Aalto University
School of Science
Degree Programme in Computer Science and Engineering

Ajay Ramaseshan

Application of Multiway Methods for Dimensionality Reduction to Music

Master's Thesis
Espoo, November 24, 2013

Supervisor:	Professor Olli Simula, Aalto University
Advisors:	Docent Francesco Corona, PhD Yoan Miche, D.Sc. (Tech)

Aalto University
School of Science

Aalto University
School of Science
Degree Programme in Computer Science and Engineering

ABSTRACT OF MASTER'S THESIS

Author:	Ajay Ramaseshan		
Title: Application of Multiway Methods for Dimensionality Reduction to Music			
Date:	November 24, 2013	**Pages:**	91
Major:	Information and Computer Science	**Code:**	T-110
Supervisor:	Professor Olli Simula		
Advisors:	Docent Francesco Corona, PhD Yoan Miche, D.Sc. (Tech)		

This thesis can be placed in the broader field of Music Information Retrieval (MIR). MIR refers to a huge set of strategies, software and tools through which computers can analyse and predict interesting patterns from audio data. It is a diverse and multidisciplinary field, encompassing fields like signal processing, machine learning, musicology and music theory, to name a few.

Methods of dimensionality reduction are widely used in data mining and machine learning. These help in reducing the complexity of the classification/clustering algorithms etc, used to process the data. They also help in studying some useful statistical properties of the dataset. In this Master's Thesis, a personalized music collection is taken and audio features are extracted from the songs, by using the mel spectrogram. A music tensor is built from these features. Then, two approaches to unfold the tensor and convert it into a 2-way data matrix are studied. After unfolding the tensor, dimensionality reduction techniques like Principal Components Analyis (PCA) and classic metric Multidimensional Scaling (MDS) are applied. Unfolding the tensor and performing either MDS or PCA is equivalent to performing Multiway Principal Component Analysis (MPCA). A third method Multilevel Simultaneous Component Analysis (MLSCA), which builds a composite model for each song is also applied.

The number of components to retain are obtained by hold-out validation. The fitness of each of these models were evaluated with the T^2 and Q statistic, and compared with each other. The aim of this thesis is to produce a dimensionality reduction which can be used for further MIR tasks like better clustering of data with respect to e.g. artists / genres.

Keywords:	Mel spectrogram, Multidimensional Scaling (MDS), Multilevel Simultaneous Component Analysis (MLSCA), multiway data, Multiway Principal Components Analysis (MPCA), music collection, Music Information Retrieval (MIR), Principal Components Analysis (PCA)
Language:	English

Acknowledgements

First of all, I would like to thank the Department of Information and Computer Science, Aalto University for providing me with an opportunity to do my Master's Thesis here. Special thanks to instructors Docent Francesco Corona and Doctor Yoan Miche, without which this work would not have been possible. They were of immense help especially when understanding the algorithms and drawing up a workflow for the thesis. I would also like to thank my supervisor Professor Olli Simula for reading and correcting the thesis. His valuable comments and suggestions were extremely helpful.

I would also like to thank my parents and my cousin Karthikesh for providing me the support to do my studies in Finland and for always being there for me. And also, thanks to all those good friends I made in Otaniemi and in the department. The discussions and fun I had with them is something that I will always remember and cherish.

Espoo, November 24, 2013

Ajay Ramaseshan

Abbreviations and Acronyms

AMG	All Music Guide
CQT	Constant Q Transform
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
EVD	Eigenvalue Decomposition
FFT	Fast Fourier Transform
GICA	Grounded Intersubjective Concept Analysis
ICA	Independent Component Analysis
ISMIR	International Society for Music Information Retrieval
LDA	Linear Discriminant Analysis
MDS	Mutidimensional Scaling
MFCC	Mel Frequency Cepstral Coefficient
MIDI	Musical Instrument Digital Interface
MIR	Music Information Retrieval
MLSCA	Multilevel Simultaneous Component Analysis
MPCA	Multiway Principal Component Anaysis
PC	Principal Component
PCA	Principal Component Anaysis
PMSC	Princial Mel Spectrum Coefficients
PRESS	Predicted Residual Sum of Squares
RESS	Residual Sum of Squares
RMS	Root Mean Square
SOM	Self Organizing Map
SVD	Singular Value Decomposition
STFT	Short Time Fourier Transorm

List of Symbols

$x(t)$	Time domain representation of Audio signal
h	Hop size
$\mathbf{x}_n$	Description of nth frame in signal
$w(t)$	Window function
α, β	Parameters of Hamming Window function
N_{win}	Number of samples in Hamming Window
f_s	Sampling rate
T_{win}	Time duration of Hamming Window
F_b	Number of frequency bins in Fourier Transform/ Spectrogram
N_b	No of frames in the signal
n_{fft}	No of points for which Fourier Transform is taken
$S_{b,n}$	STFT for frequency bin b and time frame n
$\|\mathbf{S}\|$	Magnitude Spectrum
$mel(f)$	Mel value at frequency f
$\mathbf{M}$	Mel Filter Matrix
M_{ij}	Jth element in ith filter
$\mathbf{X}$	Log Mel Spectrogram, General Dataset
$\mathbf{x}_i$	Individual data point
m	Number of instances
n	Number of features
$\mathbf{T}$	Scores
$\mathbf{P}$	Loadings
$\mathbf{E}$	Error or residual matrix
$\mathbf{C}$	Covariance matrix
$\mathbf{P}$	Orthogonal loading matrix
$\mathbf{D}$	Diagonal matrix containing Eigenvalues λ_k
$\mathbf{t}_i$	Scores for ith data point
$\hat{\mathbf{x}}_i$	Reconstructed ith data point
$\mathbf{e}_1$ $\mathbf{e}_2$ $\mathbf{e}_3$	Basis vectors of Euclidean Coordinate System

$\mathbf{v}$	Vector corresponding to a data point
$E(k)$	Reconstruction Error
$T^2_{i,k}$	T^2 distance for ith data point with k components
D_m	Mahalanobis Distance
$Q_{i,k}$	Q Distance for ith data point with k components
$T^2_{lim_k}$, Q_{lim_k}	Limiting values on $T^2_{i,k}$ and $Q_{i,k}$
$\mathbf{I}_p$	Normalized Inner Product Matrix
$\mathbb{X}$	Multiway Music Collection Data
$\mathbb{E}$	Multiway Error
K_i, K	Number of frames of ith song, $K = \sum_{i=1}^{k} K_i$.
$\mathbf{X}_i$	Matrix representation for ith song
$\mathbf{X}_{unf.songs}$, $\mathbf{X}_{unf.frames}$	Matrices obtained after unfolding 3-way data $\mathbb{X}$
$\mathbf{T}_{i,k}$	Scores for ith song
K_{sh}	Number of frames of the shortest song
$\mathbf{m}$	Global Mean
$\mathbf{t}_{b,i}$,$\mathbf{P}_b$	Between song scores and loadings for ith song
$\mathbf{T}_{w,i}$, $\mathbf{P}_w$	Within song scores and loadings for ith song
$\mathbf{E}_i$	Reconstrction Error or residual for ith song
$\mathbf{x}_{b,i}$	Mean vector of ith song
$\mathbf{W}_b$	Diagonal matrix with $\mathbf{W}_{b_i i} = \sqrt{K_i}$
R_b ,R_w	Between song components and within song components
$\mathbf{X}_{w,i}$	Within song matrix for ith song
$\mathbf{t}_{S_{b,i}}$	SVD Defined Between song scores for ith song
T^2_{b,i,R_b}	Between song T^2 distance for R_b retained components for ith song
Q_{b,i,R_b}	Between song Q distance for R_b retained components for ith song
T^2_{w,j,i,R_w}	Within song T^2 distance for R_w retained components for jth frame in ith song
Q_{w,j,i,R_w}	Within song Q distance for R_w retained components for jth frame in ith song

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 What is Music Information Retrieval?

A little over a decade ago, music was popular usually in physical media in the form of casette tapes, or CDs. The methods to listen to music were limited to radio broadcasts or casette players. Keeping a track of emerging trends in music was not as challenging as today, since the methods to distribute music were limited to physical storage media. However, all this has changed in the last decade with the rise of digital music. Digital music has now made new music more easily available, and thus the magnitude of music available is increasing at a rapid rate. It is estimated that over 10,000 new albums are released and more than 100,000 works are registered for copyright each year [33]. In 2005, legal music and mobile phone ringtone downloads witnessed a three-fold growth, and in the UK, digital music sales have overtaken sales of CDs and records [4]. With the proliferation of mp3 players, ipods, smartphones and the like, listening to music, from a community activity, is now becoming more personalised. Thus there is a growing need now to manage digital music collections and make personalised recommendations for users to discover more music. Managing and analysing digital music collections is also beneficial for record companies to keep track of emerging music trends, as also for musicians and musicologists to see which pieces of music are more similar to a given piece, or to study the evolution of a certain genre of music over the decades. The various strategies to search and organise massive digital music collections can be termed as music information retrieval (MIR) [6].

MIR is an inherently interdisciplinary field, drawing upon knowledge from various fields like signal processing, machine learning, computer science, music theory, to name a few. It is evident from the number of papers published in the proceedings of the International Society for Music Information Retrieval (ISMIR) conference, that the research activity in MIR is increasing. The first conference was held in 2000 and the proceedings included 35 papers, whereas this year more than 100 papers have been accepted [16].

1.2 Aspects of MIR

All MIR systems incorporate the following modules - query formation, description extraction, matching and finally music document retrieval [6]. Based on the kind of query, MIR systems can be categorised into:

1. High specificity systems - These require that the returned results match the query perfectly. The most common application of this is audio fingerprinting. Only a short excerpt, say 30 s of the song is given as a query, and information about the track like artist/ album name, genre, year is returned, as in the case of the popular application Shazam [35] or Microsoft Bing Audio search. Another application is plagiarism detection.

2. Mid specificity systems - The returned results share some commonalities with the query but are not identical to it. They retrieve music with similar high level music descrption like melody, harmony, rhythm etc,. but do not match the exact audio content for e.g. cover song detection, melodic similarity.

3. Low specificity systems - The returned results have little direct similarity with the query but share certain global characteristics, for e.g. genre/style/ mood retrieval, retrieving music with similar instrumentation.

Downie [9] has described many aspects of music, which interact to make MIR complex and challenging. These are pitch, temporal, harmonic, timbral, editorial, textual and bibliographic aspects. The pitch of a note is measured by its fundamental frequency F0. Note names, scale degrees or pitch class numbers are some of the methods used to represent pitch. The temporal and rhythmic aspect deals with duration of notes, musical meter, beat tracking, tempo estimation etc. The harmonic aspect of music is important when two or more pitches are played at the same time, also known as polyphony. Chords in music are a prime example of the harmonic aspect. Musical timbre can be defined as the colour of sound, or that aspect of music which helps to distinguish two notes played at identical pitches and having identical loudness [11]. Thus it is an inherent property of the musical instrument and is useful in instrument detection. The editorial aspect deals with performance instructions, like loudness levels, instrument fingerings, and other information on how to play a particular piece of music. Not all performances of a piece of music are the same, thus a single song could have many versions like unplugged versions, studio recordings, album and live concert recordings. The lyrics of a song constitute the textual aspect of music. Since the textual fact is more independent from other aspects of music, it will not be discussed further in this thesis. Finally, the bibliographic aspects deals with the metadata of music or music tags like track title, track length, artist name, genre, year, mood etc.

In the next section, we look at two methods which have been used for MIR - metadata approach or the audio content analysis approach.

1.3 Metadata vs Audio Content Analysis

Metadata, or the bibliographic aspect of music as introduced in the previous section is used by some MIR systems. Websites like All Music Guide (AMG) and Gracenote provide metadata for millions of tracks. Metadata can be broadly divided into two: *factual metadata* and *cultural metadata* [6]. Factual metadata provide technical details about music for e.g. song name, artist/album, year, length etc. while cultural metadata provide subjective information about music such as genre, mood, and style. MIR systems mine this metadata along with other information like play count of the song, or the number of 'likes' it receives to recommend similar music. Maintaining a metadata repository is useful for audio fingerprinting as in the case of Shazam and Bing Audio search described earlier.

However, there are many challenges facing metadata based methods and it cannot solve all possible MIR tasks. Some of the challenges are:

1. Metadata Consistency and Completeness - It is extremely difficult to get a consistent set of metadata. Misspellings, capitalization or punctuation differences in artist/album names make metadata consistency difficult. It is also quite possible that the metadata for the track may be incomplete or absent. This is often the case when the artist/genre is not popular.

2. Time Consuming Process - Metadata has to be entered by human experts. It is estimated that about 20-30 minutes time is taken to enter the metadata for a single song. Scaling to music collections of millions of songs makes entering metadata extremely time-consuming and infeasible.

3. Metadata cannot solve MIR tasks that deal with the aspects of music discussed earlier. For instance, a task like melodic similarity, or beat tracking, or chord detection can be performed only when one listens to the audio and identifies the melody, chords or beat. Hence, relying just on the metadata will be insufficient in these cases. Such kind of tasks can only be performed by analysis of the audio content.

Thus, metadata approaches need to be complemented with audio content analysis approaches. The next section briefly introduces the first two steps in audio content analysis - feature extraction and dimensionality reduction.

1.4 Feature Extraction and Dimensionality Reduction

In audio content analysis, the audio is converted into a feature representation using signal processing techniques. However, these features are generally in a high-dimensional space, and thus dimensionality reduction needs to be performed on the features to reduce it to lower dimensions for machine learning algorithms. Reduction to lower dimensions helps to construct algorithms with lesser number of parameters, thus reducing the algorithm complexity. It helps to understand and visualize the structure of complex datasets. Also some statistical properties like independence and uncorrelatedness which may not be in the high-dimensional space, may be observable in the lower dimensional space.

Many techniques for dimensionality reduction have been developed, which can be categorized into linear and non-linear methods. Linear methods like Principal Component Analysis (PCA) and classic metric multidimensional scaling (MDS) were the first methods to be developed, and these will be discussed in this thesis. Over the past few years, many non-linear methods like self-organizing maps, geodesic mappings, locally linear embeddings, laplacian eigenmaps etc have been developed. Also termed as manifold learning, these are discussed in great detail in [20].

1.5 Scope and Purpose of the thesis

The main goal of the thesis is to illustrate how dimensionality reduction techniques can be applied to a personalised music collection. The raw audio files are first converted to a feature representation, using signal processing techniques as mentioned earlier. A tensor representation of the music dataset is then obtained. Two ways of decomposing or unfolding this tensor are then applied. Next, dimensionality reduction techniques are performed on the unfolded tensor. Using a validation dataset, the appropriate number of components to be retained is selected. An evaluation of these dimensionality reduction methods is performed, and it is shown how effectively the dimensionality reduction techniques capture the data. Finally, a cross comparison is performed, ie certain songs selected from one dimensionality reduction technique are compared in other methods. The end goal of this thesis is to obtain an appropriate and efficient dimensionality reduction for music data, so that the features in the reduced space can be used for some audio content analysis task. Thus, this thesis can be placed in the broader domain of audio content analysis.

1.6 Structure of the Thesis

The purpose of this chapter is to provide a broad overview to this field of MIR, and introduce the problem statement of the thesis. Chapter 2 covers the process of feature extraction i.e. how to convert the raw audio into a feature representation. The dimensionality reduction algorithms are discussed in detail in Chapter 3, this is the core of the thesis work. Chapter 4 discusses the experiments performed on these dimensionality reduction methods and the results are discussed and interpreted in Chapter 5. Finally, Chapter 6 summarizes and concludes the thesis.

Chapter 2

Feature Extraction

2.1 Introduction

Feature extraction is the process where raw data is converted into a feature representation that can be used for further processing. For instance, in the case of image processing, the image is converted into feature descriptors which capture pixel intensity. In the field of text processing, the text could be converted, say into a term-document matrix. In audio content analysis, the digital audio track is converted into a feature representation that captures different aspects of the music, like pitch, loudness, duration, melody, harmony, or timbre. As discussed in Section 1.2 these aspects interact with each other, thus making music information retrieval a multifacted problem [9]. The features are extracted based on the audio content analysis task since features suitable for one task may not be suitable for another task. Thus different audio content analysis tasks like cover song detection, audio fingerprinting or instrument detection use different set of features.

Figure 2.1 shows the block diagram of a general audio content analysis system. Firstly, some preprocessing steps are applied to the digital audio track, or the song. This is next followed by the process of feature extraction where the preprocessed audio signal is usually converted into a high-dimensional feature representation. Due to the high dimensionality, this feature representation is not very suitable to be directly used. Thus using the methods discussed in Chapter 3 the features are projected into a lower dimensional space. This step is known as feature transformation. Then some machine learning algorithm is applied on these transformed features, resulting in a set of trained model parameters. Next, a test song is fed into the system, and the transformed features are obtained for the test song. The model parameters calculated earlier are applied on the transformed features of the test song

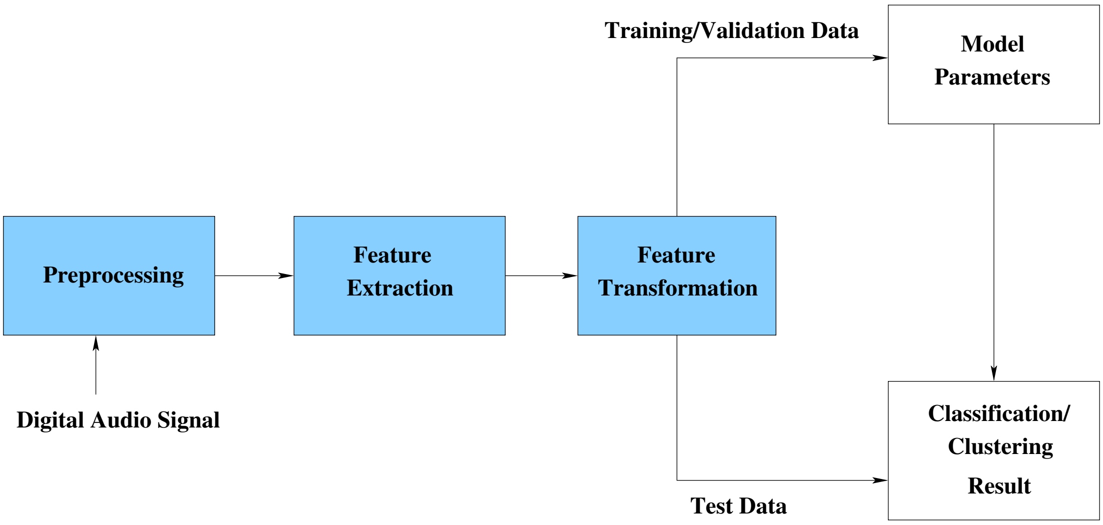

Figure 2.1: General Audio Content Analysis System. Coloured Blocks indicate the scope of the thesis.

and results of the audio content analysis task, like classification or clustering are obtained.

This chapter discusses feature extraction, and the next chapter discusses feature transformation.

2.2 Preprocessing

Before the feature extraction step, a few audio preprocessing steps are undertaken. Some common preprocessing steps are described as follows:

1. Downmixing - An audio track could be recorded in many channels. Usually, two channels are used in a stereo recording. Downmixing refers to taking the arithmetic mean of the two channels. Thus, the resultant audio track gets represented by a single channel.

2. Downsampling/Upsampling - The *sampling rate* of an audio signal is defined as the number of discrete audio samples recorded per second. Downsampling is a process where the audio signal is converted from a higher sampling rate to a lower sampling rate by means of some sample rate conversion algorithm. The reverse process is called upsampling.

3. Standardising the bit rate - The concept of bit rate is borrowed from computer networks. In audio processing field, *bit rate* refers to how

many bits are needed to represent a second of audio. In other words, bit rate is a quantifier for the number of volume levels of an audio track. Higher the bit rate, better is the sound quality, however the size of the audio file is increased.

2.3 Feature Extraction

2.3.1 Instantaneous or Low-Level Features

The purpose of feature extraction is to convert raw audio data into a numerical dataset with instances and features. These features are extracted from various properties of the audio signal, using several signal processing techniques. These features are called instantaneous features, since they are extracted from a short frame (or time block) of the audio signal, and produce a value for each frame. Segmentation into frames can be done in two ways [6]:

1. Fixed length segmentation - A fixed frame length of (10-1000 ms) is used.

2. Beat Synchronous segmentation - The frames are aligned to musical beat boundaries. This is commonly used in applications like beat tracking.

They can also be termed short-term features or low-level features. Peeters [26] proposed the following categorisation of low-level features:

1. Temporal Shape - Features computed from the waveform of the signal, for example attack time, effective duration.

2. Temporal Features - Features computed from the statistical properties of the signal are used. for example auto-correlation, zero-crossing rate.

3. Energy Features - Features referring to the energy content of the signal, for example global energy, harmonic energy, noise energy.

4. Spectral Shape Features - Features computed from the Short Time Fourier Transform (STFT) of the signal, for example spectral centroid, spectral rolloff, kurtosis, spectrogram, Mel spectrogram, Mel Frequency Cepstral Coefficients (MFCCs).

5. Harmonic Features - Features derived from the sinusoidal harmonic modelling of the signal, for example harmonic noise ratio, harmonic derivation.

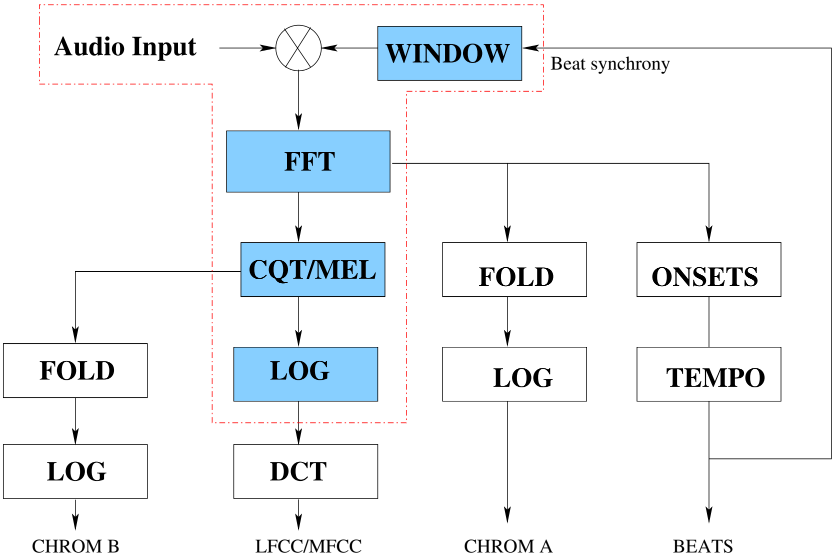

Figure 2.2: Pictorial Representation of Low-Level Audio Features [6] with the two kinds of segmentation, fixed length and beat-synchronous. Coloured blocks in the centre shows the feature used in the thesis.

