A COMPONENT BASED MUSIC CLASSIFICATION APPROACH

A Thesis

Presented to

the Faculty of the Graduate School

University of Crete

by

André Holzapfel

In Partial Fulfillment

of the Requirements for the Degree

MASTER OF SCIENCE

Computer Science

September 2006

Heraklion, Greece

CERTIFICATE OF APPROVAL OF THESIS

A COMPONENT BASED MUSIC CLASSIFICATION APPROACH

by

André Holzapfel

author	date

Graduate Advisory Committee:

Yannis Stylianou, Chairperson	date
Panagiotis Tsakalides	date
Giorgos Tziritas	date

Approved for the Faculty:

Panagiotis Trahanias
Associate President of Faculty

Date

A COMPONENT BASED MUSIC CLASSIFICATION APPROACH

André Holzapfel

Master of Science in Computer Science

This thesis introduces a new feature set based on a Non-negative Matrix Factorization approach for the classification of musical signals into genres, only using synchronous organization of music events (vertical dimension of music). This feature set generates a vector space to describe the spectrogram representation of a music signal. The space is modeled statistically by a mixture of Gaussians (GMM). A new signal is classified by considering the likelihoods over all the estimated feature vectors given these statistical models, without constructing a model for the signal itself. Cross-validation tests on two commonly utilized datasets for this task show the superiority of the proposed features compared to the widely used MFCC type of representation based on classification accuracies (over 9% of improvement), as well as on a stability measure introduced in this thesis for GMM. Furthermore, we compare results of Non-negative Matrix Factorization and Independent Component Analysis when used for the approximation of spectrograms, documenting the superiority of Non-negative Matrix Factorization. Based on our findings we give a concept for a complete musical genre classification system using matrix factorization and Support Vector Machines.

Μια Προσέγγιση Ταξινόμησης Μουσικής βασισμένη σε Συνιστώσες

Andre Holzapfel

Σεπτέμβριος 2006

ΠΕΡΙΛΗΨΗ

Παρουσιάζεται ενα καινούργιο σύνολο χαρακτηριστικών για την περιγραφή μουσικής. Περιγράφει την κάθετη δομή μουσικής στηριζόμενο σε Μη-αρνητική Παραγοντοποίηση Πινάκων φασματογραφημάτων. Τα χαρακτηριστικά σχηματίζουν μία βάση φάσματος για μουσικούς ήχους. Αυτές οι βάσεις μοντελοποιούνται στατιστικά ούτως ώστε να παρθούν περιγραφές των χαρακτηριστικών μίας συλλογής ομοίων μουσικών κομματιών. Δείχνουμε την ανωτερότητα του συγκριτικά με MFCC χαρακτηριστικά χρησιμοποιώντας μετρικές απόστασης και ακρίβεια της ταξινόμησης μουσικών ειδών. Επιπλέον, δείχνουμε ότι η Μη-αρνητική Παραγοντοποίηση Πινάκων έχει καλύτερες επιδόσεις στην προσέγγιση ενός δοθέντος φασματογραφήματος από την Independent Component Analysis. Βασιζόμενοι στα ευρήματά μας προτείνουμε την σύλληψη του συστήματος πλήρης ταξινόμησης μουσικών ειδών χρησιμοποιώντας παραγοντοποίηση πινάκων και Support Vector Machines.

DEDICATION

This thesis is dedicated to all people who could better survive without a computer than without a guitar and of course ... meiner Familie, die mich immer unterstuetzt hat.

ACKNOWLEDGMENTS

I would like to thank Yannis Stylianou for continuously supporting my studies and thesis work at University of Crete. Furthermore I would like to thank the German Academic Exchange Service for supporting this thesis work.
I also direct my acknowledgments to Günter Meier for paving the way to my Master Studies.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1
INTRODUCTION

The development of human culture has always been characterized by a different way to use and perceive music. Before the times of music recordings we were able to differ between traditional popular music and the elaborated forms of music composed at royal courts. Their musical structures differed with the traditional forms being more simple and intuitive while the latter had to follow high demands of being elaborated and they had to follow certain rules and conventions of the time. With the rise of the bourgeoisie these rules started to break up. The later symphonies of Ludwig van Beethoven (1770-1827) caused scandals because of the striking simplicity of their themes and the unconventional way to vary them throughout a movement. The composer Bela Bartok (1881-1945) made massive use of elements of traditional hungarian folk music in his orchestral works. But not only the higher forms of music broke the borders, also the traditional folk songs of the black American citizens stepped through a development that led to a harmonically rich kind of music characterized by high instrumental virtuosity. It was the development of Jazz music. Also here walls went down as the society developed. In the early years of the last century it was unacceptable for white Americans to listen to Jazz, their music was characterized by old-time and country music styles. These borders broke throughout the centuries with the most important period being the second world war, new streams in Jazz and Country music arose. Thus a musical genre is always a temporal snapshot of a cultural development. It might be impossible to strictly assign a piece of music to a particular genre.

All these problems would have stayed subject of research in musicology but from the

beginnings of the last century techniques to record and publish musical performances changed the character of perception of music. In the sixties in one of his last interviews the brillant saxophone player Eric Dolphy uttered the phrase: "When you hear music, after it's over, it's gone in the air; you can never recapture it again." Luckily he was wrong. Nowadays almost all recordings are available in digital format, we can listen to it on our computers, we can buy it from the internet. This way each kind of music went out of its traditional place of performance. We enjoy Mozart in the shopping mall and listen to the latest performance of the Rolling Stones at our computer at work. Every kind of music has gone to all the places. And the genres interact and new styles are created.

1.1 Scope of the Thesis

As we have digital forms of the music available we can try to achieve some overview in this growing confusion. We can entrust a computer program with a pre selection of the music available, specifying our idea of musical genres by examples and asking for new examples from one of these genres. In the terminology of pattern recognition we ended up in a classification problem described by a set of data

$$(x_1, y_1), ..., (x_M, y_M) \in \mathcal{X} \times \{1, ..., C\} \tag{1.1}$$

Here $\mathcal{X}$ is a non empty set of patterns or observations x_i and the y_i are the labels specifying a class the observation belongs to. In our case the labels refer to musical genres and the observations are some kind of characteristics suitable to describe the musical piece we have in our collection. As said by David Huron in

[21] such descriptors are referred to as features if they are salient, distinctive and significant for the description task at hand.

In this thesis we are going to evaluate a new kind of feature set for the classification of musical genre. For training a classifier we rely only on the acoustic data without regarding any kind of meta information such as the performing artist or some pre defined labeling. We do not give particular emphasis to the way the classification takes place. We'd rather like to see if this new feature set gives some improvement compared to some standard feature set that is widely used in presented approaches. We furtherly restrict our focus to a feature set that describes what is being referred to as timbre of music in publications on music classification. Even though the concept of timbre is somewhat ill-defined as we will see in later sections we can clearly exclude characteristics of the rhythm or key of the recording from this term. Our approach to get these descriptors is based on component analysis techniques that have found increasing interest throughout the last years. As we will see in the litarature review in Chapter 3, the possibilities of such a feature set have not been explored yet.

1.2 Thesis Outline

The thesis is organized as follows. In Chapter 2 we will outline the findings of auditory scene analysis to connect research in the way human beings perceive sounds to approaches to analyze sounds by using computational means. We will then provide an overview about techniques to decompose acoustic signals into descriptive elements and then refer to a state of the art technique to compute features of such signals. Finally we describe a way to statistically model the distributions of computed features in its space.

In Chapter 3 we give a review of the existing literature in musical content recognition

and in component decomposition techniques to bring out the current state of the art in these fields and to show the missing link between component decomposition and music classification that we will present in this work.

Chapter 4 presents the techniques of Independent Subspace Analysis (ISA) as a way to find spectral representations of the signals under consideration. We present exemplary results of these techniques and outline drawbacks.

Motivated by these findings in Chapter 5 we will bring out significant changes to the Independent Subspace Analysis and present a new kind of analysis system that is going to be proved to achieve more satisfying results than previous ones. In order not to rely only on measures of classification accuracies we also introduce a means of measuring the condition of the statistical models created by the presented system.

Chapter 6 presents the data sets for our experiments and the achieved experimental results. We describe changes to our system that have been motivated by these results and compare our final results with those taken from a baseline system that uses a standard feature set and with previously presented results.

Chapter 7 concludes the findings and shows limitations while Chapter 8 describes ways to take steps to a more accurate music genre classification system.

CHAPTER 2
BACKGROUND

2.1 Auditory Scene Analysis

The founding work in auditory scene analysis has been written by Albert S. Bregman [8]. In this work we find detailed descriptions on how sounds are grouped together to a single sound experience (integration) and when sounds are expected to be perceived as separated elements, a process called segregation. There are in general two kinds of integration: sequential and integration of simultaneous events. Many different factors have been found to have influence on the integration of sounds. Sequential integration is for example influenced by common onset times or the rate at which two pure tones of different fundamental frequency are repeated. Let us just take a simple and interesting example of the latter factor that also shows that the two kinds of integration never happen just on their own. Imagine a sequence of pure tones like shown in the spectrogram of picture 2.1. If the sequence of two tones is played repeatedly there are two possible ways of integrating: either we hear two streams each of them consisting of a single tone that is repeated or we hear a single stream of two alternating tones. The two critical factors which influence the way the sequence is perceived are the speed at which the sequence is presented and the frequency distance δf between the pure tones. Rising speed gives force to integrating the sequence into one stream while rising distance in frequency enforces the segregation into two streams.

Thus we can see that there is an interplay of vertical (i.e. frequency) and horizontal (i.e. time) factors in creating the basic streams of sounds. Things get

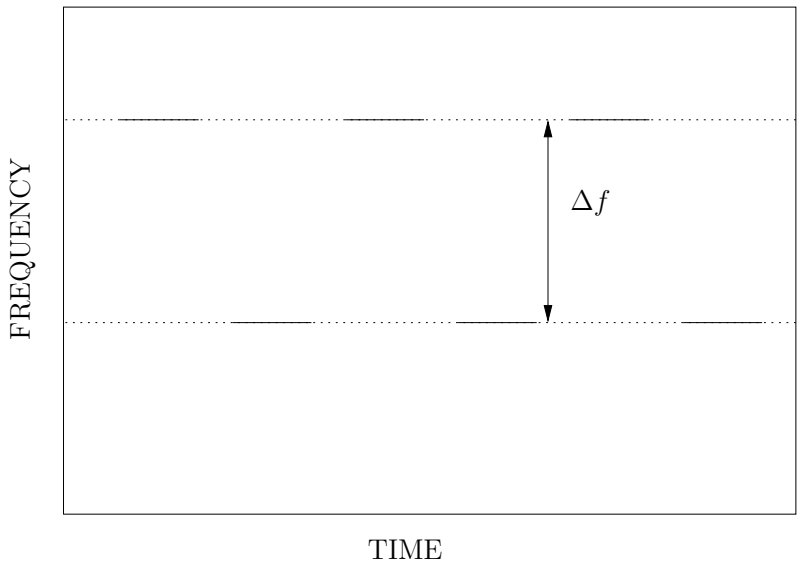

Figure 2.1. Example of stream segregation

of course more complex when making the step from pure to rich tones, that have a specific spectral structure. Then many other factors come into play as well such as grouping by common modulations and harmonic relationships.

In musicology the terms of horizontal dimension is related to the rhythmic and melodic structure of the piece while the vertical organization gives us information about chords, consonance and timbre. As in this thesis we concentrate on descriptors for the vertical structure we have to further clarify which characteristics we can possibly describe and which have to be captured by other kinds of descriptors.

At first descriptors for the vertical structure of music are often referred to as Timbral Descriptors. We have to examine what we mean when using the word timbre. According to the American Standards Association musical timbre refers to features in sounds which enable the listener to distinguish between them even though they have same pitch, loudness and perceptive duration [1]. But as mentioned by Bregman this definition fails because we would not be able to assign a timbre to percussive

instruments that do not have a pitch. There have been many experiments using different kinds of instruments in order to find the characteristics that summed up would give an explanation of the term timbre, for a line out of these experiments see [19]. In the MPEG-7 standard the results of these experiments have been transferred into the timbral descriptors for harmonic, sustained and percussive, non-sustained sounds which are shown in table 2.1. The table reveals a clear misuse of the word timbre in music information retrieval research: the standard features as described in Chapter 2 as well as the features presented in this work (see Chapter 5) are only capable of vertical structure and cannot capture attack times and temporal centroids. Thus we would like to define these features as vertical structure descriptors (VSD) while everything connected to temporal development of the sounds will be defined as horizontal structure descriptors (HSD).

