMF. The SNMF algorithm is run for 50 iterations. Individual spectrograms of the separated signals are then obtained through spectral masking followed by reconstruction of individual signal as explained in section 3.5.

For the given 25 test mixture, the number of allowable translations, k, varies between 4 and 9. Multiple tests were run for the different number of allowable time shift and the separated sources with the highest separation quality were picked.

Figure 3.5 shows the audio spectrogram of a test mixture signal and its corresponding separate sources. Figures 3.6 and 3.7 show the separation of two sources using the improved CQT method in SNMF algorithm. It can be seen from the spectrogram that the proposed algorithm achieves the separation of the signals corresponding to the sources. On hearing, the melodies played by

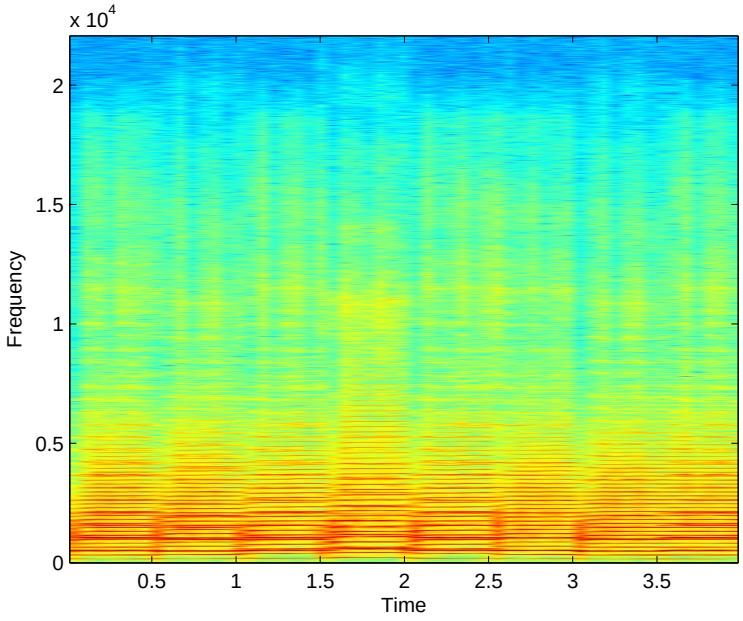

Figure 3.5: Mixture spectrogram of the two sources

both the estimated sources were found to be separated well. The algorithm was also found to separate notes simultaneously played by both the pitched instruments in the mixture with a small interference of harmonics related to that note. Performance evaluation of the SNMF algorithm, to measure the separation quality, is done in the next section.

3.8 Results

The original separated signal from the sample library [91] were used as reference to measure the performance of the SNMF algorithm. For comparison, the original SNMF algorithm is used and is denoted by $SNMF_{cqt}$. Details on the

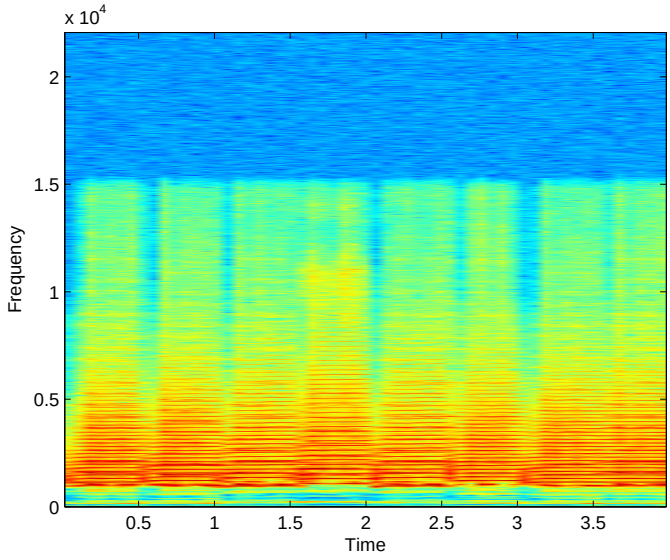

Figure 3.6: Separated source 1 using the improved CQT method in SNMF algorithm.

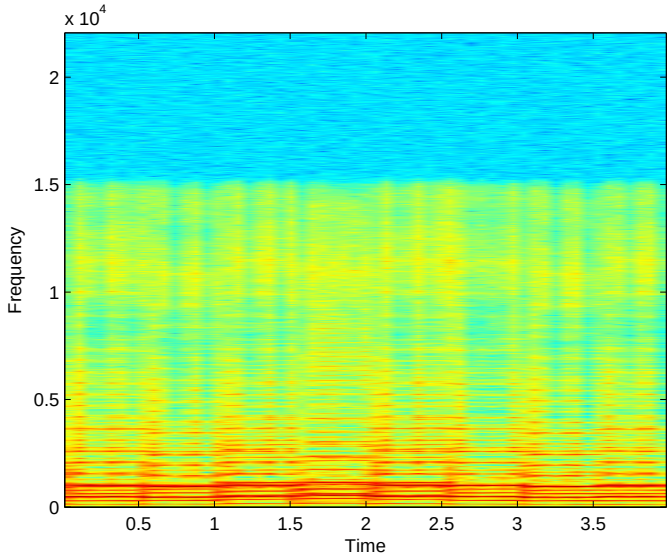

Figure 3.7: Separated source 2 using the improved CQT method in SNMF algorithm.

clustering	SDR	SIR	SAR
$SNMF_{cqt}$	-1.85	14.97	3.46
$SNMF_{ncqt}$	10.88	25.44	11.47

Table 3.1: Calculated mean SDR, SIR and SAR for separated sound sources

$SNMF_{cqt}$ algorithm can be found in [44]. $SNMF_{ncqt}$ represents the SNMF algorithm discussed in this chapter. Quality measures for both the listed algorithms were calculated. Table 3.1 shows the quality measures calculated for $SNMF_{cqt}$ and $SNMF_{ncqt}$. All the results i.e. mean SDR, SIR and SAR, are in dB. Both the SNMF algorithms, $SNMF_{cqt}$ and $SNMF_{ncqt}$, are coded in MATLAB and are tested for the same set of test mixture discussed in section 2.4. From table 3.1, we can see an improvement of mean SDR of more than 10 dB by using the new CQT in the SNMF algorithm for the same set of audio mixture and test parameters. As a result, on listening to the separated signals the sources can be clearly identified with few artefacts. These artefacts are due to interference of melodies played by one source on other in the mixture. Overall it can be stated that by replacing the method to calculate the CQT in the SNMF algorithm, considerably improves the separation quality. Hence, the algorithm $SNMF_{ncqt}$ outperforms the monaural source separation algorithm $SNMF_{cqt}$. Audio examples for the estimated audio source signals can be found at [92].

3.9 Conclusions

We have demonstrated that by replacing the method to the calculate CQT coefficients in Shifted NMF algorithm with an improved method of calculating

the CQT, the separation quality can be significantly improved. The CQT coefficients were calculated on an octave-by-octave basis to efficiently capture all the features to improve the separation quality. We have also discussed that the separation quality is directly proportional to the redundancy factor R_f and the number of frequency bins per octave B_0, of the newly proposed CQT. We tested the algorithm for multiple input mixtures of two sources that contained melodies covering wide range of frequencies. Finally, we compared the results obtained with the SNMF algorithm discussed in [44]. It was evident from the tests that by replacing the method to calculate the CQT transform, we can significantly improve the sound quality of the separated sources. However, it should be noted that this improved CQT method cannot be used for SNMF based clustering due to the lack of a direct mapping matrix from the linear domain to the log-frequency frequency domain. In the following chapter, we will modify the structure of the model in the SNMF clustering algorithm to improve the sound separation results.

Chapter 4

Incorporating the CQT in SNMF to improve clustering

4.1 Introduction

In previous chapters, we have shown that Shifted Non-negative Matrix factorisation (SNMF) based methods can be used to group NMF basis functions. However, the clustering of basis functions using SNMF uses a Constant Q Transform (CQT) of the frequency basis functions. Here, we argue that incorporating the CQT into the SNMF model can be used to better the separation quality of individual sources. An algorithm [87] is presented to estimate sound sources and will be shown to be an improvement to the existing techniques.

The system model for the proposed algorithm is shown in figure 4.1.

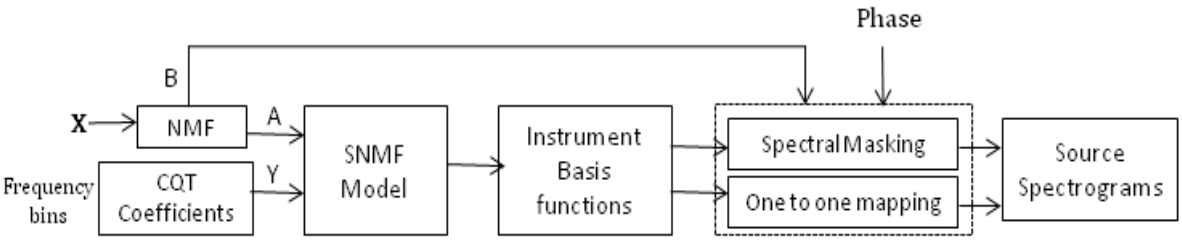

Figure 4.1: Block Diagram of the System model

Following the terminology of previous chapters, the magnitude spectrogram $\mathbf{X}$ of the input mixture is obtained by using the short-time Fourier transform (STFT). Then, the non-negative factorisation of $\mathbf{X}$ results in $\mathbf{A}$ and $\mathbf{B}$. Also a transform matrix $\mathbf{Y}$ is calculated using CQT. This is done by generating a constant Q filterbank. Then, the frequency basis functions contained in $\mathbf{A}$ and the CQT coefficients stored in $\mathbf{Y}$ are fed into the SNMF clustering model to recover the instrument basis functions. Thereafter, the source spectrograms for individual sources are recovered using two techniques. They are spectral masking and one-to-one mapping. The clustering techniques are discussed in chapter 2. The reconstruction of the individual sources is done using Weiner filtering.

As discussed earlier, the SNMF clustering algorithm uses a CQT. However, the use of CQT in the clustering algorithm makes it difficult to recover the separated sources. This is because there is no true inverse of CQT available. Although, we can use one-to-one mapping discussed in section 2.3.2 to recover the individual sources, the separation obtained using the spectral masking is considerably better than that of one-to-one mapping. Keeping this mind, the modification proposed here is principally aimed at the spectral masking

approach, as there is no need for the inverse in the one-to-one mapping. However, the approximate inverse CQT in the case of spectral masking does not give a perfect reconstruction of the signals associated with sources. This results in the deterioration of the separation quality of the individual sources. In order to avoid using frequency basis functions in the CQT domain, we propose to incorporate the CQT mapping in the SNMF model, with the aim of improving the clustering and separation. This is explained in the next section.

4.2 Methodology

4.2.1 Linear Frequency Domain Approximation of SNMF

The proposed modification of the SNMF clustering algorithm is follows:

$$\mathcal{A} \approx \langle \langle \mathcal{P}\mathcal{D} \rangle_{\{3,1\}} \mathcal{H} \rangle_{\{[2:3],[1:2]\}} \tag{4.1}$$

where, $\mathcal{P}$ is constant tensor of size $n \times k \times f$ and can be obtained using equation 4.2.

$$\mathcal{P} = \langle \mathcal{Y}\mathcal{R} \rangle_{\{1,1\}} \tag{4.2}$$

The SNMF model uses the tensors $\mathcal{A}$ and $\mathcal{Y}$ as the input parameters. The tensor $\mathcal{A}$ in equation 4.1 is the same as the matrix $\mathbf{A}$ in equation 1.33. The tensor $\mathcal{A}$ contains NMF frequency basis functions in the linear frequency domain and

is considered as a spectrogram of the linear domain frequency basis functions. Another input parameter to SNMF, the transform tensor $\mathcal{Y}$ of size $f \times n$ contains the CQT coefficients, where f is the number frequency bins. Again the transform tensor is the same as the transform matrix $\mathbf{Y}$ as discussed in section 2.3.1.

$\mathcal{R}$ is a translation tensor of dimension $f \times k \times f$ for k possible frequency translations. $\mathcal{R}$ translates the instrument basis functions in $\mathcal{D}$ up or down in frequency by half a tone to approximately cover all the notes played by the particular instrument. The tensor $\mathcal{D}$ of size $f \times s$ contains instrument basis functions for each source, where s is the number of sources. Tensor $\mathcal{H}$ of size $k \times s \times r$ denotes a time activation function. For example, $\mathcal{H}(i\,,\,j\,,\,:)$ indicates the time envelope for the i^{th} translation of the j^{th} source. It gives the temporal information about a given note that is being played by a particular instrument.

The spectrogram $\mathcal{A}$ is then factorised using the SNMF model to approximately determine the instrument basis functions as shown in equation 4.3.

The cost function used to obtain tensors $\mathcal{D}$ and $\mathcal{H}$ is the same as used for NMF. Therefore, the SNMF problem using KL divergence can be defined as

$$\langle \mathcal{L}, \mathcal{H} \rangle = \min_{\mathcal{L},\mathcal{H} \geq 0} \mathbf{D}_{KL}(\mathcal{A} || \langle \mathcal{L}\mathcal{H} \rangle_{\{2:3,1:2\}}) \tag{4.3}$$

where $\mathcal{L}$ denotes

$$\mathcal{L} = \langle \mathcal{P}\mathcal{D} \rangle_{\{3,1\}} \tag{4.4}$$

In equation 4.4, $\mathcal{P}$ is a constant tensor. Therefore, the problem defined in equation 4.3 is reduced to minimising the divergence between $\mathcal{A}$ and the product of non-negative tensors $\mathcal{D}$ and $\mathcal{H}$.

4.2.2 Update Equations

The update equations for tensor $\mathcal{D}$ and tensor $\mathcal{H}$ are derived using the cost function described in 1.36. The iterative multiplicative updates used for the translated frequency basis functions in $\mathcal{D}$ are determined in a similar manner as done in [44]. This can be formulated as follows:

$$\mathcal{D} \leftarrow \mathcal{D} \cdot \left(\frac{\langle \langle \mathcal{P}\mathcal{A} \rangle_{\{1,1\}} \mathcal{H} \rangle_{\{[1,3],[1,3]\}}}{\langle \langle \mathcal{P}\mathcal{O} \rangle_{\{1,1\}} \mathcal{H} \rangle_{\{[1,3],[1,3]\}}} \right) \tag{4.5}$$

where $\mathcal{O}$ of size $n \times r$ is a tensor of all ones. Similarly, the multiplicative updates for the activation functions in $\mathcal{H}$ are calculated as follows:

$$\mathcal{H} \leftarrow \mathcal{H} \cdot \left(\frac{\langle \langle \mathcal{P}\mathcal{D} \rangle_{\{3,1\}} \mathcal{A} \rangle_{\{1,1\}}}{\langle \langle \mathcal{P}\mathcal{D} \rangle_{\{3,1\}} \mathcal{O} \rangle_{\{1,1\}}} \right) \tag{4.6}$$

The tensors $\mathcal{D}$ and $\mathcal{H}$ are constrained to be non-negative. This is ensured by random positive initialisation and multiplicative updates. After the factorisation, the individual instrument basis functions can be reconstructed using the slices of tensor, $\mathcal{D}(:, s)$ and $\mathcal{H}(:, s, :)$. This is shown in equation 4.7.

$$\mathcal{A}_s \approx \langle \langle \mathcal{P}\mathcal{D}(:, s) \rangle_{\{3,1\}} \mathcal{H}(:, s, :) \rangle_{\{2:3,1:2\}} \tag{4.7}$$

where $\mathcal{A}_s$ denotes a spectrogram containing instrument basis functions for source

s.

It is important to note that, this method of grouping of frequency basis functions is different from previously proposed methods in chapter 2 because of the following two reasons. Firstly, the SNMF model uses the linear domain NMF basis functions as an input and the CQT transform matrix is fed into the SNMF algorithm to exploit the shift-invariant property. This is done by using the CQT transform matrix to map the linear domain NMF basis functions to the CQT domain before every iteration until the convergence is achieved. Secondly, the use of the CQT inside the SNMF model avoids the need to use the inverse CQT for recovering the NMF basis functions. As a result, the separated NMF basis functions, contained in $\mathcal{A}_s$, are in the linear domain, and so this should lead to a better approximation of the original sources. Thus, the linear frequency domain approximation of the SNMF model can be used to separate frequency basis functions corresponding to their respective sources in a given music mixture.