6. Perceptual Features - Features computed using a model of human hearing or human perception, for example loudness, sharpness, spread.

A pictorial representation of the features is given in Figure 2.2.

2.3.2 Mid-level Features

Another classification of features is given in [19]. Here the features are divided into the aforementioned low-level, mid-level and high-level features. A large variety of mid-level features have been proposed, such as the instrogram, timbregram and chromagram. These attempt to connect the low-level features to the high-level features.

2.3.3 High-level Features

High-level features are representations of music that are not generated by the audio content of the signal. These can also be called as symbolic representation of music. The most common example of this being the score or sheet music, where notes are arranged in a staff notation. This captures musical elements like melody, chords, harmony, key, and time signature. Thus is it is a very commonly used music representation. Guitar tabs, where the notes are arranged in terms of the frets of the guitar is another widely used high-level representation. Another high-level representation used is Musical Instrument Digital Interface (MIDI).

Thus to summarize, instantaneous features describe the audio content of the music, while high-level features represent the various elements of music like melody, harmony, chords, instruments used etc. Mid-level features aim to bridge the gap between these two sets of features, as depicted in Figure 2.3.

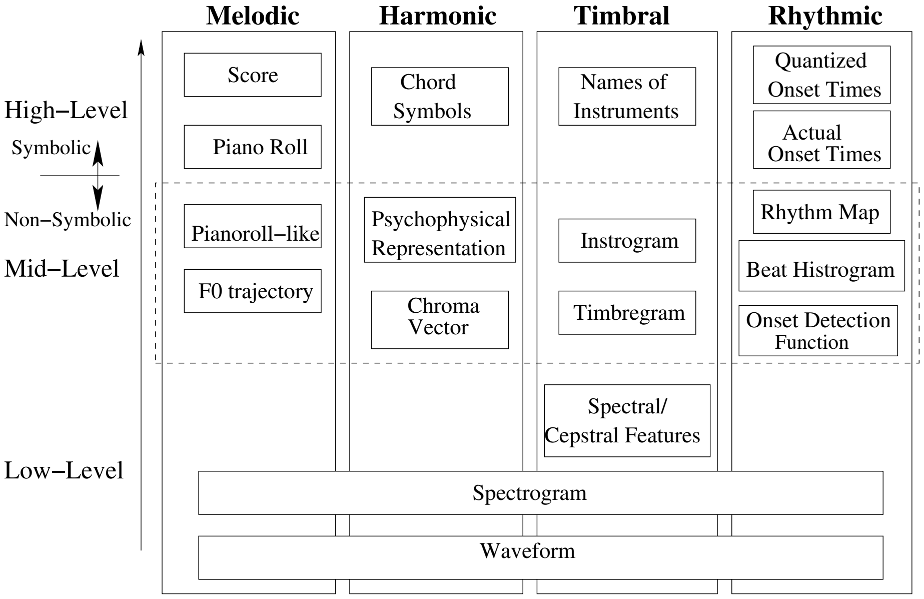

Figure 2.3: Classification of features into low-level, mid-level and high-level features as suggested in [19]. Low-level features are obtained from the audio content, while high-level features are symbolic representation of music.

The main feature used for this thesis is the mel spectrogram. First a description of the spectrogram is provided and then the mel spectrogram is described.

2.3.4 Spectrogram

The simplest among these features is the Short Time Fourier Transform abbreviated as STFT [22]. The STFT forms the basis for other features like spectral centroid, spectral roll-off etc. It can be visualised using a colour plot called *spectrogram*. Let us assume that an audio track can be represented by a discrete time signal $x(t)$, the time-domain representation of the signal. Fixed length segmentation is used, so the input signal is now split into overlapping blocks or frames of equal length by choosing a fixed window time duration T_{win}. The number of samples in the window N_{win} can be calculated by multiplying the sampling rate f_s with T_{win}, so $N_{win} = f_s T_{win}$. The number of overlapping samples is decided with a hop size parameter h. Thus the input signal gets split into N_b overlapping frames, which can be calculated as follows:

$$N_b = \left\lfloor \frac{N - N_{win}}{h} \right\rfloor + 1. \tag{2.1}$$

Hence the nth frame $\mathbf{x}_n$, $1 \leq n \leq N_b$, can be written as a N_{win} dimensional vector as follows:

$$\mathbf{x}_n = \begin{bmatrix} x[(n-1)h+1] \\ x[(n-1)h+2] \\ \vdots \\ x[(n-1)h+N_{win}] \end{bmatrix}. \tag{2.2}$$

If the Discrete Fourier Transform (DFT) of $\mathbf{x}_n$ is calculated, it is assumed that the signal is periodic in each frame. However, due to finite frame length, $\mathbf{x}_n$ is not periodic. Due to this, the spectral energy at a particular frequency gets distributed or leaks to the surrounding frequencies. This is known as spectral leakage. Window functions are multiplying functions applied to each sample in the frame to make the resulting signal periodic, thus reducing spectral leakage. A detailed discussion of different types of window functions is provided in [12]. Hamming window function has been used in this thesis [30]. The Hamming window function for the tth sample in the frame is given as follows:

$$w(t) = \alpha - \beta \cos\left(\frac{2\pi t}{N_{win} - 1}\right). \tag{2.3}$$

where $\alpha = 0.54$ and $\beta = 1 - \alpha$ [12].

The next step is to apply a Short Time Fourier Transform, or STFT to each frame. STFT converts the audio signal in the time domain representation $x(t)$ to the frequency domain representation, with F_b frequency bins and N_b frames. The number of frequency bins can be calculated as:

$$F_b = \begin{cases} \dfrac{n_{fft}+1}{2} & \text{if } n_{fft} \text{ is odd} \\ \\ \dfrac{n_{fft}}{2}+1 & \text{if } n_{fft} \text{ is even} \end{cases}. \tag{2.4}$$

Here n_{fft} refers to the number of points taken for calculating the STFT, usually a power of 2 due to the Fast Fourier Transform (FFT) algorithm.

The STFT for frequency bin b, $b = 1, ..., F_b$ and frame n is given as follows:

$$S_{b,n} = \sum_{t=0}^{N_{win}-1} x_n(t)w(t) \exp\left((-\sqrt{-1})2\pi \frac{b}{N_{win}} t\right). \tag{2.5}$$

The STFT produces a complex number, and magnitude of the complex number $|\mathbf{S}|$ is called the magnitude spectrum. A visualisation with time of song vs frequency being the two axes and the magnitude spectrum $|\mathbf{S}|$ shown in colour is called spectrogram.

Figures 2.4 and 2.5 are spectrograms of 0.5 s of a single tone at 440 Hz, and a guitar song. We can clearly see that at 440 Hz, the spectrogram shows bands which are higher in magnitude. A single tone can be easily visualised in this manner. However, music contains multiple instruments and voice at different frequencies combining with each other across time. Also, the instruments contain overtones, which are integer multiples of the fundamental frequency, and the overtones of several instruments may overlap with each other. In the time domain, the waveform of each instrument has its own attack, sustain and decay time. When the spectrogram of multiple instruments in a song is visualised, the waveforms of the individual instruments will overlap, which is observed in Figure 2.5. Thus drawing any conclusions of the high-level music content like classification of the genre, instrumentation, mood, artist or performance style from just the spectrogram of the song becomes a difficult task.

The spectrogram, thus helps to visualise both temporal and spectral evolution of audio. However, the vertical dimension (number of frequency bins) is in a linear scale, from 0 to $f_s/2$. Since humans do not perceive frequency linearly, a compressed version of the spectrogram would be a better representation. The mel spectrogram is one such representation, which is described below.

2.3.5 Mel Spectrogram

The mel spectrogram is a compressed version of the spectrogram in which the frequencies are ordered in the mel scale. The mel scale is a perceptually motivated scale based on human hearing. Given a frequency f in Hz, the mel value can be calculated as follows [23]:

$$mel(f) = 1127.01048 \log\left(1 + \frac{f}{700}\right). \tag{2.6}$$

As shown in Figure 2.6, the mel scale is approximately linear upto 1 kHz and logarithmic thereafter [27].

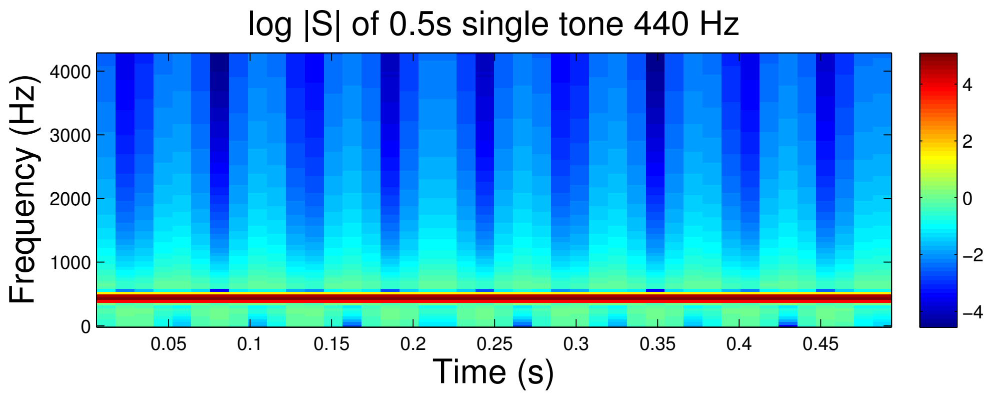

Figure 2.4: Spectrogram of 0.5 s of a single computer generated tone. We can observe that around 440 Hz there are bands of higher magnitude.

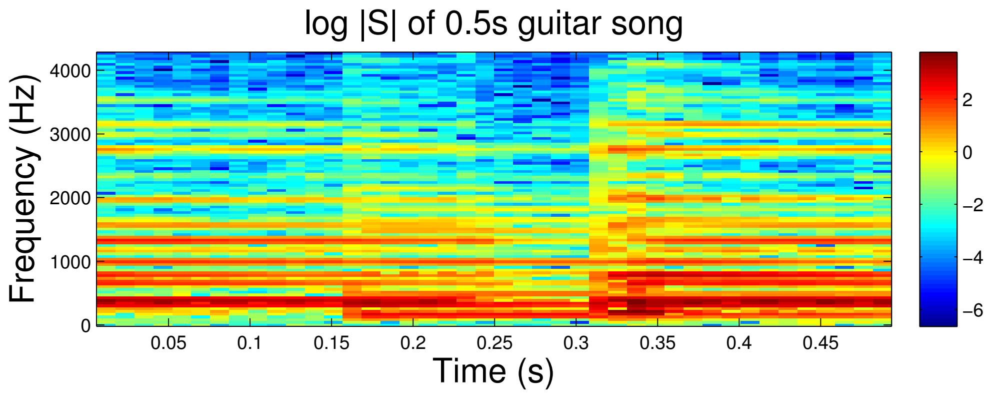

Figure 2.5: Spectrogram of 0.5 s of a guitar song. Note that bands of higher magnitude are observed parallel to each other, these are the overtones.

An overlapping set of n triangular filters is created, such that the maximum weight for each filter reduces with increase in frequency. This can be mathematically written as a mel multiplying matrix $\mathbf{M}$ of dimension $n \times F_b$. The jth element $1 \leq j \leq F_b$ of the ith filter bank can be written as:

$$M_{ij} = \begin{cases} \dfrac{2}{f(i_h) - f(i_l)} \, \dfrac{f(j) - f(i_l)}{f(i_m) - f(i_l)}, & \texttt{if } f(i_l) < f(j) \leq f(i_m) \\ \dfrac{2}{f(i_h) - f(i_l)} \, \dfrac{f(i_h) - f(j)}{f(i_h) - f(i_m)}, & \texttt{if } f(i_m) < f(j) \leq f(i_h) \\ 0 & ,\texttt{otherwise} \end{cases} . \qquad (2.7)$$

$f(i_l)$, $f(i_m)$ and $f(i_h)$ denote the lower, mid and highest frequency of each mel filter bank respectively, and $f(j)$ denotes the lowest frequency of

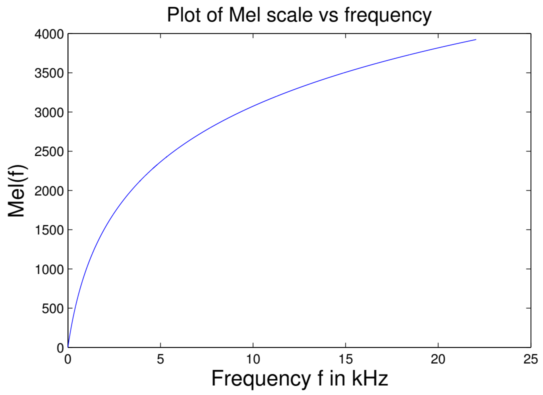

Figure 2.6: The mel scale, used to map the linear frequency scale to a logarithmic one.

each linear frequency bin as obtained through the STFT. The term in the denominator $f(i_h) - f(i_l)$ would increase for higher values of i, since the frequencies are in the mel scale. Hence, higher frequency bands get lower weight compared to lower frequency bands, as illustrated in Figure 2.7. The mel spectrogram is then scaled to the log scale. It can be written in terms of a matrix product as follows:

$$\mathbf{X} = \log\left(\mathbf{M}\ |\mathbf{S}|\right). \tag{2.8}$$

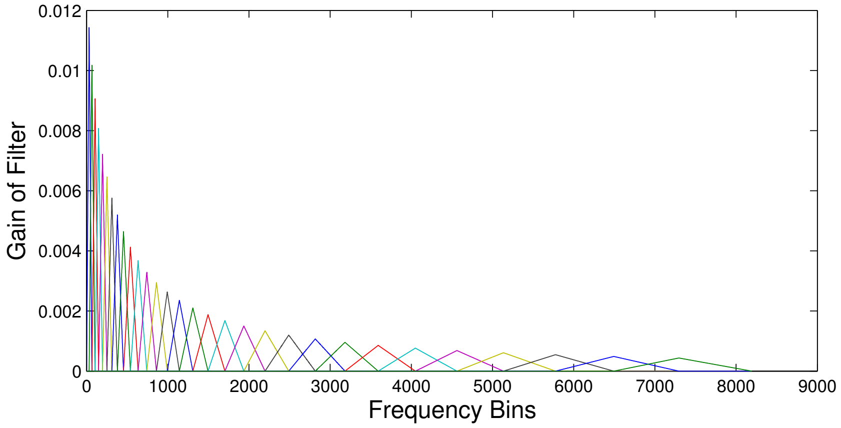

Figure 2.7: 30 Mel Filterbank used in obtaining the mel spectrogram.

A triangular filterbank of $n = 30$ mel filters is applied to the spectrogram to create the mel spectrogram. The use of mel spectrogram is justifed since many speech processing applications use the Mel Frequency Cepstral Coefficients (MFCC), which are obtained through a Discrete Cosine Transform (DCT) of the mel spectrogram. In order to calculate the MFCC, a small number 30 - 40 mel filters are used. Thus, a similar number of 30 mel filters have been used for the thesis.

2.4 Conclusion

Only two features have been described in this thesis, however many more are present in the literature. At the end of feature extraction the size of the mel spectrogram of a 3 minute song would be around 1800 × 30. This is a huge matrix, and if possible should be reduced to lower dimensions. Reduction to lower dimensionality also offers some useful statistical properties among the features. This will be dealt with in the next chapter.

Chapter 3

Feature Transformation

3.1 Introduction

Feature transformation is a process where a dataset in a high-dimensional space, is projected to a lower dimensional space. This transformation from higher dimensional space to lower dimensional space is also called dimensionality reduction.

Chapter 1 briefly mentioned the three main advantages of feature transformation. These are:

1. Reduction in training algorithm complexity - A lower-dimensional dataset requires a training algorithm with smaller number of parameters, thus also reducing the chance of overfitting.

2. Avoiding Curse of Dimensionality - As the number of dimensions increases, the search space of features grows exponentially. This is known as curse of dimensionality.

3. Reduction in memory to store the features, and computational time.

4. Some useful statistical properties like uncorrelatedness, and statistical independence can be observed in the transformed space.

The next section provides a broad overview of feature transformation. Then two feature transformation methods are discussed in detail. This chapter forms the core of the thesis work.

3.1.1 Feature Transformation - An overview

There are different techniques that can be used for dimensionality reduction, depending on application. Both supervised and unsupervised methods are used. Dimensionality reduction can be classified into two categories, feature transformation methods and feature subset selection approaches.

The main difference between subset selection and feature transformation is that in feature subset selection approaches, a subset of the original feature set is obtained. On the other hand, in feature transformation, the original features combine among themselves to form a subspace, with certain desirable properties.

One of the important feature transformation methods is *linear projection*. In this method, a linear combination of the input features in a high dimensional space is projected to a lower dimensional space. Let $\mathbf{X}$ be the input data, with m instances and n features. Each individual data point $\mathbf{x}_i$, $i = 1, ..., m$ is an n-dimensional row vector, and $\mathbf{X}^{(1)} = \begin{bmatrix} x_{11} & x_{21} & \dots & x_{m1} \end{bmatrix}$, $\mathbf{X}^{(2)} = \begin{bmatrix} x_{21} & x_{22} & \dots & x_{m2} \end{bmatrix}$, ..., $\mathbf{X}^{(n)} = \begin{bmatrix} x_{1n} & x_{2n} & \dots & x_{mn} \end{bmatrix}$ denote the original feature space.

Linear projection can be summarized as multiplication of the input data matrix $\mathbf{X}$ with a loading matrix $\mathbf{P}$ as $\mathbf{T} = \mathbf{XP}$. The loading matrix can be considered as a set of basis vectors, which projects the data to a subspace of the input space. The first component of the lower dimensional space $\mathbf{T}^{(1)}$ can be obtained by linearly combining the features $\mathbf{X}^{(l)}, l = 1, ..., n$ with n scalars P_{l1}, so

$$\mathbf{T}^{(1)} = \sum_{l=1}^{n} \mathbf{X}^{(l)} P_{l1}. \tag{3.1}$$

The second dimension $\mathbf{T}^{(2)}$ can be obtained as follows:

$$\mathbf{T}^{(2)} = \sum_{l=1}^{n} \mathbf{X}^{(l)} P_{l2}. \tag{3.2}$$

The above can be repeated for $k \leq n$ dimensions. This transformed representation of the input data is called scores. Reconstruction to the original space is achieved by multiplying the scores with the transpose of the loading matrix. However, reconstruction results in errors if $k < n$. Thus introducing an error matrix $\mathbf{E}$, we can express the reconstruction as follows: $\mathbf{X} = \mathbf{TP}^T + \mathbf{E}$. Note the error matrix $\mathbf{E} = \mathbf{0}$ if $k = n$. This can be pictorically represented in the Figure 3.1.

One of the common linear projection methods is Principal Component Analysis (PCA) [17], which finds a subspace where the features are uncorrelated. Another technique is called whitening, which is also a decorrelating transformation, with an additional property of unit variance for each feature. A stronger assumption than uncorrelatedness is statistical independence, which has been widely used in Independent Component Analysis

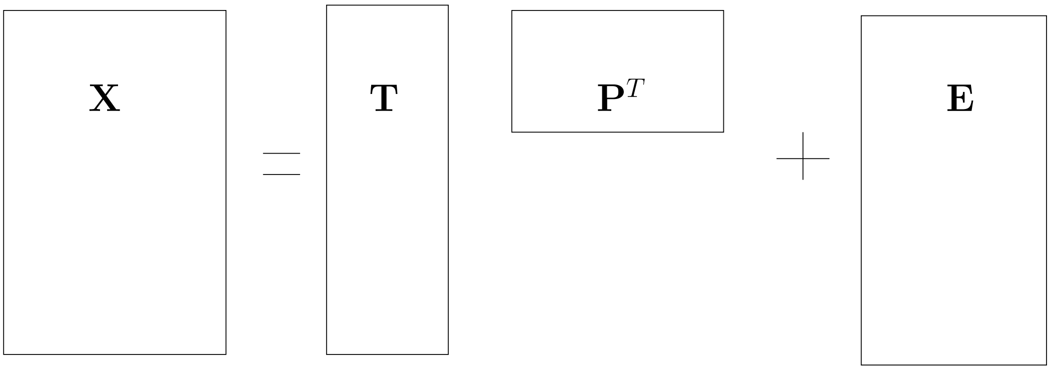

Figure 3.1: Pictorial representation of reconstruction of data matrix $\mathbf{X}$ from scores $\mathbf{T}$ and transpose of loading matrix $\mathbf{P}^T$

(ICA) [15]. ICA finds wide applications in fields such as image denoising and blind source separation. Another linear projection method that preserves distances between datapoints is Metric Multidimensional Scaling (MDS), which is equivalent to PCA.

In the recent years, many non-linear projection methods have also been used for dimensionality reduction [20]. Self-organising maps (SOM), are a good example of a non-linear method which preserves the topology of the input space. Some other examples of non-linear methods are Laplacian Eigenmaps, Locally Linear Embeddings etc.

All these methods mentioned till now are unsupervised, however if the data is provided with label or class information then dimensionality reduction can be performed with the aim to maximise separability between the classes. This forms the basis of Linear Discriminant Analysis (LDA). Thus feature transformation is a wide class of operations that can be used not only for dimensionality reduction, but to obtain other useful properties like uncorrelatedness, independence, or class separability.

The scope of this thesis is Linear Projection methods. Before any of the Linear Projection methods are applied, there are two steps which are taken to further process the data, Mean Centering and Scaling to Unit variance. These are described below.