Summing up in terms of auditory scene analysis we develop a feature set to capture vertical properties of the sounds under consideration and we will show how well the characteristics of musical genres can be distinguished by using these characteristics in a pattern recognition framework. Thus our system is capable of finding formant structures of sounds, i.e. spectral envelopes characteristic for a sound independent of its pitch. This characteristic is a strong factor in integrating sounds into a single stream as shown by Bregman, page 89. Nevertheless other important factors such as onsets or melody structures will have to be explored using other methods (see Chapter 7).

2.2 Component Decomposition

In this section we introduce two kinds of components decompositions, the Independent Component Analysis (ICA) and the Non-Negative Matrix Factorization

Harmonic	**Percussive**
HarmonicSpectralCentroid (HSC)	Log AttackTime (LAT)
HarmonicSpectralDeviation (HSD)	TemporalCentroid (TC)
HarmonicSpectralSpread (HSS)	SpectralCentroid (SC)
HarmonicSpectralVariation (HSV)	
LogAttackTime (LAT)	

Table 2.1. Low-level descriptors contained in the timbre description scheme of MPEG-7

(NMF). The theoretical foundations will be shown here while known applications will be presented in the literature review (Chapter 3). In Chapters 4 and 5 we will compare some results of these decompositions and clarify our motivation for using the NMF approach to compute the features used in our system.

Assume we observe k different mixtures of acoustic sources. This can be imagined like recording a conversation between d speakers with k microphones. If we let $\mathbf{x}$ denote this k dimensional vector of observations we can write

$$\mathbf{x}(t) = \mathbf{W}\mathbf{h}(t) \tag{2.1}$$

with $\mathbf{W}$ a $k \times d$ matrix. In general a component decomposition will try to discover the components $(h_1, ..., h_d)$ using the observations and doing some assumptions for the mixing matrix $\mathbf{W}$ and the unknown components. The goal is to find a suitable demixing matrix $\mathbf{W}^{-1}$ that will give us some meaningful components by calculating its product with the observations.

2.2.1 ICA

We find a general definition of independent component analysis in [12]:

Definition 1 (ICA)**.** *The ICA of a random vector* $\mathbf{x}$ *of size* k *with finite covariance* $\mathbf{V_x}$ *is a pair* $\{\mathbf{W}, \mathbf{\Delta}\}$ *such that*

(a) $\mathbf{V_x} = \mathbf{W}\mathbf{\Delta}^2\mathbf{W}^H$ *where* $\mathbf{\Delta}$ *is diagonal real positive and* $\mathbf{W}$ *is full column rank* d*.*

(b) the observation can be written as $\mathbf{x} = \mathbf{Wh}$ *where* $\mathbf{h}$ *is a* $d \times 1$ *random vector with covariance* $\mathbf{\Delta}^2$ *and whose components are "the most independent possible", in the sense of the maximization of a given* ***contrast function****.*

(c) the columns of $\mathbf{W}$ *have unit norm*

(d) the entries of $\mathbf{\Delta}$ *are sorted in decreasing order*

(e) the entry of largest modulus in each column of $\mathbf{W}$ *is positive real*

Items (c),(d),(e) in definition 1 are due to the indeterminacy of the result taking only (a) and (b), because without these constraints we have defined an equivalence class of decompositions as $\mathbf{W}$ and $\mathbf{\Delta}$ can be scaled and permutated and they would still represent an ICA according to the first two items. The vector $\mathbf{h}$ will be called source vector subsequently. As it has been shown in [12] $\mathbf{h}$ is restricted to have at most one Gaussian component.

When we consider $\mathbf{h}$ being a random variable in $\mathbb{R}^d$ with a probability density function $p_h(\mathbf{u})$ then its components are statistically independent if

$$p_h(\mathbf{u}) = \prod_{i=1}^{d} p_{h_i}(u_i) \tag{2.2}$$

We seek a distance measure $\delta(p_h, \prod_{i=1}^{d} p_{h_i})$ to minimize in order to get statistically independent components and find by using the *Kullback-Leibler divergence* the average

mutual information of **h** as a **contrast function**:

$$I(p_h) = \int p_h(\mathbf{u}) \log \frac{p_h(\mathbf{u})}{\prod p_{h_i}(u_i)} d\mathbf{u} \ , \ \mathbf{u} \in \mathbb{R}^d \tag{2.3}$$

As the vectors to be analyzed by the ICA are assumed to have zero mean and unit covariance the input data has to be preprocessed. This preprocessing is often referred to as *whitening*. For densities of variables with unit covariance (or in general with invertible covariance matrices) we can calculate the **negentropy**:

$$J(p_h) = S(\Phi_h) - S(p_h) \tag{2.4}$$

with the zero-mean Gaussian pdf of **h** with the covariance $\mathbf{\Delta}^2$ as

$$\Phi_h(\mathbf{u}) = (2\pi)^{-\frac{d}{2}} |\mathbf{\Delta}^2|^{-\frac{1}{2}} \exp\{-\mathbf{u}^H (\mathbf{\Delta}^2)^{-1} \mathbf{u}\}/2 \tag{2.5}$$

and the **differential entropy**

$$S(p_h) = -\int p_h(\mathbf{u}) \log p_h(\mathbf{u}) d\mathbf{u} \tag{2.6}$$

Using equation 2.4 in equation 2.3 we can write mutual information as

$$I(p_h) = J(p_h) - \sum_{i=1}^{d} J(p_{h_i}) + \frac{1}{2} \log \frac{\prod \mathbf{\Delta}_{ii}^2}{det \mathbf{\Delta}^2} \tag{2.7}$$

Negentropy has the advantage of being invariant under linear transformations and gives us a measure of non-Gaussianity of the distribution of the random variable in question. The reason for maximizing non-Gaussianity lies in the *Central Limit Theorem* which simplified states that a sum of an increasing amount of independent

identically distributed variables will approximate a normal distribution. Taking note of the fact that the Gaussian distribution has the largest entropy among all random variables distributed with equal variance, by minimizing mutual information we are going to find a demixing matrix $\mathbf{W}^{-1}$ that is going to rotate the variables in $\mathbf{x}$ in the directions which are least Gaussian.

Relation 2.7 provides a mean to approximate the mutual information. For this the authors of [12] expand then p_h in the neighborhood of Φ_h by using an edgeworth expansion of the pdf of order four. The approximation of distributions is well explained in [48]. If done so the authors implement an algorithm to minimize function 2.7 to find the matrix $\mathbf{W}$. They cancel at first the last term of 2.7 which they proved to be equivalent to standardize the data. This steps incorporates second order moments only. Then they compute sequential orthogonal transforms for which then higher order cumulants are used.

The ICA algorithm used in the experiments of this thesis approximates the mutual information is minimized in a different way because using a cumulant based approach has been shown not to be robust to data outliers. The authors use

$$J(p_h) \approx [E\{G(p_h)\} - E\{v\}]^2 \tag{2.8}$$

as an approximation to negentropy. v is a Gaussian zero mean unit variance distributed variable and $G(\cdot)$ is chosen to be a function that does not grow "too fast", for example

$$G_1(u) = \frac{1}{a_1} \log \cosh a_1 u \ , \quad G_2(u) = -\exp(-u^2/2) \tag{2.9}$$

with $1 \leq a_1 \leq 2$. It has been proven by the authors that a wide range of functions provide consistency [22]. In this publication the authors show as well that this approach is equivalent to minimizing mutual information.

2.2.2 NMF

The NMF bases on the same generation model as described in equation 2.1. It takes as a starting point an amount of observations $(\mathbf{x_1}, ..., \mathbf{x_M})$ summing them up in a matrix $\mathbf{X} \in \mathbb{R}^{k \times M}$ containing the observations in its columns. Then we have to solve

$$\mathbf{X} \approx \mathbf{WH} \tag{2.10}$$

with the mixing matrix $\mathbf{W} \in \mathbb{R}^{k \times d}$ and the resulting component matrix $\mathbf{H} \in \mathbb{R}^{d \times M}$. The additional constraint that is the reason for the name of this approach is that all the three matrices have to be non-negative. In contrast to the ICA concept no assumptions about statistics are made and the approximation to the observation matrix is performed by minimizing either the square of the euclidian distance

$$||\mathbf{X} - \mathbf{WH}||^2 = \sum_{i,j} (\mathbf{X}_{ij} - \mathbf{WH}_{ij})^2 \tag{2.11}$$

or

$$D(\mathbf{X}||\mathbf{WH}) = \sum_{i,j} \left(\mathbf{X}_{i,j} \log \frac{\mathbf{X}_{i,j}}{[\mathbf{WH}]_{i,j}} - \mathbf{X}_{i,j} + [\mathbf{WH}]_{i,j} \right) \tag{2.12}$$

The latter cost function will be referred to as divergence as it is not symmetric. Both functions are lower bounded by zero. This is obvious for the Euclideann cost function

and for the second function it can be shown with $x, y \geq 0$

$$\begin{aligned} x \log \frac{x}{y} - x + y &\geq 0 \\ \log \frac{x}{y} + \frac{y}{x} &\geq 1 \\ \log \frac{y}{x} &\leq \frac{y}{x} - 1 \end{aligned}$$

which is true for $\frac{y}{x} > 0$ and holds with equality when $\frac{y}{x} = 1$. Thus all the $i \times j$ summands in 2.12 must be bigger or equal than zero. For the case that $\sum_{i,j} \mathbf{X}_{i,j} = \sum_{i,j} \mathbf{WH}_{i,j} = 1$ the second cost function reduces to the relative entropy. Algorithms for performing NMF have been given in [25], implementations are also available for matlab [20]. These algorithms are steepest decent algorithms with multiplicative update rules. Thus there is no step size parameter to be determined to perform the parameter updates. The updates when using the divergence criterium function are

$$\mathbf{H}_{\alpha,j} \leftarrow \mathbf{H}_{\alpha,j} \frac{\sum_i \mathbf{W}_{i,\alpha} \mathbf{X}_{i,j} / (\mathbf{WH})_{i,j}}{\sum_\beta \mathbf{W}_{\beta,\alpha}} \tag{2.13}$$

$$\mathbf{W}_{i,\alpha} \leftarrow \mathbf{W}_{i,\alpha} \frac{\sum_i \mathbf{H}_{\alpha,i} \mathbf{X}_{i,j} / (\mathbf{WH})_{i,j}}{\sum_k \mathbf{H}_{\alpha,k}} \tag{2.14}$$

A geometric interpretation of the NMF is that the columns of the matrix $\mathbf{W}$ give the generating vectors of a simplicial cone that contains all columns of the target matrix $\mathbf{X} \approx \mathbf{WH}$. The positivity constraints only require the cone to be in the positive orthant as shown in figure 2.2. This shows that the constraints might be insufficient and as we will see in the literature review much effort has been made to further constrain the problem depending on the specific application. Nevertheless we recognize that the columns of the matrix $\mathbf{W}$ represent a basis of the columns space of the matrix $\mathbf{X}$ which are linearly combined by coefficients contained in the columns

of the matrix **H**. Vice versa we can interpret the rows of **H** a basis of the row space of **X** with its coefficients in the rows of **W**.

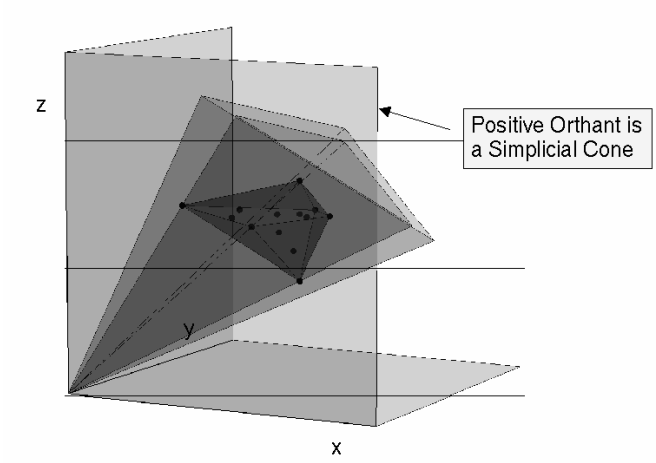

Figure 2.2. The columns of **W** define a simplicial cone containing the columns of the approximated matrix, simplified example in three dimensions, from [14]

2.3 MFCC

The Mel-frequency Cepstral coefficients (MFCC) are a commonly used set of features to describe the short time characteristics of a signal. They have also been successfully applied in the field of music genre classification as we will see in Chapter 3. Because of this reason we decided to implement a system based on this features to compare the system presented in this thesis with this baseline system.