4.3 Signal Reconstruction

Having obtained the clustering of the basis functions, the individual source spectrograms can be reconstructed by the two techniques used in section 2.3.2. However, the estimated source spectrograms here are in linear domain rather than in log-frequency domain. Therefore, the processing step of converting the frequency basis functions from log-frequency domain to linear domain will vanish. Thus, the two methods for the reconstruction of the sound sources are

as follows.

4.3.1 One-to-one mapping

The first method of reconstruction is one to one mapping. Having obtained the individual source spectrograms for frequency basis functions $\mathbf{A}_s$ corresponding to s, the energy of the individual frame in each spectrogram $\mathbf{A}_s$ for P number of sources is calculated. Subsequently, the frequency basis functions in the original NMF matrix $\mathbf{A}$ is assigned to the source that has the highest energy at that frame, thus the grouping of frequency basis functions is done. After, the frequency basis functions $\mathbf{A}$ are grouped together corresponding to their sources, the individual complex valued spectrograms corresponding to the sources are obtained as detailed in section 2.3.2.

Finally, the individual sources are obtained using inverse STFT.

4.3.2 Spectral Masking

The second method of source reconstruction is that of spectral masking. Here, the individual source spectrograms are reconstructed using the spectral masking as detailed in 2.3.2. The estimated source spectrograms $\mathbf{A}_s$ are used to generate individual masks. These masks are then applied to the original spectrogram $\mathbf{A}$ that contains the frequency basis functions obtained using NMF. The calculation of the individual mask M_s associated with s is shown in equation:

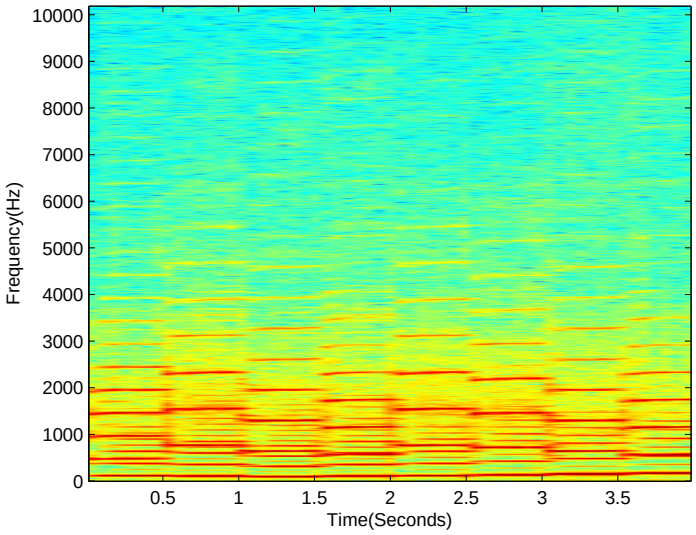

Figure 4.2: Spectrogram of a input mixture signal

$$\mathbf{M}_s = \left(\frac{\mathbf{A}_s^{\cdot 2}}{\sum_{p=1}^{P} \mathbf{A}_p^{\cdot 2}} \right) \tag{4.8}$$

Then, the mask M_s is applied on the original basis function spectrogram $\mathbf{A}$ to obtain $\hat{\mathbf{A}}_s$ that contains the frequency basis functions corresponding to the source s as shown in equation 4.9. This is done to improve the quality of the separation.

$$\hat{\mathbf{A}}_s = \mathbf{A} \cdot \left(\frac{\mathbf{A}_s^{\cdot 2}}{\sum_{p=1}^{P} \mathbf{A}_p^{\cdot 2}} \right) \tag{4.9}$$

As each row vector in $\mathbf{A}$ has a corresponding column vector in $\mathbf{B}$, clustering of the time activations is handled automatically. Then, the source magnitude

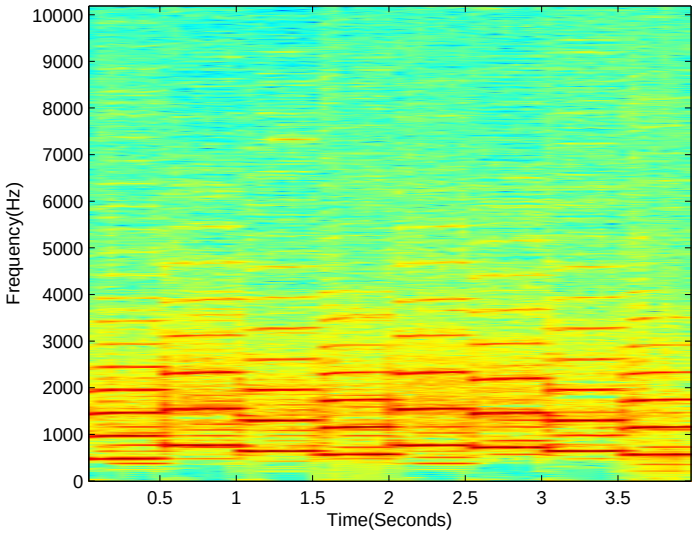

Figure 4.3: Spectrogram of the separated source 1.

spectrogram is obtained as follows:

$$\mathbf{X}_s = \hat{\mathbf{A}}_s \mathbf{B}_s \tag{4.10}$$

Thereafter, the generalised Weiner filter is used to obtained the complex-valued source spectrograms as shown in equations 2.18 and 2.19.

Then, the individual sound sources can be reconstructed using inverse STFT. In the following section, we will discuss the details about the test mixture and simulation setup used for the experiments detailed in this chapter.

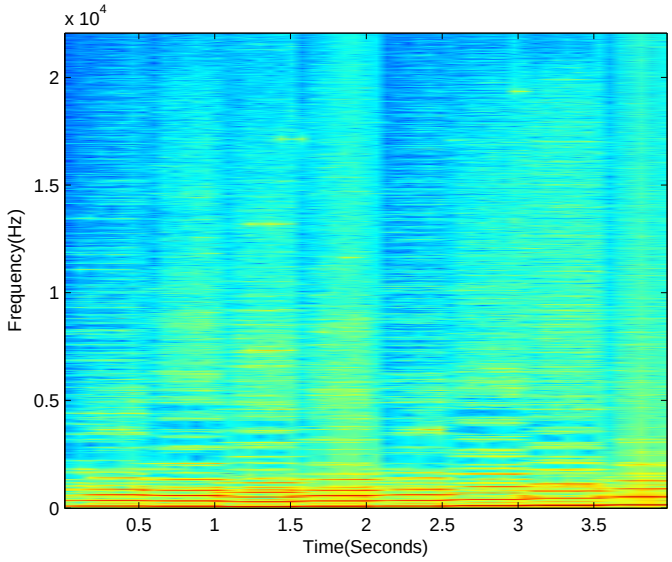

Figure 4.4: Spectrogram of the separated source 2

4.4 Results and Discussion

The simulations are done on the same experimental set up as previous chapters to compare the results. Figure 4.2 shows the magnitude spectrogram of a test signal containing music signals of two pitched instruments. Figure 4.3 and figure 4.4 show the spectrogram of reconstructed sound source of a test mixture. Through visual inspection, it can be concluded that the linear frequency domain approximation of the SNMF model can be used to separate frequency basis functions corresponding to sources in monaural mixture. The performance of the proposed SNMF algorithms is evaluated in the following section.

4.4.1 Results

We will compare the results of the proposed method i.e. the linear frequency domain approximation of the SNMF algorithm with the recently proposed SNMF clustering algorithms (SNMF_{mask} and SNMF_{map}) discussed in chapter 2. We will use SNMF_{lmap} and SNMF_{lmask} to represent the SNMF clustering algorithms proposed in this chapter, where $lmap$ and $lmask$ in the subscript indicate the use of one-to-one mapping and spectral masking respectively. It is important to note that SNMF_{mask} and SNMF_{map} use log-frequency mapped versions of the NMF basis functions as an input to the SNMF model while SNMF_{lmap} and SNMF_{lmask} use the linear domain frequency basis functions.

SNMF algorithm	SDR	SIR	SAR
SNMF_{map}	7.69	20.61	8.83
SNMF_{mask}	10.25	27.15	10.87
SNMF_{lmap}	5.75	24.86	6.88
SNMF_{lmask}	11.11	32.13	11.47

Table 4.1: Mean SDR, SIR and SAR for separated sound sources using SNMF algorithms.

A summary of the results for all the SNMF clustering algorithms are listed in table 4.1. The results are calculated by averaging the quality measures over frequency shifts k and the number of sources P, present for each mixture in the test dataset.

From table 4.1, we can see that SNMF_{lmask} outperforms the other listed SNMF algorithms. We can see that there is a significant improvement of separation quality with the use of SNMF_{lmask} over SNMF_{map}. It can be

concluded from the SIR score that the SNMF$_{lmask}$ performs considerably better than SNMF$_{mask}$ to remove interference between the sources in a given mixture. There is approximately 1 dB improvement on the SAR and SDR scores but on listening, the separated sound sources using SNMF$_{lmask}$ were audibly better than those of SNMF$_{mask}$. This highlights the fact that the quality measures do not correlate well with human perception of separation quality. Audio examples for the estimated source signals using SNMF$_{lmask}$ can be found at [92].

Furthermore, we found that the use of one-to-one mapping in SNMF$_{lmap}$ does improve the separation as compared to the previously proposed algorithms. Although we were able to reduce the interference using SNMF$_{lmap}$ as compared to SNMF$_{map}$, it resulted in increased artefacts and lower overall performance in terms of SDR. Overall, the SNMF$_{lmask}$ was found to perform best. This further proves the point that spectral masking is a better way to re-synthesise the separated sources as it adds to the improvement achieved through the separation algorithm. Hence, we can argue that, a better separation algorithm may further be improved by using spectral masking. We will discuss the effect of masking on separation performance in chapter 6.

4.5 Conclusions

A SNMF based algorithm has been proposed to group NMF frequency basis functions corresponding to their respective sources. We have implemented the clustering algorithm to use the NMF frequency basis function in the linear

frequency domain as an input to the SNMF model. This avoids the need for the use of the inverse CQT after the clustering stage for the reconstruction of the individual source signals. This is because the CQT is now incorporated in the SNMF model to obtain the clustering of the frequency basis functions. The SNMF clustering algorithm proposed in this chapter demonstrates improved performance over previous attempts at clustering basis functions for sound source separation. Having discussed various SNMF algorithms, we will introduce a group sparsity technique, motivated by the work done in [53], to improve the working of the SNMF clustering algorithms.

Chapter 5

Group Sparsity with Shifted NMF

5.1 Introduction

In this chapter, we will continue our discussion with NMF based algorithms discussed in previous chapters. Here, we will see the effect of various cost functions used to evaluate the basis functions and further evaluate its effect on SNMF algorithms. Also, we will try to incorporate the group sparsity technique in the SNMF clustering method. We argue that the incorporation of group sparsity to the NMF based methods may benefit the clustering algorithms. We will test this on various SNMF clustering algorithms to evaluate the separation quality of individual sources.

As discussed earlier in section 1.6.2, a property of NMF is that it typically

gives a sparse representation of the given audio data. This makes the frequency basis function sparse in nature. However, the NMF does not impose any quantitative constraint on the nature of sparsity. Until this point in thesis, NMF has been used to generate a compressed representation of the given audio data with no control on the sparseness. Therefore, additional constraints may be imposed to control the degree of sparseness to identify components in mixtures. Such a constraint was proposed in [53] that generates a set of NMF basis functions which benefits from sparsity at a group level. Here, we will attempt to incorporate the group sparsity technique inside the SNMF clustering algorithms. As mentioned earlier in 1.6.2, power spectrograms were used to calculate the frequency basis functions in the original implementation of GS [53]. However, many recent works in audio have used NMF of magnitude spectra instead of power spectra because it gave better sound separation quality [28, 68, 41]. Therefore, we will use magnitude spectrograms for the the calculation of the frequency basis functions.

To this end, we propose that this incorporation of GS in NMF of magnitude spectra may improve the clustering in recently proposed SNMF-based clustering algorithm discussed in chapter 2. The use of GS in NMF is motivated by the fact that the activation of the NMF results in frequency basis functions that correspond to the instruments (groups) present in the mixture. Therefore, we want the NMF basis functions to be sparse in a group sense, hence the prior knowledge of group sparsity may yield better grouping of the frequency basis functions. Here, we use the relation between KL-NMF (NMF using

KL divergence) and ML problem of estimating $\mathbf{A}$ and $\mathbf{B}$ using the Poisson distribution [29] as explained in section 5.3. We also propose that the GS constraint can further be integrated in the SNMF model for better separation of the individual sources. Here, the SNMF model refers to the SNMF stage in the SNMF clustering algorithm.

Here, we will explain the significance of the incorporation of GS at the two stages of the SNMF clustering algorithm discussed in this chapter. The first stage refers to the calculation of the frequency basis functions using the NMF and the second stage refers to the clustering stage where the SNMF model is used. In [53], it is mentioned that, in general, clustering of the basis functions using group sparsity, close to that of the ideal, can be achieved for temporal overlapping of sources up to 66%. However, in many cases there will be more overlap of the sources than this percentage. Therefore, it can be concluded that the GS in the first stage alone will not give good clustering of the basis functions hence, the use of second stage to improve clustering.

We have discussed the implementation of the second stage alone in chapter 3 i.e. the standard SNMF algorithm (SNMF_{ncqt}) where the frequency basis functions in log domain are obtained directly from the time domain signal using CQT. However, after testing, we did not get any significant improvement on the application of GS on SNMF_{ncqt} as evident from table 5.1. However, GS in NMF at first stage did appear to reduce the amount of temporal overlapping in the separated frequency basis functions. Further, with the application of GS at the clustering stage, we argue that the prior knowledge of a particular group

(source) will activate the SNMF model to force the corresponding frequency basis function to iterate towards the group it belongs and thus improve the quality of separation. This leads us to use the two stage process. Hence, GS may assist the SNMF clustering algorithms discussed in this chapter and the two stage process was necessary for improving the quality of separation [86].

As noted above, grouping of NMF basis functions is needed to segregate sound sources. To this end we propose that the prior information of these groups may be incorporated while calculating the NMF basis functions. Furthermore, we can assume that an individual sound source present in the mixture is sparse in nature, i.e, at a particular time, the contribution of the other instruments compared to the currently active one is negligible and can be ignored. This group-sparsity can further be integrated in the SNMF clustering algorithm to improve the quality of the separation.

The structure of the chapter is as follows: Section 5.2 gives the flowchart of the proposed algorithm. Section 5.3 illustrates the penalized ML estimation method for GS in KL-NMF. Section 5.4 gives a overview of the SNMF algorithm and gives the update equations for the proposed SNMF algorithm with GS. A comparison of various SNMF algorithms is done in section 5.5. Finally, the results of the proposed SNMF algorithm are compared against a previously proposed algorithm in section 2.3.