Mean Centering Mean centering as the term suggests, is to remove the mean of each feature from the original data. Thus the resultant features become zero mean. Mean centering can be expressed as follows:

$$\mathbf{X} := \mathbf{X} - \boldsymbol{\mu}. \tag{3.3}$$

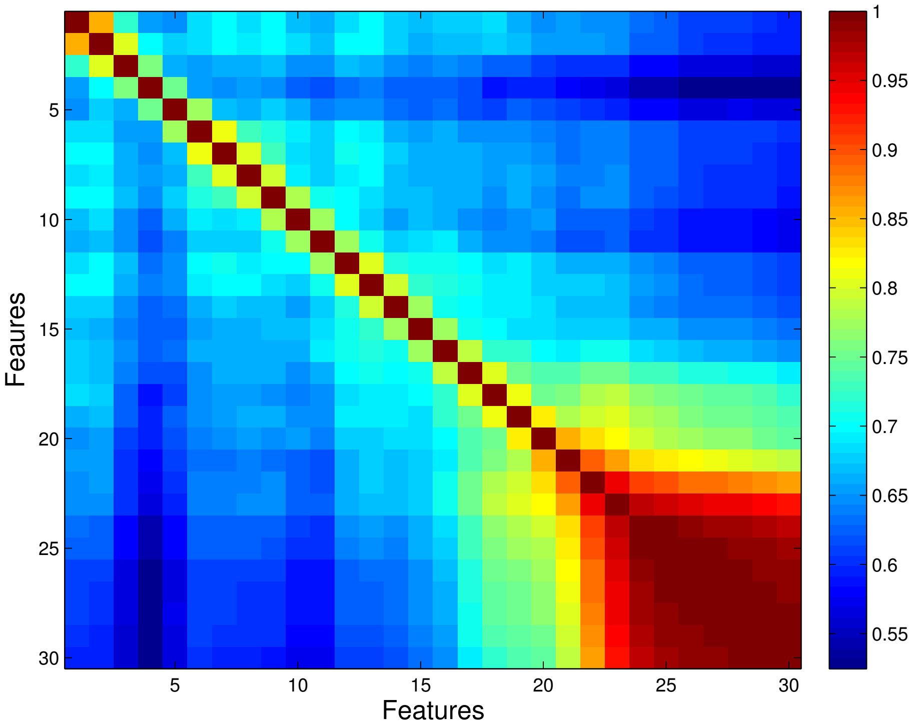

Figure 3.2: Correlation plot of an example song. Higher dimensional features 20-30 are highly correlated.

Scaling to Unit Variance The mean centered data points are divided by their individual standard deviations for each feature. This produces data points with unit variance along each dimension.

$$\mathbf{X} := \mathbf{X}\mathbf{W}_u. \tag{3.4}$$

$\mathbf{W}_u$ is a $n \times n$ digaonal matrix containing the reciprocals of the individual standard deviations $\sigma_i, i = 1, ..., n$.

Next, two linear projection methods for dimensionality reduction, Principal Component Analysis (PCA) and Multidimensional Scaling (MDS) are discussed in detail. The motivation for using PCA is that the music data is highly correlated, especially among the higher dimensional features. A correlation plot of a sample music file is provided in Figure 3.2.

It is hoped that PCA can decorrelate such data. Also we have seen that the Mel Frequency Cepstral Coefficients (MFCC), mentioned in Chapter 2 are obtained by performing a Discrete Cosine Transform (DCT) on the mel spectrogram. Logan has shown in his paper [21] that the DCT is an approximation of PCA. Also in another paper by Hamel et al for annotation and ranking of music [10], PCA has been applied on the mel spectrogram to obtain a set of Principal Mel Spectrum Coefficients (PMSC). Thus, the use of PCA is justified and appropriate.

3.2 Principal Component Analysis

Principal Component Analysis (PCA), also known as Karhunen-Loeve Transform or Proper Orthogonal Decomposition, is a linear projection method where the input data is projected to a lower dimension using the covariance matrix of the data [2]. It is an unsupervised algorithm, so no addtional information regarding the classes or clusters is needed. Let us take the general dataset $\mathbf{X}$, as described in Section 3.1.1. After mean centering it, or scaling it to unit variance, the first step is to estimate the covariance matrix.

$$\mathbf{C} = \frac{1}{m-1}(\mathbf{X}^T\mathbf{X}). \tag{3.5}$$

Here onwards, $\mathbf{X}$ shall denote either mean centered data, or mean centered and scaled data. Then, the next step is to perform eigenvalue decomposition (EVD) of the covariance matrix. This produces a product of an orthogonal matrix $\mathbf{P}$, a diagonal matrix and transpose of the orthogonal matrix $\mathbf{P}$, with the diagonal matrix $\mathbf{D}$ containing the eigenvalues λ_k, $k = 1, ..., n$. The eigenvalues, are infact equal to the variances of different dimensions, and $\mathbf{P}$ contains the eigenvectors arranged in the decreasing order of their eigenvalues, known as the loading matrix.

$$\mathbf{C} = \mathbf{PDP}^T. \tag{3.6}$$

To project into lower dimensional space of $k < n$ components choose k columns from $\mathbf{P}$ thus getting a smaller subspace $\mathbf{P}_k$ and multiply with the mean centered data point, thus the scores $\mathbf{t}_i$, $i = 1, ..., m$, are obtained as follows

$$\mathbf{t}_i = \mathbf{x}_i\mathbf{P}_k. \tag{3.7}$$

Reconstructing $\mathbf{x}_i$ from $\mathbf{t}_i$ is just the reverse process. Here the scores are taken and the transpose of the loading matrix $\mathbf{P}_k$ is used.

$$\hat{\mathbf{x}}_i = \mathbf{t}_i\mathbf{P}_k^T. \tag{3.8}$$

PCA can also be thought of as rotation of the axis of reference, in such a way that the resulting scores $\mathbf{t}_i$ become uncorrelated. A simple example is the 3 dimensional toy data, depicted in Figure 3.3.

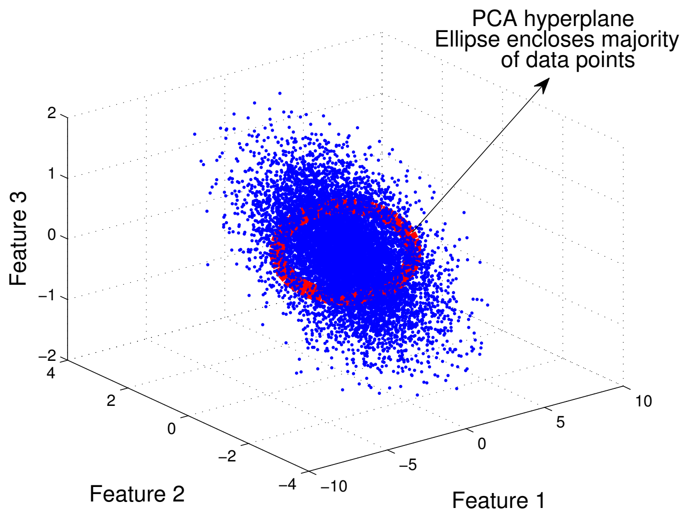

Figure 3.3: Toy Data in 3 dimensions. PCA transforms this data into a 2 dimensional representation shown in Figure 3.4.

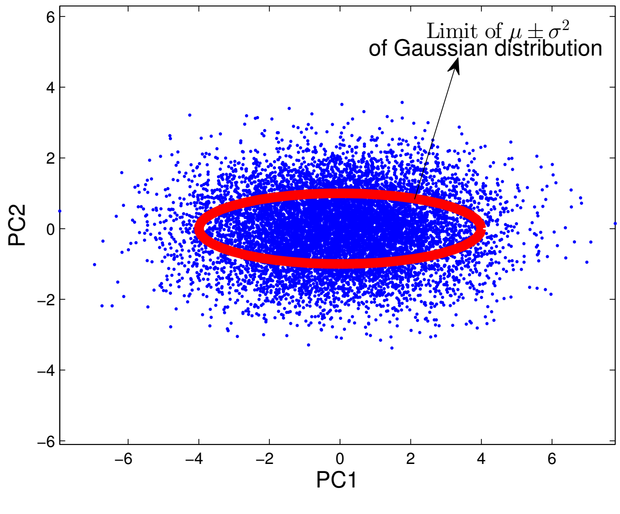

Figure 3.4: Features in Figure 3.3 transformed using PCA into scores in a 2 dimensional space.

The inital frame of reference is the Euclidean coordinate system with basis vectors $\mathbf{e}_1 = \begin{bmatrix}1\\0\\0\end{bmatrix}$, $\mathbf{e}_2 = \begin{bmatrix}0\\1\\0\end{bmatrix}$ and $\mathbf{e}_3 = \begin{bmatrix}0\\0\\1\end{bmatrix}$. Any vector $\mathbf{v}$, corresponding to a data point in this 3-dimensional space can be written as a linear combination of these basis vectors. But it is clear that most of the points are enclosed by the plane in which the ellipse, depicted by red colour in Figure 3.3 lies. This ellipse is, in fact, the boundary of the $\mu \pm \sigma^2$ extent of the normal distribution with unequal variances lying in this plane. Next, the frame of reference is now rotated to obtain a new reference system which lies along the plane of this ellipse. Now the two basis vectors of this new reference system aligned parallel to the principal components (PC) PC1 and PC2 are enough to represent the data, since all the data lies along the plane given by these two basis vectors. Thus a dimensionality reduction from 3 to 2 is achieved, as shown in Figure 3.4, and a 2 dimensional embedding of the 3 dimensional data has been found.

It can be observed from Figure 3.4 that the 2 dimensional scores are actually Gaussian distributed variables with unequal variances along the PCs. If the variance along the second PC can be ignored, then all the data can now be represented by one dimension along PC1, and the new basis vectors are orthogonal to each other. These new basis vectors are the eigenvectors of covariance matrix $\mathbf{C}$. Hence, we observe that by rotation of the axis of reference, we can perform a linear transformation such that the resulting scores are uncorrelated. Thus, PCA corresponds to rotation of axis of reference.

It can also be proved that the objective of PCA is to find directions that maximise the variance. Appendix A provides a proof.

3.2.1 Measures of Fitness

As illustrated in the earlier section, PCA can be used to reduce dimensionality. But how many dimensions is enough? To answer this question, there is a need to develop some measures of fitness for the PCA model. As the number of PCs k increases, the amount of variance explained would also increase. As illustrated in Figure 3.4, with one component, the variance only along PC1 can be captured, with two components however, the entire variance is captured. This leads to the formulation of fraction of explained variance as a measure of fit for the PCA model.

$$\text{Fraction of Explained Variance}_k = \frac{\lambda_k}{\sum_{k=1}^{n} \lambda_k}. \tag{3.9}$$

When $k = n$ the fraction of variance $= 1$. A curve of fraction of explained variance vs k is drawn and a cut-off value for fraction of explained variance is taken. For instance, the number of components which capture say 0.85 or 0.9 of the total variance $\sum_{k=1}^{n} \lambda_k$ is chosen. Plotting a fraction of variance curve for the 3 dimensional toy dataset of Figure 3.3 is not very useful, since the number of dimensions is just 3 and with 2 dimensions all the variance is captured. Hence we take another toy dataset of 10 dimensions. The first 5 dimensions are Gaussian distributed variables, and the next 5 are the square of these variables.

Figure 3.5 shows the plot of fraction of variance for the 10 dimensional toy data described above.

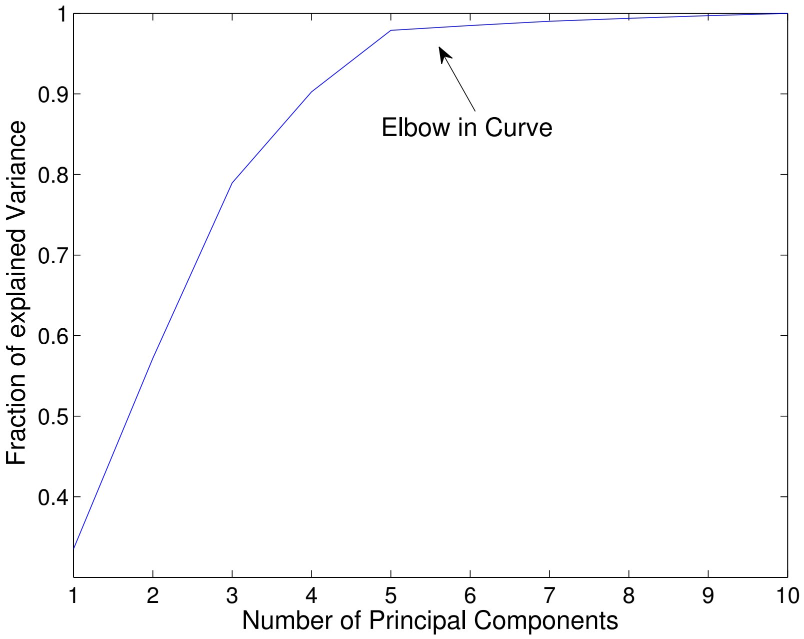

Figure 3.5: Fraction of Variance plot for 10 dimensional toy data. Elbow in the curve is an indicator for choosing appropriate number of PCs.

We can observe an elbow in the curve, which means that adding another component increases the fraction of variance only marginally. Then the number of components which correspond to the elbow in the curve is taken.

Another method to measure PCA fitness is the reconstruction error, defined as the L^2-norm between the reconstructed data points and the original data points.

$$E(k) = \frac{1}{m}\sum_{i=1}^{m} ||\hat{\mathbf{x}}_i - \mathbf{x}_i||_2. \tag{3.10}$$

The reconstruction error is a decreasing curve, and reaches 0 when $k = n$. The reconstruction error for the 10 dimensional toy dataset is shown in Figure 3.6.

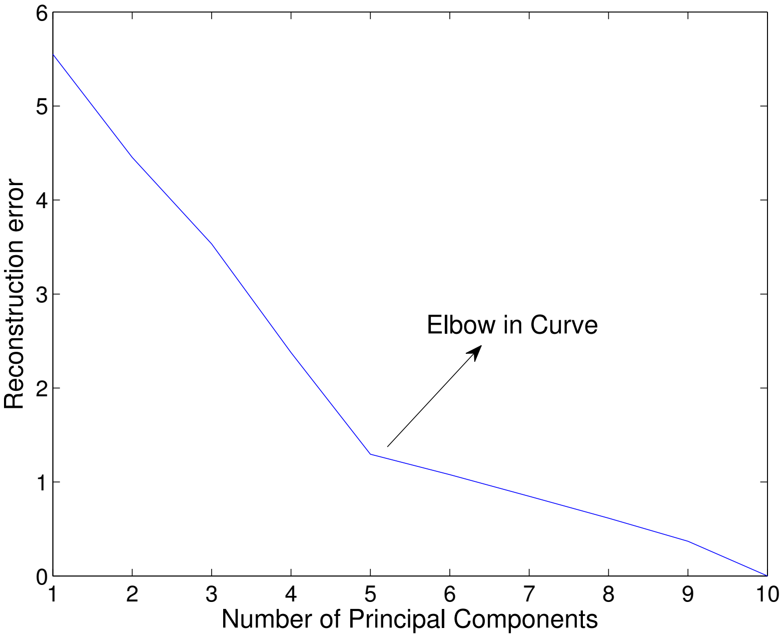

Figure 3.6: Reconstruction error for 10 dimensional toy data. Similar to Figure 3.5, the elbow in the curve is an indicator for choosing appropriate number of PCs.

Similar to the previous figure, there will be an elbow in the curve, where the rate of decrease is marginal. This point is taken as the optimal number of components.

3.2.2 T^2 and Q statistic

The above two measures of fitness are a characteristic of the entire dataset. To characterize each data point and to determine how well the PCA model fits each data point, two measures of fitness with respect to distance of points have been used in the thesis. These are T^2 and Q statistic [17],[18]. T^2 distance measures the distance of a data point score from the origin of the

PC space, while Q distance is the perpendicular distance of a data point from the PCA hyperplane. These are illustrated for the 3 dimensional toy dataset, in Figures 3.7 and 3.8.

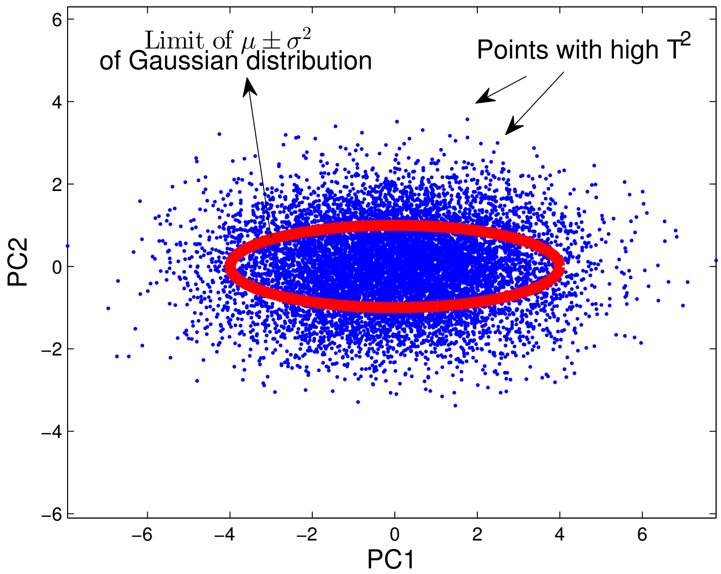

Figure 3.7: Illustration of T^2 distances for 2 dimensional scores. $\mu \pm \sigma^2$ limit of the Gaussian distribution is shown in the red ellipse.

Most of the scores of data points in Figure 3.7 lie close to the origin of the 2 dimensional PC plane, but there are certain points which are far away from the origin, thus showing a large T^2 distance, and variation within the model. In Figure 3.8 there are certain points which lie at high distances above or below the PC plane, and thus have a large Q distance, hence showing large variation outside the model.

The T^2 and Q distance for a PCA model with k components for the ith data point with scores $\mathbf{t}_i$ is given as follows:

$$T^2_{i,k} = \sum_{l=1}^{k} \frac{t^2_{i,l}}{\lambda_l}. \tag{3.11}$$

The sum in the right hand side of Equation (3.11) is called the Mahalanobis distance, since the components of the scores are normalized by their corresponding eigenvalues. The general equation for the Mahalanobis distance for data point $\mathbf{x}_i$ and with score $\mathbf{t}_i$ is given by:

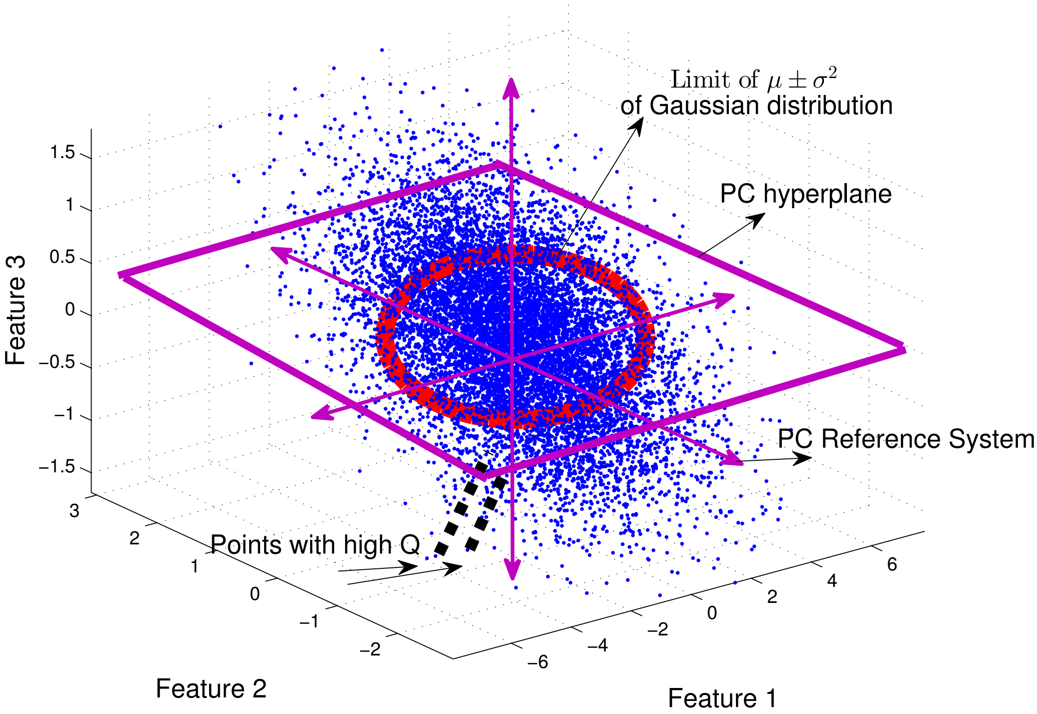

Figure 3.8: Illustration of Q distances for 3 dimensional data points. $\mu \pm \sigma^2$ limit of the Gaussian distribution in the red ellipse, and PC hyperplane are shown.

$$D_M = \mathbf{t}_i \mathbf{D}^{-1} \mathbf{t}_i^T. \tag{3.12}$$

It can be seen that Equation (3.12) is the same formulation as Equation (3.11).

The Q distance for ith datapoint and k PCs is defined as follows:

$$Q_{i,k} = ||\mathbf{x}_i - \boldsymbol{\mu} - \mathbf{t}_i \mathbf{P}_k^T||_2. \tag{3.13}$$

The Q statistic is another name for the reconstruction error that was described in Equation (3.10). The difference being that Q statistic is calculated per data point, while reconstruction error is calculated for the entire dataset.

To detect if a data point has an unusually high value of T^2 or Q, there is a need for a bound on T^2 distance and Q distance. Thus a cut-off distance is defined such that the probability of finding a point beyond this distance is low. In other words, the cut-off distance must ensure that majority of the data points are present at distances less than the cut-off distance. Hubert et al [14] have provided a probabilistic method to estimate the cut-off distances. The cut off on the T^2 distance for k components, $T^2_{lim_k}$ can be calculated by assuming that the squared value $T^2_{i,k}$ of normally distributed scores are approximately k-variate chi-square distributed. Thus the square root of the inverse of the Chi-square distribution for a confidence value of 97.5% can be used to calculate $T^2_{lim_k}$. To calculate the Q distance limit for k components Q_{lim_k}, it is assumed that the squares of the cube roots of the Q distances, $Q^{\frac{2}{3}}_{i,k}$ are normally distributed. Thus the inverse of the normal distribution with mean $\mu(Q^{\frac{2}{3}}_{i,k})$ and standard deviation $\sigma(Q^{\frac{2}{3}}_{i,k})$ with the same probability value of 0.975 is calculated to obtain Q_{lim_k}.

A point could be an outlier due to its T^2 distance, its Q distance or both. Hence, an outlier map can be drawn for all the scores with respect to the ratio $\frac{T^2_{i,k}}{T^2_{lim_k}}$ and $\frac{Q_{i,k}}{Q_{lim_k}}$ for all i,k.

This completes the discussion of PCA. Next, we move to Multidimensional Scaling (MDS).

3.3 Multidimensional Scaling

Multidimensional Scaling (MDS) refers to a family of dimensionality reduction techniques in which the main objective is to preserve pairwise distance among instances [5]. The initial methods which were developed used the Euclidean Distance between data points, this is known as classic metric MDS [32]. It can be proved that classic metric MDS and PCA are equivalent. Please refer to Appendix B for a simple proof. Over the years, many non-linear methods have been used for MDS, the most common method among them being Sammon's mapping [28]. This thesis is restricted to classic metric MDS.