The term cepstrum was invented back in 1963 by Bogert, Healy and Tukey [7]. It originates from the idea to separate convolutive elements of a time domain signal. This is done by using the sequence of operations Fourier transform → Spectral density → logarithm → filtering → inverse Fourier transform (referred to as Cepstrum). This way the convolutive elements become factors of a multiplication in the Fourier

transform and then additive elements by applying the logarithm. By applying a low pass on this signal we can smoothen the spectral envelope as far as the additive components differ from each other in the resulting cepstrum. Figure 2.3 illustrates this process of filtering which is also known as liftering applied to the deconvolution of vocal tract and excitation in human speech. In [13] the Cepstral coefficients received

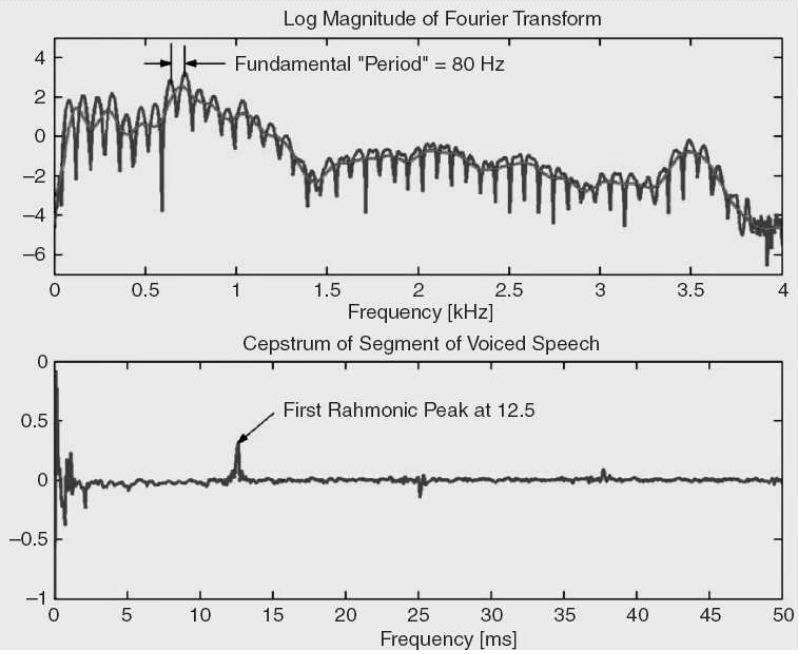

Figure 2.3. Example of liftering applied to a speech signal. The upper graph shows the unliftered and the liftered log magnitude of the frequency spectrum. The smooth line is the liftered spectrum. As we can see in the lower graph the cepstrum before liftering has two well separated regions: the lower ones caused by the spectral envelope of the vocal tract and the higher so-called rahmonics which have been caused by the excitation with a period of 80 Hz. Figure taken from [32]

from the described procedure have been shown to perform best in comparison with other signal representations when applied to speech recognition. They also applied a non-linear frequency scale called the Mel scale. The characteristic of this scale is a linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz. It is motivated by the critical band rates of the inner ear and differs by a linear factor

from the bark scale, see section 5 for more details. A block diagram of the calculation procedure of these features is shown in figure 2.4. Here the general Fourier transform is suplied by a Short Time Fourier Transform of the windowed time signal. The form of the filterbank is shown in figure 2.5. The logarithmic outputs X_j of these filters are then inverse transformed by applying a DCT transform as shown in equation 2.15.

$$MFCC_i = \sum_{j=1}^{20} X_j \cos\left[i(j-\frac{1}{2})\frac{\pi}{20}\right], \quad i = 1, ..., N_{cep} \tag{2.15}$$

with N_{cep} the number of Cepstral coefficients and $j = 1, ..., 20$ the index addressing the triangular filters.

Figure 2.4. Procedure for calculating MFCC

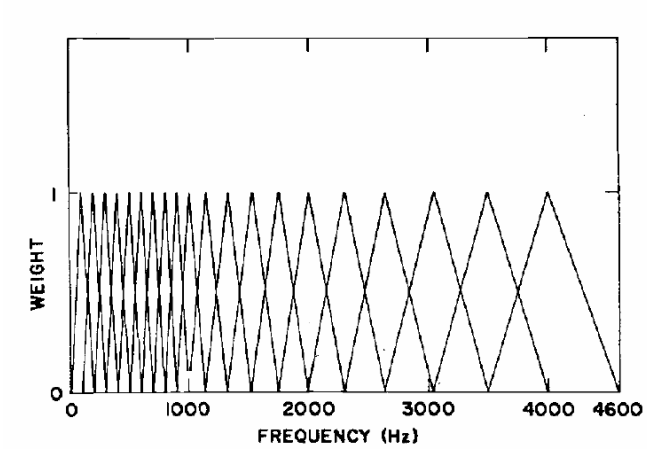

Figure 2.5. Filters for generating th Mel-scale Cepstral coefficients

2.4 Gaussian Mixture Models

The system presented in this thesis constructs a statistical model for features taken from the non-negative matrix decomposition. This model neglects the temporal order of the features found throughout songs and thus follows the bag of frame model as mentioned in [28]. This way songs with a similar group of feature vectors in different order are judged to be similar. For modeling the distributions of the vectors a Gaussian Mixture Model (GMM) was applied. A GMM models a density with a linear combination of N_g Gaussian distributions:

$$p(\mathbf{x}|\Theta) = \sum_{i}^{N_g} \alpha_i p_i(\mathbf{x}|\mu_i, \mathbf{\Sigma}_i) \tag{2.16}$$

so that the parameter of the model are $\Theta = (\alpha_1, \mu_1, \mathbf{\Sigma}_1, ..., \alpha_{N_g}, \mu_{N_g}, \mathbf{\Sigma}_{N_g})$ and the parameters $(\alpha_1, ..., \alpha_{N_g})$ sum to one. The parameters of the distributions have been found by a maximum likelihood estimation using the Expectation Maximization algorithm (EM), see appendix B.

2.4.1 Initialization

The initialization performed as default in the used GMMBayes toolbox [33] consists of a kmeans clustering of a randomly chosen number of vectors from the training set which is equal to the number of Gaussian components to be estimated later on by the EM algorithm. This method has been found insufficient as it produces varying accuracies on the same data on different runs and also suboptimal average accuracies in the cross validations. Because of this two alternatives have been tested. The first is initializing several clusterings on different vectors using kmeans. The algorithm is stopped after some iterations for each vector selection and the likelihoods

of the training data on the produced clusters are computed. The covariances are calculated from the data points belonging to the clusters and the weight of a cluster is defined by the fraction of training data belonging to the according cluster. For the clusters that produced the highest likelihood we then perform again a kmeans initializing with the mean vectors of the clusters but now taking a higher iteration number.

As this procedure is still somewhat greedy it was tried to introduce a deterministic procedure in order to get exactly the same results when the same training data is used. For this reason we used a clustering approach referred to as Gaussian means clustering presented by Greg Hamerly in [18]. This approach starts from a single cluster with the mean vector being the statistical mean of the data. As Gaussian distribution of the data is assumed later on in the EM algorithm we can make afford of this assumption also in the clustering. Thus in each iteration the algorithm only splits one cluster into two whose data seem to be least Gaussian distributed. For testing this condition the Anderson-Darling statistic is used as a measure. For this the data is being projected onto one dimension and converted to zero mean and unit covariance. Then the values z of the $N(0,1)$ cumulative distribution function are computed on the ordered data. Then the statistic is

$$A^2(Z) = -\frac{1}{M}\sum_{i=1}^{M}(2i-1)\left[\log(z_i) + \log(1 - z_{M+1-i})\right] - M \tag{2.17}$$

where M is the number of data points. The splitting of one cluster is then determined by the direction of the main principal component of the data belonging to the cluster and the position of the new center candidates on this line is determined by the size of the according eigenvalue. Then again the statistical test is performed to see if the splitting produced more Gaussian like data clusters. This way the algorithm produces

a deterministic clustering that proceeds until further splittings are rejected.

As mentioned in [16] the above described method suffers problems from overlapping data clusters. The algorithm overestimates the number of clusters then. As this is the case for our data and in any way we want to define a number of clusters we stop the algorithm when a specified number of centers is reached.

A comparison of the two methods shows that running several kmeans and choosing the clustering with the highest likelihood clearly improves the results in relation to a single run of kmeans on randomly chosen vectors. But for a high number of runs this method gives in the average equal results as the Gaussian means method (method 2). The computational cost of running many clusterings is higher though and the number of kmeans initializations depends also on the number of training data points. Because of these reasons the Gaussian means method was preferred because it represents a fast and deterministic way to initialize the EM algorithm for the GMM estimation.

CHAPTER 3
LITERATURE REVIEW

This Chapter presents the recent findings in the field of music classification and in component decomposition techniques. We will work out the state of the art and show the need for a new set of characteristics to classify music. We describe which ways of finding meaningful components in music have been explored and in which applications they have been used. Note that in this Chapter despite the discrepancy of the term timbral feature for characteristics based on MFCC (see section 2) we use this terminology here as it is common throughout the described articles.

3.1 Music Classification

As mentioned in the introduction musical genres are the products of a cultural development. Thus a founding work that has been done only recently is to define a ground truth to rely upon. An approach to create a reliable basis for performing tests has been done in [2] where a large scale evaluation among web users has been initiated. Due to copyright restrictions though this data is not available as a collection of audio files. A way to facilitate at least comparability of results is to use data which is publicly available. A database which is available for non-commercial projects has been used in the contest for music information retrieval in ISMIR 2004[1]. A very important step in music similarity research has been done in [34] where from many publications a standard system for computing music similarity based on vertical structure descriptors is derived and differences between existing approaches are shown. They experimentally examine the systems and end up with the conclusion

[1] http://ismir2004.ismir.net

of the existence of a recognition precision upper border for all the existent systems no matter of how the parameters are changed. The authors describe the structure of the existent standard system as shown in figure 3.1. Block 1 (windowing) is usually

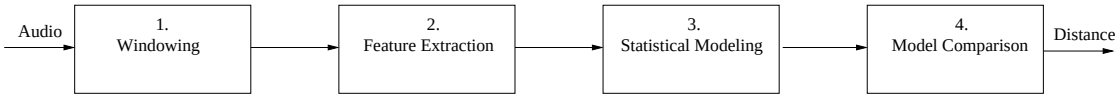

Figure 3.1. General Standard System for Music Similarity based on VSD

the multiplication of the signal with a window of length 20 to 50 ms and an overlap of 25% to 50%. Block 2 typically contains a STFT and a conversion to MFCC where usually 8 to 20 coefficients are used. Block 3 (Statistical Modeling) tries to compute a statistical model of the MFCC distributions where typically either K-Means or GMM are used where the GMM parameter can be learned by using Expectation Maximation algorithm. The model comparison can be done using Earth Mover's distance, sampling or likelihood approximation. The most important results are the following: Using the EMD, taking the distance using Mahalanobis performs worse than Kullback Leibler. It is assumed that the reason for this lies in the property of Mahalanobis to judge two Gaussians having the same mean but different covariance as having zero distance. A Monte Carlo Sampling approach followed by a maximum likelihood choice for the distance shows similar results as EMD with Kullback Leibler while it is computationally much more expensive as mentioned in [28]. Also they found that a STFT window longer than 30ms decreases the precision. They propose a spectral flatness measure in order to filter out percussive frames which are poorly modelled by MFCC. They also propose computing a spectral contrast measure for each Mel subband which gives the difference between the peak and the noise floor in each band. A very significant finding is also that in contrast to sound recognition there is no enhancement of results when incorporating HMM to model

temporal structure. Comparing the results of all their experiments they end up with the conclusion that a system for music similarity based on MFCC features hits a glass ceiling of recognition performance at about 65 % on the dataset used. This states the need for finding a set of features superior to MFCC.