5.2 Overview of Statistical Model

Figure 5.1 shows the flowchart for the algorithm proposed. The spectrogram of the input signal is obtained by using the short-time Fourier transform (STFT). Then, the NMF basis functions are obtained from the magnitude spectrogram of a given mixture. Thereafter, the NMF basis functions are then converted into log frequency domain using CQT to exploit the shift-invariant property

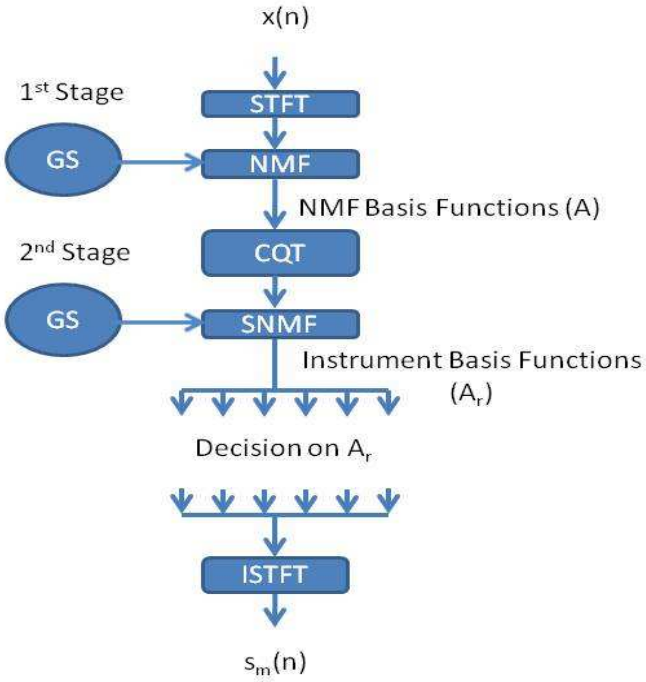

Figure 5.1: Signal flowchart of the System model

It can be seen from the figure 5.1 that we have incorporated group sparsity at

two stages of the proposed algorithm. The first of the two stages is calculating the NMF basis functions and the second stage is the activation of the SNMF clustering algorithm to determine the instrument basis functions. However, it can be noted that no knowledge of GS at first stage is used to model the SNMF clustering at second stage and vice versa. In the following section we will explain the incorporation of the NMF method using KL divergence to determine the NMF basis functions.

5.3 Group sparsity with KL-NMF

5.3.1 Equivalence between KL-NMF and ML estimation

The minimising of the KL divergence cost function in equation 1.36 to determine **A** and **B** can be derived from a probabilistic model described in [29]. This can be illustrated as follows. Given the magnitude spectrogram **X** of the input signal **x**, we assume that at every time-frequency interval, the sum of the magnitude of individual source signals $x^r_{m,n}$ is the total magnitude of the observed signal $x_{m,n}$, such that:

$$x_{m,n} = \sum_{r=1}^{R} x^r_{m,n} \tag{5.1}$$

where $x^r_{m,n}$ represents the time-frequency atom in the instrument spectrogram x^r produced by the r^{th} source. R is the number of sources in the mixture. Also, we make the hypothesis that signals in $x^r_{m,n}$ follow the Poisson distribution.

Thus, the magnitude of each $x^r_{m,n}$ can be represented as:

$$x^r_{m,n} \sim \mathcal{P}(x^r_{m,n}; A_{m,r}B_{r,n}) \tag{5.2}$$

$$\mathcal{P}(k;\lambda) = \frac{\lambda^k e^{-\lambda}}{\Gamma(k+1)!} \tag{5.3}$$

where $B_{r,n}$ is the activation gain for the basis function $A_{m,r}$. Equation 5.3 defines the Poisson distribution, $\mathcal{P}(k;\lambda)$. It can be noted that the summation of the statistically independent Poisson random variable is also a Poisson random variable. Further, as mentioned in [1] the determination of basis functions can be modelled as

$$p(\mathbf{X}|\mathbf{A},\mathbf{B}) = \mathcal{P}(\mathbf{X};\mathbf{AB}) \tag{5.4}$$

Alternatively, it can be written as:

$$p(\mathbf{X}|\mathbf{A},\mathbf{B}) = \prod_{mn} \frac{e^{-(AB)_{mn}}(AB)_{mn}{}^{X_{mn}}}{\Gamma(X_{mn}+1)!} \tag{5.5}$$

The ML solution can be given by taking the log and solving which is as follows:

$$\begin{aligned}(\mathbf{A},\mathbf{B}) &= arg\ \max_{\mathbf{A},\mathbf{B}} \log p(\mathbf{X}|\mathbf{A},\mathbf{B}) \\ &= \sum_{mn} -(AB)_{mn} + X_{mn}\log(AB)_{mn} - \log(\Gamma(X_{mn}+1)) \\ &\equiv -\mathbf{D}_{KL}(\mathbf{X}||\mathbf{AB})\end{aligned} \tag{5.6}$$

Thus, we derived a ML estimation of the basis vectors using the probability model in equation 5.6. Here, we can see that the problem definition in equation

5.6 is same as the cost function $\mathbf{D}_{KL}(\mathbf{X}||\hat{\mathbf{X}})$ defined in equation 1.36 up to a set of constant terms. Hence, we find that this objective is same as minimising the cost function $\mathbf{D}_{KL}(\mathbf{X}||\hat{\mathbf{X}})$. In the next section we will incorporate group sparsity with the ML estimation that would favour NMF using KL divergence. Following the assumption made in section 1.6.2, a source can be characterised by a subset of components g. Therefore, the source spectrogram corresponding to a group, $\mathbf{X}^g$, where

$$\mathbf{X}^g = \sum_{r \in g} \mathbf{X}^r, \tag{5.7}$$

can be estimated by the following Wiener filter estimator [53]:

$$E\left(\mathbf{X}^g|\mathbf{X}, \mathbf{A}, \mathbf{B}\right) = \mathbf{X} \cdot \left(\frac{\mathbf{A}_g\mathbf{B}_g}{\mathbf{AB}}\right). \tag{5.8}$$

5.3.2 ML with Group Sparsity

Given r basis functions, we need to group them into g groups, where each of these non-overlapping groups contains all the basis functions that correspond to a particular source. The sparsity constraint has been previously applied on both $\mathbf{A}$ and $\mathbf{B}$ or either $\mathbf{A}$ or $\mathbf{B}$ for many SSS algorithms but until the introduction of group sparsity, this was done on individual basis functions because setting the correct level of sparsity at the basis function level was problematic, as the level of sparsity varied from signal to signal [17]. In our case, we want to make a given source active for as little time as possible. Therefore, following the principle used in [53], for a given time-frequency frame n, if a source (group) is

not *on*, then the corresponding activation gain $\mathbf{B}_{gn}$ should be set to zero. Here, $\mathbf{B}_{gn}$ is a vector of basis functions r_i such that r_i is a member of a given group g ($r_i \in g$ where $1 \leq i \leq m$). Let B_n^g be defined as a time envelope of the given source for a given time frame n such as

$$B_n^g = ||\mathbf{B}_{g,n}||_1 \tag{5.9}$$

where $||.||_1$ is the L_1 norm function. Furthermore, it is assumed that the activation gains B_n^g for all the individual sources are statistically independent inverse gamma random variables. Thereafter, by using the conditional probability on the activation function $\mathbf{B}$ at frame n for r basis functions, the activation gains can be factorized into groups to determine respective sources. Hence, the marginal distribution of $\mathbf{B}_n$:

$$p(\mathbf{B}_n|B_n^g) = \prod_g \prod_{r \in g} p(B_{rn}|B_n^g) \tag{5.10}$$

The prior of the activation functions $\mathbf{B}_n$ can be calculated using the marginal distribution as follows:

$$p(\mathbf{B}_n) = \prod_g \frac{\Gamma(g+\eta)}{\Gamma(\eta)} \frac{\alpha^\eta}{(\alpha + B_n^g)^{(\eta+g)}} \tag{5.11}$$

where α is the scaling factor and the parameter η defines the shape of the gamma distribution. The ML estimation of basis functions $\mathbf{A}$ and gains $\mathbf{B}$ is done using this prior and the term defined in equation 5.6. This introduction of

the penalized term, i.e. the prior information, for the ML estimation is known as MAP (maximum a posterior) estimation. Therefore, the MAP estimation technique can be formulated as:

$$(\mathbf{A}, \mathbf{B}) = \min_{\mathbf{A},\mathbf{B} \geq 0} \mathbf{D}_{KL}(\mathbf{X}||\mathbf{AB}) + \lambda\Phi(\mathbf{B}) \tag{5.12}$$

where the 2^{nd} term $\Phi(\mathbf{B})$ is an optimisation term and is used to uniquely define the grouping pattern. A definition of $\Phi(\mathbf{B})$, as shown in equation 5.14, was proposed in [53]. The regularisation term $\lambda \in [0, 1)$ tunes the quality of factorisation obtained and can be set to zero to obtain standard KL-NMF solution.

The update equation for the activation function $\mathbf{A}$ and $\mathbf{B}$ are follows:

$$\mathbf{B} \leftarrow \mathbf{B} \cdot \left(\frac{\mathbf{A}^T(\mathbf{X} \circledast \hat{\mathbf{X}}^{-\delta})}{\mathbf{A}^T(\hat{\mathbf{X}}^{-(\delta-1)}) + \lambda\Phi'||\mathbf{B}_{gn}||_1)} \right) \tag{5.13}$$

where

$$\Phi(z) = \log(\alpha + z) \tag{5.14}$$

$$\mathbf{A} \leftarrow \mathbf{A} \cdot \left(\frac{(\mathbf{X} \circledast \hat{\mathbf{X}}^{-\delta})\mathbf{B}^T}{(\hat{\mathbf{X}}^{-(\delta-1)})\mathbf{B}^T + \lambda \sum_n B_{rn}\Phi'||\mathbf{B}_{gn}||_1} \right) \tag{5.15}$$

were δ is set to 1 for KL divergence. The operator $\cdot$ indicates elementwise matrix multiplication. The derivation of update equations can be found in [53] where δ was set to 2 for the IS divergence. All operations in equations 5.13 and 5.15 are done elementwise. Using these equations the basis functions with GS constraints can be obtained. The obtained frequency basis functions need to be clustered

to respective sources for SSS. In chapter 2, we have proposed a SNMF based clustering algorithm to segregate the frequency basis functions to their sources. We argue that further incorporating GS in SNMF would better the quality of separated sources as it would guide the basis function obtained using NMF with GS towards the sources. In section 5.5, we will show how the choice of method to calculate the NMF basis functions affects the SNMF clustering stage. Also, we mention that we are not using GS grouping at the first stage to guide the SNMF clustering at the second stage.

Next, we will discuss the implementation of GS in KL-SNMF.

5.4 Group sparsity with KL-SNMF

Having obtained the basis functions using group sparsity in KL-NMF, a knowledge of groups and their sparseness can be introduced in SNMF when clustering these basis functions. This enforcing of the basis functions towards their respective groups will further improve the clustering and hence improving the separation quality of the individual sources. This can be done in the same way as explained in section 5.3. Here, we will use the principles and techniques used in SNMF clustering algorithm to derive the update equations in KL-SNMF.

5.4.1 Shifted NMF with Group Sparsity

To incorporate shift-invariant property, the Constant Q spectrogram $\mathcal{C}$ is obtained by multiplying a transform matrix $\mathbf{Y}$ with matrix $\mathbf{A}$ in a similar

manner as done in section 2.3. Here, transform matrix $\mathbf{Y}$ acts as a warping function which translates the linear frequency representation in $\mathbf{A}$ into Constant Q domain.

$$\mathcal{C} = \mathbf{YA} \tag{5.16}$$

The spectrogram $\mathcal{C}$ is then factorised using the SNMF model to approximately determine the instrument basis functions as shown in equation 5.17.

$$\mathcal{C} \approx \langle \langle \mathcal{RD} \rangle_{\{3,1\}} \mathcal{H} \rangle_{\{2:3,1:2\}} \tag{5.17}$$

The parameter definition of the given SNMF model can be found in section 1.5.6. The cost function used to obtain tensors $\mathcal{D}$ and $\mathcal{H}$ is same as used for NMF. Therefore, the equivalence between ML estimation of tensors $\mathcal{D}$ and $\mathcal{H}$ and minimising the KL divergence between tensors $\mathcal{C}$ and $\langle \mathcal{DH} \rangle$ can be exploited. The cost function for the decomposition described in equation 1.76 can be defined as:

$$\mathbf{D}_{KL}(\mathcal{C}||\langle \mathcal{PH} \rangle_{\{2:3,1:2\}})$$

$$= \sum_{i,j} (\mathcal{C}_{ij} log \frac{\mathcal{C}_{ij}}{\langle \mathcal{PH} \rangle_{\{2:3,1:2\}}} - \mathcal{C}_{ij} + \langle \mathcal{PH} \rangle_{\{2:3,1:2\}}) \tag{5.18}$$

where

$$\mathcal{P} = \langle \mathcal{RD} \rangle_{\{3,1\}} \tag{5.19}$$

where tensor $\mathcal{P}$ contains the translated instrument basis functions. The basis functions in $\mathcal{D}$ are translated using the translation tensor $\mathcal{P}$ as shown in equation 5.19.

5.4.2 Update equations for $\mathcal{H}$ and $\mathcal{D}$ with Group Sparsity

Assuming that the number of groups is equal to the number of instruments, we can get the required clustering of frequency basis functions. The GS in SNMF can be incorporated by applying the group sparsity constraint on $\mathcal{H}$ and determining the priors using the gamma distribution as done in equation 5.10 and 5.11. For a given time-frequency frame, let the activation gain $\mathcal{H}_{g,k}$ in SNMF model be the summation of all the components defined by $\mathcal{H}(k,:,:)$ for a particular g. This can be expressed as:

$$\mathcal{H}_g^k = \sum_k \mathcal{H}(k, g, :) \tag{5.20}$$

where k is the number of frequency shifts. Further, with the knowledge of priors of the activation function $\mathcal{H}$, the SNMF problem can be reduced to the ML estimation of the tensors $\mathcal{D}$ and $\mathcal{H}$. The penalised ML solution for the KL-SNMF problem can be defined as:

$$\langle \mathcal{P}, \mathcal{H} \rangle = \min_{\mathcal{P},\mathcal{H} \geq 0} \mathbf{D}_{KL}(\mathcal{C} || \langle \mathcal{P}\mathcal{H} \rangle_{\{2:3,1:2\}}) + \lambda \Phi(\mathcal{H}) \tag{5.21}$$

The optimisation term $\Phi(\mathcal{H})$ is again used to define the group sparsity constraint. The interactive multiplicative update equations for $\mathcal{P}$ and $\mathcal{H}$ can be

derived in a manner similar to [44]. This can be formulated as follows:

$$\mathcal{H} \leftarrow \mathcal{H} \cdot \left(\frac{\langle \langle \mathcal{R}\mathcal{D} \rangle_{\{3,1\}} \mathcal{Q} \rangle_{\{3,1\}}}{\langle \langle \mathcal{R}\mathcal{D} \rangle_{\{3,1\}} \mathcal{O} \rangle_{\{1,1\}} + \lambda \Phi'(\mathcal{H}_g^k)} \right) \tag{5.22}$$

where

$$\mathcal{Q} = \frac{\mathcal{C}}{\langle \mathcal{P}\mathcal{H} \rangle_{\{2:3,1:2\}}} \tag{5.23}$$

and $\mathcal{O}$ is a tensor of all ones. The multiplicative updates for the translated basis functions in $\mathcal{D}$ can be found by using following equations:

$$\mathcal{W} = \langle \mathcal{R}\mathcal{O} \rangle_{\{1,1\}} \tag{5.24}$$

$$\mathcal{D} \leftarrow \mathcal{D} \cdot \left(\frac{\langle \mathcal{Z}\mathcal{H} \rangle_{\{1:3,1:3\}}}{\langle \mathcal{W}\mathcal{H} \rangle_{\{1:3,1:3\}} + \lambda \sum_n \mathcal{H}_{r,n} \Phi'(\mathcal{H}_g^k)} \right) \tag{5.25}$$

where

$$\mathcal{Z} = \langle \mathcal{R}\mathcal{Q} \rangle_{\{1,1\}}$$

where function $\Phi(z)$ is same as stated in equation 5.14. The multiplicative updates and the positive random initialization for $\mathcal{D}$ and $\mathcal{H}$ ensures the positive tensor factorisation. The number of translations k in $\mathcal{R}$ is chosen such that the translated (frequency-shifted) instrument basis functions cover all the notes or chords corresponding to basis functions in the mixture.

The performance evaluation of the various SNMF clustering algorithms discussed in this chapter is based on two factors. The first one is the use of cost functions i.e. KL divergence and IS divergence. The second one is the use of GS

at the two stages discussed through the course of this chapter. We will define some notations to denote the two stages and the SNMF clustering algorithms. The prefix g has been used in subscript of SNMF or NMF to indicate the use of GS in the given SNMF clustering algorithm. In other words, gkl represents KL divergence with GS and kl refers to KL divergence without GS. For example, $\text{SNMF}_{gis-gis}$ represents two stages of SNMF clustering algorithm with group sparsity at both the stages with IS divergence. - in the subscript divides the two stages where the left side refers to the first stage and the right side represents the second stage. However, NMF_{kl} denotes standard NMF method with KL divergence for calculating frequency basis functions and SNMF_{gkl} represents the 2^{nd} stage of the SNMF clustering algorithm with KL divergence incorporated with group sparsity. We will use these notations for rest of the chapter.