Classic Metric MDS is not a true distance-preserving method since it preserves pairwise scalar products and not the distances. However, pairwise Euclidean distances can be transformed into scalar products and then Classic Metric MDS can be applied. It is a useful technique when the number of features per instance n is very high, for instance $n \geq 10^4$ and the number of instances is much less than the number of features ($m << n$). Calculating the covariance matrix of such a dataset would involve calculating a huge

$n \times n$ matrix, which would be cumbersome and may not fit into memory. So instead of using the covariance matrix, the first step is to calculate the $m \times m$ normalized inner product matrix $\mathbf{I}_p$.

$$\mathbf{I}_p = \frac{1}{m}(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^T. \tag{3.14}$$

Then EVD of $\mathbf{I}_p$ is performed. Thus we get the following equation after EVD similar to Equation (3.6).

$$\mathbf{I}_p = \mathbf{P}\mathbf{D}\mathbf{P}^T. \tag{3.15}$$

A k dimensional representation of the dataset, where $k < m$, is obtained by taking k columns out of m from $\mathbf{D}$, thus obtaining a $m \times k$ smaller matrix $\mathbf{D}_k$. The MDS scores $\mathbf{t}_i$ for the ith data point are obtained as follows:

$$\mathbf{t}_i = \mathbf{p}_i\mathbf{D}_k^{1/2}. \tag{3.16}$$

However, the above equation cannot be used for test data. For test data, we use the equivalance of classic metric MDS and PCA and obtain the eigenvectors of the covariance matrix $\mathbf{C}$ from EVD of $\mathbf{I}_p$, and then use Equation (3.7). This is becuase the eigenvalues of $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}\mathbf{X}^T$ are equal [7], and thus the eigenvectors of $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}\mathbf{X}^T$ are related by a linear transformation. More details regarding this can be found in Appendix B.

Reconstruction of the original data point from the MDS scores is done as follows:

$$\hat{\mathbf{x}_i} = \mathbf{t}_i\mathbf{D}_k^{1/2^T}. \tag{3.17}$$

For reconstruction of test data, first eigenvectors of $\mathbf{C}$ are calculated from EVD of $\mathbf{I}_p$ and then Equation (3.8) is used.

To calculate an error measure for MDS, the reconstruction is calculated according to equation (3.17) and the reconstruction error can be calculated as illustrated in Equation (3.10). Also, the T^2 and Q statistics that were defined in Section 3.2.2 can be similarly defined for MDS.

In the next section, the music dataset and the feature transformation methods used on the dataset are described.

3.4 Multiway Data for Music Collection

As the previous chapter demonstrated, each song is represented by its mel spectrogram. Thus a collection of songs would be represented by their individual mel spectrograms. Since the length of each song is different, a 3-way

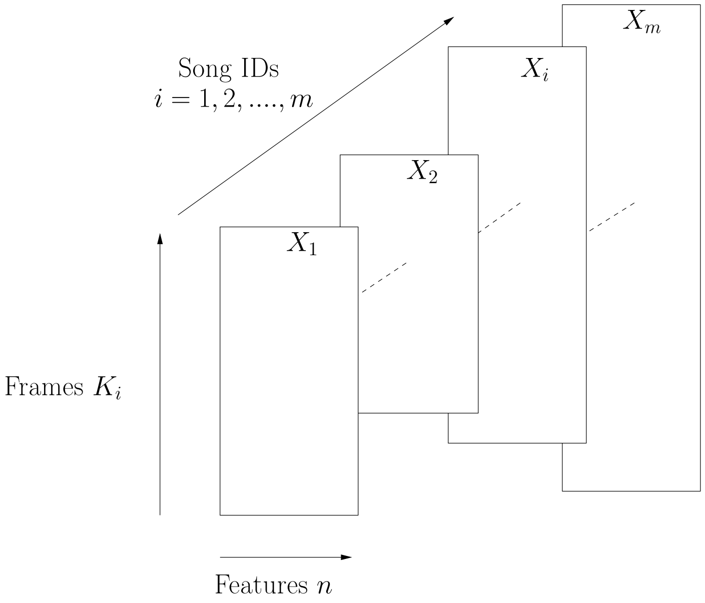

Figure 3.9: 3 way Tensor Representation of Music Collection $\mathbb{X}$. Each song has K_i frames and n features.

tensor data $\mathbb{X}$ for the music collection is obtained, which is represented in Figure 3.9.

Hence, each audio track or song, gets represented by a data matrix $\mathbf{X}_i$, $i = 1, ..., m$ of K_i frames and n features. To decompose this 3-way tensor, two linear projection methods have been used, Multiway Principal Component Analysis (Multiway PCA) and Multilevel Simultaneous Component Analysis (MLSCA).

3.5 Multiway PCA

Multiway PCA is a generalization of Principal Components Analysis to multiway arrays. Multiway data occurs where there is a set of batches or iterations, and in each batch there is a multiway matrix of variables. In the case of 3-way data, there is a set of batches and a set of 1-way variables varying across time. The decomposition of 3-way data is done by allowing the batches direction to be expressed as vectors and the other directions by a 2-way array, and the Kronecker product between the two is calculated. Thus the 3-way music collection $\mathbb{X}$ can be decomposed for k components as follows [37], [36]:

$$\mathbb{X} = \sum_{i=i}^{k} \mathbf{t}_i \otimes \mathbf{P}_i + \mathbb{E}. \tag{3.18}$$

$\mathbf{t}_i$ denotes the score vector and $\mathbf{P}_i$ denote the loading matrices for each component respectively. Note the presence of Kronecker product, thus making it a sum of k 3-way arrays and a residual $\mathbb{E}$. It can be shown that the decomposition given in Equation (3.18) can be converted to a 2-way PCA by unfolding.

3.5.1 Unfolding for 3-way PCA

As demonstrated in Section 3.4, the input data is not 2-way matrix, but a 3-way tensor with one direction having unequal number of dimensions per song, the frames direction. For regular PCA to be applied, this 3-way tensor must be unfolded. Unfolding is the process where a 3-way tensor is collapsed into a 2-way data matrix. This unfolding method has been described in detail for a different application in Timo Honkela's Grounded Intersubjective Concept Analysis (GICA) model [13]. Since there are 3 directions to view the tensor, unfolding could be done in 3 ways. However, for this dataset, there are only 2 unfoldings of interest and practicality. The first type of unfolding is along the songs direction, and the second one along the frames direction. Both methods are described below.

3.5.1.1 Unfolding along songs

Here the dataset $\mathbb{X}$ is unfolded in such a way that the data matrix of each song is placed beneath the other, thus getting a matrix $\mathbf{X}_{unf.songs}$ of dimension $K \times n$, K denotes the sum of all frames of all the songs in the collection. Thus, $K = \Sigma_{i=1}^{m} K_i$. Figure 3.10 provides a pictorial representation of the same.

In this case, regular PCA algorithm as described in Section 3.2, can be used. Thus for each song matrix $\mathbf{X}_i$ a song wise score representation $\mathbf{T}_{i,k}$ with k PCs is obtained, it is thus the Equation (3.7) applied to a matrix.

$$\mathbf{T}_{i,k} = \mathbf{X}_i \mathbf{P}_k. \tag{3.19}$$

The reconstruction of the song is obtained similar to Equation (3.8).

$$\hat{\mathbf{X}}_i = \mathbf{T}_{i,k} \mathbf{P}'_k. \tag{3.20}$$

The calculation of the reconstruction error, however undergoes a slight modification due to the presence of song matrices. The Average Root Mean

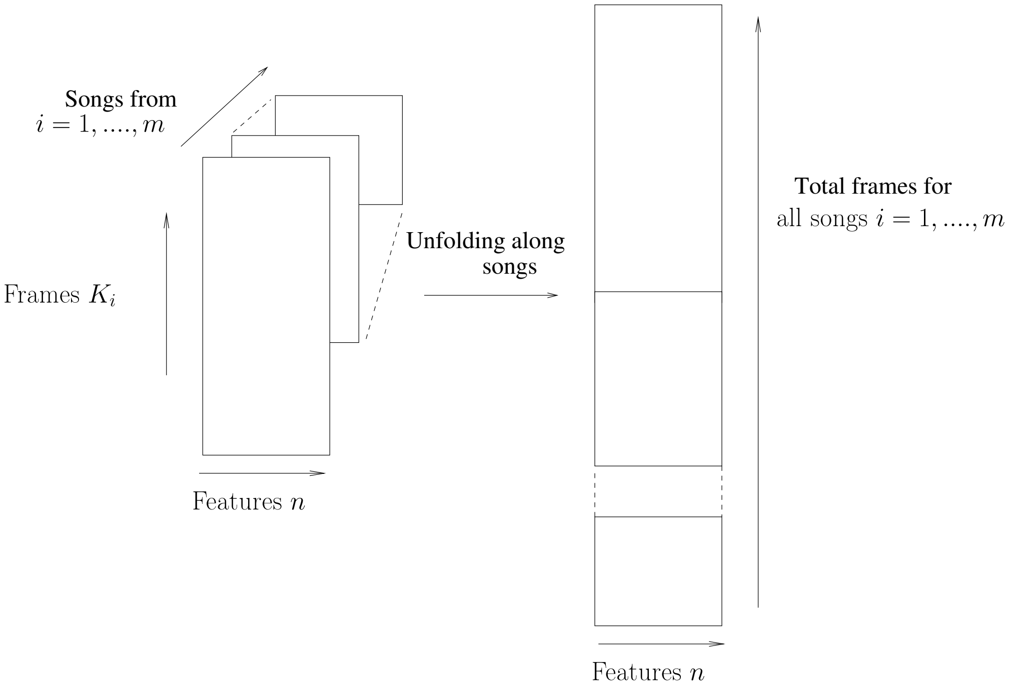

Figure 3.10: Dataset $\mathbb{X}$ unfolded along songs. Each song matrix $\mathbf{X}_i$ is placed beneath other song matrices to produce a $\sum_{i=1}^{m} K_i \times n$ matrix $\mathbf{X}_{unf.songs}$.

Square (RMS) Error measure is used. This error is defined as the RMS error taken across all frames, and then averaged by the number of features. Mathematically,

$$\mu RMS(i) = \frac{1}{n} \sum_{j=1}^{n} \sqrt{\frac{\Sigma_{k=1}^{K_i} (\mathbf{X}_{i,j,k} - \hat{\mathbf{X}}_{i,j,k})^2}{K_i}}. \tag{3.21}$$

Another error measure which could be used is the Frobenius norm between $\mathbf{X}_i$ and $\hat{\mathbf{X}}_i$.

Thus, unfolding along songs helps to characterise the framewise evolution of each song and build a PCA model for all the frames in all the songs. No data is truncated before applying the PCA model. Since each song is represented by its own data matrix $\mathbf{X}_i$, measuring distances or similarities betweeen two songs cannot be done by using the PCA scores, since the matrices are not of the same size. One possibility is to calculate a mean vector for each song and use a distance measure, such as the Euclidean distance, between the two mean vectors. However, since the mean is highly sensitive to outliers, this method would not give satisfactory results. Thus probabilistic methods have to be used to characterise the frame-wise behaviour of each song, in order to build a global song model. One such probabilistic method based upon Gaussian Mixture Models of MFCCs has already been developed by Accourtier and Pachet [3], and more improvements have been made by subsequent researchers, yielding promising results. Thus applying a global song model on the PCA scores looks to be an interesting work for the future.

3.5.1.2 Unfolding along frames

In this method, the dataset $\mathbb{X}$ is unfolded along the frames direction. This is achieved by concatenating all the frames of a particular song adjacent to each other. This would mean that in the resultant dataset matrix, every row vector corresponds to a single song. Hence it is a convenient approach to represent each song as a single feature vector and not as a collection of frames. For tasks like song similarity the distance between the individual feature vectors can be calculated. However, this method suffers from a weakness. Since all the songs are not of the same length, the number of features in this new dataset is different per instance. Thus, thresholding needs to be performed. Again, thresholding can be done in a variety of ways, thresholding upto the length of the shortest song.

1. Start from the beginning of each song,
2. Start from the end of each song and go backward, cut off at threshold point described above, or
3. Cut off from the middle of each song.

The unfolding has been pictorially described in Figure 3.11.

Thresholding method 1 has been used in this thesis, in order to obtain a matrix $\mathbf{X}_{unf.frames}$ of dimension $m \times nK_{sh}$, where K_{sh} denotes the number of frames of the shortest song. Usually, the order of magnitude of $K_{sh} = 10^3$. Then the order of magnitude of the number of features nK_{sh} would be 10^4, which is an extremely high value. Also, the number of features $nK_{sh} >>$ number of instances m. If classic PCA as described in Section 3.2 is applied, a covariance matrix of order of $10^4 \times 10^4$ would have to be calculated. Since this calculation would be cumbersome, PCA is not applied, instead MDS as described in Section 3.3 is used. This results in a dimensionality reduction where number of reducible dimensions k is in the range $1 \leq k \leq m$. It differs from the classic PCA case where $1 \leq k \leq n$.

The scores for a particular song i, can be defined similar to Equation (3.16) and the reconstruction can be performed similar to Equation (3.17).

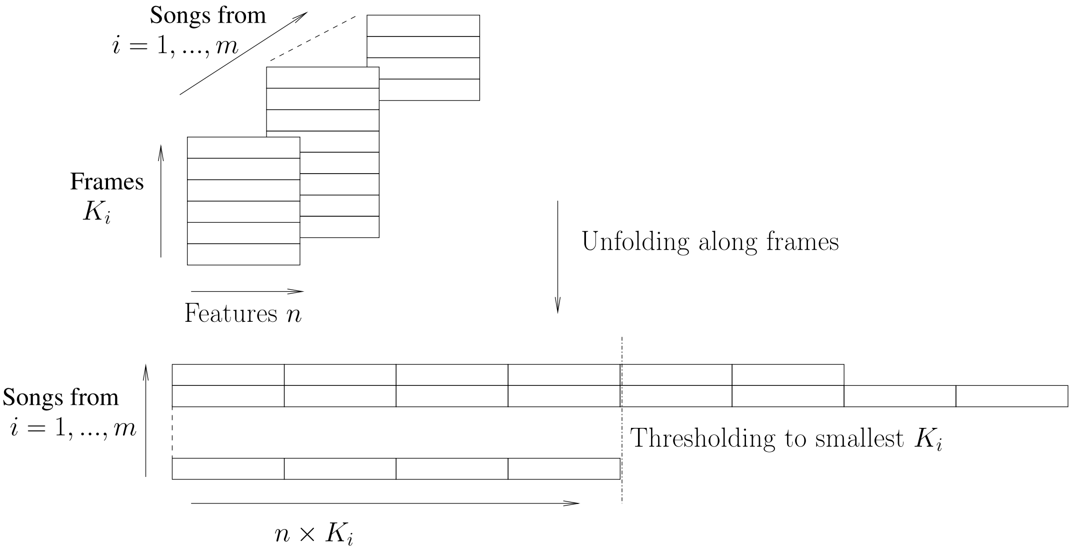

Figure 3.11: Unfolding Along frames. Each song is represented by a single feature vector. Then thresholding to the shortest number of frames K_{sh} is done, obtaining a $m \times nK_{sh}$ data matrix $\mathbf{X}_{unf.frames}$.

For test data, we use the equivalence between classic metric MDS and PCA as described earlier.

$$\mathbf{t}_i = \mathbf{p}_i \mathbf{D}_k^{1/2}. \tag{3.22}$$

$$\hat{\mathbf{x}}_i = \mathbf{t}_i \mathbf{D}_k^{1/2^T}. \tag{3.23}$$

3.6 Multilevel Simultaneous Component Analysis

The main weakness in Multiway PCA described above is that the 3-way data needs to be unfolded. Instead of trying to develop a model for the entire song collection, a model for each song can be developed. This provides the basis for Multilevel Simultaneous Component Analysis (MLSCA). In this method, a model is developed for each song as an additive combination of a global mean, between song components and within song components. It has been used in chemometrics to analyse process or chemical experiment data [8]. A rigorous mathematical understanding of simultaneous component analysis is provided in [31].

Every song $\mathbf{X}_i$, each of K_i frames and n features uses the following model:

$$\mathbf{X}_i = \mathbf{1}_{K_i}\mathbf{m} + \mathbf{1}_{K_i}\mathbf{t}_{b,i}\mathbf{P}_b^T + \mathbf{T}_{w,i}\mathbf{P}_w^T + \mathbf{E}_i. \tag{3.24}$$

The right hand side of Equation (3.24) has three addition terms and a reconstruction error or residual term $\mathbf{E}_i$. The main objective of MLSCA is to minimize the reconstruction error in the least square sense over all songs in the training set. In other words, the model parameters are calculated in such a way that the sum of squares of the L^2-norm of the reconstruction error over all m songs $\sum_{i=1}^{m} ||\mathbf{E}_i||_2^2$ is minimized. Thus a set of model parameters is obtained as shown below:

$$F(\mathbf{m}, \mathbf{t}_{b,i}, \mathbf{P}_b, \mathbf{T}_{w,i}, \mathbf{P}_w) = \sum_{i=1}^{m} ||\mathbf{X}_i - \mathbf{1}_{K_i}\mathbf{m} - \mathbf{1}_{K_i}\mathbf{t}_{b,i}\mathbf{P}_b^T - \mathbf{T}_{w,i}\mathbf{P}_w^T||_2^2. \tag{3.25}$$

subject to constraints $\Sigma_{i=1}^{m} K_i\mathbf{t}_{b,i} = \mathbf{0}$ and $\mathbf{1}'_{K_i}\mathbf{T}_{w,i} = \mathbf{0}$, for $i = 1, ..., m$. The three additive terms on the RHS of Equation (3.24) from left to right are the global mean, between song part, and within song part respectively. Mathematically it can be proven that these three parts of the MLSCA model can be solved separately [31]. Next, each part of the model is described in detail below.

Global Mean The global mean $\mathbf{m}$ is calculated by unfolding the dataset $\mathbb{X}$ along the songs direction as described in Section 3.5.1.1 and then calculating the mean vector of the resultant unfolded matrix $\mathbf{X}_{unf.songs}$. $\mathbf{1}_{K_i}$ in Equation (3.24) denotes a vector of ones as long as the number of frames in the current song K_i. Thus, the first addition term consists of the global mean repeated as many times as there are frames in the current song K_i.

Between Songs Part To calculate the second part or the between song part of the model, the mean of each song is calculated, and the mean vectors of all songs $\mathbf{x}_{b,i}$, $i = 1, ..., m$ are stacked together, obtaining a $m \times n$ matrix $\mathbf{X}_b$ of means. Next, a diagonal matrix $\mathbf{W}_b$ is constructed, such that $\mathbf{W}_{b_{ii}} = \sqrt{K_i}$. Then Singular Value Decomposition (SVD) is performed on the matrix $\mathbf{W}_b\mathbf{X}_b$.

$$\mathbf{W}_b\mathbf{X}_b = \mathbf{U}_b\mathbf{S}_b\mathbf{V}_b^T. \tag{3.26}$$

Here $\mathbf{U}_b$ denotes the left singular matrix, $\mathbf{S}_b$ a diagonal matrix of singular values and $\mathbf{V}_b$, the right singular matrix. In order to perform feature transformation, the right singular vectors corresponding to the first $R_b \leq n$ singular values arranged in decreasing order are taken. Thus the first R_b

columns of $\mathbf{V}_b$ and $\mathbf{U}_b$ producing smaller submatrices $\mathbf{V}_{R_b}$ and $\mathbf{U}_{R_b}$, and an $R_b \times R_b$ subset of $\mathbf{S}_b$, $\mathbf{S}_{R_b}$ are required.

The betweeen song scores $\mathbf{T}_b$ of all songs and between song loadings $\mathbf{P}_b$ are calculated as follows [8]. Note that $\mathbf{t}_{b,i}$, the between song scores for the ith song, is the ith row vector of $\mathbf{T}_b$.

$$\mathbf{T}_b = \mathbf{W}_b^{-1}\mathbf{U}_{R_b}. \tag{3.27}$$

$$\mathbf{P}_b = \mathbf{V}_{R_b}\mathbf{S}_{R_b}^T. \tag{3.28}$$

However, the betweeen song scores in Equation (3.27) needs to be expressed in terms of the right singular matrix $\mathbf{V}_{R_b}$ and not the left singular matrix $\mathbf{U}_{R_b}$, so that it can be generalized to new test data points. This is because the number of rows in $\mathbf{U}_{R_b} = m$, but $\mathbf{V}_{R_b}$ is of dimension $n \times R_b$, independent of number of songs m. Using some matrix manipulations, this can be achieved. Proceeding from Equation (3.26) but replacing the matrices on RHS with R_b components,

$$\mathbf{W}_b\mathbf{X}_b = \mathbf{U}_{R_b}\mathbf{S}_{R_b}\mathbf{V}_{R_b}^T. \tag{3.29}$$

Multiplying both sides by $\mathbf{V}_{R_b}$, it can be further simplified as:

$$\mathbf{W}_b\mathbf{X}_b\mathbf{V}_{R_b} = \mathbf{U}_{R_b}\mathbf{S}_{R_b}. \tag{3.30}$$

Note: $\mathbf{V}_{R_b}^T\mathbf{V}_{R_b} = \mathbf{I}$, the identity matrix. This is becuase since $\mathbf{V}_b$ is orthogonal, the columns of $\mathbf{V}_{R_b}$ are orthogonal to each other.

Since $\mathbf{S}_{R_b}$ is a $R_b \times R_b$ diagonal matrix, we have

$$\mathbf{S}_{R_b}^{-1}\mathbf{S}_{R_b} = \mathbf{S}_{R_b}\mathbf{S}_{R_b}^{-1} = \mathbf{I}. \tag{3.31}$$

Thus, multiplying both sides of Equation (3.30) by $\mathbf{S}_{R_b}^{-1}$, we have

$$\mathbf{W}_b\mathbf{X}_b\mathbf{V}_{R_b}\mathbf{S}_{R_b}^{-1} = \mathbf{U}_{R_b}\mathbf{S}_{R_b}\mathbf{S}_{R_b}^{-1}. \tag{3.32}$$

Using the result of Equation (3.31) and substituing that in the RHS, we get

$$\mathbf{U}_{R_b} = \mathbf{W}_b\mathbf{X}_b\mathbf{V}_{R_b}\mathbf{S}_{R_b}^{-1}. \tag{3.33}$$

Mutiplying on the LHS of the terms by $\mathbf{W}_b^{-1}$, we obtain

$$\mathbf{W}_b^{-1}\mathbf{U}_{R_b} = \mathbf{X}_b\mathbf{V}_{R_b}\mathbf{S}_{R_b}^{-1}. \tag{3.34}$$

The LHS of the Equation above is the between songs scores as given in Equation (3.27). So finally, the between songs scores and loadings can be expressed as follows:

$$\mathbf{T}_b = \mathbf{W}_b^{-1}\mathbf{U}_{R_b} = \mathbf{X}_b\mathbf{V}_{R_b}\mathbf{S}_{R_b}^{-1}. \tag{3.35}$$

$$\mathbf{t}_{b,i} = \mathbf{x}_{b,i}\mathbf{V}_{R_b}\mathbf{S}_{R_b}^{-1}. \tag{3.36}$$

$$\mathbf{P}_b = \mathbf{V}_{R_b}\mathbf{S}_{R_b}^{T}. \tag{3.37}$$

In other words, the between song scores are obtained by multiplying the mean vector for the ith song $\mathbf{x}_{b,i}$ by the loading matrix $\mathbf{V}_{R_b}$, and then scaling down by the singular values present in $\mathbf{S}_{R_b}$.