Also Giorgos Tzanetakis has presented an approach for classifying pieces of music into a hierarchical genre structure [44]. He proposes usage of a threefold feature set containing features for timbre, rhythm and pitch. The tested features contain the following:

1. Timbre: spectral centroid, spectral rolloff (below which frequency 85% of the energy is concentrated), time zero crossings, MFCC; all of these are described by means and variances throughout a texture window of length one second which was found to be optimal. These result in 19 features describing timbre.
2. Rhythm: An algorithm for pitch detection is applied for getting rhythm descriptors using larger periods. A discrete Wavelet transformation is applied to decompose the signal into bands, in each of them an envelope is extracted. These are then added and the peaks of the autocorrelations are found. Beats are detected in a range from 40 to 200 beats per second. A three second window is required therefor. The resulting feature vector consists of different amplitude ratios and periods and the all over sum of the beat histogram.
3. Pitch: frequencies are transferred to MIDI notes, and also a wrapped set mod 12 is computed. Different characteristics of these folded and unfolded histograms are put to the pitch vector.

The all over feature vector has 30 dimensions. As the pitch and rhythm features have to be computed for the whole file the musical pieces have to be homogenous. The features found most effective are the sum of the beat histogram, the peak of the folded pitch histogram (which represents the most significant tonal step in a piece of music), the variance of the spectral centroid and the mean of the first MFCC coefficient. This represents a step beyond the framework as described in [34] as the horizontal dimension of music is considered as well. An important finding is the length of the applied timbre window, which is almost the same as the window length found to be optimal in our system as well, even though the used features differ. Nevertheless we have to mention that the idea of combining vertical and horizontal structural descriptors could not outperform the system presented by Dan Ellis [28] which uses MFCC as features. We could assume the reason for this in the elaborated classifier based on a Support Vector Machine (SVM). As we will see in our conclusions first experiments with SVM have shown improvement to a simple Maximum Likelihood classification as well. Apart from that they show that modeling a single song instead of artist or genres is the most promising approach as it is quite reasonable that the characteristics of the song do not change to much within the same song. They use only a timbral description based on the means and variances of the MFCC throughout a song thus regarding songs with the same MFCC in a different order as equal. They compare different ways to model the song level description: using Mahalanobis measure on the unwrapped vector of variances and means, using Kullback Leibler describing the means and variances of the song as single Gaussians, or a more complex GMM calculating the distances using a Monte Carlo approach. They use a Kernel based on the calculated distances between two songs $D(X_i, X_j)$:

$$K(X_i, X_j) = e^{\gamma D(X_i, X_j)} \tag{3.1}$$

This kernel satisfies the Mercer conditions when the symmetric Kullback Leibler divergence is slightly modified as shown in [30]. An SVM algorithm which is specialized for a multiclass decision [39] was chosen for classifying a song to one of 18 artists. They decrease training times by using song instead of artist level features. Comparing SVM with a kNN classifier and all the possible distance measures on song and artist level features they find that SVM on song level features is best.

In [37] approaches following the model described by Aucouturier ([34] and above) are compared with their own method which is based on computing a sonogram on a bark scale and computing simple frequency histograms which are compared by taking an euclidian distance measure. Their method seems to outperform the more complex approaches. A crucial point seems to be the differing feature structure as also a model of the outer and inner ear is applied in order to get values for the amplitude spectrum in the bark frequency bins which have a closer relation with the actually perceived loudness. The outer and middle ear model has been developed by Terhardt [43] and its frequency response is shown in figure 5.2. The influence of the inner ear model consits of a spectral masking (see Chapter 5 for more details on the psychoacoustic model). They also introduce a user friendly interface to compare similarity measures using Self Organizing Maps (SOM) [23] which map the high dimensional feature space onto the plane while preserving the existing distance relations as far as possible. They give a more detailed overview of their own approach in [36]. Their frequency histogram computes how often in a certain frequency band a certain loudness level has been reached. Along with that they present a Period histogram which captures information about the beat of the song. Their approach is not very different from that taken by Tzanetakis and in order to judge which is more suitable experimental comparison is necessary. Where in [44] a wavelet transform is

used the authors of this approach use comb filters.

The last approach we describe went again quite a different way and puts emphasis on an automatic feature selection. It was presented by Bergstra [3] for the contest in music information retrieval MIREX in 2005 and figured out to be the best performing system. Their approach is to divide a piece of music into non-overlapping frames of 13.5 seconds length each and to calculate a big set of features in each of those frames. Note that this block length does not lead to optimal performance but is due to the limitation of the MIREX contest that the system has to be trained on the database within 24 hours. The authors mention in [4] that the optimal block length of 3.5 seconds would have exceeded this limit by far, even without doing cross validation. The feature vector contains means and variances of all the calculated features which are:

- 256 RCEPS (Real Cepstral Coefficient, on linear frequency scale)
- 64 MFCC
- 32 Linear predictive coefficients
- 32 Low-frequency Fourier magnitudes
- 16 Rolloff
- 1 Linear prediction error
- 1 Zero-crossing rate

This results in a feature vector with 804 dimensions. Because of this high dimensional feature vector they rely on AdaBoost as classification method. This enables them to detect the relevance of the incorporated features in the same time as getting classification results. The weak learners that are part of the AdaBoost are simple two leaf trees. Even with this simple classifier they outperformed all other systems in genre classification. This leads to the conclusion that the information content of

other feature sets must have been suboptimal. They also mention their experimental result that the classes seemed not to be well separated in their feature space, a conclusion that agrees with our observations both on MFCC and our own feature sets.

3.2 Component Decomposition

In his publication Eronen describes the use of a three stage HMM to model onset, steady and decay phase of musical instruments [15]. He uses MFCC and ΔMFCC as features and rotates them by using ICA. The features are contained in a matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M]$ with $\mathbf{x}_i$ containing the MFCC and ΔMFCC coefficients. The training of the HMM is not performed by involving the *Baum-Welch* algorithm to do a baseline Maximum Likelihood training. Instead a discriminative training is incorporated because the authors assume that the statistical model fits the data poorly. So they involve a Maximum Mutual Information method which tries to maximize the ability to distinguish between the observation belonging to the particular class and the other ones. The mutual information is approximated to make the algorithm feasible. They showed that the classification was significantly better with the usage of the ICA. We will see in Chapter 4 that this approach is similar to the Independent Subspace Analysis.

Even though there have been many more promising publications for the application of ICA for the classification of separated instrumental sounds or other sounds as well things get more difficult when the sound we are interested in is contained within a mixture of other musical sounds. In [45] it was tried to separate a singing voice from a mixture by applying either NMF or ICA. They judge the perceptual quality

of separation better in case of using NMF than in case of using ICA. At first a vocal/non vocal classification on a piece of music is applied by using a feature vector built from MFCC, perceptual linear prediction coefficients and log frequency power coefficients in a Support Vector Machine classifier. Then they apply either ICA or NMF for determining components from an amplitude spectrum. The number of the needed components is estimated in a pre processing step which after whitening the signal applies a PCA and calculates for a chosen $\phi \in [0, 1]$

$$\frac{\sum_{i=1}^{d} e_i}{\sum_{i=1}^{k} e_i} \geq \phi \tag{3.2}$$

where ϕ represents the percentage of information kept and k is the initial matrix dimension, $\mathbf{e}$ are the eigenvalues and d gives the component number used in the later decomposition stages. This approach has motivated the determination of components as used in our system (see Chapter 5).

Many other publications seem to certify that the application of the NMF yields better results when the observed signal is a musical piece containing several instruments. T. Virtanen has been doing assumptions why NMF seemingly outperforms ICA/ISA in separation of sound sources [47]. He claims that exactly the independence assumption could be the weak point. This is because in music there are harmonical dependencies in the vertical structure and temporal dependencies. The latter are created because several instruments have more or less synchronous onsets because they are following the same rhythm. In his algorithmic approach he assumes that sound components consist of a finite length convolution of an onset parameter and the DFT of a component so that the whole spectrogram can be written as a summation of these convolved elements. It is also mentioned that a psychoacoustic loudness model improves the performance of his system. So he does not use the standard mean

square error measure to determine the quality of the approximation but uses a diagonal weighting matrix in order to get an error according to the perceived loudness in the critical bands. Also he introduces sparsity into the temporal weights.

Also Paris Smaragdis [41] extends the original idea of NMF. He incorporates time dependent changes in the components. The goal is to separate the distinct elements of a drum set. The drum set includes bass drum, snare and hi-hat. These components have been found remarkably well by the approach. This approach tries to approximate the amplitude spectrogram by

$$\mathbf{X} \approx \sum_{t=0}^{T-1} \mathbf{W}_t \cdot \overset{t\rightarrow}{\mathbf{H}} \tag{3.3}$$

This method approximates the TFD $\mathbf{X}$ by a convolution of a spectral temporal basis $\mathbf{W}$ with temporal components $\mathbf{H}$. For all the matrices the constraint of non-negativity has to be fulfilled. The approximation in this model can be performed by incorporating slight changes to the algorithm given in [25]. In [42] Smaragdis introduces a heuristic to make the rows of $\mathbf{H}$ more sparse in order to improve the separation of components found in a mixture of instruments. He shows that $\mathbf{W}_i$ contain harmonic series describing the musical notes, a finding also confirmed in [46].

An extension of the algorithm presented in [41] is shown in [40]. There the convolution is taking place in temporal and in frequency direction. Update rules for the least squares error and the divergence error functions are presented where the algorithm by Smaragdis is a special case for the frequency span of the convolution equal to zero. It seems that this algorithm is preferable to the one presented by Smaragdis as it implicitely solves the problem of grouping the notes created by different pitches

of one instrument. Nevertheless the structure is even less practicable for music similarity measures as the ith component is now presented by the T columns of the W_t and also by the ith row in each of the H_f.

3.3 Conclusions

For the applications of music classification it has been shown that the existing standard feature set (MFCC) for the description of sound characteristics cannot exceed a certain upper bound when described by any kind of statistical model. Even though there have been attempts to include also features describing the horizontal dimension of music it is necessary to explore new kinds of features for the vertical dimension in order to surpass this upper bound in order to pave the way for better performing music classification systems.

A possible approach is to find meaningful features by decomposing a signal. ICA seems a promising method for the classification of short duration sounds or for the recognition of independent sources in a mixture such as the separation of speakers in a cocktail party. But for musical mixtures NMF finds more meaningful sources as we have seen in the reviews. Even though we have to take note of the fact that NMF might have to be further constrained and introduction of sparsity and the inclusion of the temporal dimension might also improve the approach, we can conclude that a horizontal description computed by NMF could be a good starting point for a music genre classification system.

So in Chapter 4 we will refer to a system that has been presented in order to find representative elements in sounds by incorporating ICA. In the following chapters we will introduce changes motivated by the recent research results. These changes will include substituting ICA by NMF, usage of psychoacoustic models and a statistical model for the computed features.

CHAPTER 4

INDEPENDENT SUBSPACE ANALYSIS

The term Independent Subspace Analysis was coined by Michael Casey in his PhD thesis [10]. He develops the theory that sound generally consists of invariants that are caused by the objects that produce the sound. According to his description sound objects consist of two kinds of invariants:

1. Structural invariants, either of spectral type describing things like material or the size of objects or of excitation type like scraping or striking a surface
2. Transformational invariants are functions that represent higher order combinations of structures.

These invariants have the group property. Let us recall the definition of a group:

Definition 2 (Group)**.** *A group is a set of elements together with a binary operation that together satisfy the four properties of closure, associativity, identity property and inverse property for each element.*

Based on Lie groups a theory is developed in which independent features in a sound are altered by transformations that follow the group properties. These transforms then represent changes in shape or size of the objects creating the sounds. In order to find these independent features Independent Component Analysis is suggested.

In Casey's thesis the approach of ecological acoustic research plays an important role, because in this field acoustic streams have been described by object structures like the ones described above. The goal is now to find appropriate signal descriptions and processing methods to find these invariants that describe the sound objects. If the assumption holds that these invariants describe the physical form of the sound

sources along with description of their combination we could have a good tool at hand also for a representation of musical signals. In the following we will see the suggested approach that was developed by Michael Casey in his patents and publications.

4.1 System Description

In [9] a system for general sound classification is presented and applied to environmental sounds as well as music. The system applies the theory presented in [10] in a framework referred to as Independent Subspace Analysis. The general idea is to describe the space of the sound object invariants by a dimensionality reduction of the spectrogram of a sound based on SVD and ICA, see figure 4.1. As we can consider the variables in the frequency bins of the spectra as random variables the spectrogram is also referred to as Time Frequency Distribution (TFD). This approach implies that the individual frequency bins can be considered the separate observation signals in the variable $\mathbf{x}$ from equation 2.1.