Having obtained the source spectrograms in $\mathbf{C}$, signal reconstruction can be carried out in the similar manner as done in section 2.3.2.

Figure 5.2 shows the log-frequency spectra of the NMF_{gkl} basis functions of a test mixture of two sources. The x-axis shows the frequency basis functions for all the notes played by the instruments present in mixture. The application of the SNMF clustering algorithm separates the basis functions into two groups corresponding to the individual sources.

The separated basis functions of source 1 and source 2 respectively can be seen in figures 5.3 and 5.5. Here, the SNMF_{kl} was used for the clustering stage. Figures 5.4 and show the separated frequency basis spectrograms using SNMF_{gkl}. The separated basis functions are more visible for respective sources

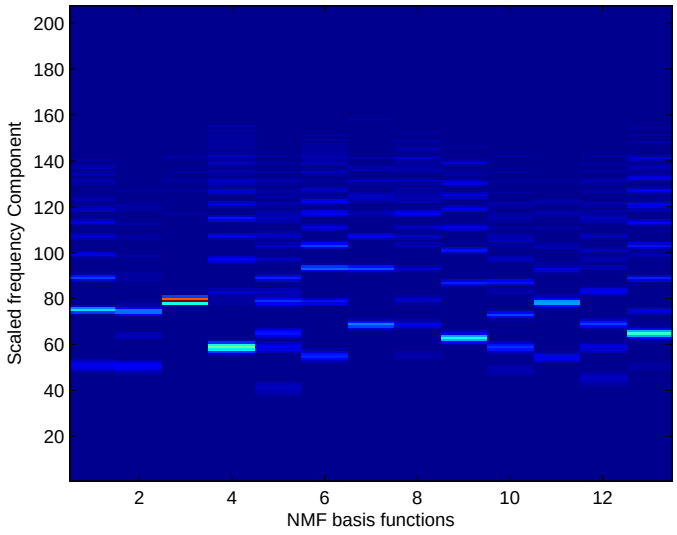

Figure 5.2: NMF_{kl} basis function of input mixture in constant Q domain.

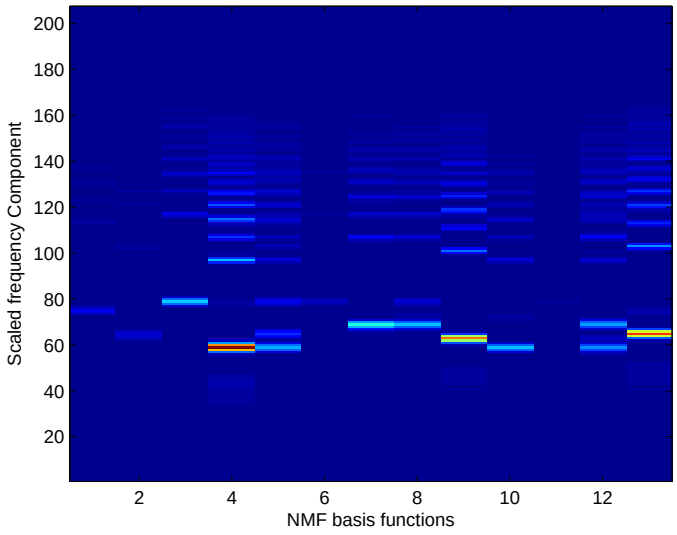

Figure 5.3: Recovered NMF_{kl} basis functions using SNMF_{kl} for source 1.

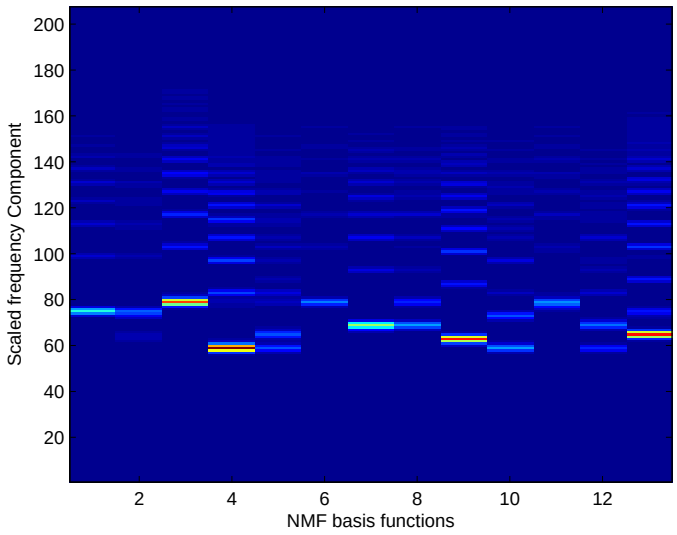

Figure 5.4: Recovered NMF_{kl} basis functions using SNMF_{gkl} for source 1
.

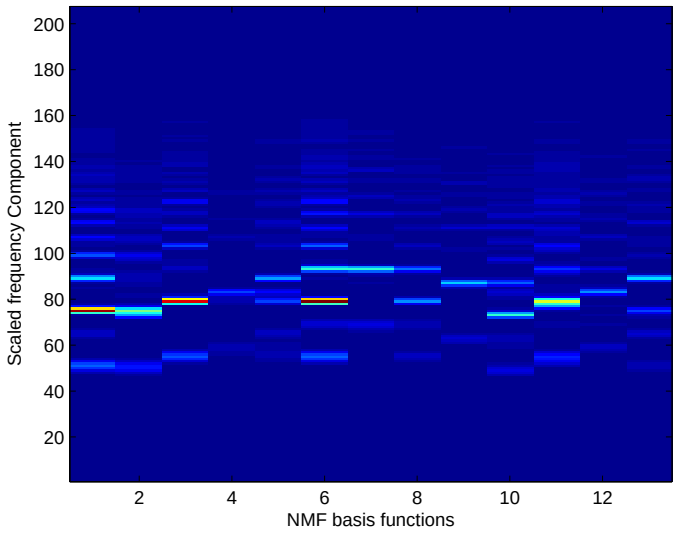

Figure 5.5: Recovered NMF_{kl} basis functions using SNMF_{kl} for source 2.

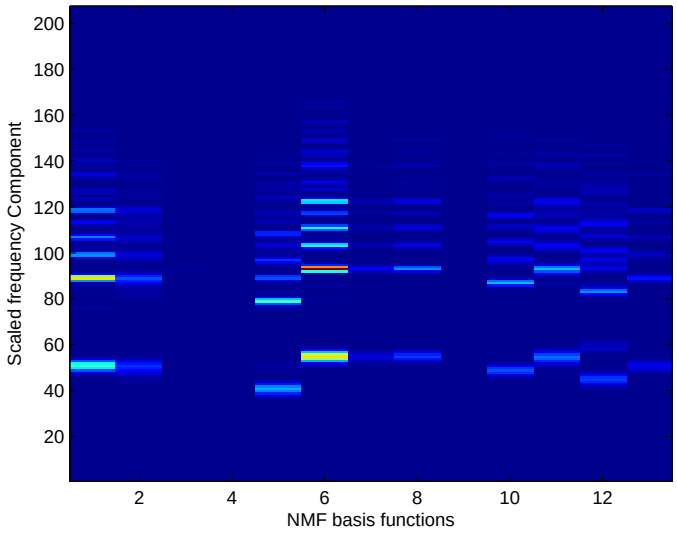

Figure 5.6: Recovered NMF_{kl} basis functions using SNMF_{gkl} for source 2

.

for SNMF_{gkl} as compared against SNMF_{kl}. Thus, by inspecting above figures, we can show that SNMF_{gkl} works better than SNMF_{kl} to obtain distinct groupings of basis functions and can further be used to separate sources in the mixture. We will further prove this point in the result section. Also, we can conclude that SNMF with GS constraint can be used to cluster basis functions in monaural mixtures. We

5.5 Experiments

A number of different tests were conducted to efficiently determine the frequency basis functions using NMF and to determine the effect of the number of different translations, k, in frequency on various SNMF algorithms. We were hoping that

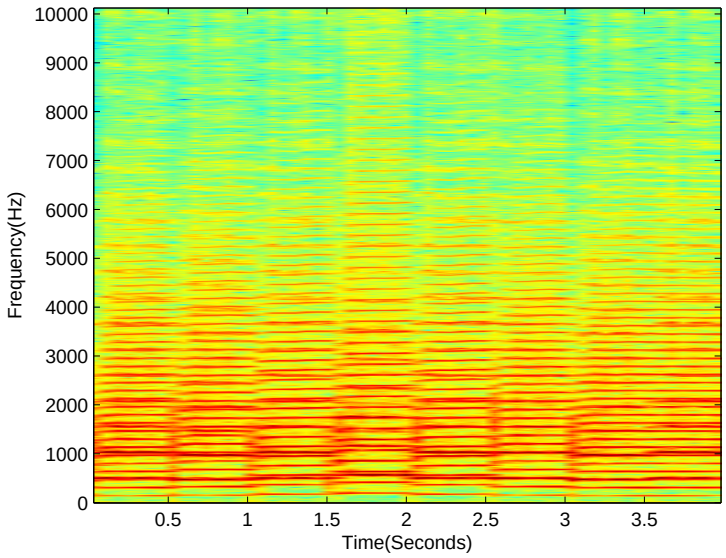

Figure 5.7: Mixture spectrogram of the two sources

the use of various frequency shifts would give some insights to the clustering obtained and would give a clear comparison of the various SNMF methods. The number of frequency shifts ranged from 5 to 12. Again, we use the same set of test signals as in all previous chapters. The number of groups in GS was limited to 2 for the given tests.

To demonstrate the improvement by using group sparsity, a comparison is done between the SNMF$_{kl-kl}$ clustering algorithm and the SNMF$_{kl-gkl}$ clustering method using spectrograms of the mixture signal and the estimated sources. Figure 5.7 shows the mixture spectrogram of a mono signal which consists of two sources. Figures 5.8 and 5.10 show the spectrograms of the separated sources using SNMF$_{kl-kl}$ clustering algorithm. The spectrograms of the estimated sources obtained using SNMF$_{kl-gkl}$ clustering algorithm is shown in figures 5.9

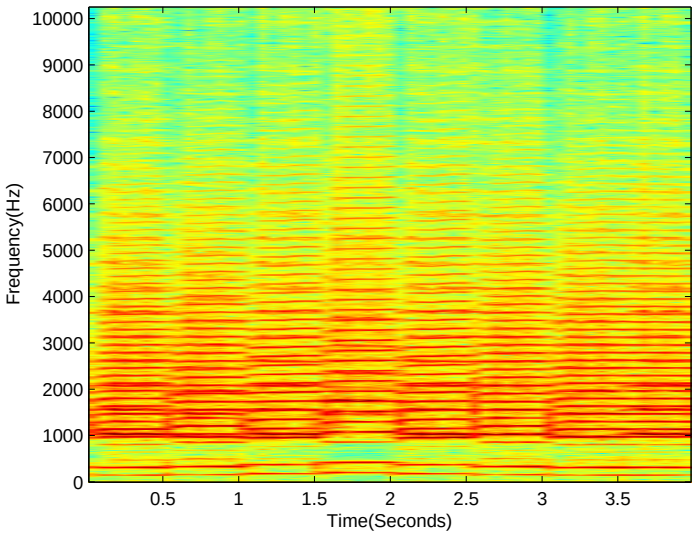

Figure 5.8: Separated source 1 using SNMF$_{kl-kl}$ clustering algorithm.

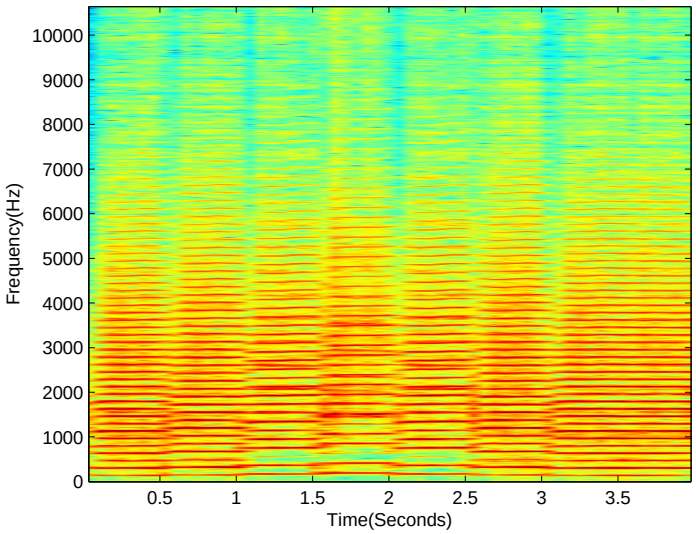

Figure 5.9: Separated source 1 using SNMF$_{kl-gkl}$ clustering algorithm.

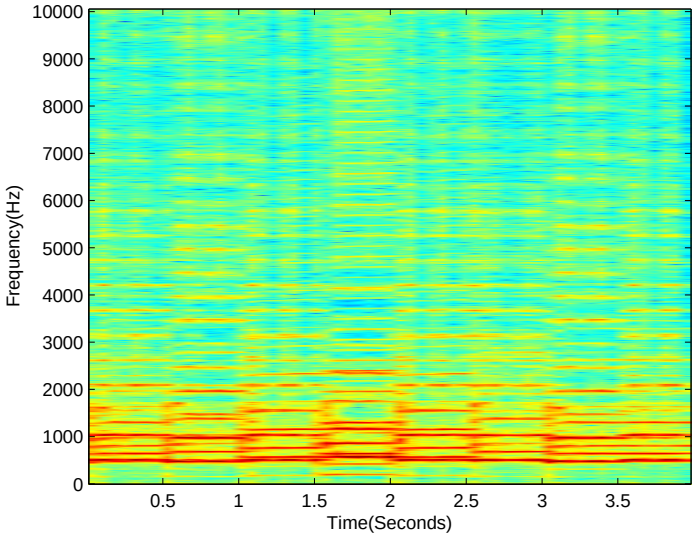

Figure 5.10: Separated source 2 using SNMF$_{kl-kl}$ clustering algorithm.

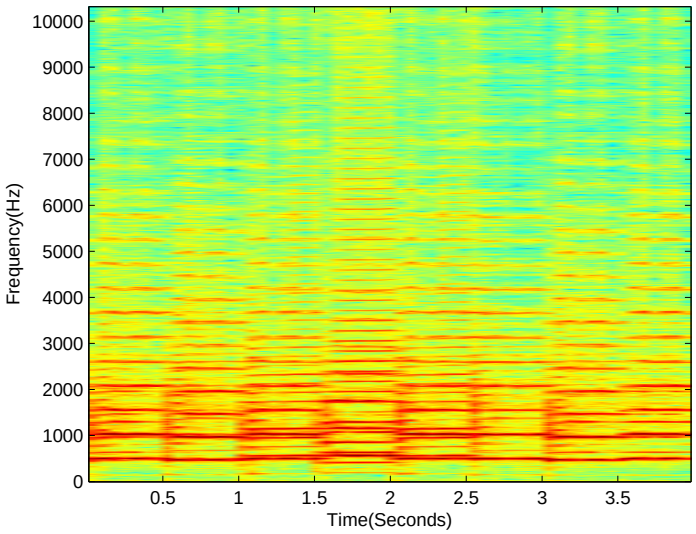

Figure 5.11: Separated source 2 using SNMF$_{kl-gkl}$ clustering algorithm.

and 5.11 respectively. From figure 5.10 it can be said that by using SNMF clustering algorithm without GS completely rejects the upper harmonics of the source 2 while SNMF_{kl-gkl} recovers the upper harmonics to give better separation. Hence, the incorporation of group sparsity helps in improving the separation of the sources and after listening to the separated sources it was found that the notes and the melodies played by the sources in mixture have separated better than those of SNMF_{kl-kl} clustering algorithm.