To reconstruct the mean vector of the song data matrix multiply the between song scores $\mathbf{t}_{b,i}$ with the transpose of the between songs loading matrix $\mathbf{P}_b$. Then the reconstructed mean vector is replicated for K_i frames. This corresponds to the second additive term of Equation (3.24). The between songs part of the model tries to model the inter song variations and is controlled by the number of between song components R_b. Higher is the value of R_b, better is the reconstruction.

Within Songs Part The third additive part of Equation (3.24) is the within song part, and tries to model the intra song data. Each song is now centered around its own mean $\mathbf{x}_{b,i}$ (calculated in the previous part), and then all the mean centered versions of each song is unfolded along the songs direction. This is similar to $\mathbf{X}_{unf.songs}$, but here the song is centered around its own mean. So, $\mathbf{X}_{w,i} = \mathbf{X}_i - \mathbf{1}_{K_i}\mathbf{x}_{b,i}$ for $i = 1, ..., m$, is obtained and then all the $\mathbf{X}_{w,i}$ are stacked together, giving the $K \times n$ within song matrix $\mathbf{X}_w$. Next, SVD is performed on $\mathbf{X}_w$.

$$\mathbf{X}_w = \mathbf{U}_w\mathbf{S}_w\mathbf{V}_w^T. \tag{3.38}$$

Similar to the between songs part, the number of within song components $R_w \leq n$ are chosen, to get smaller submatrices of the right singular orthogonal matrix $\mathbf{V}_w$ and diagonal matrix $\mathbf{S}_w$, taking the first R_w singular values. This orthogonal loading matrix $\mathbf{V}_{R_w}$, is used to project the within song data. Thus the within song scores for the ith song $\mathbf{T}_{w,i}$ and loadings $\mathbf{P}_w$ are obtained as follows:

$$\mathbf{T}_{w,i} = \mathbf{X}_{w,i}\mathbf{V}_{R_w}. \tag{3.39}$$

$$\mathbf{P}_w = \mathbf{V}_{R_w}. \tag{3.40}$$

To reconstruct the within song data, the transpose of $\mathbf{P}_w$ is taken and multiplied with $\mathbf{T}_{w,i}$, as depicted in third part of Equation (3.24).

The three parts of the model are now added to calculate the reconstructed song $\hat{\mathbf{X}}_i = \mathbf{1}_{K_i}\mathbf{m} + \mathbf{1}_{K_i}\mathbf{t}_{b,i}\mathbf{P}_b^T + \mathbf{T}_{w,i}\mathbf{P}_w^T$.

3.6.1 Measures of Fitness

Since MLSCA returns a reconstruction for each song, the average RMS error for each song as introduced in Equation (3.21) can be calculated, and a plot of the mean average RMS error over the entire dataset can be drawn. The average RMS error plot is now a surface, since there are two parameters R_b and R_w controlling the amount of reconstruction. The proportion of explained variance for the singular values in the between part and the within part can be plotted separately, to choose appropriate values for R_b and R_w. Details of the validation part and selecting appropriate number of components is provided in the next chapter.

3.6.2 T^2 and Q statistic

The T^2 and Q statistics can be defined similar to Section 3.2.2. However, it should be noted that unlike PCA or MDS, there are two constituent submodels in MLSCA, the between song part and the within song part. Each has its own scores and loadings, and thus T^2 and Q statistics have to be developed for both between song hyperplane and within song hyperplane separately.

3.6.2.1 Between song hyperplane

In the between song part of Section 3.6, we had seen the between song scores and loadings defined according to the paper [8]. But for the between song T^2 calculation, we do not use these scores. Instead, we use the SVD defined scores which do not involve scaling down by singular values unlike Equation (3.36). We use a new set of scores $\mathbf{t}_{S_{b,i}}$, defined as follows:

$$\mathbf{t}_{S_{b,i}} = \mathbf{w}_{b,i}\mathbf{X}_b\mathbf{V}_{R_b}. \tag{3.41}$$

Here the term $\mathbf{w}_{b,i}\mathbf{X}_b$ is the same as scaling up each component of the mean song vector $\mathbf{x}_{b,i}$ by $\sqrt{K_i}$. Then similar to Equation (3.11), we can define the between song T^2 distance. But we also require the eigenvalues of the covariance matrix of $\mathbf{W}_b\mathbf{X}_b$. How do we obtain eigenvalues, when we only have the singular values present in $\mathbf{S}_{R_b}$? For this, we need to utilize the connection between eigenvalues and singular values.

Given any matrix $\mathbf{X}$, it can be proved that non-zero singular values of $\mathbf{X}$ are the square roots of the non-zero eigenvalues of outer product matrix $\mathbf{X}^T\mathbf{X}$

and inner product matrix $\mathbf{X}\mathbf{X}^T$, provided $\mathbf{X}$ is mean centered. Appendix B provides a simple proof.

Also, when $\mathbf{X}$ is mean centered, the outer product matrix is just a scaled version of the covariance matrix, the scaling factor being $\frac{1}{m-1}$. So, we mean center the matrix $\mathbf{W}_b\mathbf{X}_b$, calculate the SVD of the mean centered matrix, and store the vector of singular values $\boldsymbol{\sigma}_{b_{mc}}$. Then the eigenvalues of the covariance matrix of $\mathbf{W}_b\mathbf{X}_b$ can be calculated as:

$$\boldsymbol{\lambda}_b = \frac{1}{m-1}\text{diag}(\boldsymbol{\sigma}_{b_{mc}})\boldsymbol{\sigma}_{b_{mc}}. \tag{3.42}$$

The expression on the RHS of the previous equation refers to squaring each element of the vector $\boldsymbol{\sigma}_{b_{mc}}$. Now, once the eigenvalues are obtained, we can define the between song T^2 distance for ith song and R_b retained between song components, similar to Equation (3.11).

$$T^2_{b,i,R_b} = \sum_{l=1}^{R_b} \frac{t^2_{b,i,l}}{\lambda_{b,l}} \tag{3.43}$$

where $t_{b,i,l}$ and $\lambda_{b,l}$ are the lth elements in $\mathbf{t}_{b,i}$ and $\boldsymbol{\lambda}_b$ respectively.

The between song Q distance for the ith song can also defined similar to Equation (3.13). It can be defined as follows for R_b retained between song components. Here we use the between song scores and loadings as defined in Equations (3.36).

$$Q_{b,i,R_b} = ||\mathbf{w}_{b,i}\mathbf{X}_b - \mathbf{t}_{b,i}\mathbf{P}_b^T||_2. \tag{3.44}$$

Thus, the between song hyperplane characterizes each song with a T^2 distance and a Q distance. An outlier map can be drawn for each song based on their T^2_{b,i,R_b} and Q_{b,i,R_b}. Hence it is similar to the unfolding along frames described in Section 3.5.1.2.

3.6.2.2 Within song hyperplane

For the within song hyperplane, the scores and loadings as defined in Equations (3.39) and (3.40) are used. However, unlike the previous section where a T^2 and Q distance is obtained per song, a T^2 and Q distance is obtained per frame of the song. The eigenvalues of the covariance matrix of within songs matrix $\mathbf{X}_w$ are needed for computing the T^2 distance, these can be calculated in a similar manner as was calculated for the between song hyperplane. Then the T^2 and Q distance for the jth frame in the ith song can be defined as follows for R_w retained within song components:

$$T^2_{w,j,i,R_w} = \sum_{l=1}^{R_w} \frac{t^2_{w,j,i,l}}{\lambda_{w,l}}. \tag{3.45}$$

$$Q_{w,j,i,R_w} = ||\mathbf{x}_{w,j,i} - \mathbf{t}_{w,j,i}\mathbf{P}^T_w||. \tag{3.46}$$

where $t_{w,j,i,l}$ denotes the lth component of the within song score vector for jth frame in ith song $\mathbf{t}_{w,j,i}$ and $\lambda_{w,l}$, $l = 1, ..., R_w$ are the eigenvalues of the covariance matrix of $\mathbf{X}_w$.

Thus, the within song hyperplane characterises each song with a set of T^2 and Q distances. A limit can be placed on each distance measure and we can calculate for each song the number of frames exceeding the T^2 and Q limits. Thus, it is similar to the unfolding along songs described in Section 3.5.1.1.

Thus, it is evident that MLSCA is a model that tries to capture both between song variations and within song variations. The inter song variations are modelled by the between song sub-model, which is the second addition term of Equation (3.24), where the mean vector of each song is reconstructed. This portion is similar to unfolding along frames. The intra song variations are modelled by the within song sub-model, represented in the third addition term of Equation (3.24), where the song mean centered around its own mean is modelled. This replicates the unfolding along songs.

Thus, MLSCA is a composite model that tries to capture both the unfoldings of multiway PCA as described in section 3.5.1.

3.7 Conclusion

In this chapter, a brief overview of the methods for dimensionality reduction used in the thesis was provided. Then the methods Multiway PCA, and MLSCA were discussed in detail. The next chapter discusses the experimental setup to test these methods.

Chapter 4

Experiments

4.1 Introduction

In the previous Chapter 3 we had seen how we can apply multiway techniques for dimensionality reduction. Some measures of fitness were described in Section 3.2.1, to choose the number of components. In this chapter, we describe the process to select the appropriate number of components using a separate validation dataset. There are several validation techniques available like Hold out Cross Validation, k-fold Cross Validation, Leave One Out Cross Validation etc. Due to the large number of data points, we chose to do Hold out Cross Validation. This is explained in Section 4.4 and Section 4.5.

Once the validation process is complete and appropriate number of components is selected, we retrain the model using both the training and validation datasets, and report the results on the test datasets. Then we calculate how well the model fits the data using the T^2 and Q statistics described in section 3.2.2.

4.2 Dataset Description

The personal music collection consists of 903 songs, divided across 5 major classes according to language: English Pop/Rock music, Hindi film soundtracks and Hindi Pop/Rock music, soundtracks from two regional languages and some classical/ instrumental music. Since the average length of the songs in the collection was around 3 minutes, it was decided to partition the dataset into songs greater than 3 minutes and songs less than 3 minutes. Then, from the set of songs greater than 3 minutes, a random sample of 50% of the songs was chosen for the training dataset. The next 25% were chosen for the validation dataset and the next 25% were segregated for the test set.

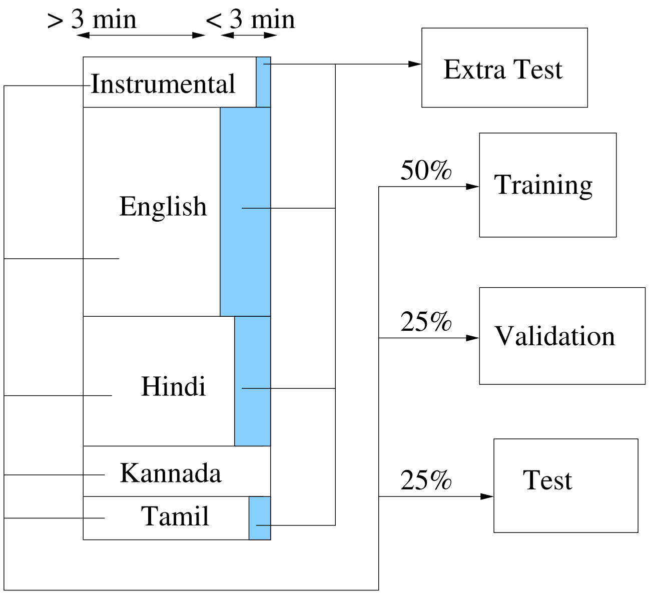

Figure 4.1: Pictorial representation of Dataset. Shaded portion shows songs less than 3 minutes, which are part of extra test dataset.

The songs less than 3 minutes in duration formed part of an extra test set. This is pictorially shown in Figure 4.1.

4.3 Implementation Details

Each song, which was stored as an mp3 file, was read into the Matlab programming environment using the mp3read function developed by Dan Ellis [1]. Next, the audio was downmixed to a single channel and all songs were resampled to a sampling rate of 44.1 kHz. Then, a Hamming window of length 200 ms and with 50% overlap was used to calculate the spectrogram, and a 30 mel filter bank was used to calculate the mel spectrogram. The code for the mel filter bank was based on Malcom Stanley's Auditory Toolbox [29].

The training set was then mean centered. An additional training set with mean centered and unit variance was also created. The two datasets were then trained using the different approaches described in Chapter 3. The dimensionality reduction toolbox developed by Laurens van der Maaten [34] was used for implementing the dimensionality reduction algorithms.

[1] http://labrosa.ee.columbia.edu/matlab/mp3read.html

4.4 Validation Procedure for multiway PCA

4.4.1 Method 1

To choose the number of Principal Components for the two unfoldings of Multiway PCA described in Section 3.5.1, we use the Predicted Sum of Squares of Residuals (PRESS statistics) for validation [17]. The square of the Q statistic as described in Section 3.2.2 when applied to a validation set, is known as PRESS. A simple procedure for cross-validation is provided in Page 354 of [17], for a general dataset $\mathbf{X}$ with m instances, and n features. It is described in the following steps:

1. Divide the dataset into training and validation datasets as described in Section 4.2, and perform the unfolding and subsequent PCA on the training set, as described in Section 3.5.1.

2. For each data element $\mathbf{x}_{val}$ in the $m_{val} \times n_{val}$ validation dataset, calculate its Q statistic $Q_{val,k}$ for all components. The PRESS statistic for k, $0 \leq k \leq n$ is defined as follows:

$$\text{PRESS}(k) = \begin{cases} \dfrac{\sum \mathbf{x}_{val}\mathbf{x}'_{val}}{m_{val}n_{val}}, \text{ if } k = 0 \\ \\ \dfrac{\sum Q^2_{val,k}}{m_{val}n_{val}}, \text{ if } k > 0 \end{cases}. \tag{4.1}$$

   PRESS(0) refers to normalized sum of squares of the original data, and PRESS($k > 0$) refers to the normalised sum of squares of Q statistics with k components.

3. Calculate the value of statistic $W_1(k)$ as given by the following equation for all k PCs, $1 \leq k \leq n$ [17]

$$W_1(k) = \frac{\text{PRESS}(k-1) - \text{PRESS}(k)/D_m}{\text{PRESS}(k)/D_r}. \tag{4.2}$$

   where $D_m = m + n - 2k$ and $D_r = n(m-1) + \sum_{i=1}^{k}(n + m - 2i)$.

4. Retain all those PCs for which $W_1(k) > 1$.

4.4.2 Method 2

Another method of choosing the number of validation components is by using both the PRESS and RESS [1]. Here RESS denotes residual sum of squares over the training set. It can be defined similar to Equation 4.1 as follows, for the $m_{tr} \times n_{tr}$ training set $\mathbf{X}_{tr}$

$$\mathrm{RESS}(k) = \begin{cases} \dfrac{\sum \mathbf{x}_{tr}\mathbf{x}'_{tr}}{m_{tr}n_{tr}}, \text{ if } k = 0 \\ \\ \dfrac{\sum Q^2_{tr,k}}{m_{tr}n_{tr}}, \text{ if } k > 0 \end{cases} . \tag{4.3}$$

Using the RESS and PRESS statistics, we define this term $W_2(k)$ for k retained components [1]:

$$W_2(k) = 1 - \frac{\mathrm{PRESS}(k)}{\mathrm{RESS}(k-1)}. \tag{4.4}$$

Only the components with $W_2(k) \geq 0$ are retained.

4.4.3 Method 3 - Fraction of Variance

If both methods as described above do not yield any result, then the fraction of explained variance plot as described in Section 3.2.1 can be used. A cut off threshold of 0.9 on the fraction of explained variance is used to select an appropriate number of components.

4.5 Validation Procedure for MLSCA

For MLSCA, the same procedure as presented in Section 4.4 can be used. However, it is performed twice, one for the selection of between song components R_b and another for within song components R_w. Keeping R_b constant, the change in PRESS statistic is observed to select an appropriate R_w. Next, R_w is kept constant, and an appropriate value of R_b is chosen using the aforementioned PRESS statistics.

4.6 T^2 and Q statistics

To find out how well the models fit the data, the T^2 and Q statistics as described in the previous chapter are used. An outlier map is created which shows the distribution of the data points and their T^2 and Q cutoffs. For the unfolding along songs method, each song has its own outlier map, whereas for the unfolding along frames a single outlier map can be drawn for the entire dataset. The MLSCA analysis needs to be broken down into the within song part and the between song part. The within song part corresponds to the unfolding along songs while the between song part corresponds to the unfolding along frames.

From each method, we select 6 representative songs. These are as follows:

1. song with maximum T^2,
2. song with maximum Q,
3. song with maximum T^2 and Q,
4. song with minimum T^2,
5. song with minimum Q and
6. song with minimum T^2 and Q.

We then perform a cross comparison: i.e. for the 6 representative songs selected by one method, how did they appear in another method. For instance, the song with maximum T^2 in one method, is it also the song with maximum T^2 in another method and vice-versa.

4.7 Conclusion

Thus in this chapter, the purpose and motivation for the experiments has been explained and all the experiments described. The next chapter infers the results of these experiments.

Chapter 5

Results

This chapter discusses and interprets the results for the experiments carried out in Chapter 4. Each method has results for three experiments: selection of number of components to be retained, error on test set with retained components, and results of T^2 and Q statistics. Additional observations are discussed separately.

5.1 Multiway PCA Unfolding along songs

5.1.1 Selection of number of components

Section 4.4 describes two methods to choose an appropriate number of components. Method 1 uses only PRESS statistics, while Method 2 uses both PRESS and RESS statistics. Both these methods are applied, and the results of these are provided in Figures 5.1 and 5.2.

We can see that for Method 1 $W_1(k) > 1$ and in Method 2 $W_2(k) \geq 0$ for all k, so both these methods fail to select an appropriate number of components to retain. Thus we use the fraction of explained variance, shown in Figure 5.3. Based on the fraction of explained variance, we see that 90% of the variance is explained by both mean centered and mean centered unit variance datasets when 5 components are selected.

5.1.2 Test Set Generalisation

Thus we retain 5 components and check the error on the validation set with 5 components. Figure 5.4 shows how the average validation error decreases with increase in number of components. Next, we move to the test set. With 5 retained components, we obtain an error of 0.4862 on the mean centered

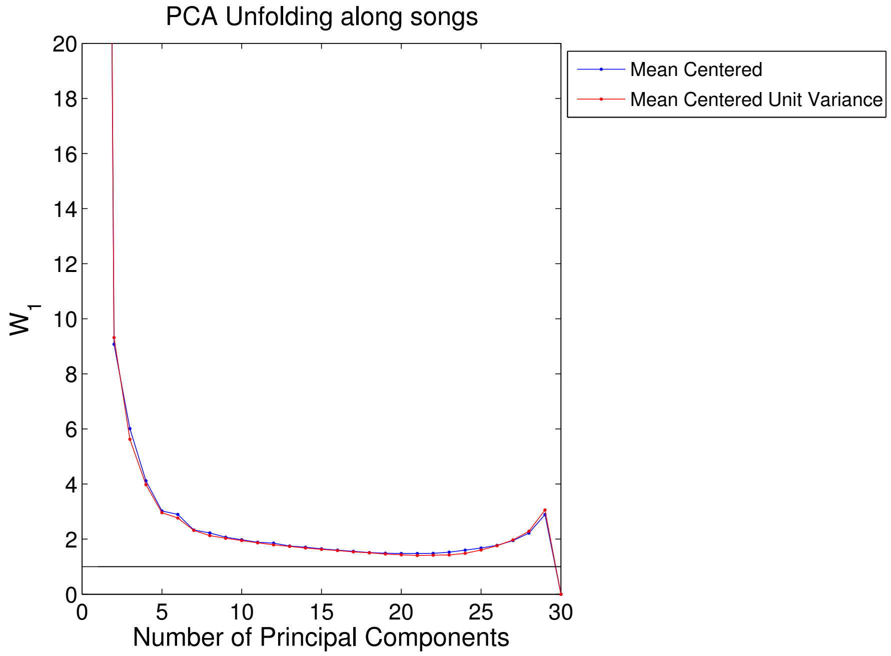

Figure 5.1: Result of using Method 1 used to select number of components in Unfolding along songs. $W_1(k) > 1$ for all k, so this method fails.

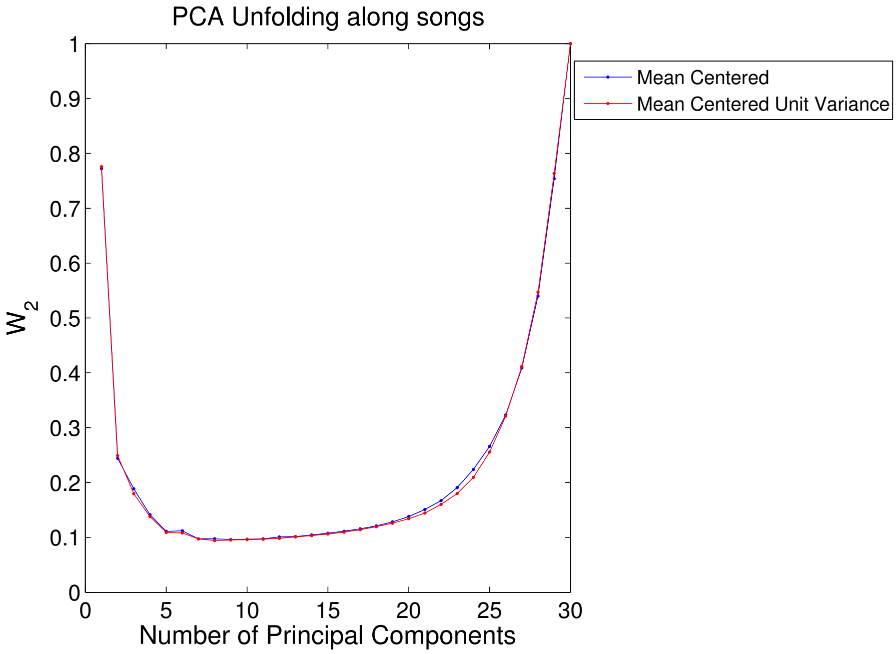

Figure 5.2: Result of using Method 2 used to select number of components in Unfolding along songs. $W_2(k) \geq 0$ for all k, so even this method fails.