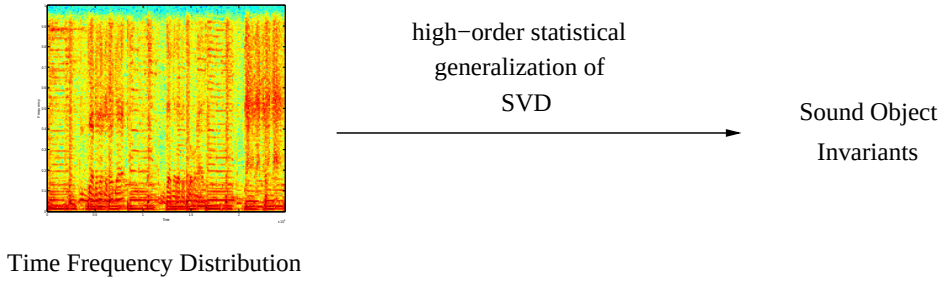

Figure 4.1. The basic idea of ISA

The output of the ISA consists of features that have been projected to a space that has been found characteristic for the sound class under consideration. In figure 4.2 we get a more detailed overview of how these base projection is being computed. A short time Fourier transform is applied to the signal and then the normalized

power spectrum on a db scale is calculated. Several of these spectral envelopes are then put together into a spectrogram $\mathbf{X}$ and processed by a dimensionality reduction based on SVD and ICA. The sound is assumed to remain stationary within this TFD and the length is mentioned to lie between half a second up to 30 second depending on the signal. The SVD performs the reduction of the dimensionality to the desired number of components d by factoring

$$\mathbf{X} = \mathbf{USV}^T \tag{4.1}$$

where $\mathbf{U}$ contains the row and $\mathbf{V}$ the transposed column basis of $\mathbf{X}$. The number of components is determined in analogy to equation 3.2 using the elements of the diagonal singular value matrix $\mathbf{S}$ instead of the eigenvalues. In [9] the number of components for this application is mentioned to range from 3 to 10. Optionally the calculated column bases are now rotated into directions that are maximally statistically independent by ICA. The projected features are now calculated by multiplying the spectrogram matrix with the spectral base matrix.

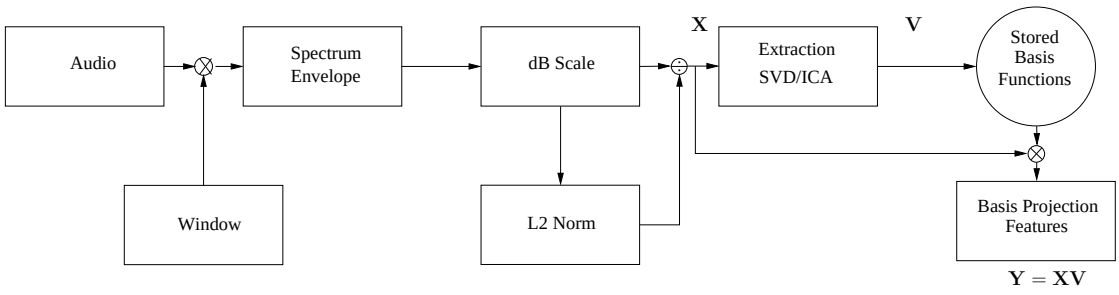

Figure 4.2. Computation of Spectral Bases and the projected features

In order to get a classifier for the sounds the next step is training Hidden Markov Models (HMM) on the projected features. This results in a HMM for each

of the sound classes. For classifying a sound it is then at first projected against the classes' spectral bases. Then the sound is assigned to the class for that its HMM received a Maximum Likelihood Score.

4.2 Evaluation

In order to evaluate the approach described above we relied on the MPEG-7 reference implementation as referred to in [29]. This provides us with matlab code for the computations of the TFDs and with an interface to some ICA algorithms and an HMM toolbox. When applying the algorithm to the speaker gender classification by using the supplied example files the accuracy was high, but using a four class subset of music database 1 with examples from the four classes classical, disco, metal and rock the accuracies did not exceed 60 % of correct classification (see table 6.1 for details of the database). This is contradictory to the classification performance presented in [9]. The reason could be that the authors might have chosen a too simple data set but as they do not reveal any details of their data no definite conclusion can be drawn. The length of the audio samples used in our experiments was 30 seconds and as such the same as in [9]. Changing the number of components for the SVD to compute and choosing different numbers of states in the HMM did not change the picture.

In order to get a more detailed insight into the qualities of the decompositions we integrated a decomposition based on NMF as well and compared the results of the algorithms.

We will have a look at some exemplary results now. The most important result is that the approximation is much better in the sense of mean squared error for the NMF than for ISA independent of the kind of signal under consideration. Figure 4.3 shows

a TFD for a utterance taken from the TIMIT speech corpus. On the abscissa the phoneme alignment of the file is shown while the ordinate has a logarithmic division using eight bands per octave and the values shown are in Hz. Figures 4.5 and 4.4 show the decompositions computed by the two algorithms, using three components in each case. The NMF approximation preserves much more of the temporal structure and in comparison to the ISA approximation harmonic structures in the upper frequency regions are still recognizable.

As an example for a decomposition of a music signal we will show a 2 seconds excerpt from a heavy metal piece. The limitation to a short snapshot has been motivated by the idea of the Independent Subspace analysis as described by Casey in [11] where the timbre of a sound is considered stationary within a certain temporal range. A change in timbre in music could be caused by the beginning of a vocal part that was not present at the beginning of the mixture or by the start of an instrumental solo. An exact determination of the optimal length of this timbre window for NMF is outlined in section 6. Also for music the NMF seems to yield superior results compared to the ISA decomposition.

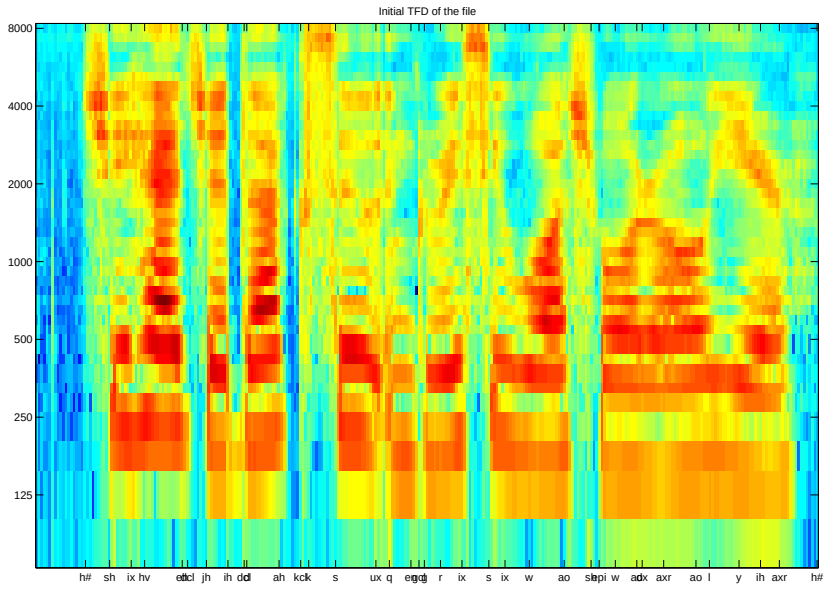

Figure 4.3. Input TFD for a speech signal

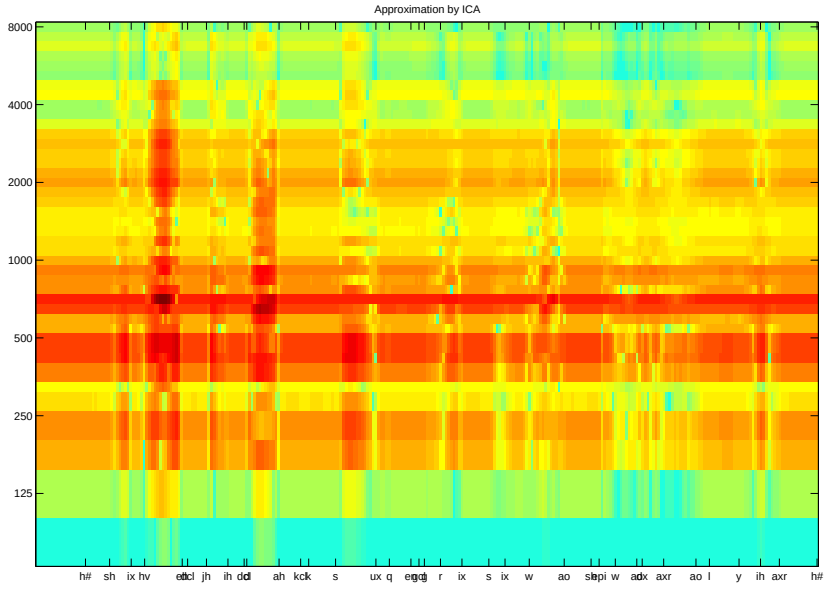

Figure 4.4. Approximation of the TFD in figure 4.3 computed with three ISA components

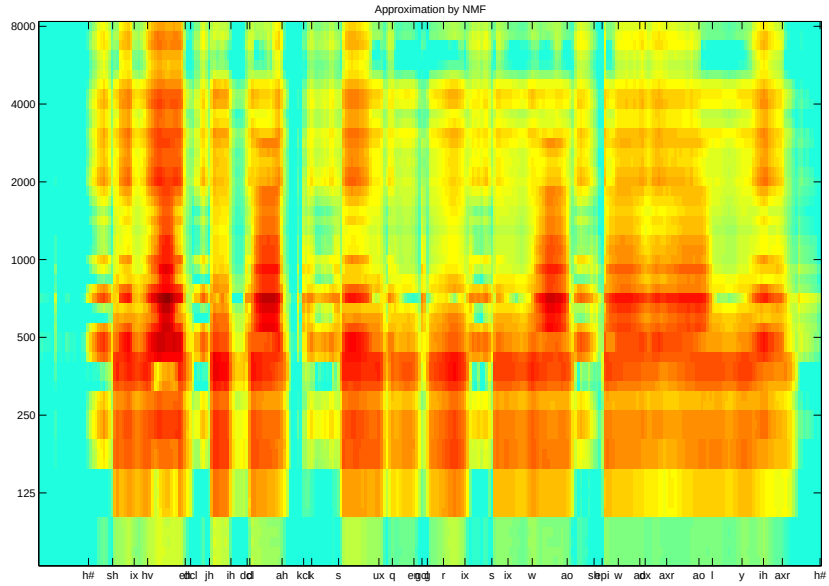

Figure 4.5. Approximation of the TFD in figure 4.3 computed with three NMF components

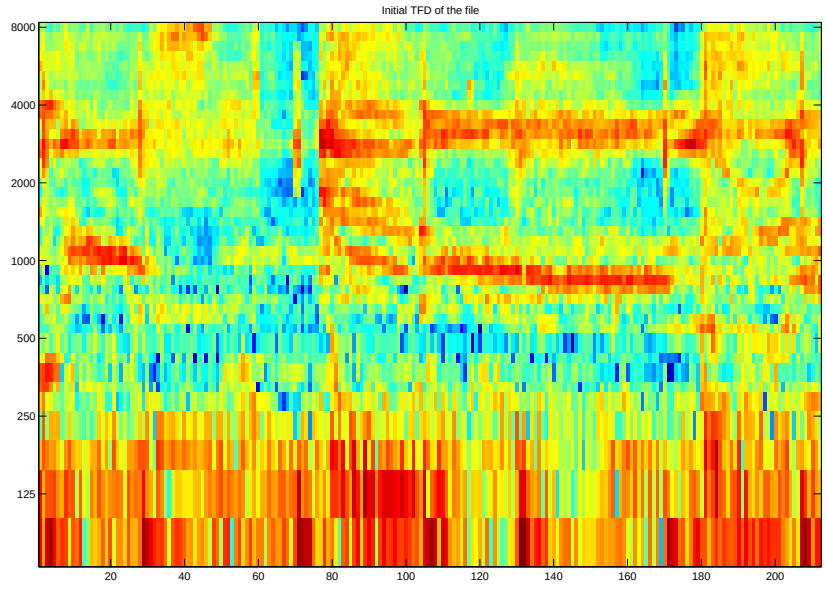

Figure 4.6. Input TFD for a two second excerpt from a piece of heavy metal music

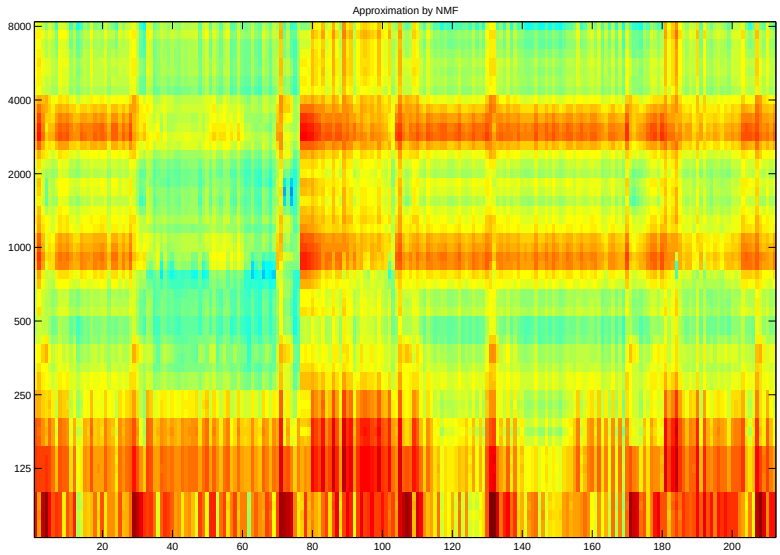

Figure 4.7. Approximation of the TFD in figure 4.6 computed with three NMF components