A summary of the results for all the SNMF algorithms are shown in figure 5.12. The scores for all the quality measures were calculated and graphed against the allowable frequency shifts k. The results were determined by finding the average of the quality measures obtained for each separated source for each input mixture. Each set of quality measure, say SDR in figures (a), (d) and (g), illustrates the comparison of all the listed SNMF algorithms for NMF_{kl}, NMF_{gkl} and NMF_{gis} basis functions respectively. Although, the GS constraint in SNMF_{gis} helps in enhance the clustering of NMF_{gkl} basis functions as compared against SNMF_{is} but it fails to improve the grouping of for NMF_{gis} basis functions than that of SNMF_{is}. Also, it can be concluded from the figure 1 (a) and (c) that the clustering results obtained using SNMF_{kl-gis} are not good as compared to the other proposed algorithms. On informal listening, we observed that the SNMF clustering algorithm using KL divergence were found to give typically good separation of the individual sources as compared to that of the IS divergence. Also, through visual inspection it can be concluded that SNMF algorithms with KL divergence (SNMF_{kl} and SNMF_{gkl}) completely outperforms

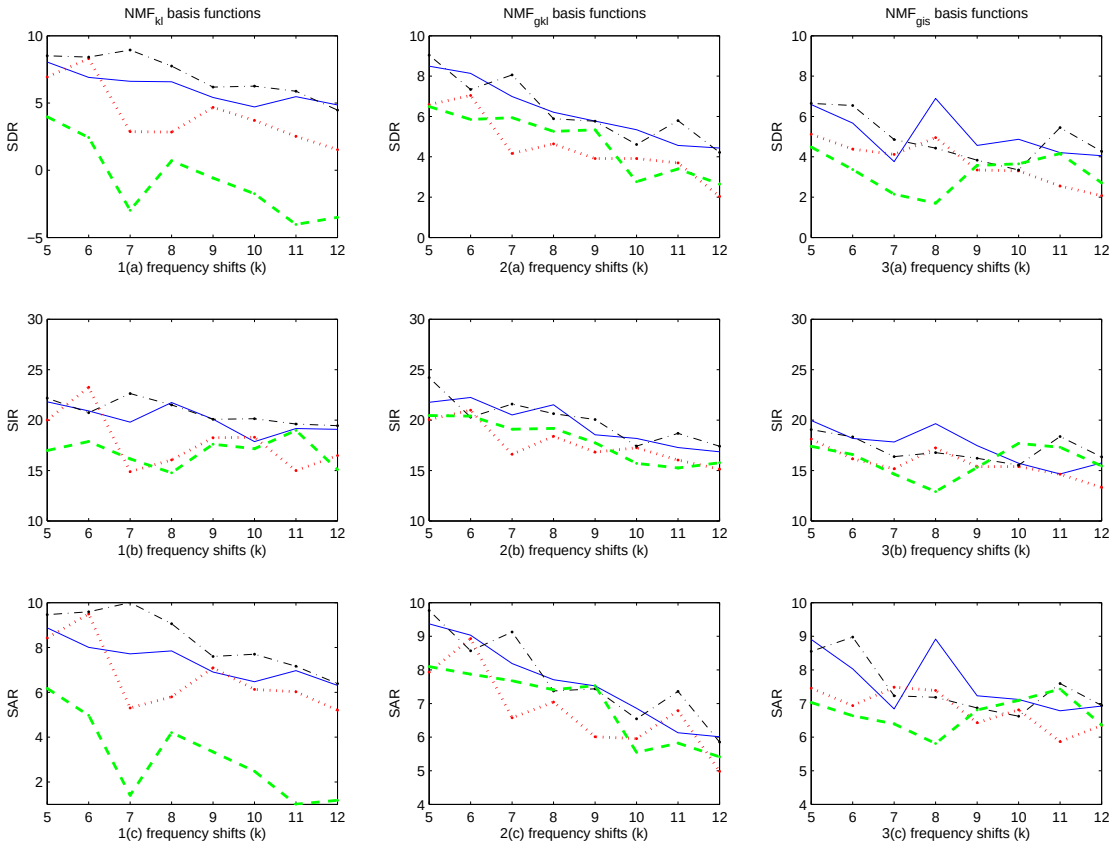

Figure 5.12: Performance evaluation of SNMF$_{kl}$(blue solid line), SNMF$_{is}$(red dotted line), SNMF$_{gkl}$(black dash-dot line) and SNMF$_{gis}$(green dashed line) to group basis functions generated by NMF$_{kl}$(1st column), NMF$_{gkl}$(2nd column) and NMF$_{gis}$(3rd column) for different number of frequency shifts

SNMF model with IS divergence (SNMF_{is} and SNMF_{gis}). As a result, we will elaborate more on SNMF algorithms based on KL divergence in section 5.6.

5.6 Results

In this section, we will compare the result of the proposed SNMF clustering algorithm with GS constraint against the SNMF clustering algorithm (SNMF_{mask}) discussed in chapter 2. It is important to note that the SNMF_{mask} algorithm is same as SNMF_{kl-kl} as denoted in this chapter. As discussed in section 5.5 the SNMF clustering algorithms with KL divergence work better for clustering the basis functions when compared against the SNMF clustering algorithms with IS divergence. Therefore, we will focus on SNMF clustering algorithms with KL divergence. It can be concluded from figure 5.12 that for NMF_{kl} basis functions, SNMF_{gkl} improves the grouping of basis functions as compared to SNMF_{kl}. Also, SNMF_{gkl} is marginally better than SNMF_{kl} to group the NMF_{gkl} basis functions. However, both the SNMF clustering algorithms, $\text{SNMF}_{gkl-gkl}$ and SNMF_{gkl-kl} scores lower as the number of frequency shifts increases. This is due to the over estimation of active notes within the mixture because with the increase in frequency shifts the number of estimated notes present in the mixture increases. It can potentially split one 'original note' into two or more 'notes', thus deteriorating the timbre of the original note present in the mixture and hence this adversely affects the separation quality. Audio examples of the estimated source signals can be found

at [92].

SNMF algorithm	SDR	SIR	SAR
SNMF_{ncqt}	10.88	25.44	11.47
SNMF_{gncqt}	10.75	25.19	11.39

Table 5.1: Mean SDR, SIR and SAR for separated sound sources using the standard SNMF algorithm

SNMF algorithm	SDR	SIR	SAR
SNMF_{kl-kl} (SNMF_{mask})	10.25	27.15	10.87
SNMF_{kl-gkl}	11.79	27.09	12.38
SNMF_{gkl-kl}	10.83	26.04	11.43
$\text{SNMF}_{gkl-gkl}$	10.98	25.81	11.64

Table 5.2: Mean SDR, SIR and SAR for separated sound sources using SNMF algorithm

To compare the results listed in [2], the highest scores of the quality measures for the separated sound sources for each mixture were hand-picked for the given range of frequency shifts such that

$$SDR = \max_{k} SDR_k,\ k \in K \tag{5.26}$$

where K is the number of frequency shifts. The results were then calculated by averaging the metrics (SDR, SIR and SAR) over each of the separated sources for all the test mixtures. Thereafter, the mean SDR, SIR and SAR were obtained by finding the average over each of the input mixture.

As mentioned earlier in section 5.1 that we did not get much improvement in the separation performance by incorporating GS in the standard SNMF

algorithm, SNMF$_{ncqt}$. Let SNMF$_{gncqt}$ denote the algorithm in which GS is incorporated in SNMF$_{ncqt}$. A comparison of the quality measures between SNMF$_{ncqt}$ and SNMF$_{gncqt}$ is done in table 5.1. It can be concluded from the table 5.1 that, in context of the standard SNMF algorithm, the incorporation of the group sparsity does not give much improvement of the separation quality of the individual sources.

Table 5.2 give the quality measures for the SNMF clustering algorithms. It can be seen from the table 5.2 that each of the SNMF clustering algorithm with group sparsity performs better than SNMF$_{kl-kl}$. We can also see that SNMF$_{gkl}$ performs better clustering for basis functions generated by NMF$_{kl}$ and is marginally better for NMF$_{gkl}$. Hence, the GS in SNMF improves clustering for NMF basis functions. In general, for the SNMF clustering algorithms with GS, the separated sound sources contained melodies corresponding a source with less interference of the notes corresponding to the other source, as compared to that of the SNMF clustering algorithm without GS, SNMF$_{kl-kl}$. This observation was made from the informal listening of the separated sources.

5.7 Conclusions

We have presented a Shifted NMF based clustering technique to cluster the frequency basis functions. We have incorporated group sparsity at two stages of the SNMF clustering algorithm. We have explained how the incorporation of group sparsity at the first stage can potentially improve the clustering

of frequency basis functions by reducing the overlapping of basis functions. Subsequently, at the second stage, the group sparsity would guide the basis functions to their respected groups corresponding to instruments in the given mixture. A probabilistic model is used to exploit the equivalence between the ML problem and minimising the KL divergence cost function, using the Poisson distribution, to estimate the frequency basis functions. Group sparsity was incorporated in the activation gain functions $\mathbf{B}$ and $\mathcal{H}$ respectively for the first and second stages of the SNMF clustering algorithm. An optimisation term was used to tune the grouping criteria. Results show that incorporating GS improves the clustering of frequency basis function in the SNMF model, thus improving the separation quality. In the next chapter we will discuss a family of masks that may be used on the separation algorithms to improve the reconstruction of the individual source signals.

Chapter 6

Masking filters for Reconstruction of Signals

6.1 Introduction

Through the course of this thesis, we have discussed the implementation of various NMF based sound source separation (SSS) techniques that make use of the magnitude or the power spectrograms and disregard the phase information of the given audio signal. In general, the SSS algorithms filter out the phase information from the audio signals [1, 33] and reduce the algorithm to a subset of an image signal processing problem. This helps in reducing the complexity in the analysis of the signal in order to separate meaningful identities to reconstruct the magnitude spectrograms of the desired signals. However, a notable shortcoming with such methods is that there is no phase information available to recover the

time-domain signals from the separated source spectrograms. Many attempts have been made to overcome this problem. Griffin and Lim proposed a phase estimation technique to recover the phase of the source spectrograms [35]. Le Roux et al [36] have used explicit consistency constraints on the STFT spectrograms for the phase reconstruction. An alternative approach to resynthesize the recovered signals was to simply reuse the phase of the given original mixture.

In recent years, the most commonly used method has been to create generalised Wiener filters using the estimated source spectrograms. Then, these Wiener filters can be used as soft masks to the original complex valued spectrogram to obtain the complex-valued individual source spectrograms. The generalised Wiener filter in the context of monaural separation was first proposed by Benaroya et al [34]. Recently, Le Roux et al [37] have utilised a spectrogram consistency constraint to obtain better performing masks for phase estimation of the recovered spectrograms. It can be noted that the creation of soft masks is the same as spectral masking as discussed in section 2.3.2. The method can be formulated as follows:

$$X_s = X \cdot \mathbf{M}_s \tag{6.1}$$

where for generalised Wiener filter, the soft masks $\mathbf{M}_s$ is defined as

$$\mathbf{M}_s = \left(\frac{\mathbf{X}_s^{\cdot r}}{\sum_{p=1}^{P} \mathbf{X}_p^{\cdot r}} \right) \tag{6.2}$$

Here X_s is the estimated complex spectrogram of the s^{th} source, the block letter X represents the original complex mixture spectrogram, $\mathbf{X}_s$ is the estimated magnitude spectrogram of the s^{th} source and the number of sources in the mixture is P. Here, p is used to index the sources in P, The exponent r is 1 for power spectrograms and is set to 2 for magnitude spectrograms. The operator $\cdot$ represents elementwise operation and all the divisions in all the equations in the chapter is done elementwise.

An advantage of using the Wiener filter approach is that the separated sources sum together to give the original mixture signal. Thus, we will not lose any part of the signal due to re-synthesis. This property is of particular benefit for one of the application of separating sound sources from a mono mixture i.e. remixing from mono to stereo. This is because the total sum of the separated sources is equal to the original mixture, the interference due to the other sources and errors in separation will often be masked and will be less prominent in the upmix stereo space [40].

In effect, the Wiener filter method allocates energy in a given time-frequency bin to the sources according to a least-square best fit. Thus, the masks obtained are optimal in the lease square sense [54]. However, the masks generated does not give any quantitative measure to justify that they are equally good in the perceptual sense. Hence, it can be argued that from a perceptual point of view, other masks may be more optimal for re-synthesis. Also, no work has been done on how the performance of the masks vary with the number of iterations performed by the separation algorithms. Instead, the masking is carried out at

the end of the separation algorithm to recover the spectrograms. Having said that, it is proposed to investigate the above mentioned issues in the remainder of the chapter.

It is important to note that my contribution to this chapter is limited to testing and evaluating the performance of the proposed family of masks using various separation algorithms. I was also involved in the discussion of the results obtained using PEASS toolbox [38]. However, the original idea and the derivation of the divergence based masks is done by Derry Fitzgerald [39].

6.2 Divergence-Based Masks

As mentioned previously in section 6.1, the generalised Wiener filtering approach is optimal in a least-square sense. However, in case of sound source separation algorithms, especially NMF based methods, the cost function defined by the least- square approximation has typically not performed better than other cost functions. The two most widely used cost functions for audio applications are the KL divergence and the IS divergence. The definition of the KL-based and IS-based cost functions is detailed in section 1.5.4.

To this end, we propose to develop masks based on these divergences and see if they improve the separation quality of the individual sources as compared to the generalised Wiener filter masks. Hence we define a family of divergence based masks:

$$\mathbf{M}_s = 1 - \frac{D(\mathbf{X}_s, \mathbf{X})^t}{\sum_{p=1}^{P} D(\mathbf{X}_p, \mathbf{X})^t} \tag{6.3}$$

where the mask associated with the s^{th} source is represented by M_s, X denotes the estimated mixture spectrogram. The letter D symbolises any suitable divergence metric and the parameter t is used to vary the properties of the mask. More details on the derivation of equation 6.3 can be found in [39]. It is important to note that both the KL and IS divergence used to create these masks approach zero when the corresponding data points are similar. Therefore the variable term in equation 6.3 defines a mask that removes the source from the given mixture. Hence, subtracting this value from 1 generates the mask to separate the source in consideration. Thereafter, for all the family of masks generated, the complex valued source spectrogram X_s can be obtained using equation 6.2.

Here, the allocation of the energy in a given time-frequency bin is based on best fit according to the chosen divergence metric. It is also worth noting that the sources separated using these masks will sum together to reconstruct the original mixture to a constant term $P - 1$, where P indicates the number of sources present in the mixture. However, the result will not vary as it is invariant to the amplitude changes. Hence, the resynthesis using the proposed family of masks is also suitable for remixing or upmixing.

6.3 Masking Testsets

Here, we use three previously proposed NMF based algorithms with their respective testsets to test the performance. This is done to ensure that the applicability of the proposed family of masks is not limited to a given algorithm.

The first algorithm used is the Source-Filter based Sinusoidal Shifted Non-negative Tensor Factorisation (SSNTF) algorithm. The algorithm is based on the assumption that a given instrument can be uniquely modelled as a frequency invariant set of harmonic weights along with a corresponding formant filter. This set up allows the timbre of the instrument to change with pitch. The details of the algorithm can be found in [20].

The second algorithm is the NMF based user-assisted source separation algorithm (UA)as detailed in [18]. In this algorithm, the user sings along with the given song and records the source to be separated. The recording is then factorised to obtained frequency basis functions using NMF. The resultant frequency basis functions are then used as priors to influence the factorisation of the mixture signal to recover the desired source. Here, the source can be any instrument or vocal present in the original mixture. The priors are used to guide the factorisation for the first 20 iterations, with the influence of the priors reduced with each subsequent iteration until the updates reduce to those of standard NMF after 20 iterations. The test used here was created from a set of recordings by the Beach Boys. Here, the vocals and the backing tracks were available separately. Further, these recordings were used to create mono mixtures by manually synchronising and mixing the tracks. The details of the

testset can be found in [19].