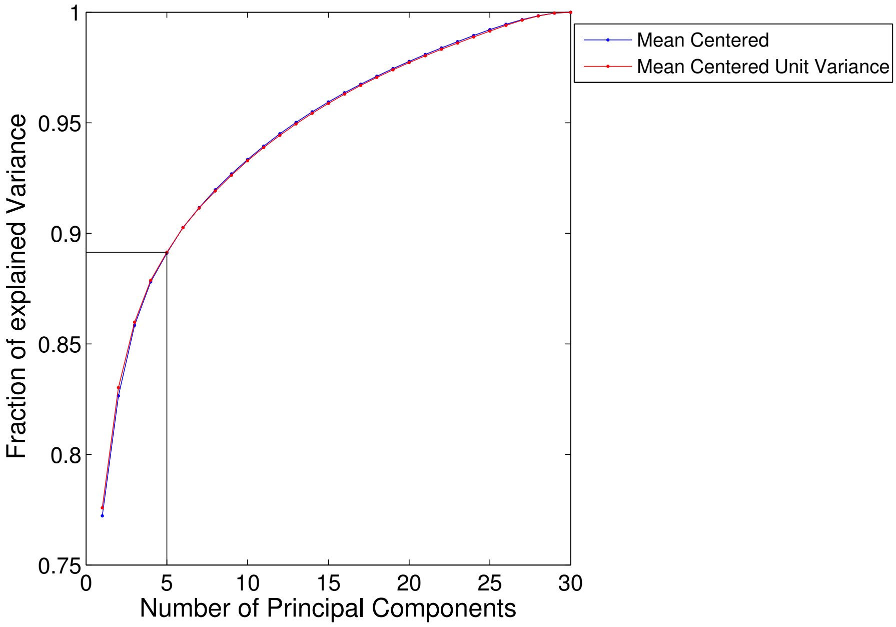

Figure 5.3: Fraction of Explained Variance with 90% cut off suggests 5 retained components.

test set and 0.3229 on the mean centered unit variance test set. Both results for test and extra test dataset are summarized in Table 5.1.

5.1.3 T^2 and Q statistics on Test Data

Next, we calculate the T^2 and Q statistics on the test set. Unfolding across songs produces a T^2 and Q statistic for each frame in each song, so for each song we find the fraction of frames that exceed the prescribed T^2 and Q limits. These limits are calculated according to Section 3.2.1. Thus an

Dataset	**Optimal PCs**	**Average Test Error**	**Average Extra Test Error**
Mean Centered	5	0.4862	0.4539
Mean Centered Unit Variance	5	0.3229	0.3013

Table 5.1: Test Error and Extra Test Error for Unfolding along songs with 5 retained components.

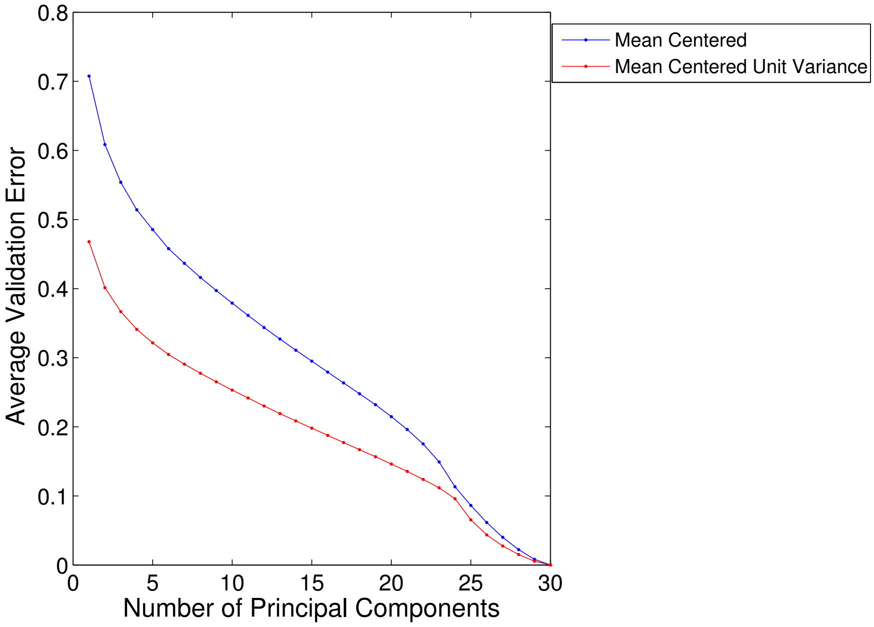

Figure 5.4: Average Validation Error decreases with increase in number of components. When all 30 components are used, the error on validation set goes to 0.

outlier map can be drawn for each song in the dataset. Two representative outlier maps are shown in Figures 5.6 and 5.8. Their corresponding T^2 and Q plots are shown in Figures 5.5 and 5.7 respectively. These are the songs that have maximum and minimum number of frames exceeding both the limits. We observe that, in both songs there are very few frames that break the T^2 and/or Q limits, illustrating that the PCA model with 5 retained components was able to capture most of the data. Hence, it shows that the dimensionality reduction from 30 to 5 is appropriate.

5.2 Multiway PCA Unfolding along frames

5.2.1 Selection of number of components

The results are summarized in Table 5.2, and Figures 5.9 and 5.10.

Method 1 suggests a higher number of components than Method 2. To choose between the two methods, we use the fraction of explained variance. We observe that there is a 6-7% increase in the fraction of explained variance if Method 1 is chosen instead of Method 2. Also the fraction of variance values

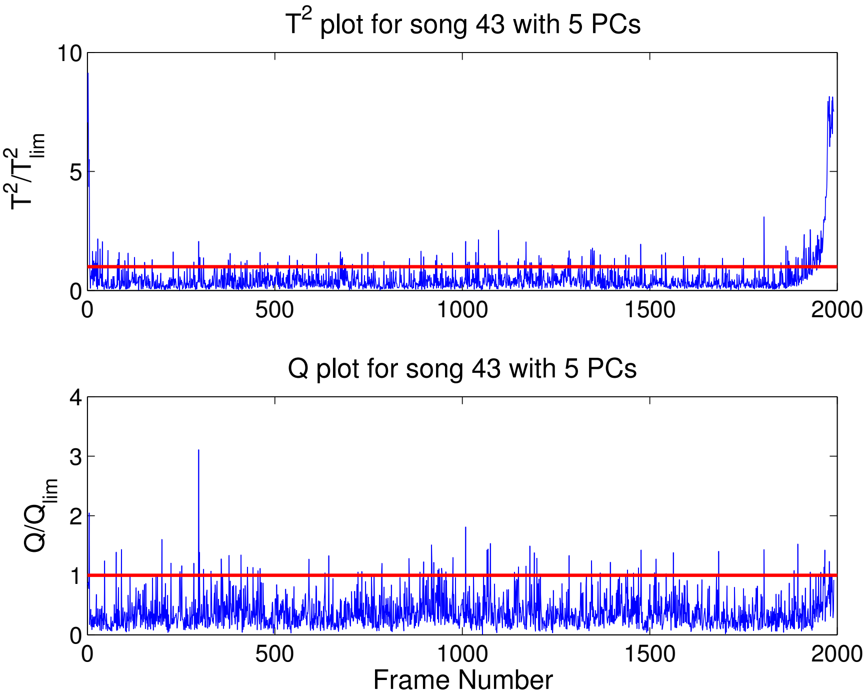

Figure 5.5: Individual T^2 and Q plots for song with maximum no of frames exceeding T^2 and Q limits.

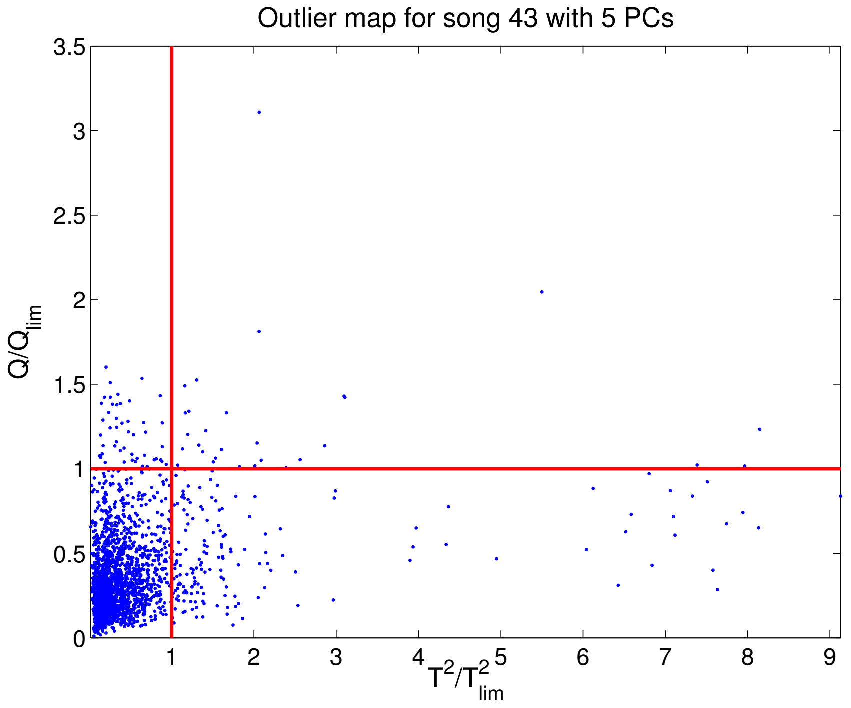

Figure 5.6: Outlier map for song with maximum no of frames exceeding T^2 and Q limits.

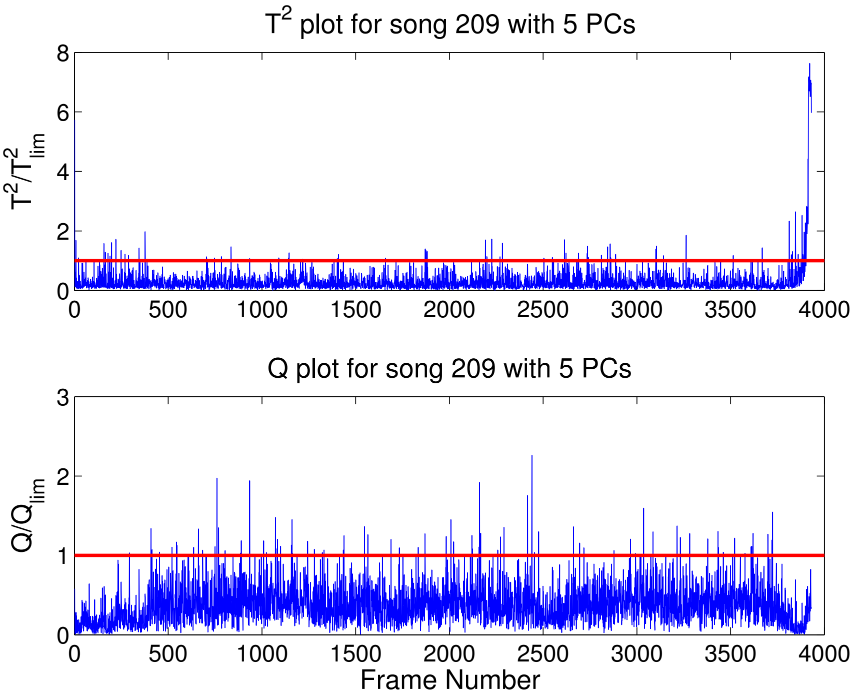

Figure 5.7: Individual T^2 and Q plots for song with minimum no of frames exceeding T^2 and Q limits.

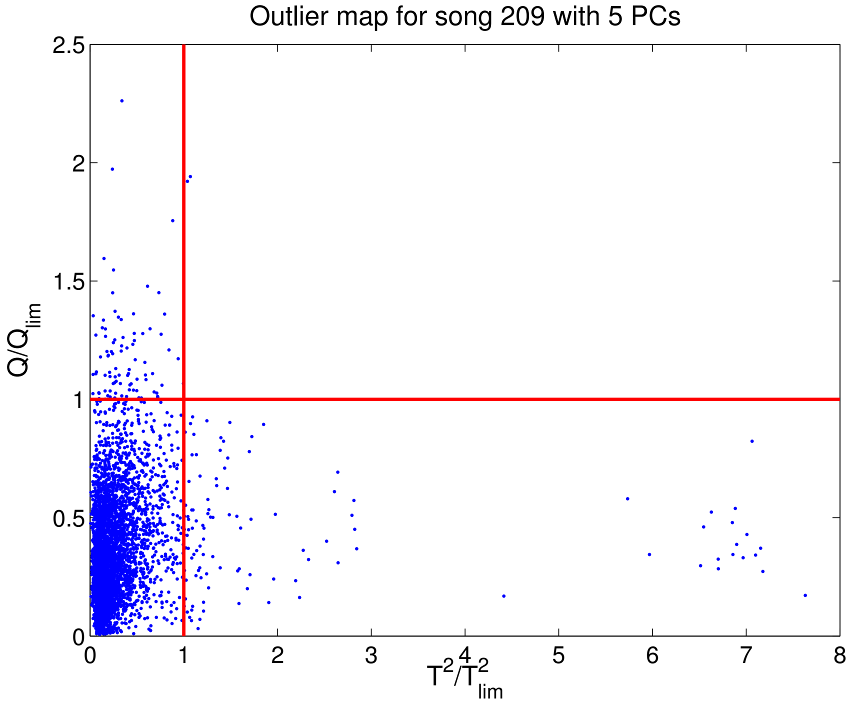

Figure 5.8: Outlier map for song with minimum no of frames exceding T^2 and Q limits.

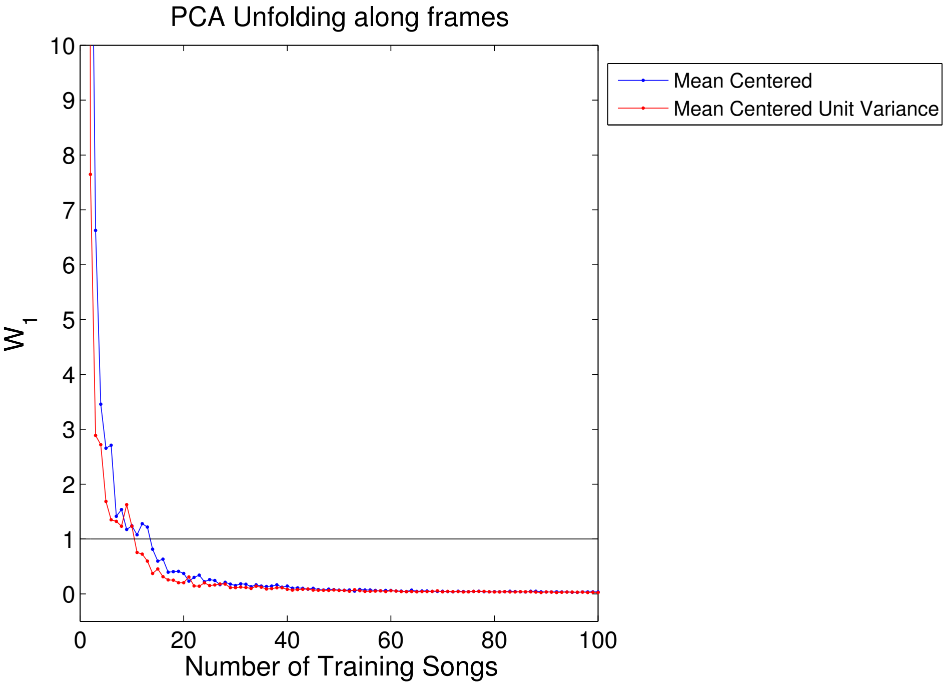

Figure 5.9: Unfolding along Frames Selection of number of components by Method 1.

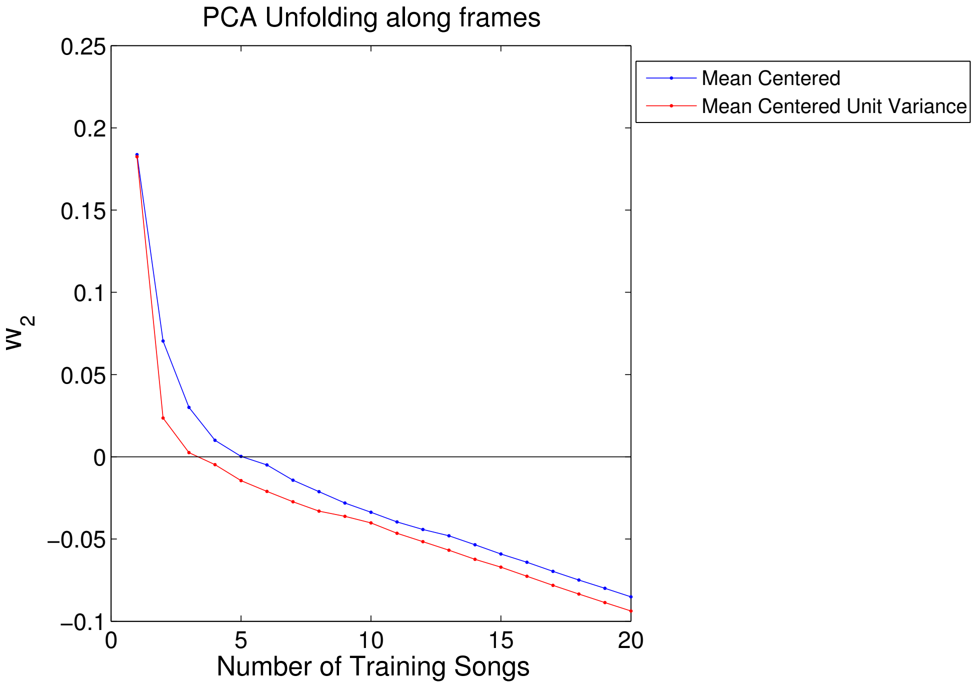

Figure 5.10: Unfolding along Frames Selection of number of components by Method 2.

Dataset	**Optimal Features Method 1**	**Optimal Features Method 2**	**Percentage explained variance - Method 1**	**Percentage explained variance - Method 2**
Mean Centered	13	5	35.8%	29.45%
Mean Centered Unit Variance	10	3	29.45%	23.47%

Table 5.2: Suggested number of retained components using both validation techniques described in Section 4.4 for Unfolding along frames.

are quite less, the maximum being only 0.36. This is becuase the maximum number of reducible dimensions is equal to the number of instances m, which is greater than 400, and not the number of features $n = 30$ as in the previous case. Thus we retain 13 components for the mean centered dataset and 10 components for the mean centered unit variance dataset. We then check the validation error obtained with these components. The validation error for all components is shown in Figure 5.11. Note that unlike PCA, the validation error does not decrease to 0.

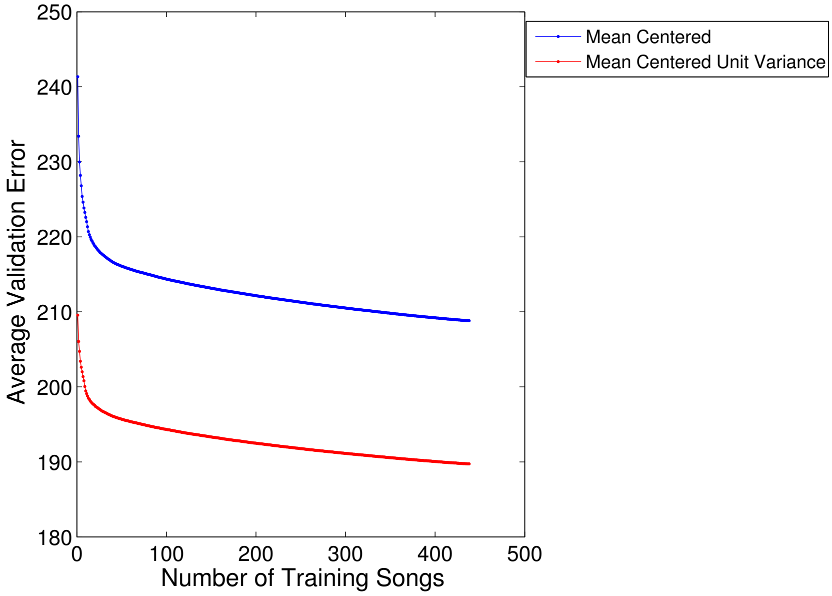

Figure 5.11: Unfolding along Frames Validation error for all components. The error descreases but does not reach 0 when all components are selected.

5.2.2 Test Set Generalisation

Now after selection of appropriate components, we move to the test set. The mean centered test dataset gives an error of 221.7 with 13 retained components and the mean centered unit variance dataset an error of 227.6. Notice that here, with all m compnents retained, the test error does not go to 0. This is because the maximum number of reducible dimensions m, which is in the order of 10^2, is much less than the initial dimension nK_{sh} whose order of magnitude is 10^4.

5.2.3 T^2 and Q statistics

Here, a T^2 and Q statistics can be calculated for each song. Based on the limit set on T^2 and Q distances, we can obtain a single outlier map for the entire dataset. This is illustrated in Figure 5.12. The songs that were used to draw outlier maps for the previous unfolding are depicted in red. Again, we can observe from Figure 5.12 that very few songs are present beyond the T^2 or the Q limits, indicating the MDS model fits the data well.

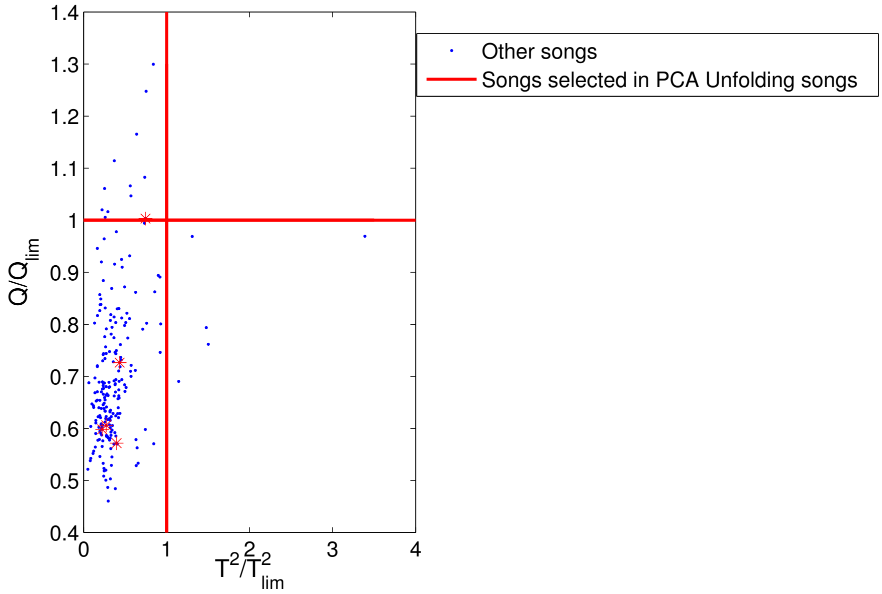

Figure 5.12: Unfolding along Frames Outlier Map for Test Mean Centered Dataset. For the purpose of cross-comparison, songs that were selected from PCA Unfolding songs are shown in red.