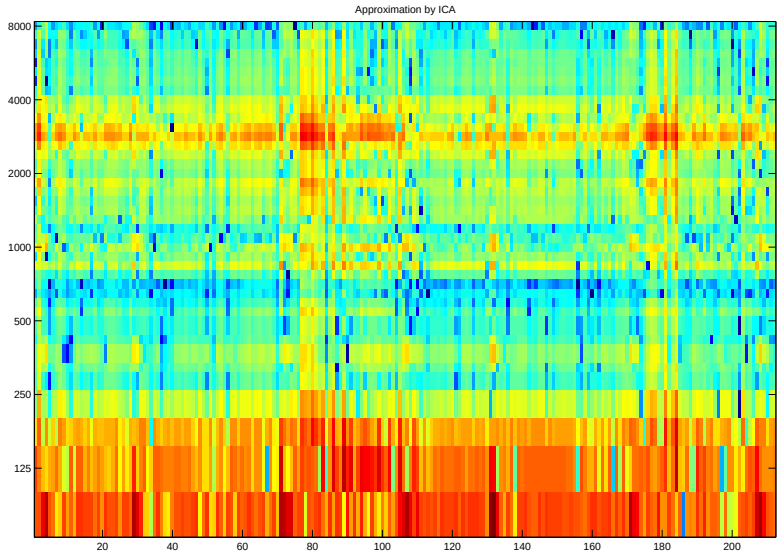

Figure 4.8. Approximation of the TFD in figure 4.6 computed with three ISA components

The superiority of NMF is also supported by measurements of the sum of squared errors of the approximations which are always smaller for the NMF independent of the number of component chosen. As can be seen in figure 4.9 even an ISA with a much larger number of components does not perform better than NMF. The shown figure is derived from the measurements of the metal music example but it is representative as its main characteristics remain unchanged for other music files. Not astonishingly rising the number of components gives a better approximation in both methods. A major difference lies in the length of the TFD that is given for the approximation. While the ISA performs increasingly worse for short time TFD's as we can observe in figure 4.9 for lengths smaller than three seconds, the opposite is the case for the NMF as it is also displayed in figure 4.10 for the NMF separately. In the examples we have always taken the metal piece which had a length of 30 seconds. For a length of i.e. 7.5s for a TFD we separated the TFD of the whole piece into four sub-TFD's. Then we applied the decomposition separately to each sub-TFD and then we summed up the errors. So from this we conclude that the best we could do to receive a good approximation is to use NMF instead of ISA and to use short extracts of the signal for the decomposition. We will refer in Chapter 5 to the length of these windows we apply to the whole TFD that we found to be optimal when focused on the classification of music.

As a general result of the comparison we conclude that the NMF decomposition yields components that preserve the temporal structure much better. In the harmonic structure we could observe the tendency of a smearing of details in the approximation by the NMF. Also a reconstruction of the time signal from the approximations using the phase of the original signal supported the found results. While only some elements like vocals of speech sounds could be recognized in the reconstruction from ISA the

signal built from the NMF approximated TFD was much closer to the initial signal. Considering computational power the NMF is much faster as the way to maximize the target function is by far more simple than in ICA.

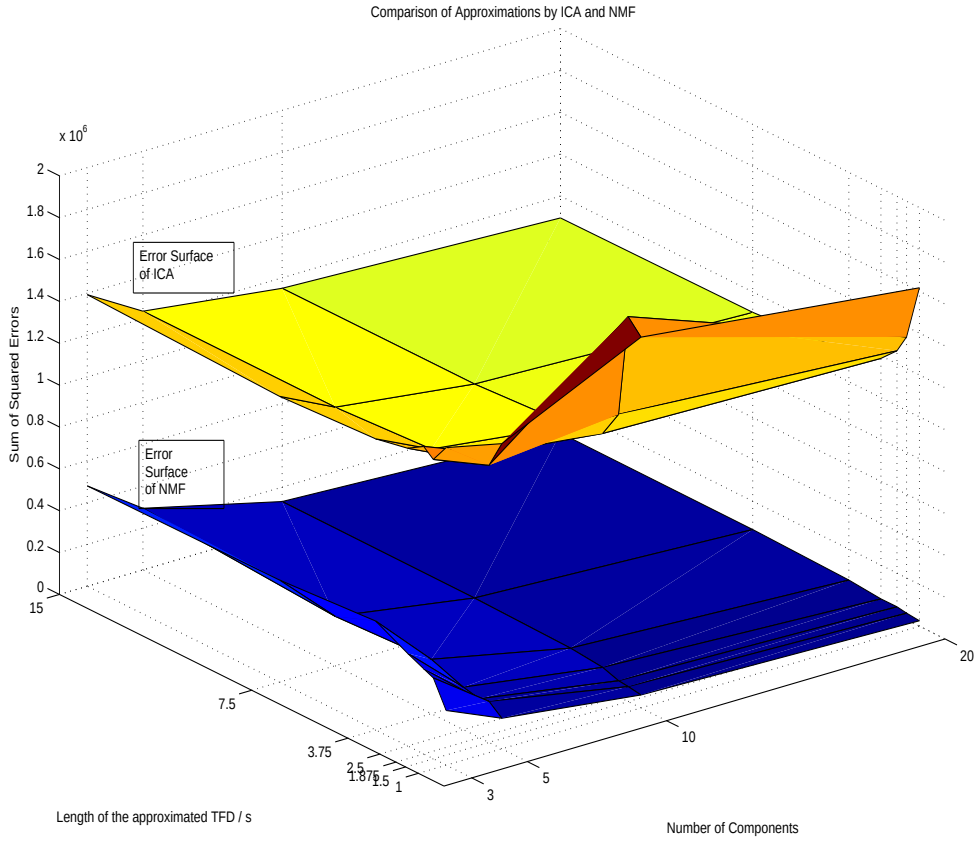

Figure 4.9. An example of error surfaces of ISA and NMF for a piece of metal music. Approximation by NMF has generally a smaller error than approximation by ISA

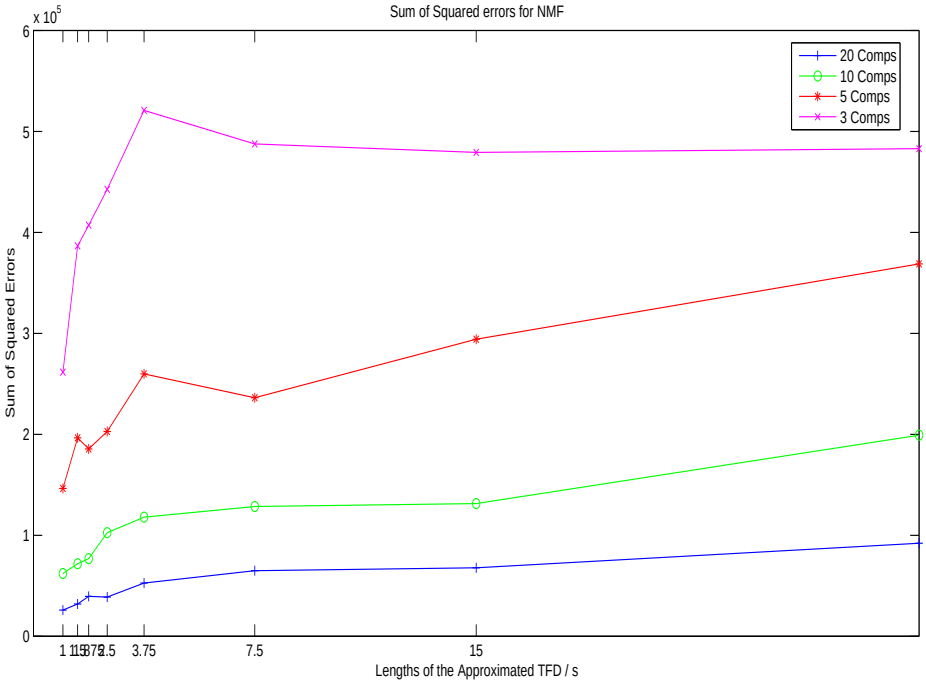

Figure 4.10. Sum of mean square errors for NMF using different number of components. The piece of music has 30 seconds length and cutting it into short pieces decreases the error.

CHAPTER 5

ALTERNATIVE SUBSPACE ANALYSIS BASED ON NMF

Above NMF has been shown to give a good lower dimensional decomposition of a signal into components. Compared with ISA the major changes concerning the feature extraction are the introduction of a psychoacoustical model and the usage of NMF instead of ICA. The first one was motivated by the publications of E. Pampalk [35] and T. Virtanen [47] while the latter was strongly motivated by the findings outlined in the chapter before. From the NMF we keep the columns of $\mathbf{W}$ in equation 2.10 as a characteristic representation for the spectral basis of the TFD and use them as our feature vectors. A GMM is trained for each class. For classification of a piece of music its features are calculated. Then we do a maximum likelihood decision in order to determine the genre of the piece. We will now describe the whole system in detail and explain the building blocks of the psychoacoustic model. The final section in this chapter introduces a way to judge the stability of the built models.

5.1 Initial System Description

Considering the results taken from our comparison between ICA and NMF figure 5.1 shows the initial form of the system presented in this thesis. Along the parameters of the system are listed in the figure together with their default values. Stages 6 to 8 will be described in more detail in the following section.

- 2: Removes mean of the wave data
- 3: Adjust average SPL to a dB
- 4: Compute the STFT of the signal. A Hamming window of length 40ms is applied to the signal first. The length has been chosen in order to give

a higher frequency resolution in order not to introduce much redundancy into the lower frequency bands when the abscissa is converted to a non linear scale.

- 5: Keep the amplitude only and neglect the phase
- 6: Calculate outer ear function according to Terhardt
- 7: Change to nonlinear bark scale
- 8: Calculate critical band spectral masking
- 9: Create TFD's from the calculated vectors that correspond to a signal length of t_{Block} seconds. As our experiments in Chapter 4 have shown a short length of this window improves the results. The optimal value was found to be half a second, see Chapter 6 for details on the experiment. Note that this is quite similar to the value of one second as found by Tzanetakis [44].
- 10: Calculate SVD of the TFD and calculate the number of singular vectors needed to retain $(100 * \phi)\%$ of the information
- 11: Compute the NMF of the TFD. We keep the columns $(w_1, ..., w_d)$ of $\mathbf{W}$ as a characteristic representation of the spectral space that contains the signal. We refer to these vectors as spectral base.
- 12: Store the spectral bases found for all TFD's of all songs. These represent the input feature set for the GMM to be trained.
- 13: Initialize the GMM with the Gaussian means procedure as described in Chapter 2
- 14: Train the GMM with a standard Expectation Maximization. Full covariance matrices are used.

The training is done for each of the C classes, giving us C GMM's. For classifying a piece of music at first we calculate its features (see block Feature Calculation in figure 5.1). Then all the calculated spectral bases are taken and for each of its vectors the likelihoods of being produced by the class models are calculated. The likelihoods of all the base vectors of the songs are then summed up and the song is assigned to the class having the maximum likelihood. This represents a music classification system that uses a new set of vertical features instead of the standard MFCC based set.