The third algorithm is the SNMF clustering algorithm (SNMF$_{mask}$) discussed in chapter 2. The testset used for the first and the third algorithm are same as detailed in section 2.4. Here SNMF was used to identify the clustering of the frequency basis functions obtained by NMF. Then, NMF was run again on the spectrogram using the same initialisation as the original NMF stage, so that the NMF will converge to the same point as the initial NMF. This was done to ensure that the clustering obtained using SNMF still applies. Further, the use of standard NMF without any constraints helps in analysing the effects of masking on the standard NMF algorithm.

The use of three algorithms to evaluate the performance ensures the following two points. Firstly, all the NMF based source separation algorithm are using different methods and constraints and secondly, the two testsets used here are very different from each other. Therefore, in context of using masks, the results obtained should generalise well.

6.4 Experiments and Results

We have used the values of t equal to 1 and 2 (see equation 6.3) to evaluate the performance of the divergence based masks for both KL and IS divergences. The three algorithms were run for 100 iterations with KL divergence as a cost function. All the audio test signals in the testset were mono mixtures with a sample rate of $44.1kHz$. The separation performance of the proposed family of

masks and the Wiener filter mask were calculated after every 10 iterations.

The evaluation of the working of the masks was done using the PEASS toolbox, which attempts to measure the perceptual quality of the audio source separation by calculating a set of objective measures [38]. The metrics used were the target-related perceptual score (TPS), the artefacts-related perceptual score (APS), the interference-related perceptual score (IPS), and the overall perceptual score (OPS).

TPS determines how well the separated source matches the spatial positioning of the original source. APS measures the amount of artefacts perceived in the estimated source. IPS calculates the perceived interference of the other sources in the separated source and finally OPS measures the perceived overall quality of the separated sources.

Figure 6.1 shows the obtained mean OPS values for the source separation achieved using the SSNTF algorithm for the corresponding testset. Through visual inspection, it can be seen that the proposed family of masks outperform the generalised Wiener filter. Also, the KL mask with t equal to 2 has the highest perceptual value, thus performing the best. It is very interesting but surprising to note that the peak is achieved after 10 iterations and then the performance decreases for all the masks. We will comment on this in the later part of this section.

Figure 6.2 shows the average OPS values for UA and its associated testset. It can be seen that again the proposed masks outperforms the generalised Wiener filter and KL mask with $t = 1$ performs best among all the masks and KL

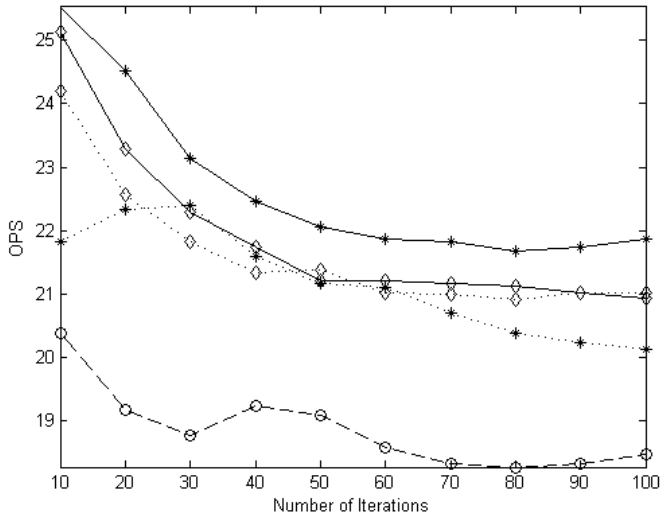

Figure 6.1: Overall Perceptual Scores for the SSNFT algorithm. A line with diamonds indicates the performance of the use of the IS divergence mask, the circle-dashed line denotes the perceptual score obtained due to the generalised Wiener filter mask and stars indicates the use of a KL divergence mask. The use of solid line is for $t = 2$ and a dotted line indicates the use of $t = 1$ for the corresponding mask. The same legends is used for all subsequent figures in this chapter

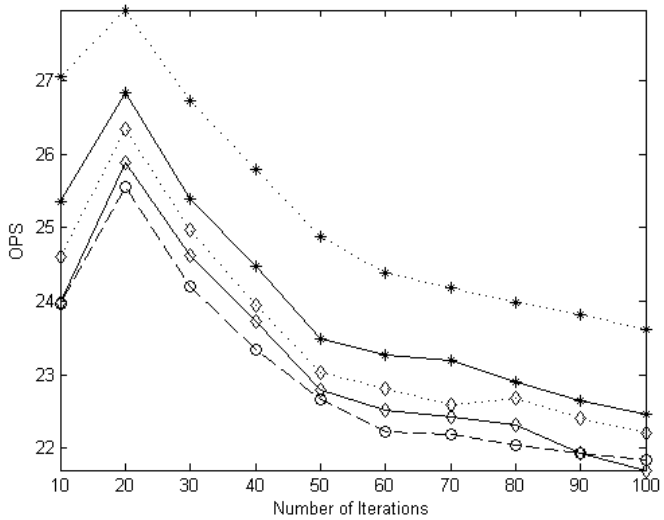

Figure 6.2: Overall Perceptual Scores for the UA algorithm. Legend as per figure 6.1.

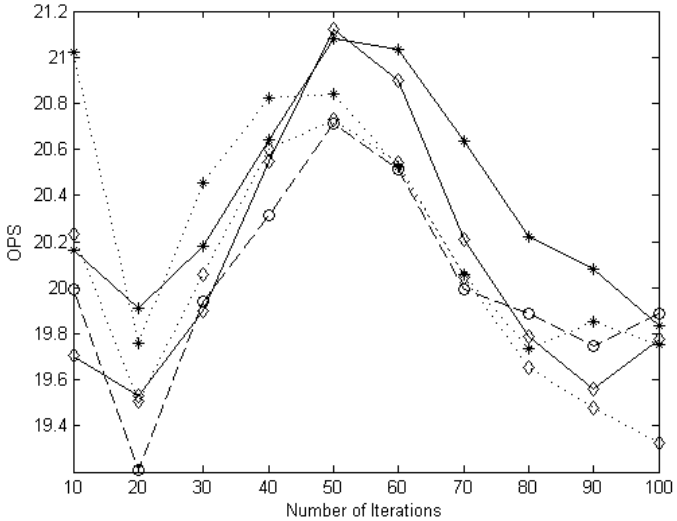

Figure 6.3: Overall Perceptual Scores for the standard NMF algorithm. Legend as per figure 6.1.

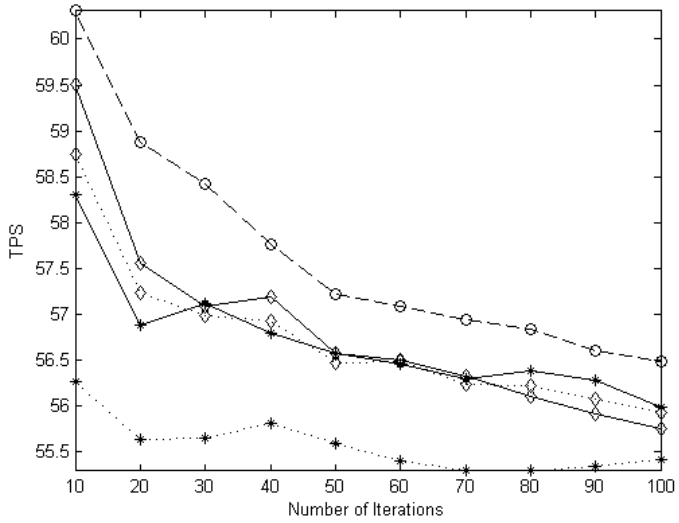

Figure 6.4: Target-related Perceptual Scores for the SSNTF algorithm. Legend as per figure 6.1.

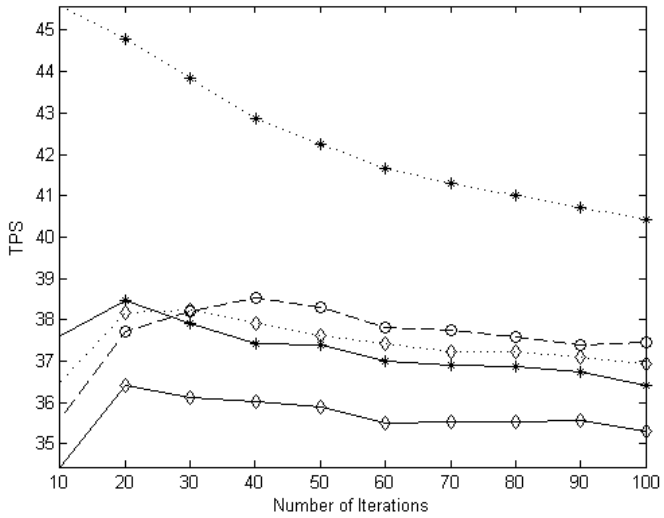

Figure 6.5: Target-related Perceptual Scores for the UA algorithm. Legend as per figure 6.1.

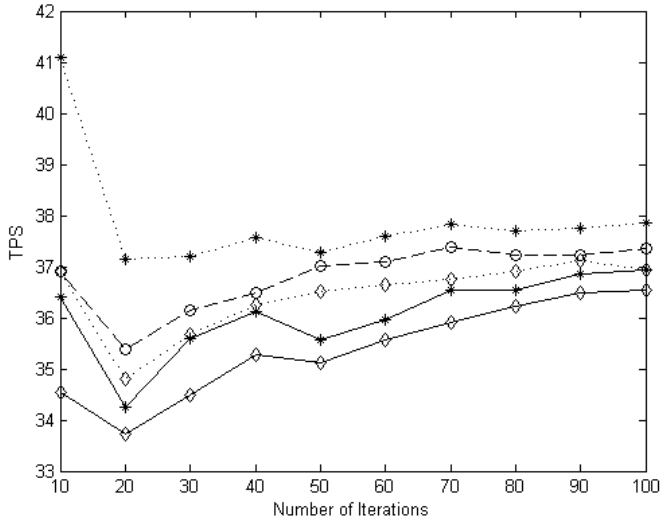

Figure 6.6: Target-related Perceptual Scores for the standard NMF algorithm. Legend as per figure 6.1.

mask with $t = 2$ is the second best in performance. Another interesting point to note is that around 20 iterations, all the masks give best performance. As mentioned earlier, it is at this point (20 iterations), the guidance of the priors is removed from the update equations. This suggests that after this point the NMF frequency basis functions begin to capture other parts of the mixture signal along with the signals associated with the individual sources in question, hence, the decline in performance of the masks.

Figure 6.3 shows the results obtained for OPS for standard NMF on its testset. Again, like previous algorithms the peak is achieved at around 50 iterations, long before the convergence is achieved. Again, the generalised Wiener filter is outperformed by the proposed divergence masks. However, the improvement is smaller as compared to previous algorithms. Here, the IS masks

with $p = 2$ performs the best with KL mask with $t = 2$ is the second best. Therefore, in context of OPS, it can be said that the KL divergence mask with $t = 2$ performs consistently well for all the algorithms.

Figure 6.4, 6.5 and 6.6 show the TPS for the SSNTF, the UA and the NMF algorithms respectively. It was found that similar trends were followed for TPS as compared to OPS for the SSNTF and the UA algorithms i.e. the proposed masks outperform the generalised Wiener filter. However, the best performing mask was for individual algorithms were different. In case of the standard NMF algorithm TPS values increase gradually with the increase in iterations. The KL mask with $t = 1$ performs best but the KL mask with $t = 2$ falls somewhere in the centre in terms of performance.

Figure 6.7, 6.8 and 6.9 show the results calculated for IPS for the SSNTF, the UA and the standard NMF algorithms respectively. Again, SSNTF shows the highest peak in performance at the lowest iterations and follows a downward trend with the increasing number of iterations and peaks can be seen for UA at 20 iterations. IPS values again reach a peak at 50 iterations and further varies with iteration number. It can be seen that IS mask with $t = 2$ performs the best and again KL mask with $t = 2$ is amongst the two masks. This suggests that to minimise the interference of the other sources in the separated source, the IS mask with $t = 2$ is optimal for resynthesis.

Finally, figures 6.10, 6.11 and 6.12 show the values obtained of APS for all the three algorithms respectively. Here, the KL mask with $t = 1$ performs the best and the Wiener filter is the second best. It is important to note that both

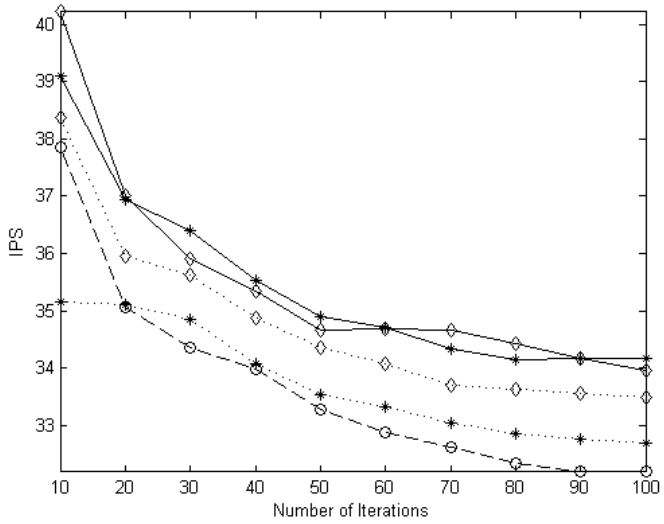

Figure 6.7: Interference-related Perceptual Scores for the SSNFT algorithm. Legend as per figure 6.1.

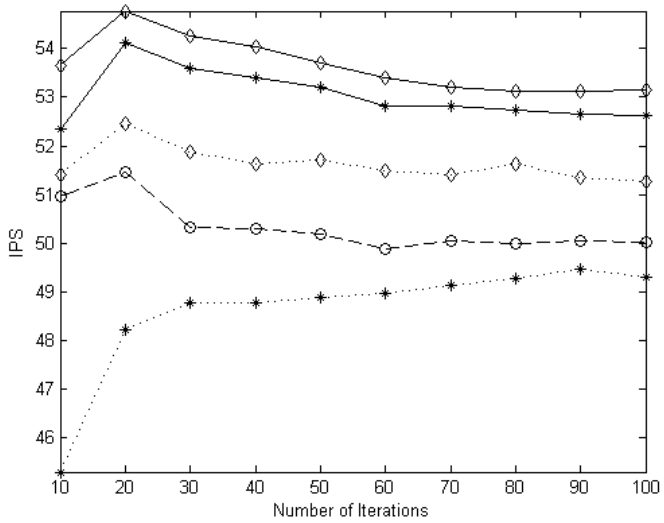

Figure 6.8: Interference-related Perceptual Scores for the UA algorithm. Legend as per figure 6.1.

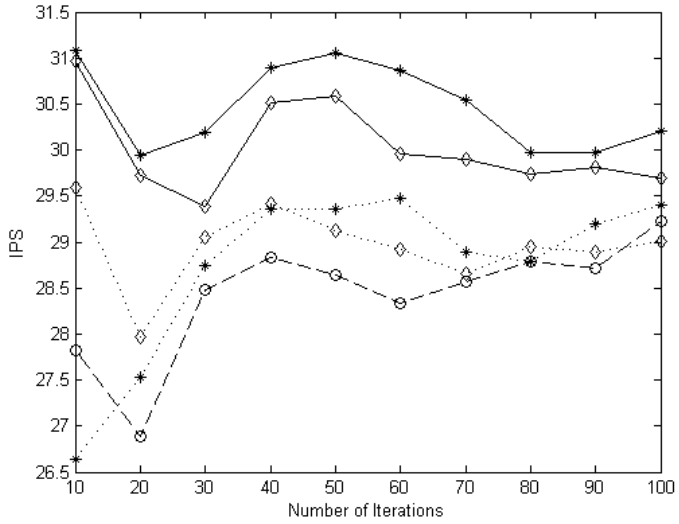

Figure 6.9: Interference-related Perceptual Scores for the standard NMF algorithm. Legend as per figure 6.1.