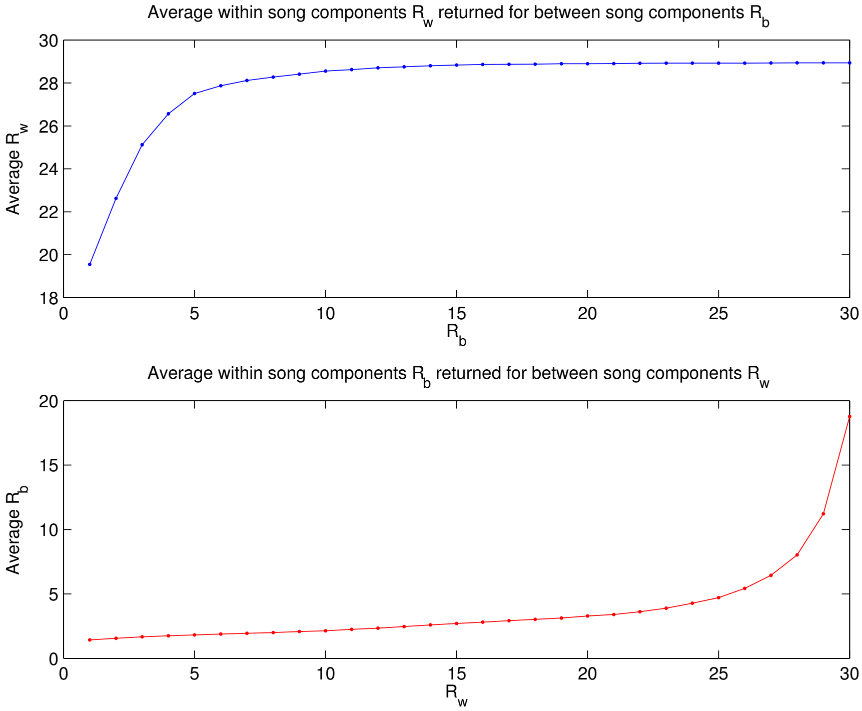

Figure 5.13: MLSCA selection of between song components R_b and within song components R_w. When R_b is fixed and R_w is selected then Model2 with (R_b,R_w)= (10,29) is suggested, and in reverse case Model3 with (R_b,R_w)= (4,22) is suggested.

5.3 MLSCA

5.3.1 Selection of number of components

There are two sub-models in MLSCA - the between song sub-model and the within song sub-model. We need to choose an appropriate R_b and R_w separately. As discussed in Section 4.5, first R_b is kept constant and an appropriate R_w is chosen using Method 1. We thus obtain a plot of appropriate R_w for each value of R_b. Similarly, if R_w is kept constant and appropriate R_b is calculated, we obtain a plot of appropriate R_b for all values of R_w. These plots for the mean centered dataset are shown in Figure 5.13. The results for the mean centered unit variance dataset are very similar.

We can observe that in the first figure of Figure 5.13 when we increase R_b beyond 10, R_w does not increase beyond 29. So we can fix one candidate model with $(R_b, R_w) = (10, 29)$. From the second figure in Figure 5.13, we

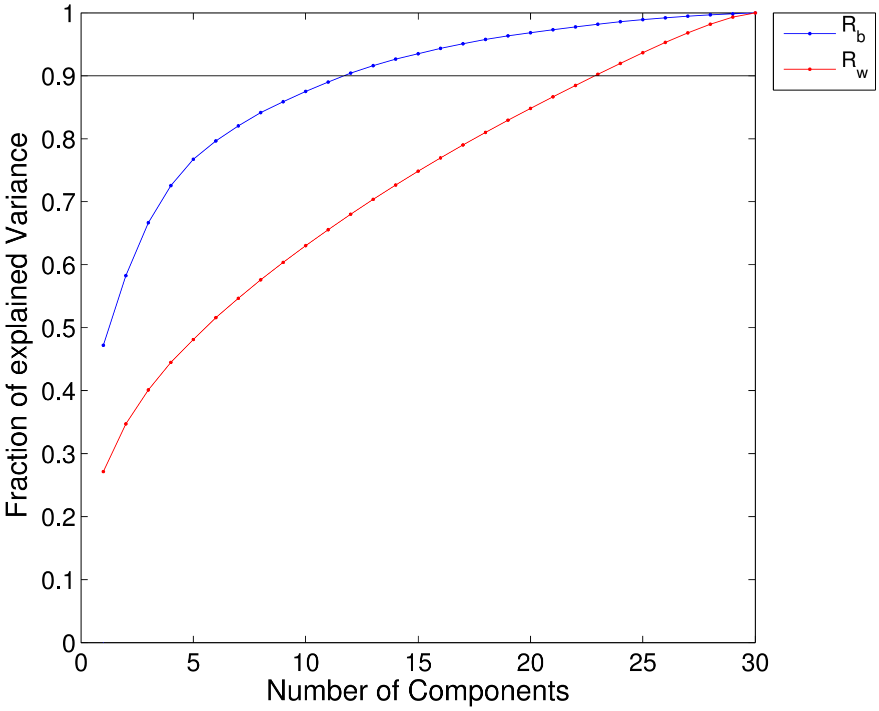

Figure 5.14: Fraction of explained variance with the between and the within parts done separately, with the 90% cut-off threshold shown.

can observe a sudden increase in R_b, after $R_b \geq 4$, once $R_w \geq 22$. Thus we can fix another candidate model with $(R_b, R_w) = (4, 22)$. Also from the 90% cut off on the fraction of explained variance for the between part and the within part separately, shown in Figure 5.14, we obtain a third model with $(R_b, R_w) = (12, 23)$.

To decide between these three models, we look at the average RMS error surface with the selected components. This is shown in Figure 5.15, with the three candidate models shown in red. We can observe that an increase in within song components R_w decreases the RMS error better than increase in R_b. Thus we choose the final model with $R_b = 10$ and $R_w = 29$.

5.3.2 Test Set Generalisation

After the appropriate between songs components and within songs components have been selected, we extend this model to the test set. The errors on the test and extra test dataset are summarized in Table 5.3.

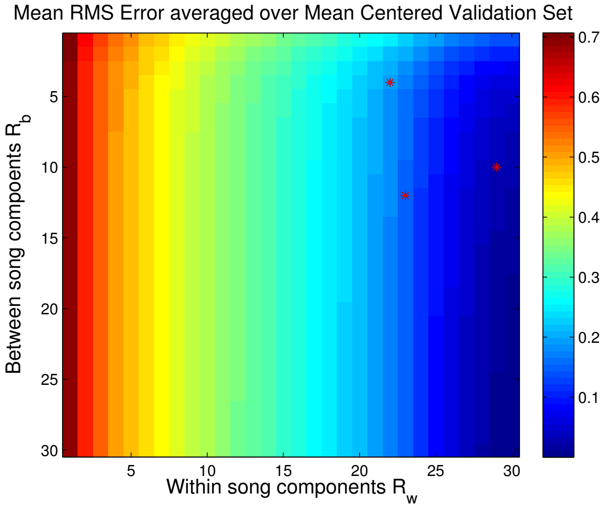

Figure 5.15: Reconstruction Error Surface for Mean Centered dataset, with the 3 candidate models shown in Red.

Dataset	**Optimal** R_b	**Optimal** R_w	**Average RMS Error (Test)**	**Average RMS Error (Extra Test)**
Mean Centered	10	29	0.0313	0.0362
Mean Centered Unit Variance	10	29	0.0209	0.0241

Table 5.3: Test Error and Extra Test Error for MLSCA with $R_b = 10$ and $R_w = 29$.

5.3.3 T^2 and Q statistics - Between songs hyperplane

Next, we calculate T^2 and Q statistics for the between songs hyperplane. This is similar to the Unfolding across frames, so we get an outlier map for the entire dataset, similar to Section 3.5.1.2. This outlier map for the mean centered dataset is shown in Figure 5.16.

5.3.4 T^2 and Q statistics - Within songs hyperplane

The T^2 and Q statistics are now calculated for the within songs hyperplane. Similar to Section 3.5.1.1, we obtain an outlier map for each song. The outlier maps of two representative songs - the song with maximum fraction of frames exceeding T^2 and Q limits, and minimum frames exceeding T^2 and

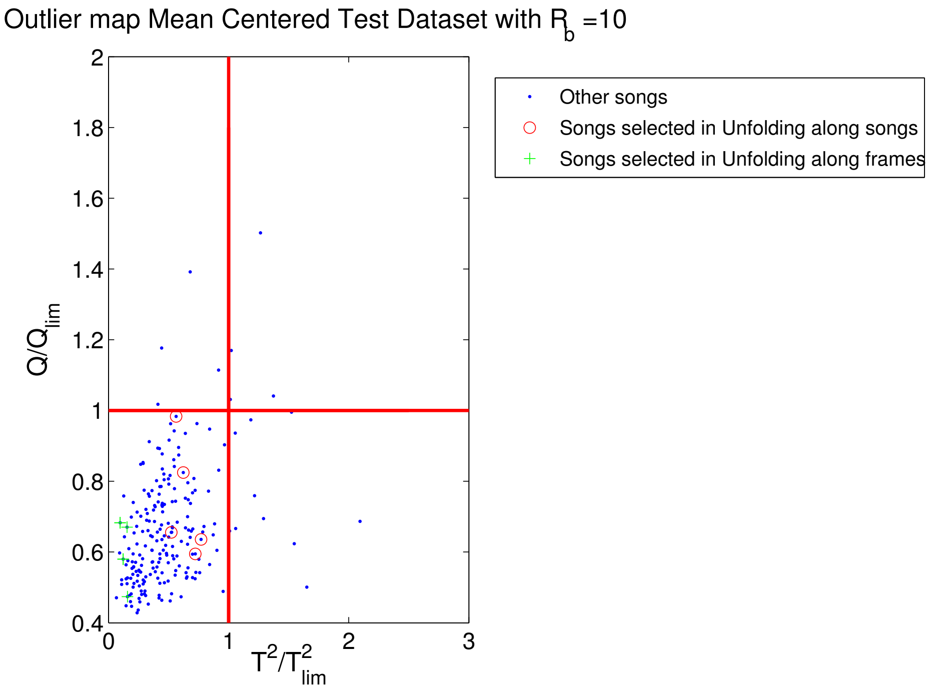

Figure 5.16: Outlier map for the MLSCA along the between songs hyperplane for mean centered dataset. Songs selected in the unfoldings of Multiway PCA are shown in red and green. As in the previous case of Unfolding along frames, most songs are present within the T^2 and Q cut off limits.

Q limits are shown in Figures 5.17 and 5.18 respectively.

The between part of MLSCA is similar to PCA Unfolding along frames. Hence Figures 5.12 and 5.16 are similar to each other. Similarly, the within part of MLSCA is similar to PCA Unfolding along songs. If we compare the sets of outlier maps in Figures 5.6 and 5.8 and Figures 5.17 and 5.18, we see that MLSCA within part has more fraction of frames (5.48%) breaking the T^2 and Q limits compared to PCA Unfolding along songs model (1.46%).

5.3.5 Cross Comparison

Now a cross comparison is performed, i.e. the songs that were obtained in the PCA Unfoldings are compared with each other and with the two MLSCA sub-models. There are 6 songs for each model as described in Section 4.6. First, we start with PCA Unfolding songs, the 6 songs which are calculated

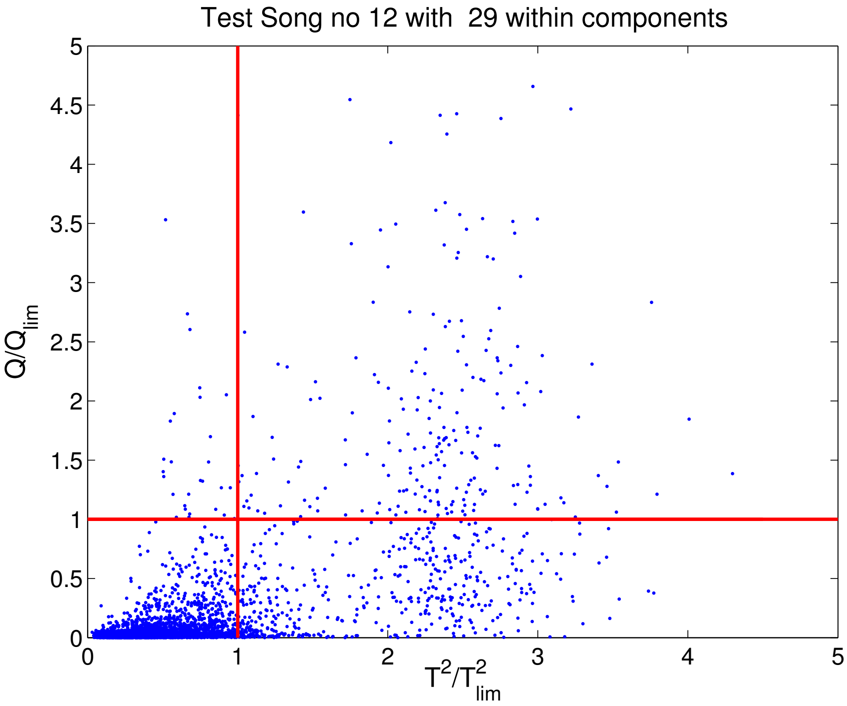

Figure 5.17: Outlier map for the MLSCA within songs hyperplane for mean centered dataset for song with maximum fraction of frames exceeding T^2 and Q limits.

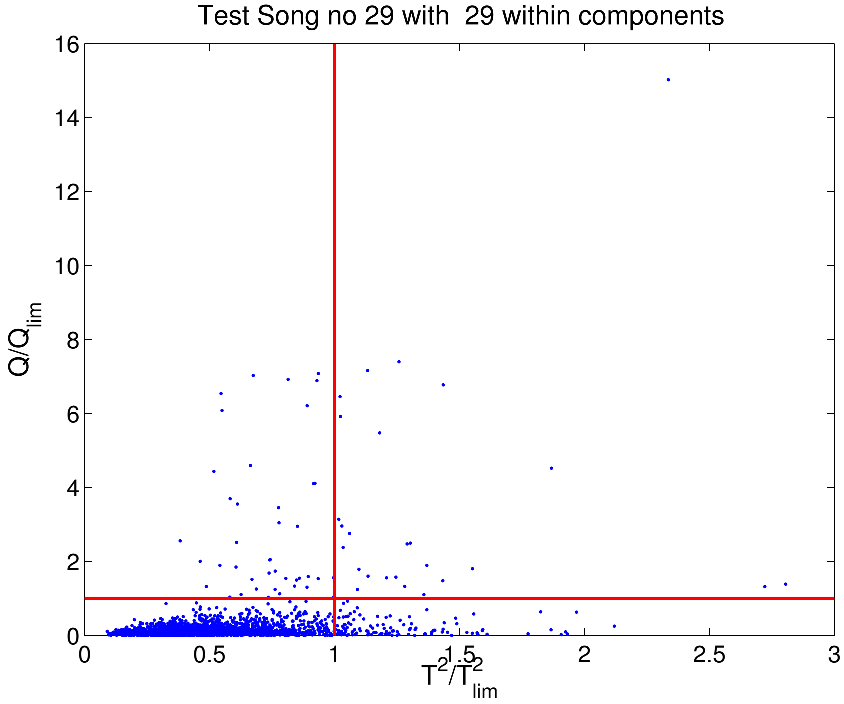

Figure 5.18: Outlier map for the MLSCA within songs hyperplane for mean centered dataset for song with minimum fraction of frames exceeding T^2 and Q limits.

as per Section 4.6 are now used in the other unfolding along frames. Table 5.4 in the second column lists these songs for the mean centered dataset $S_{s1}, ..., S_{s6}$ and Table 5.5 lists the T^2/T^2_{lim}, Q/Q_{lim} ratios and the ranking in terms of descending order of $\sqrt{(T^2/T^2_{lim})^2 + (Q/Q_{lim})^2}$ (which corresponds to how far away a datapoint in terms of its T^2 and Q distance is from the origin) of these 6 songs when unfolded across frames. If the ranking is lower, then it is closer to the origin.

We observe that song S_{s1} had the highest fraction of frames exceeding T^2 distance limit, however when unfolded along frames its ranking in terms of decreasing order of distance is 142 (out of 219). It is interesting to note that song S_{s5} with the minimum numer of frames exceeding the Q distance limits has the 12th highest distance in terms of T^2 and Q when unfolded along frames. Song S_{s5} if unfolded along the frames is an outlying point, however when it is unfolded along the songs, it is not an outlier since it has the least fraction of frames exceeding Q limits. Thus, we can conclude that if a song is an outlier in one unfolding, it may not necessarily be an outlier in the other unfolding.

Next, we do the reverse process i.e. take selected songs $S_{f1}, ..., S_{f6}$ from the Unfolding along frames and unfold them along songs. Each song will produce an outlier map, from which we calculate the fraction of frames exceeding T^2, Q and both T^2 and Q distance limits. Also, we get a ranking of the song based on fraction of frames breaking both T^2 and Q distance limits. These results are shown in Table 5.6. Here we see that song S_{f2} when unfolded along the frames is an outlier, and remains an outlier when unfolded along songs direction.

	Unfolding across songs (5 PCs)	**Unfolding across frames (13 PCs)**
Max T^2	$S_{s1} = 84$	$S_{f1} = 117$
Max Q	$S_{s2} = 43$	$S_{f2} = 125$
Max T^2 and Q	$S_{s3} = 43$	$S_{f3} = 117$
Min T^2	$S_{s4} = 88$	$S_{f4} = 110$
Min Q	$S_{s5} = 203$	$S_{f5} = 62$
Min T^2 and Q	$S_{s6} = 209$	$S_{f6} = 110$

Table 5.4: Song IDs for PCA Unfoldings along songs and along frames for the mean centered dataset. These songs are compared from one unfolding onto the other and the MLSCA.

The last step is to compare these selected songs with the MLSCA between song and within song sub-models. Tables 5.7 and 5.8 show the two song sets with the MLSCA between songs model and Tables 5.9 and 5.10

Song ID	T^2/T^2_{lim}	Q/Q_{lim}	**Ranking (descending order of distance from T^2,Q origin**
S_{s1}	0.3976	0.5716	142
S_{s2}	0.4345	0.7264	72
S_{s3}	0.4345	0.7264	72
S_{s4}	0.2236	0.5981	188
S_{s5}	0.7458	1.0032	12
S_{s6}	0.2726	0.6061	169

Table 5.5: Songs selected along PCA Unfolding songs from the mean centered dataset now unfolded along frames direction.

Song ID	**Fraction of frames exceeding T^2 limits**	**Fraction of frames exceeding Q limits**	**Fraction of frames exceeding both T^2 and Q limits**	**Ranking (descending order of values in previous column)**
S_{f1}	0.0412	0.0282	0.0027	142
S_{f2}	0.0692	0.0227	0.0072	8
S_{f3}	0.0412	0.0282	0.0027	142
S_{f4}	0.0217	0.0217	0.0013	205
S_{f5}	0.0364	0.0325	0.0034	96
S_{f6}	0.0217	0.0217	0.0013	205

Table 5.6: Songs selected along PCA Unfolding Frames from the mean centered dataset now unfolded along Songs direction.

show the cross-comparison on the MLSCA within song model. We observe that the song set which was obtained through PCA Unfolding along songs $S_{s1}, ..., S_{s6}$ have higher T^2 and Q distances in the MLSCA between songs model and higher fraction of frames exceeding T^2 and Q distance limits in the MLSCA within songs model compared to the Unfolding along frames song list $S_{f1}, ..., S_{f6}$.

5.4 Other Observations

Another interesting pattern was observed for songs which contain silence. The T^2 and Q ratios show an extremely high value in the silence section. This was true for these ratios extracted from both PCA unfolding along

Song ID	T^2/T^2_{lim}	Q/Q_{lim}	**Ranking (descending order of distance from T^2,Q origin**
S_{f1}	0.1215	0.6685	152
S_{f2}	0.1579	0.4626	214
S_{f3}	0.1215	0.6685	152
S_{f4}	0.0967	0.6575	161
S_{f5}	0.1543	0.7203	132
S_{f6}	0.0967	0.6575	161

Table 5.7: Songs selected along PCA Unfolding along Frames from the mean centered dataset now used on the MLSCA between songs model.

Song ID	T^2/T^2_{lim}	Q/Q_{lim}	**Ranking (descending order of distance from T^2,Q origin**
S_{s1}	0.5632	0.9692	26
S_{s2}	0.5229	0.7504	69
S_{s3}	0.5229	0.7504	69
S_{s4}	0.6232	0.8474	37
S_{s5}	0.7712	0.5811	53
S_{s6}	0.7227	0.5236	77

Table 5.8: Songs selected along PCA Unfolding along Songs from the mean centered dataset now used on MLSCA between songs model.

Song ID	**Fraction of frames exceeding T^2 limits**	**Fraction of frames exceeding Q limits**	**Fraction of frames exceeding both T^2 and Q limits**	**Ranking (descending order of values in prev column)**
S_{f1}	0.1626	0.0499	0.0260	128
S_{f2}	0.1301	0.0344	0.0287	72
S_{f3}	0.1626	0.0499	0.0260	128
S_{f4}	0.1305	0.0560	0.0201	193
S_{f5}	0.1806	0.0595	0.0308	38
S_{f6}	0.1305	0.0560	0.0201	193

Table 5.9: Songs selected along PCA Unfolding along Frames from the mean centered dataset now used on MLSCA Within songs model.

Song ID	**Fraction of frames exceeding T^2 limits**	**Fraction of frames exceeding Q limits**	**Fraction of frames exceeding both T^2 and Q limits**	**Ranking (descending order of values in prev column)**
S_{s1}	0.1889	0.0506	0.0503	2
S_{s2}	0.0648	0.0357	0.0261	126
S_{s3}	0.0648	0.0357	0.0261	126
S_{s4}	0.1327	0.0540	0.0211	187
S_{s5}	0.1692	0.0575	0.0371	6
S_{s6}	0.1473	0.0547	0.0298	52

Table 5.10: Songs selected along PCA Unfolding along Songs from the mean centered dataset now used on MLSCA Within songs model.

songs model and the MLSCA Within songs model.

To illustrate this, we took a song which has 30 s of silence in the audio in the middle of the song, and plotted the T^2 and Q ratios of MLSCA within song model and PCA Unfolding along songs. These are shown in Figure 5.19.