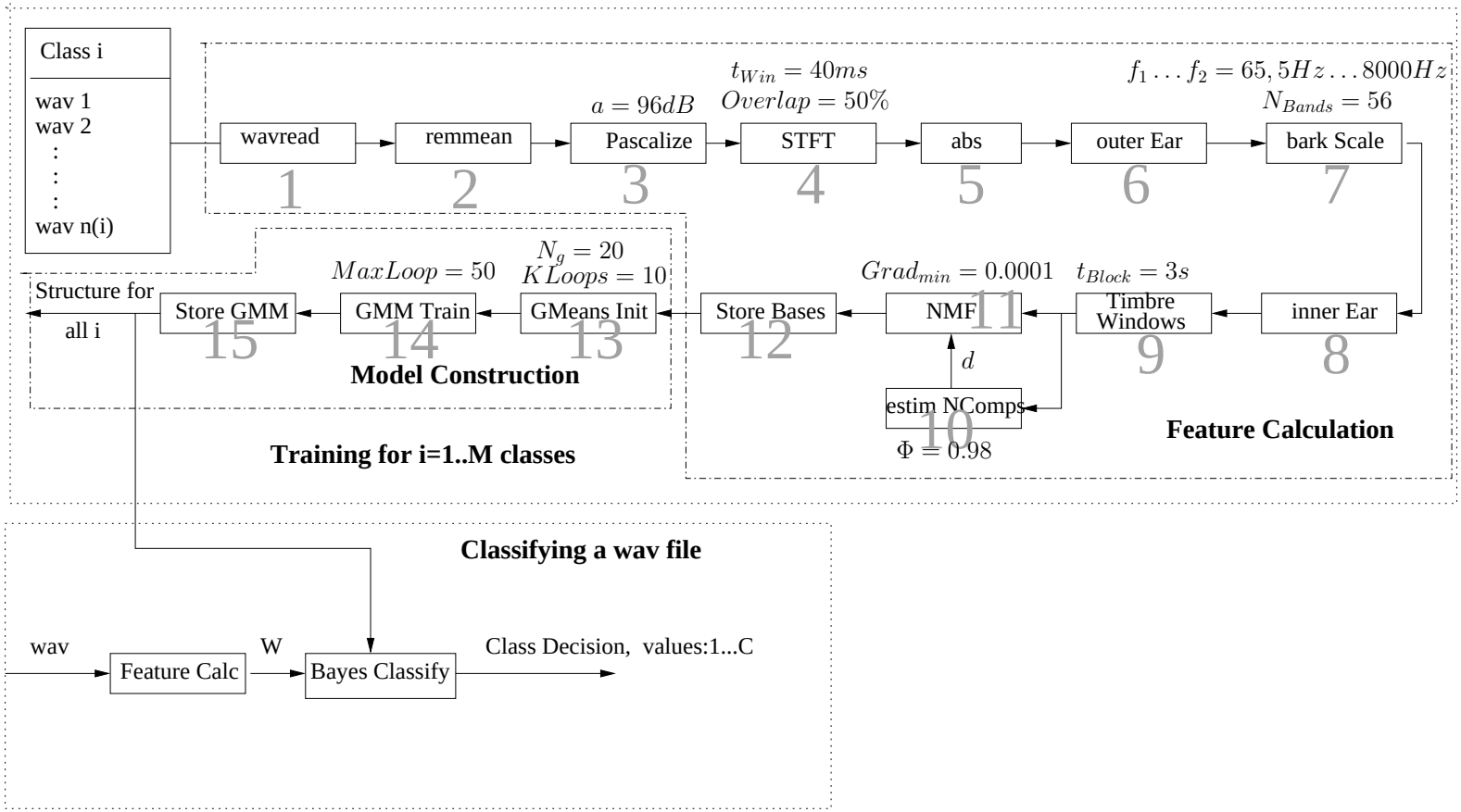

Figure 5.1. Initial concept for the Component Based Music Similarity System

5.2 Psychoacoustical Model

The stages six to eight in figure 5.1 represent the acoustic model included into the system. The implementation makes use of the MA toolbox presented by Elias Pampalk [35]. This toolbox provides one with functions for music information retrieval. We use the Short Time Fourier Transform of the toolbox and the successing stages for transforming a spectrogram on a linear frequency scale to a sonogram. The first of these transforms applies an outer ear model to each of the frequency slices. This adapts the calculated Fourier coefficients to the actually perceived loudness of the signal. Therefor the function as presented by Terhardt [43] is being used:

$$A_{dB}(f) = -3.64(10^{-3}f)^{-0.8} + 6.5\exp\left(-0.6(10^{-3}f - 3.3)^2\right) + 10^{-3}(10^{-3}f)^4 \quad (5.1)$$

where f is assumed to be given in kHz. So the model is applied by multiplying the fourier coefficient $Y(\omega)$ with the outer ear function $A_{dB}(\frac{\omega}{2*\pi}f_s)$ where f_s is the sample frequency in kHz. A plot of the outer ear function is shown in figure 5.2. It has the effect of emphasizing frequencies round 3kHz and damping low frequencies.

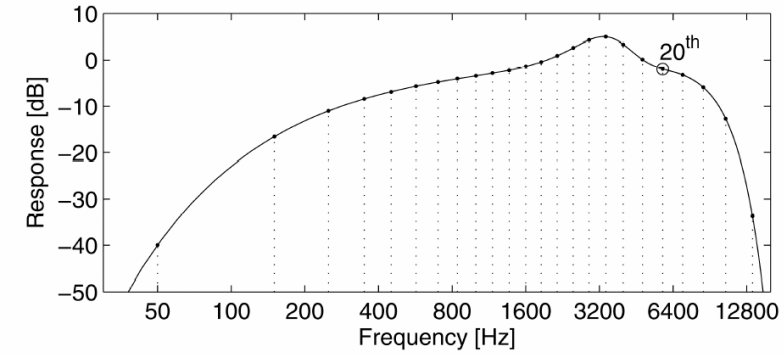

Figure 5.2. Outer and middle ear model according to Terhardt [43]

The next algorithm step is the conversion of the linear frequency scale to the *bark* scale or critical band rate scale. This scale describes best the critical bandwidths of the human ear. Basically there is a phenomenon of spectral masking when two frequencies are close enough to stimulate the same region of the basilar membran. For an exact definition of these terminology see [49]. The critical bandwidths remain constant for frequencies below $500Hz$ and grow then in a non linear fashion. This leads to a conversion from frequency to bark as shown in figure 5.3 which can be calculated as

$$z_{bark}(f) = 13\arctan(0.76f) + 3.5\arctan(f/7.5)^2 \tag{5.2}$$

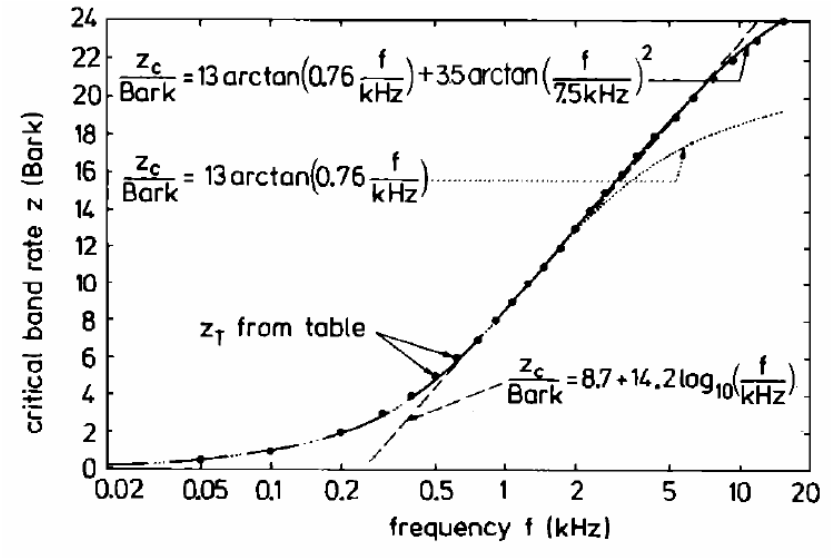

Figure 5.3. Critical bandrate z as a function of frequency f, see [50]

Input arguments to the functions of the MA toolbox are also the lower and upper frequency bound and the number of bands per octave to be contained in the sonogram. The next calculation to be done for the psychoacoustic model is the inner ear function which includes effects of spectral masking within the critical bands

because of the structure of the ear's basilar membrane. It is applied because the perceived loudness can be only approximated by the simple outer ear filter in case of sinusoidal tones. For example for a bandpass noise of constant overall sound pressure level the perceived loudness will rise when its bandwidth steps over the critical bandwidth at the center frequency of the noise. The basilar membrane spreading function used to model this effect was derived by Schroeder in [31]:

$$10\log_{10} B(z) = 15.81 + 7.5(z + 0.474) - 17.5(1 + (x + 0.474)^2)^{1/2} dB \qquad (5.3)$$

The form of the spread function is shown in figure 5.4 for the 10th bark band. In this representation it is clear to see that the function is steeper to the side of high frequencies which indicates that spectral masking is more present towards lower frequencies. This means that a loud signal is more likely to mask a signal that has a lower frequency than to mask a signal higher in frequency than itself. The last

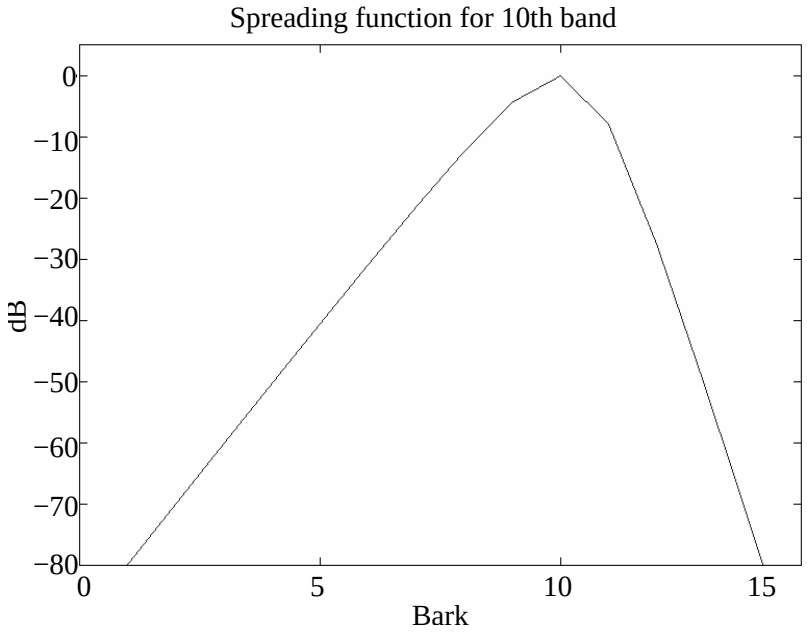

Figure 5.4. Spreading function for the 10th band

calculation is the conversion from dB to sone. The conversion is done using a formula from [6]:

$$S_{sone}(l_{dB-SPL}) = \begin{cases} 2^{(l-40)/10} & l \geq 40dB \\ (l/40)^{2.642} & otherwise \end{cases} \tag{5.4}$$

5.3 Judgment of Classifier Stability

In order to judge the stability of the trained GMM model a method based on Kullback Leibler divergence was implemented. The Kullback Leibler divergence between two distributions f and g is given by

$$KL(f||g) = \int f(x) \log \frac{f(x)}{g(x)} dx \tag{5.5}$$

However for GMM there is no closed form expression to solve this. A possible way to get a distance measure in this case is by taking M samples generated by $f(x)$ and then calculate

$$KL(f||g) \approx \frac{1}{M} \sum_{t=1}^{M} \log \frac{f(x_t)}{g(x_t)} \tag{5.6}$$

As this does not give us a distance measure because it is not symmetric the final measure for the distance between two GMM's is calculated by

$$D_{KL}(f, g) = KL(f||g) + KL(g||f) \tag{5.7}$$

Our problem consists of the classification into one of C classes. Performing an n-fold cross validation we will get a set of $n \times C$ GMM's described by their parameters $\theta_i^j, 1 \leq i \leq n, 1 \leq j \leq C$. For convenience we display this set as a $n \times C$ matrix in figure 5.5. We can now determine the distances between the GMM's of different classes using equation 5.7 for each of the n cross-validation runs separately. For example for the first run we would consider the row marked by the red ellipse. The minimum of these values gives us the least distance D_{inter} between two different classes in this cross validation. Then we calculate the distances within the classes throughout the different cross validation runs. The biggest value along all classes D_{intra} gives us a measure of how much the model differs throughout the crossvalidation due to problems with the data set. We can now define a condition measure for the GMM computed by

$$Cond_\theta = \frac{D_{inter}}{D_{intra}} \tag{5.8}$$

which for values smaller than one implies that a classification with this model might be unreliable due to models, that varied strongly and had a relatively small distance between the different class models.