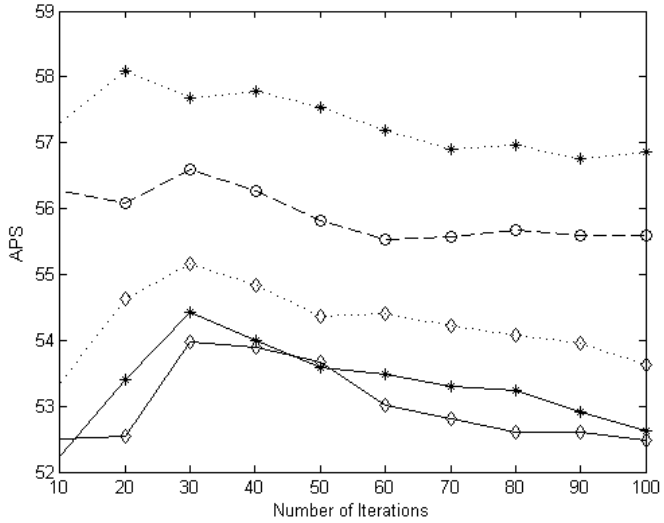

Figure 6.10: Artefacts-related Perceptual Scores for the SSNTF algorithm. Legend as per figure 6.1.

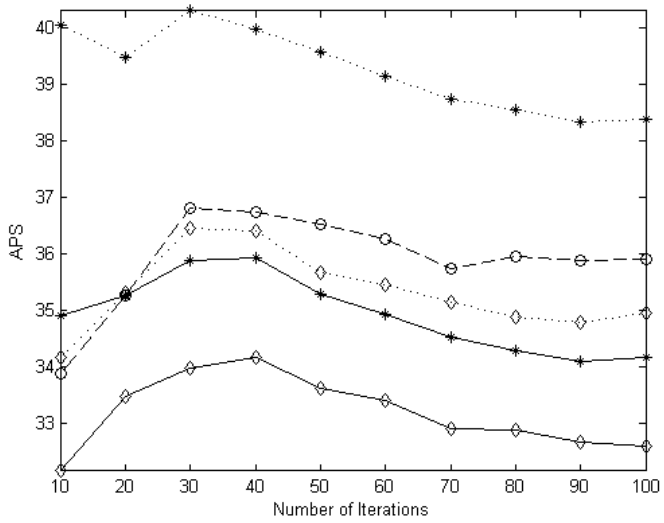

Figure 6.11: Artefacts-related Perceptual Scores for the UA algorithm. Legend as per figure 6.1.

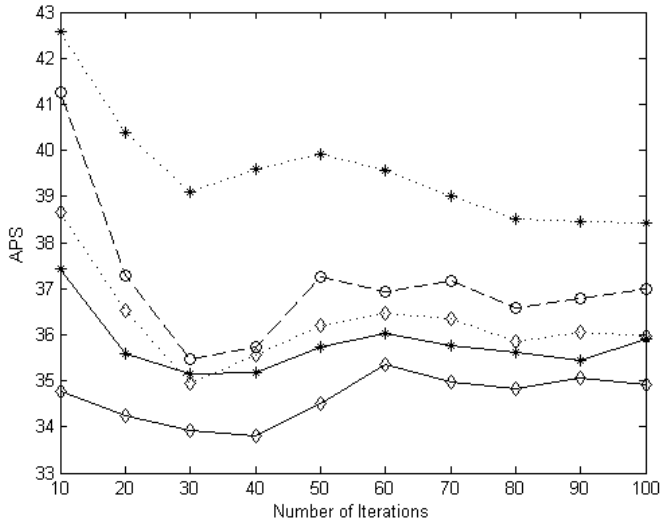

Figure 6.12: Artefacts-related Perceptual Scores for the standard NMF algorithm. Legend as per figure 6.1.

the KL and IS mask with $t = 2$ perform worst. Hence it can be concluded that there is a trade off between the presence of artefacts and the interference of the other sources in separation i.e. with the increase in artefacts reduces the interference of the other sources and improves the IPS score.

Hence, we have seen that no individual mask performs the best for all the metrics discussed in the chapter. However, in general the divergence based masks outperforms that of the Wiener filter. Also, it can be said that the mask should be chosen optimally according to the application for which the separation of sources is done. Also, the KL mask with $t = 2$ is optimal for the overall separation. Further, the IS mask with $t = 2$ is best suited for a separation algorithm where the rejection of the other sources is important.

As noted earlier, we obtained an interesting but surprising result from these tests at low number of iterations. We expected that the perceptual scores would gradually increase as we increase the number of iterations and would score highest once the convergence is achieved. This expectation was based on the fact that more accurate modelling of the sources would lead to more accurate separation, hence the high perceptual scores. On the contrary, in most cases, we obtained highest perceptual scores at low number of iterations, long before the separation algorithms are converged. After investigating this issue, we found that a more accurate modelling of sources has actually resulted in high perceptual scores, provided that the sources are reconstructed from the estimated spectrograms directly, instead of using the estimated source spectrograms to generate a mask to apply to the original mixture spectrogram.

Hence, it can said that the highest perceptual score at low number of iterations is a direct influence of using the masks when resynthesising the source signals.

We can further argue that the use of masking directly affects the quality of separation and can give better separation before the convergence is achieved. It can be potentially explained as follows. We are well aware of the fact that an audio spectrogram representation is sparse in nature. Hence, we can say that there will be a little or no energy present in many bins in the given spectrogram. However, the estimated source spectrograms are more likely to contain significant amount of energy at low number of iterations because of the random initialisation of the basis functions. This is due to the fact that the basis functions will not have adapted enough to remove this energy at low number of iterations. In contrast, if we use the estimated source spectrograms to generate masks then the masks would allocate energy of the corresponding bins in the original spectrogram in proportion to that of the estimated source spectrograms, and a proportion of a small number (in the original spectrogram) only yields a smaller number. Therefore, it can be said that the masks gives a better separation by removing the noise present in the estimated source spectrograms obtained at low numbers of iterations, particularly for bins with low energy in the original mixture spectrogram.

However, the above explanation is not valid for the bins having significant energy. Nonetheless, the difference between the energy content in these bins and the initial values obtained from the random initialisations will be considerably high. This high energy difference results in higher rescaled gradient in the

multiplicative updates used in these algorithms. As a result, the bins containing high energy are more likely to converge faster at the lower iterations. Hence, we can say that these bins are more likely to contain reasonable estimates of the actual source energy than the other bins at low numbers of iterations. Further, as discussed earlier, the use of masks is more efficient when the proportion of energy content, not the actual energy, is correct. Therefore, a good separation can be obtained once the proportion is approximately correct and the number of iterations performed or the errors in the actual energy estimates no longer matter. Therefore, it is reasonable to say that the use of masks can give good separation at low iterations and the highest perceptual scores at low number of iterations does make sense.

Hence, we can conclude that it may not be optimal to run the factorisation based algorithms to fully converge in order to obtain good separation. This is of particular benefit to reduce the run-time for the separation algorithms by reducing the number of iterations while still getting better separation performance. Audio examples related to the masking filters can be found at [92].

6.5 Conclusions

We first discussed the use of the generalised Wiener Filter as a means of resynthesis when performing the source separation. It was noted that although Wiener filter is optimal in a least square sense, it may not guarantee good

separation performance from a perceptual point of view. Hence, a new family of masks based on the KL divergence and IS divergence were introduced. These masks were shown to outperform the generalised Wiener filter for overall perceptual quality when tested using three NMF based separation algorithm and two testsets. We also discussed that a good separation performance can be achieved with these algorithms at low number of iterations long before convergence is achieved. Areas for future work include extending the family of masks to include the Beta divergence to attempt further improvements and also these masks may used to improve the results of the various NMF based algorithms discussed throughout the course of the thesis.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In chapter 1, first, we gave the motivation for blind sound source separation and defined the Blind SSS problem. Then, a review of sound source separation techniques was presented including the standard ICA algorithm, DUET and ADRess. Then, we discussed how these techniques would require more than one sensors and would fail in the case of mono signals. Further, we showed that how the factorisation based approaches such as NMF attempt to give a part-based decomposition of the audio spectrograms where the individual parts (basis functions) often correspond to the notes in the given mixture. This led to the problem of clustering frequency basis functions (the principal focus of our thesis) because these basis functions are usually greater in number than the active sources, hence the need of clustering. Then, we have stated how

the Shifted NMF algorithm attempts to use a single instrument basis function per source to avoid the need for clustering of the basis functions. Thereafter, we outlined the previous techniques used for clustering of the frequency basis functions, including the MFCC based clustering using the source filter model and the technique involving the incorporation of GS in NMF. We found that the previous clustering techniques were able to reduce the overlapping of sources in the output of the separation algorithms, but these techniques failed to give distinct separation of notes (source signals) corresponding to the sources. Hence, much research needed to be done in this area and so provided the main focus of the research in this thesis. Here, we give a brief summary of our contributions.

Firstly, we have introduced two novel SNMF based clustering algorithms in chapter 2 to improve the clustering of the NMF frequency basis functions. Here, we have used the SNMF model discussed in 1.5.6 for clustering the basis functions in such a way that it finds shift invariance in sets of CQ domain frequency basis functions where these frequency basis functions were obtained using NMF. In the process, we also dealt with the problem related to the change in timbre due to change in pitch by assuming a separate NMF frequency basis function for each note present in the given mixture. Then, the frequency basis functions were used to determine the instrument basis functions which were then used to cluster the original frequency basis functions to sources. The reconstruction of the source signals was performed using the two techniques, one-to-one mapping and spectral masking, as detailed in section 2.3.2. It was found that the source separation obtained through the proposed methods was

considerably better than the previously proposed clustering algorithms.

As noted previously in chapter 2, a limitation of using SNMF algorithms was that it used a log-frequency spectrogram and no true inverse of a log-frequency spectrogram was possible. The non-availability of an exact inverse of the CQT transform was found to result in a deterioration of the separation quality. Our next two contributions deal with this issue. We showed that the performance of the original SNMF separation algorithm can be improved considerably by incorporating a recently proposed improved method of obtaining an inverse CQT [85]. This was done to obtain an improved approximation of the inverse CQT and thus improve the reconstruction of the estimated signals. Further, on testing we have showed that the performance was significantly improved, in the order of 10 dB over the original implementation of the SNMF separation algorithm. It is important to note that more recently, a new invertible CQT method was proposed by Velasco et al [55], where it uses the non-stationary Gabor frames [56] to reconstruct more efficient and near to perfect inverse CQT from the CQT transform. However, we have not tested the proposed CQT method [55] to evaluate the performance in context of the separation algorithms proposed in this thesis.

Another attempt was made to deal with the lack of true CQT inverse by incorporating the CQT inside the signal model. In order to do so, we have proposed a modified SNMF clustering algorithm that uses the NMF frequency basis functions in the linear domain as an input to the SNMF model (clustering stage) rather than in the log-frequency domain. This was done

by incorporating the transform from linear to log frequency domains into the SNMF algorithm as detailed in chapter 4. Here, the CQT transform matrix was used to map the linear domain NMF basis functions to the CQT domain before every iteration until the convergence was achieved. This incorporation of the CQT transform allowed the SNMF model to measure the reconstruction error from the divergence in the linear domain, thereby allowing a better fit to the original linear domain frequency basis spectrogram. This use of the CQT inside the SNMF model ensured that the need of the inverse CQT for recovering the frequency basis functions in linear domain was avoided. The new SNMF model was found to improve the separation quality of the individual sources as compared to the SNMF clustering algorithms discussed in chapter 2.

Furthermore, we have introduced a group sparsity technique motivated by the work done in [53]. The SNMF clustering algorithm discussed in 2 was considered to have two stages, the NMF stage, where the frequency basis function is calculated and the clustering stage, where the SNMF model is used for clustering the frequency basis functions. Here, first we have incorporated the GS in NMF because we wanted the NMF basis functions, that correspond to instruments (groups), to be sparse in a group sense. Hence, the prior knowledge of groups while calculating the NMF basis functions would result in better grouping. Originally the GS in NMF [53] was shown to work well with mixtures of sources with temporal overlapping of up to 66%, which is not true in general. However, the application of GS in NMF at first stage was found to reduce the amount of temporal overlapping in the separated frequency basis functions.

Therefore, we thought that this reduction of temporal overlapping of sources and the prior knowledge of a particular group (source) would help the SNMF model to force the corresponding frequency basis function to iterate towards the group (or source) it belongs and thus result in better separation. Hence, the two stage process was implemented. A probabilistic model was used to exploit the equivalence between the ML problem and minimising the KL divergence cost function to estimate of frequency basis functions. This was based on the assumption that components in a magnitude spectrogram X are distributed according to a Poisson noise model [29] as explained in section 5.3. A similar probabilistic model was used for the IS divergence also as mentioned in [53]. A number of SNMF clustering algorithms were implemented based on the chosen divergence cost functions for the two stages in the SNMF clustering algorithm. A summary of quality measures for all the SNMF algorithms were calculated to evaluate the performance. Overall, just applying GS to the clustering stage of the SNMF clustering algorithm when using the KL divergence was found to perform best (see section 5.5).

Finally, we have presented a family of masks based on IS and KL divergences to improve the separation quality of the individual sources for various separation algorithms. This work was motivated by the fact that the commonly used Wiener filter masks were a least square best fit and was optimal in a least square sense, however it did not give a guarantee to be optimal in the perceptual sense. Also, we have tested the performance of the proposed masks with the number of iterations performed in the separation algorithm. The performance evaluation of

this new family of masks was done using the PEASS toolbox, which measures the perceptual quality of the audio source separation by calculating a set of objective measures. After evaluation, we found that no individual mask has performed the best for all the metrics discussed in the chapter 6. However, in general the divergence based masks have outperformed the masks based on the Weiner filter. Overall, it was concluded that the choice of mask should be optimally done according to the application for which the separation of sources is done. The best overall separation performance was obtained using the KL-based masks with $t = 2$, but that best rejection of the other sources was obtained using the IS divergence based masks with $t = 2$. On further investigation, we found that a good separation performance can be achieved with these algorithms at low number of iterations long before the convergence is achieved. Taking this fact into account, we can considerably reduce the run-time for the separation algorithms, while still getting better separation performance.

The above research work demonstrates a considerable improvement in dealing with the problem of monaural sound source separation. Through this work, we have overcome many of the problems present in previous research. We have demonstrated that by optimizing or modifying individual stages in the NMF based algorithms we can considerably improve the clustering of the frequency basis functions, and that the clustering obtained can be used to re-synthesis the individual sources with reasonable quality by the use of a new family of divergence based masks.

7.2 Future Work

Although, the algorithms described in this thesis represent an advance when attempting SSS on single channel mixtures of musical instruments, there remains a number of open issues and room for future research that may further improve the clustering of the frequency basis functions for sound source separation in music.

The algorithms described here are designed to work for pitched instruments only, possible future work would be an extension of the proposed algorithms by incorporating or developing new methods, such that, the algorithms work for both pitched and percussive instruments. The systems presented are limited to the fact that the number of the frequency basis functions, r corresponding to the notes are chosen manually and r varies with the mixture in question. A topic for future work would be in finding ways to automatically detect the number of frequency basis functions required for a particular mixture. Further, all the proposed separation algorithms were tested for the testset comprising of a mixture of two sources. However, with the extension of the proposed designs to overcome the complex overlapping of additional sources, these clustering algorithms can be extended for input mixtures of more than two sources.

We have shown how improved signal reconstruction can be obtained by using a family of masks obtained through the KL and IS divergences. Future work may include evaluating the performance of the family of masks on all the SNMF algorithms proposed in the thesis. This may improve the robustness in the separation of sources and may give a clear idea that how the use of masks vary

with the separation algorithms, number of iterations and the testsets used.

We have simulated all the proposed algorithms in Matlab, which is in general not time-efficient. Future work may include the implementation of these algorithms in high level language such as C or C++. This would help in optimising the time and memory required to run these algorithms.

In conclusion, the research undertaken has developed a number of possible ideas to improve the source separation algorithms. We have shown that by improving the clustering of the frequency basis functions we can successfully re-synthesize the individual sources with better quality. It is hoped that the techniques outlined in this thesis will provide a basis for further advances in sound source separation.

Bibliography

[1] T. Virtanen, A. T. Cemgil and S. Godsill, "Bayesian extensions to non-negative matrix factorisation for audio signal modelling," *in Proceedings of IEEE International Conference on Acoustic Speech and Signal Processing ICASSP*, pp. 1825,1828, April 2008.