For Unfolding along songs, the deviation in Q ratio is not very apparent, however for MLSCA within song model. we can clearly see a set of high Q ratios from frames 3200-3700. When we look at the T^2 ratios, we can see a set of high values for the same frame range. When the audio file was played, there was absolute silence in the region indicated above. The silence corresponds to the pause between two songs, which were recorded in the same audio file.

The log mel spectrogram of this audio file is plotted in Figure 5.20. Note the presence of a blue region in between frames 3200-3700 which corresponds to silence. The same area is detected in the T^2 and the Q statistics, hence verifying the observation.

5.5 Conclusion

We have thus performed a thorough data analysis of the audio dataset, using the different multiway methods. This concludes the thesis work. The thesis work is summarized in the next chapter 6.

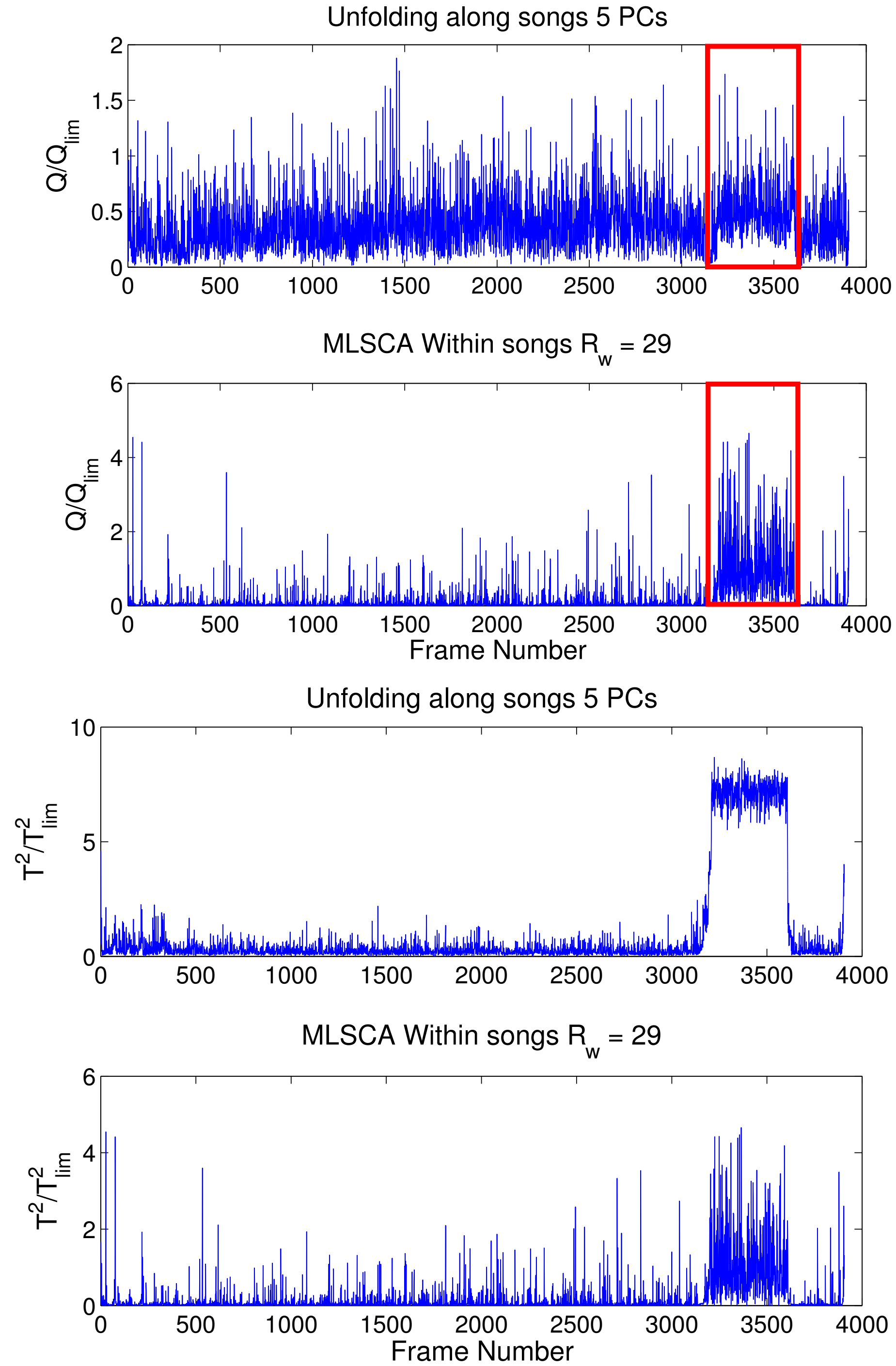

Figure 5.19: T^2 and Q ratios from both Unfolding along songs and MLSCA Within part for a song which has a section of silence in the audio. It can be observed that both statistics are unusually high in the silence section.

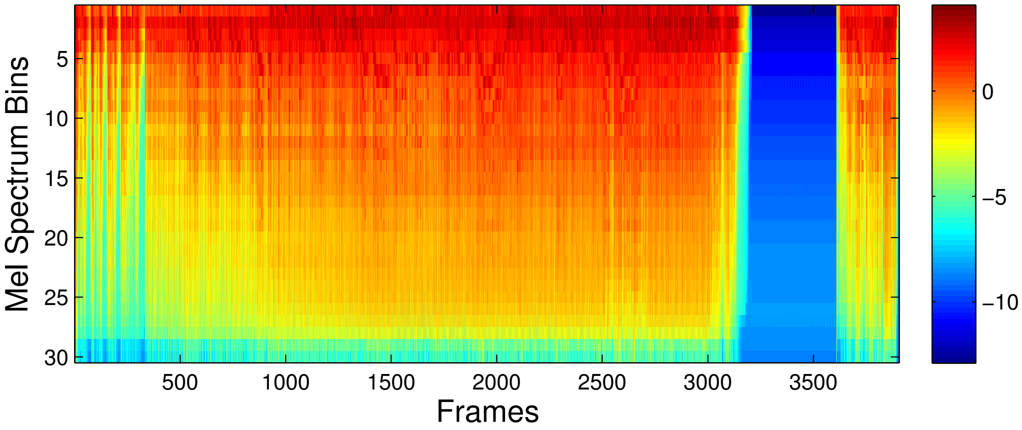

Figure 5.20: Log mel spectrogram of the audio file with silence zone in the centre, indicated by a blue region.

Chapter 6

Conclusions

This thesis work proposed the application of two dimensionality reduction techniques, namely multiway PCA and MLSCA on mel spectrogram features extracted from a personal music collection, which can be used for audio content analysis tasks like genre classification, cover song retrieval etc. First, all the songs were downsampled to a single channel and resampled to 44.1 kHz. Then, the mel spectrogram for all these songs were calculated.

A music tensor for the collection was built by aggregating the mel spectrogram features for all songs. Two ways of unfolding this 3-way tensor into a 2-way a data matrix were studied : Unfolding along songs, and unfolding along frames. For the first unfolding, PCA was used and for the 2nd unfolding classic metric MDS was used. Then, a third method MLSCA was used which built a model for each song as an additive combination of global mean, within song components and between song components.

It was observed in Figure 5.3 that Unfolding along songs does a good performance in dimensionality reduction since we can cover 90% of the explained variance with just 5 dimensions, 1/6th of the original 30 dimensional feature space. However, it characterizes the frame wise evolution of each song, and thus we cannot represent the whole song by a single feature by these dimensionality reduction techniques. Summarization of the feature vectors needs to be done.

Unfolding along frames, however can represent the entire song as a single feature vector, however, some thresholding based on the length of the shortest song has to be done. This method also suggests a low number of components to retain through the validation methods using PRESS and RESS statistics (Table 5.2), however the fraction of explained variance is very low. This is because classic metric MDS and not PCA has been applied, and thus the maximum number of reducible dimensions equals the number of training instances, which is much less than the number of features.

MLSCA is a composite model that is a combination of the two models described above. The analysis of MLSCA can be broken down into the between song part, which corresponds to the unfolding along frames, and within song part, which corresponds to unfolding along songs. It can be observed from Figure 5.15 that increasing within song components R_w reduces the recostruction error faster than increasing the between song components R_b. Hence, we arrive at a value of $R_w = 29$ and $R_b = 10$ as the final number of components. The fitness of each of these models were evaluated with the T^2 and Q statistic, and compared with each other.

There are several areas in which this thesis work can be further extended. We can validate these features by using them on some audio content analysis task like genre classification. We need to build a global song model from these features. Using the *bag-of-frames* model, the frames of a particular song can be summarized into a single feature vector with the help of probability distributions. Then some clustering can be applied to categorize song features into genres. The accuracy of the genre recognition task can be taken as a parameter to validate the number of retained features. Other linear and non-linear dimensionality reduction techniques like SOM, geodesic mappings, locally linear embeddings etc. and tensor factorization methods like Parallel Factor analysis (PARAFAC), non-negative tensor factorisation (NTF), higher order Singular Value Decomposition (HOSVD) etc could also be attempted. There has been some recent work in tensor analyis for tasks like genre classification [24], [25]. Application of these tensor analysis methods seems to be an interesting work for the future.

Bibliography

[1] Abdi, H., and Williams, L. J. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics 2*, 4 (2010), 433–459.

[2] Alpaydin, E. *Introduction to machine learning.* MIT press, 2004.

[3] Aucouturier, J.-J., and Pachet, F. Music similarity measures: What's the use. In *Proc. ISMIR* (2002), vol. 2.

[4] bbc. Digital music sales outstrip cds and records. `http://www.bbc.co.uk/news/entertainment-arts-18278037`, May 2012.

[5] Borg, I., and Groenen, P. *Modern Multidimensional Scaling: Theory and Applications.* Springer Series in Statistics. Springer, 2005.

[6] Casey, M. A., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., and Slaney, M. Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE 96*, 4 (2008), 668–696.

[7] Chatfield, C., and Collins, A. *Introduction to multivariate analysis.* London [etc.]: Chapman and Hall, 1980.

[8] de Noord, O. E., and Theobald, E. H. Multilevel component analysis and multilevel pls of chemical process data. *Journal of chemometrics 19*, 5-7 (2005), 301–307.

[9] Downie, J. S. Music information retrieval. *Annual review of information science and technology 37*, 1 (2003), 295–340.

[10] Hamel, P., Lemieux, S., Bengio, Y., and Eck, D. Temporal pooling and multiscale learning for automatic annotation and ranking of music audio. In *ISMIR* (2011), pp. 729–734.

[11] Handel, S. Timbre perception and auditory object identification. *Hearing* (1995), 425–461.

[12] Harris, F. J. On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE 66*, 1 (1978), 51–83.

[13] Honkela, T., Raitio, J., Lagus, K., Nieminen, I. T., Honkela, N., and Pantzar, M. Subjects on objects in contexts: Using gica method to quantify epistemological subjectivity. In *Neural Networks (IJCNN), The 2012 International Joint Conference on* (2012), IEEE, pp. 1–9.

[14] Hubert, M., Rousseeuw, P. J., and Branden, K. V. Robpca: a new approach to robust principal component analysis. *Technometrics 47*, 1 (2005).

[15] Hyvärinen, A., Karhunen, J., and Oja, E. *Independent Component Analysis.* Adaptive and Learning Systems for Signal Processing, Communications and Control Series. Wiley, 2004.

[16] ISMIR. Cumulative ismir proceedings. `http://www.ismir.net/proceedings/`.

[17] Jackson, J. *A User's Guide to Principal Components.* Wiley Series in Probability and Statistics. Wiley, 2005.

[18] Jolliffe, I. T. *Principal Component Analysis*, second ed. Springer, Oct. 2002.

[19] Kitahara, T. Mid-level representations of musical audio signals for music information retrieval. In *Advances in Music Information Retrieval*, Z. Ras and A. Wieczorkowska, Eds., vol. 274 of *Studies in Computational Intelligence.* Springer Berlin Heidelberg, 2010, pp. 65–91.

[20] Lee, J. J. A., and Verleysen, M. *Nonlinear dimensionality reduction.* Springer, 2007.

[21] Logan, B., et al. Mel frequency cepstral coefficients for music modeling. In *ISMIR* (2000).

[22] Oppenheim, A. V., Schafer, R. W., Buck, J. R., et al. *Discrete-time signal processing*, vol. 2. Prentice hall Englewood Cliffs, NJ:, 1989.

[23] Pampalk, E. *Computational Models of Music Similarity and their Application in Music Information Retrieval.* PhD thesis, Vienna University of Technology, Vienna, Austria, March 2006.

[24] Panagakis, Y., Kotropoulos, C., and Arce, G. R. Music genre classification using locality preserving non-negative tensor factorization and sparse representations. In *ISMIR* (2009), pp. 249–254.

[25] Panagakis, Y., Kotropoulos, C., and Arce, G. R. Non-negative multilinear principal component analysis of auditory temporal modulations for music genre classification. *Audio, Speech, and Language Processing, IEEE Transactions on 18*, 3 (2010), 576–588.

[26] Peeters, G. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Tech. rep., IRCAM, 2004.

[27] Quatieri, T. *Discrete-time speech signal processing.* Pearson Education, 2002.

[28] Sammon Jr, J. W. A nonlinear mapping for data structure analysis. *Computers, IEEE Transactions on 100*, 5 (1969), 401–409.

[29] Slaney, M. Auditory toolbox. *Interval Research Corporation, Tech. Rep 10* (1998), 1998.

[30] Smith, J. O. *Spectral audio signal processing.* Stanford University, CCRMA, 2008.

[31] Timmerman, M. E. Multilevel component analysis. *British Journal of Mathematical and Statistical Psychology 59*, 2 (2006), 301–320.

[32] Torgerson, W. Multidimensional scaling of similarity. *Psychometrika 30*, 4 (1965), 379–393.

[33] Uitdenbogerd, A., and Zobel, J. Melodic matching techniques for large music databases. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)* (1999), ACM, pp. 57–66.

[34] Van der Maaten, L. An introduction to dimensionality reduction using matlab. *Report 1201* (2007), 07–07.

[35] Wang, A., et al. An industrial strength audio search algorithm. In *ISMIR* (2003), pp. 7–13.

[36] Wise, B. M., Gallagher, N. B., Butler, S. W., White, D. D., and Barna, G. G. A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process. *Journal of Chemometrics 13*, 3-4 (1999), 379–396.

[37] Wold, S., Geladi, P., Esbensen, K., and Ohman, J. Multi-way principal components-and pls-analysis. *Journal of Chemometrics 1*, 1 (1987), 41–56.

Appendix A

First appendix

A.1 Derivation of PCA

In section 3.2 we had approached PCA from a viewpoint of change of axis of reference. PCA can also be looked at from another viewpoint of maximization of variance. The principal components project data into directions that maximize the variance.

Let us take a mean centered $m \times n$ data matrix $\mathbf{X}$, with m instances and n features. We first estimate the covariance matrix $\mathbf{C}$ according to Equation (3.5). Let $\mathbf{w}$ be a vector that projects $\mathbf{X}$ into PC space. The projection gives component $\mathbf{t}$.

$$\mathbf{t} = \mathbf{X}\mathbf{w}. \tag{A.1}$$

We can calculate the variance of $\mathbf{t}$ as follows:

$$\begin{aligned} \operatorname{var}(\mathbf{t}) &= \frac{1}{m-1}((\mathbf{X}\mathbf{w})^T(\mathbf{X}\mathbf{w})). \\ &= \mathbf{w}^T\left(\frac{1}{m-1}(\mathbf{X}^T\mathbf{X})\right)\mathbf{w}. \\ &= \mathbf{w}^T\mathbf{C}\mathbf{w}. \end{aligned}$$

Now, this variance needs to be maximised. To obtain a unique solution for $\mathbf{w}$, we put a constraint that $||\mathbf{w}|| = 1$. This can be expressed in terms of a Lagrange multiplier and the objective function can be written as follows

$$J(\mathbf{w}) = \mathbf{w}^T\mathbf{C}\mathbf{w} - \alpha(\mathbf{w}^T\mathbf{w} - 1). \tag{A.2}$$

To maximise, set $\frac{\partial}{\partial \mathbf{w}} J(\mathbf{w}) = 0$. Thus we obtain

$$2\mathbf{C}\mathbf{w} - \alpha\mathbf{w} = 0. \tag{A.3}$$

The above equation simplifies to $\mathbf{C}\mathbf{w} = \alpha\mathbf{w}$. This is an eigenvalue equation of $\mathbf{C}$ with α as an eigenvalue and $\mathbf{w}$ as the corresponding eigenvector. To maximise this, we choose the maximum value of α. This is the highest eigenvalue of $\mathbf{C}$ and hence, $\mathbf{w}$ is the eigenvector corresponding to the highest eigenvalue. So, the first PC is the leading eigenvector of $\mathbf{C}$.

For the next PC, it has to chosen such that it maximizes the variance, be of unit norm and an additional constraint is that it is orthogonal to the first PC $\mathbf{w}$. Solving a similar optimization equation gives again an eigenvalue equation. This can be continued upto $k < n$ desired principal components.

Hence, we have proved that PCA is equivalent to finding projections that maximize the variance.

Appendix B

Second appendix

B.1 Proof that PCA and classic metric MDS produce the same solution

Let us take data matrix $\mathbf{X}$ of m instances and n features. We ensure that it is mean centered. First, we perform a singular value decomposition (SVD) of $\mathbf{X}$.

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T. \tag{B.1}$$

The covariance matrix can be written according to Equation (3.5).

$$\mathbf{C} = \frac{1}{m-1}(\mathbf{X}^T\mathbf{X}). \tag{B.2}$$

Ignoring the denominator $(m-1)$, we can rewrite the above equation as follows:

$$\mathbf{C} \propto \mathbf{X}^T\mathbf{X}. \tag{B.3}$$

Inserting the expression of SVD into the previous equation, we get

$$\begin{aligned}\mathbf{C} &\propto (\mathbf{U}\mathbf{S}\mathbf{V}^T)^T(\mathbf{U}\mathbf{S}\mathbf{V}^T).\\ &\propto \mathbf{V}\mathbf{S}^T\mathbf{U}^T\mathbf{U}\mathbf{S}\mathbf{V}^T.\\ &\propto \mathbf{V}\mathbf{S}^T\mathbf{S}\mathbf{V}^T.\end{aligned}$$

Thus we get,

$$\mathbf{C} \propto \mathbf{V}\mathbf{S}^2\mathbf{V}^T. \tag{B.4}$$

The above equation is now the EVD of $\mathbf{C}$ with loading matrix $\mathbf{V}$. Now, next we follow the procedure of PCA, ie take the columns of $\mathbf{V}$ corresponding to the k highest diagonal elements in $\mathbf{S}^2$, thus obtaining an $n \times k$ submatrix $\mathbf{V}_k$ and project the data. Thus our PCA solution is as follows:

$$\mathbf{T} = \mathbf{X}\mathbf{V}_k. \tag{B.5}$$

Next, we look at the normalized inner product matrix $\mathbf{I}_p$. According to Equation (3.14)

$$\mathbf{I}_p = \frac{1}{m}\mathbf{X}\mathbf{X}^T. \tag{B.6}$$

Ignoring the denominator m and substituting SVD expansion of $\mathbf{X}$, we can rewrite the above equation as follows:

$$\begin{aligned} \mathbf{I}_p &\propto \mathbf{X}\mathbf{X}^T. \\ &\propto \mathbf{U}\mathbf{S}\mathbf{V}^T(\mathbf{U}\mathbf{S}\mathbf{V}^T)^T. \\ &\propto \mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{V}\mathbf{S}^T\mathbf{U}^T. \\ &\propto \mathbf{U}\mathbf{S}\mathbf{S}^T\mathbf{U}^T. \end{aligned}$$

Thus we get,

$$\mathbf{I}_p \propto \mathbf{U}\mathbf{S}^2\mathbf{U}^T. \tag{B.7}$$

Now the above equation corresponds to the EVD of $\mathbf{I}_p$. Next, we take the k highest eigenvalues and obtain $k \times k$ submatrix $\mathbf{S}_k^2$. Then, we project the data for MDS using Equation (3.16).

$$\mathbf{T} = \mathbf{U}(\mathbf{S}_k^2)^{1/2} = \mathbf{U}\mathbf{S}_k. \tag{B.8}$$

Now, let us take the solution of PCA in Equation (B.5) and try to derive the MDS solution of Equation (B.8) from it. Substituting the expansion of SVD in Equation (B.5), we get:

$$\mathbf{T} = \mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{V}_k. \tag{B.9}$$

Note that $\mathbf{V}^T\mathbf{V}_k = \mathbf{I}_k$, a $k \times k$ identity matrix. And $\mathbf{S}\mathbf{I}_k$ will produce the $k \times k$ diagonal matrix $\mathbf{S}_k$. Thus, we can clearly see that the equation above simplifies to Equation (B.8).

Thus, we have proved that classic metric MDS and PCA produce the same solution, and are thus equivalent.

We can observe that the term $\mathbf{S}^2$ appears in both equations (B.4) and (B.7). Thus the eigenvalues of unnormalized outer product matrix $\mathbf{X}^T\mathbf{X}$ and unnormalized inner product matrix $\mathbf{X}\mathbf{X}^T$ are contained in the leading diagonal of $\mathbf{S}^2$, which denotes the squares of the singular values of $\mathbf{X}$.

In Section 3.3 we realized that to extend MDS for new test data, we have to calculate the eigenvectors of the covariance matrix from the EVD of $\mathbf{I}_p$ obtained from training data. The equivalence of classic metric MDS and PCA allows us to do this. Let us calculate $\mathbf{X}^T\mathbf{U}$.

$$\begin{aligned}\mathbf{X}^T\mathbf{U} &= (\mathbf{U}\mathbf{S}\mathbf{V}^T)^T\mathbf{U}.\\ &= \mathbf{V}\mathbf{S}^T\mathbf{U}^T\mathbf{U}.\end{aligned}$$

Since $\mathbf{S}^T = \mathbf{S}$ and $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, we get

$$\mathbf{X}^T\mathbf{U} = \mathbf{V}\mathbf{S}. \tag{B.10}$$

Right multiplying the above equation with $\mathbf{S}^{-1}$, we obtain

$$\mathbf{V} = \mathbf{X}^T\mathbf{U}(\mathbf{S}^2)^{-1/2}. \tag{B.11}$$

We observe that we have obtained the eigenvectors of unnormalized outer product matrix $\mathbf{V}$ from the eigenvalue decomposition of unnormalized inner product matrix (Equation (B.7)). Finally, we normalize $\mathbf{V}$ by $(m-1)$ to obtain eigenvectors of covariance matrix $\mathbf{C}$. Thus, this can be used as a loading matrix to project unseen test data.