[2] R. Jaiswal, D. FitzGerald, E. Coyle and S. Rickard, "Clustering NMF basis functions using Shifted NMF for monaural sound source separation," *in Proceedings of IEEE International Conference on Acoustic Speech and Signal Processing ICASSP*, pp. 245,248, May 2011.

[3] G. J. Brown and M. Cooke, "Computational auditory scene analysis," *Journal of Computer Speech and Language*, vol. 8, pp. 297,336, 1994.

[4] G. J. Brown, "Computational auditory scene analysis : a representational approach," *PhD thesis*, University of Sheffield, 1992.

[5] J. Paulus and T. Virtanen, "Drum transcription with non-negative spectrogram factorisation," *In Proceedings of European Signal Processing Conference*," pp. 4,8, Turkey, 2005.

[6] S. Itahashi, S. Makino and K. Kido, "Discrete-word recognition utilizing a word dictionary and phonological rules," *in Proceedings of IEEE Transactions on Audio and Electro-acoustics*, vol. 21, pp. 239,249, Jun 1973.

[7] D. Fitzgerald, "The Good Vibrations Problem," *134th International Audio Engineering Society Convention*, Rome, Italy, 2013.

[8] L. Girin, J. L. Schwartz and G. Feng, "Audio-visual enhancement of speech in noise," *Journal Acoustic Society America*, vol. 109, pp. 3007,3020, Jun 2001.

[9] SM. R. Schroeder and B. S. Atal, "Generalized short-time power spectra and autocorrelation functions," *Journal Acoustic Society America*, vol. 34, pp. 1679,1683, 1962.

[10] J. E. Youngenberg and S. F. Boll, "Constant-Q signal analysis and synthesis," *in Proceedings of IEEE International Conference Acoustic, Speech, Signal Processing*, vol. 3, pp. 375,378, 1978.

[11] G. Gambardella, "A contribution to the theory of short-time spectral analysis with non-uniform bandwidth filters," *in Proceedings of IEEE Transactions Circuit Theory*, vol. 18, pp. 455,460, Jul 1971.

[12] G. Gambardella, "The Mellin transforms and constant-Q spectral analysis," *Journal Acoustic Society America*, vol. 66, pp. 913,915, 1979.

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

[14] D. J. Field, "What is the goal of sensory coding?," *Neural Computation*, vol. 6, pp. 559,601, 1994.

[15] B. A. Olshausen and D. F. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?," *Vision Research*, vol. 37, pp. 3311,3325, 1997.

[16] P. Smaragdis, B. Raj and M.V. Shashanka, "Supervised and semi-supervised separation of sounds from single channel mixtures," *In Proceedings of International Conference on ICA and Signal Separation*, pp. 414,421, London, UK, Sept 2007.

[17] P. O. Hoyer, "Non-negative Matrix Factorization with Sparseness Constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457,1469, 2004.

[18] D. Fitzgerald, "User assisted source separation using non negative matrix factorisation," *in proceedings of IET Irish Signals and Systems Conference,* Dublin, 2011.

[19] D. Fitzgerald, "NMF-based algorithms for user assisted sound source separation," *EURASIP Journal on Audio, Speech, and Music Processing*, 2012.

[20] D. FitzGerald, M. Cranitch and E. Coyle, "Extended non-negative tensor factorisation models for musical sound source separation," *Computational Intelligence and Neuroscience*, Hindawi Publishing Corporation, 2008.

[21] B. Rivet, L. Girin and C. Jutten, "Mixing audio-visual speech processing and blind source separation for extraction of speech," *IEEE transactions on audio, speech, and language*, vol. 15, pp. 96,108, 2007.

[22] Cédric Févotte, Nancy Bertin and Jean-Louis Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence: with application to music analysis," *Journal of Neural Computation*, vol. 21, pp. 793,830, Mar 2009.

[23] D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorisation," *Advances in neural information processing system*, pp. 556,562, 2000.

[24] S. A. Abdallah and M. D. Plumbley, "Polyphonic transcription by non-negative sparse coding of power spectra," *in Proceedings of the* 5^{th} *International Conference on Music Information Retrieval*, Barcelona, Spain, 2004.

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

[26] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of Royal Statistical Society* vol. 39, pp. 1,38, 1997.

[27] L. Saul and F. Pereira, "Aggregate and mixed-order Markov models for statistical language processing," *in Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pp. 81,89, ACL Press, 1997.

[28] T. Virtanen, "Monaural sound source separation by Non-negative matrix factorisation with temporal continuity and sparseness criteria," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 3, pp. 1066,1074, 2007.

[29] T. Virtanen, "Monaural sound source separation by perceptually weighted non-negative matrix factorisation," *Technical report at Tampere University of Technology*, Institute of Signal Processing, 2007.

[30] E. Zwicker and H. Fastl, "Psychoacoustics: Facts and Models," *Springer, Berlin*, Second Edition, pp. 141,144, Berlin, 1999.

[31] E. Vincent and X. Rodet, "Music transcription with ISA and HMM," *in Proceedings 5th International Symposium Independent Component Analysis Blind Signal Separation*, pp. 1197,1204, Granada, Spain, 2004.

[32] S. Rickard, R. Balan and J. Rosca, "Real-time time frequency based blind source separation," *in Proceedings of International Conference on*

Independent Component Analysis and Signal Separation, pp. 651,656, 2001.

[33] P. Smaragdis, "Discovering auditory objects through non-negativity constraints," *in Proceedings of Research Workshop on Statistical and Perceptual Audio Processing*, 2004.

[34] L. Benaroya, L. Mc Donagh, F. Bimbot and R. Gribonval, "Non negative sparse representation for Wiener based source separation with a single sensor," *in proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 6, pp. 613,616, 2003.

[35] D. Griffin and J. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 32, pp. 236,243, Apr 1984.

[36] J. LeRoux, N. Ono and S. Sagayama, "Explicit consistency constraints for STFT spectrograms and their application to phase reconstruction," *in Proceedings of Workshop on Statistical and Perceptual Audition*, pp. 23,28, 2008.

[37] J. LeRoux, E. Vincent, Y. Mizuno, H. Kameoka, N. Ono and S. Sagayama, "Consistent wiener filtering: Generalized time-frequency masking respecting spectrogram consistency," *in Proceedings 9th International Conference on Latent Variable Analysis and Signal Separation*, pp. 89,96, 2010.

[38] V. Emiya, E. Vincent, N. Harlander and V. Hohmann, "Subjective and objective quality assessment of audio source separation," *IEEE Transactions on Audio, Speech and Language Processing*, vol.19, pp. 2046,2057, Sept 2011.

[39] D. Fitzgerald and R. Jaiswal, "On the use of masking filters in sound source separation," *15th International Conference on Digital Audio Effects*, York, England, 2012.

[40] D. FitzGerald, "Upmixing from mono - a source separation approach," *in proceedings of 17th International Conference on Digital Signal Processing*, pp. 721,727, Greece, 2011.

[41] M. Spiertz and V. Gnann, "Source-Filter based clustering for monaural blind source separation," *in Proceedings of the 12th International Conference on Digital Audio Effects*, Italy, 2009.

[42] D. FitzGerald, M. Cranitch and E. Coyle "Using Tensor Factorization model to Separate drums from Polyphonic music," *in Proceedings of the International Conference on Digital Audio Effects*, Como, Italy, 2009.

[43] D. FitzGerald, "Automatic drum transcription and source separation," *Ph.D. dissertation, Conservatory of Music and Drama*, Dublin Institute of Technology, Ireland, 2004.

[44] D. FitzGerald, M. Cranitch and E. Coyle "Shifted Non-negative matrix factorisation for sound source separation," *IEEE Workshop of Statistical Signal Processing, Bordeaux*, France, 2005.

[45] J. C. Brown, "Calculation of a Constant Q spectral transform," *Journal of the Acoustic Society of America*, vol. 90, pp. 60,66, 1991.

[46] G. N. Abras and V. L. Ballarin, "A Weighted K-means Algorithm applied to Brain Tissue Classification," *Journal of Computer Science and Technology*, vol. 5, pp. 121,126, 2005.

[47] S. Rickard and O. Yilmaz, "On the approximate W-Disjoint Orthogonality of Speech," *IEEE International Conference on Acoustics, Speech, and Signal*, vol. 1, pp. 529,532, May 2002.

[48] S. Makino, T. Lee and H. Sawada, "Blind Speech Separation," *Book published, Springer,* chapter 8, July 2007.

[49] O. Yilmaz and S. Rickard, "Blind Separation of Speech Mixtures via Time-Frequency Masking," *The IEEE Transactions on Signal Processing*, vol. 52, pp. 1830,1847, July 2004.

[50] A. Jourjine, S. Rickard and O. Yilmaz, "Blind Separation of Disjoint Orthogonal Signals: Demixing N Sources from 2 Mixtures," *In Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 2985,2988, Istanbul, Turkey, June 2000.

[51] A. Lefèvre, "Dictionary Learning methods for single-channel audio source separation," *Ph.D thesis*, Uinversité Catholique de Louvain-la-Neuve, Oct 2012.

[52] E. M. Burns, "Intervals, scales and tuning," *The Psychology of Music*, D. Deutsch, Ed. Academic Press, Chapter 7, 1999.

[53] A. Lefèvre, F. Bach and C. Févotte, "Itakura-Saito nonnegative matrix factorization with group sparsity," *in Proceedings of IEEE International Conference on Acoustic Speech and Signal Processing ICASSP*, pp. 21,24, May, 2011.

[54] R. G. Brown and P. Y. C. Hwang, "Introduction to random signals and applied Kalman Filtering (3^{rd} edition)," *New York: John Wiley & Sons*, ISBN 0-471-12839-2, 1996.

[55] G. A. Velasco, N. Holighaus, M. Dörfler and T. Grill, "Constructing an invertible constant-Q transform with non-stationary Gabor frames," *in Proceedings of the 14th International Conference on Digital Audio Effects*, Paris, France, Sept 2011.

[56] F. Jaillet, P. Balazs, M. Dörfler, and N. Engelputzeder, "Nonstationary Gabor frames," *in proceedings of International Conference on Sampling Theory and Applications*, pp. 227,230, 2009.

[57] A. Hyv̈arinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks*, vol. 10, pp. 626,634, 1999.

[58] J. F. Cardoso, "Blind signal separation: statistical principles," *in Proceedings of the IEEE, special issue on blind identification and estimation*, vol. 86, pp. 2009,2025, Oct 1998.

[59] A. J. Bell and T. J. Sejnowski, "An information maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7(6), pp. 1129,1159, 1995.

[60] C. Févotte and J. Idier, "Algorithms for nonnegative matrix factorization with the beta-divergence," *Neural Computation*, 2011.

[61] A. S. Bregman, "Auditory Scene Analysis," *MIT Press*, 1990.

[62] D. Barry, B. Lawlor and E. Coyle, "Sound Source Separation: Azimuth Discrimination and Resynthesis," *Proceedings of the* 7^{th} *International Conference on Digital Audio Effects*, Naples, Italy, Oct 2004.

[63] A. Hyv̈arinen, "Survey on Independent Component Analysis," *Neural Computing Surveys 2*, pp. 94,128, 2009.

[64] D. Barry, D. Fitzgerald, E. Coyle and B. Lawlor, "Single channel Source Separation using short-time independent component analysis," *AES Convention paper, 119th Convention*, New York, 2005.

[65] A. Hyv̈arinen, J. Karhunen and E. Oja, "Independent Component Analysis," *John Wiley*, New York, 2001.

[66] P. Comon, "Independent component analysis, a new concept?," *Signal Processing*, Elsevier, vol. 36, pp. 287,314, 1994.

[67] S. A. Abdallah and M. D. Plumbley, "An independent component analysis approach to automatic music transcription," *in Proceedings of Audio Engineering Society 114th Convention*, Amsterdam, The Netherlands, Mar 2003.

[68] T. Virtanen, "Sound source separation using sparse coding with temporal continuity objective," *Proceedings of the International Computer Music Conference*, pp. 231,234, Singapore, 2003.

[69] R. Jaiswal, D. FitzGerald, E. Coyle and S. Rickard, "Shifted NMF Using an Efficient Constant Q Transform for Monaural Sound Source Separation," *22nd IET Irish Signals and Systems Conference*, Dublin, Jun 2011.

[70] Chen Wen- Hsiung, C. H. Smith and S. C. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform," *IEEE Transactions on Communications*, vol. 25, pp. 1004,1009, Sept 1977.

[71] B. Logan, "Mel frequency cepstral coefficients for music modelling," *In proceedings of International Symposium on Music Information Retrieval*, Cambridge Research Laboratory, Compaq Computer Corporation, 2000.

[72] S. Molau, M. Pitz, R. Schluter, and H. Ney, "Computing Mel-frequency cepstral coefficients on the power spectrum," *in Proceedings at IEEE International conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 73,76, 2001.

[73] L. K. Saul and S.T. Roweis, "An introduction to locally linear embedding," 2001, Available from http://www.cs.toronto.edu/ roweis/lle/.

[74] L. K. Saul and S.T. Roweis, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323,2326, 2000.

[75] M. Brand, "Charting a manifold," *In Advances in Neural Information Processing Systems*, MIT Press, 2003.

[76] H. Chang and D.Y. Yeung, "Robust locally linear embedding," *Pattern Recognition*, vol. 39, pp. 1053,1065, 2006.

[77] D. Ridder and R.P. Duin, "Locally Linear Embedding for classification," *in the pattern recognition Group Technical Report Series*, Delft University of Technology, 2002.

[78] J. S. Tobias, "Foundations of Modern Auditory Theory," *Academic Press*, vol. 1, New York, 1970.

[79] Z. Duan, Y. Zhang, C. Zhang and Z. Shi, "Unsupervised single-channel music source separation by average harmonic structure modeling," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, pp. 766,778, May 2008.

[80] Hui Kanghua and Wang Chunheng, "Clustering-based locally linear embedding," *19th international conference on Pattern Recognition*, pp. 1,4, Dec 2008.

[81] E. Vincent, R. Gribonval and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, pp. 1462,1469, 2006.

[82] B. Wang and M. D. Plumbley, "Investigating single-channel audio source separation methods based on non-negative matrix factorization," *in Proceedings of the ICA Research Network International Workshop*, pp. 17,20, 2006.

[83] C. Schörkhuber and A. Klapuri, "MATLAB toolbox for the CQT," `http://www.elec.qmul.ac.uk/people/anssik/cqt/`.

[84] J. C. Brown and M. S. Puckette, "An efficient algorithm for the calculation of a Constant Q Transform," *Journal Acoustic Society America*, vol. 92, no. 5, pp. 2698,2701, 1992.

[85] C. Schörkhuber and A. Klapuri, "Constant-Q Transform toolbox for music processing," *7th Sound and Music Computing Conference*, Barcelona, Spain, 2010.

[86] R. Jaiswal, D. Fitzgerald, E. Coyle and S. Rickard, "Shifted NMF with Group Sparsity for clustering NMF basis functions," *Proceedings at 15th International Conference on Digital Audio Effects*, York, UK, Sept 2012.

[87] R. Jaiswal, D. Fitzgerald, E. Coyle and S. Rickard, "Towards Shifted NMF for improved monaural separation," *Proceedings at 23rd IET Irish Signals and Systems Conference,* LYIT Letterkenny, Ireland, June 2013.

[88] D. FitzGerald, M. Cranitch and M. T. Cychowski, "Towards an inverse Constant Q Transform," *120th Audio engineering Society Convention*, Paris, France, 2006

[89] E. Vincent, R. Gribonval and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, pp. 1462,1469, 2006.

[90] B. W. Bader and T. G. Kolda, "Algorithm 862: MATLAB tensor classes for fast algorithm prototyping," *ACM Transactions on Mathematical Software*, vol. 32, pp. 635,653, 2006.

[91] P. Siedlaczek, "Advanced Orchestra Library Set," Available at `http://eleceng.dit.ie/derryfitzgerald/index.php?uid=489&menu_id=52`.

[92] Audio Examples `http://eleceng.dit.ie/rajeshjaiswal`