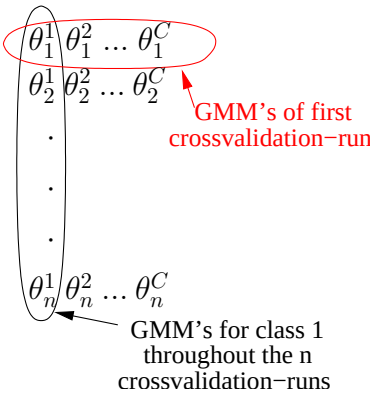

Figure 5.5. Resulting GMM's for cross validation

CHAPTER 6
EXPERIMENTS

In this chapter we present the results of our system and compare them with a baseline system that uses a MFCC feature set and with previously published results. In the first section the used databases are described. Then we introduce our baseline system that is motivated by the state of the art as referred to in the literature review of this thesis. We show the results of the baseline system and thereafter we line out the experiments that have been conducted in order to find the optimal form of our component based classification system. Then the actual form of the system is described and its results are shown. Finally we draw conclusions from our results. If not mentioned otherwise all the classification accuracies are the average of the accuracies from 5 fold cross validations on the whole set of the particular data base. For each of the five steps of the cross validation 80% of the data have been used for training and the other 20% of data have been used to get the classification accuracy. For this the data had been divided into five non-overlapping sets and the training and test sets in each iteration had been arranged as shown in figure 6.1.

6.1 Database Description

For the experiments two different data sets have been used. All the audio files of the data bases have been converted to monaural wave files at a sampling frequency of 16000 Hz and a quantization of 16 bit.

The first is a data set consisting of ten classes, each containing 100 subsections of musical pieces of 30 seconds length. The data base was collected by Giorgos Tzanetakis and has been used for performance evaluation also by other researchers.

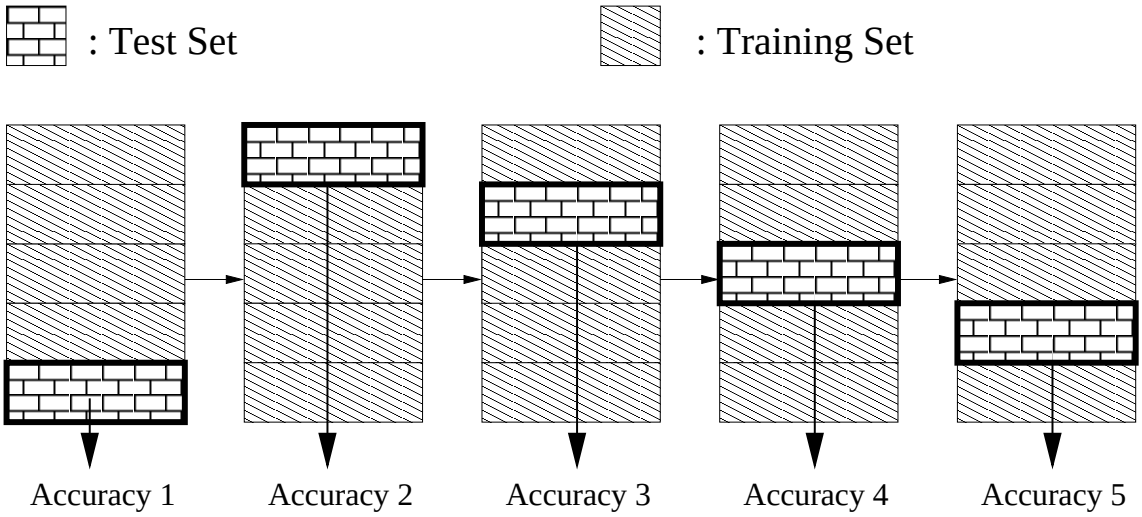

Figure 6.1. 5 fold cross validation on a data set

This data base will be referred to as database 1. Table 6.1 sums up the characteristics of the database.

The second data base was downloaded from the website of the ISMIR contest in 2004[1]. It will be referred to as database 2. Database 2 consists of six classes that are not equally distributed as they are in database 1. Also the pieces are full musical pieces and not snapshots as in database 1, due to this the lengths of the pieces differ. In order to get a quick idea of the system's performance also a subset of this data base has been created in which the sum of durations for each class is about 100 minutes. Results on this data will be referred to as subset results of database 2. See table 6.2 for details of the database.

[1]http://ismir2004.ismir.net

Class	Number of pieces	Sum of durations / minutes
blues	100	50
classical	100	50
country	100	50
disco	100	50
hiphop	100	50
jazz	100	50
metal	100	50
pop	100	50
reggae	100	50
rock	100	50

Table 6.1. Properties of database 1

Class	Number of pieces	Sum of durations
classical	320	17 hours
electronic	115	10 hours
jazz	26	96 minutes
metal/punk	45	182 minutes
rock	101	6 hours
world	121	10 hours

Table 6.2. Properties of database 2

6.2 Results of MFCC Baseline

The simplified form of the baseline system is shown in figure 3.1. A more detailed insight is given in figure 6.2. In the feature extraction an additional step called pascalize is introduced that normalizes the each piece of music to a common average sound pressure level. For the training and the testing exactly the same procedures as for the NMF feature based system are used in order to be able to compare the performance of the feature sets. The results on the first database are

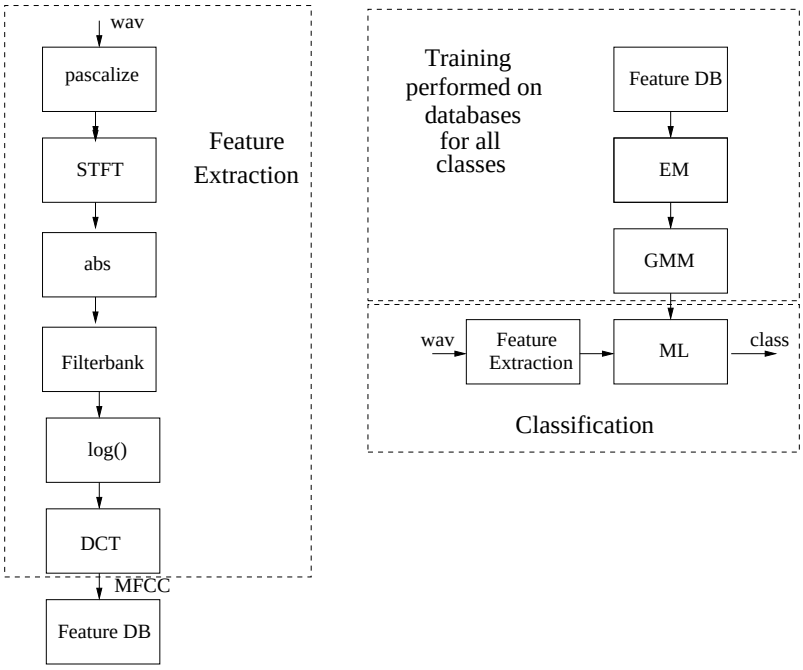

Figure 6.2. Implemented baseline system

shown in figure 6.3. Two five fold cross-validations have been performed using 20 and 30 Gaussian components with full covariance matrices on the first database. Using 30 instead of 20 components slightly increases performance from a mean of 71.5% to 72%. Nevertheless the condition numbers calculated according to equation 5.8 are smaller than one, we calculate a value of 0.32 for 20 Gaussians and a value of 0.44 for 30 Gaussians. This implies that the distances between different classes are small compared to the changes between the models in the cross-validation runs. Figure 6.4 shows a inter class distance matrix for one of the cross validation runs with the 20 component model. We can recognize that the distance from the model for classical music to all other models is large. The class disco music has the smallest distances to other classes. These findings agree with the classification accuracies that are best for classical music and worst for disco. Regarding the distances between the models of one class throughout the different cross validations (intra class distances) we find

that the classes classical and jazz differ most. Also we notice big differences for these changes comparing the different classes. In figure 6.5 we show the sum of these distances for all the classes.

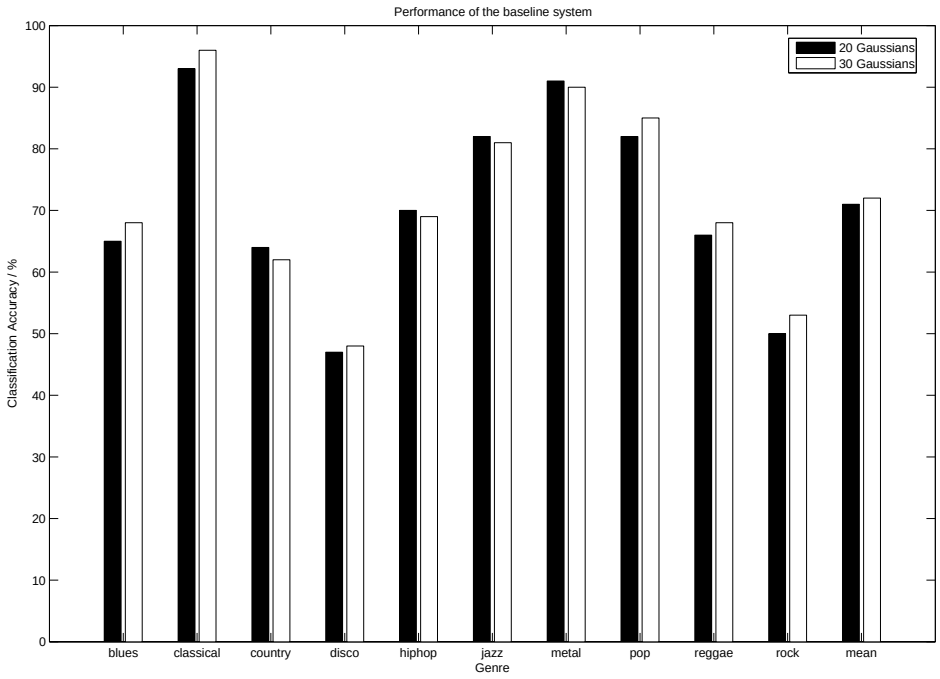

Figure 6.3. Results of the baseline system on the first database

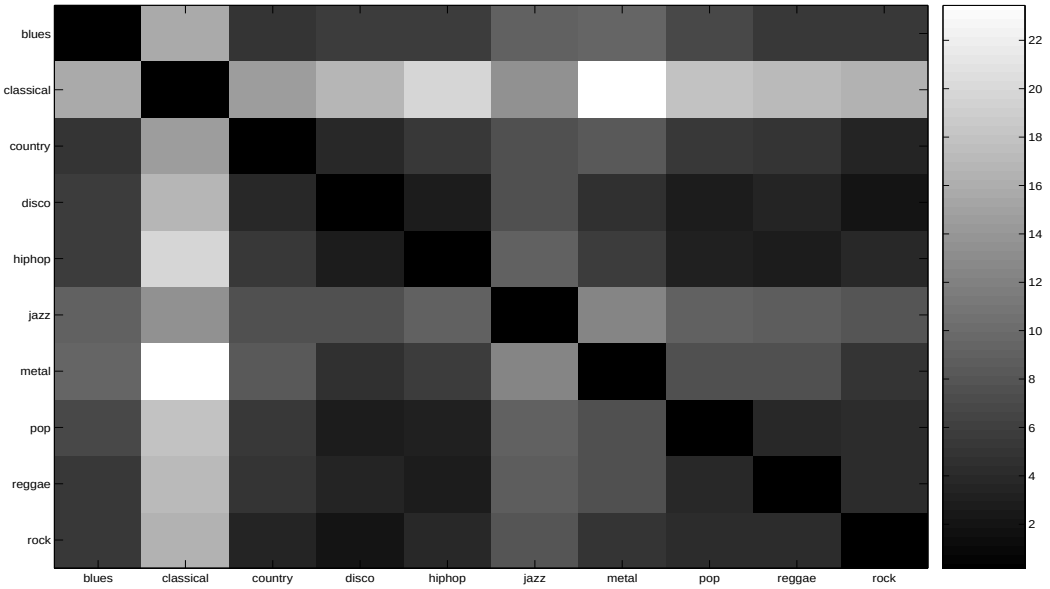

Figure 6.4. Distance matrix of the 20 component model on database 1

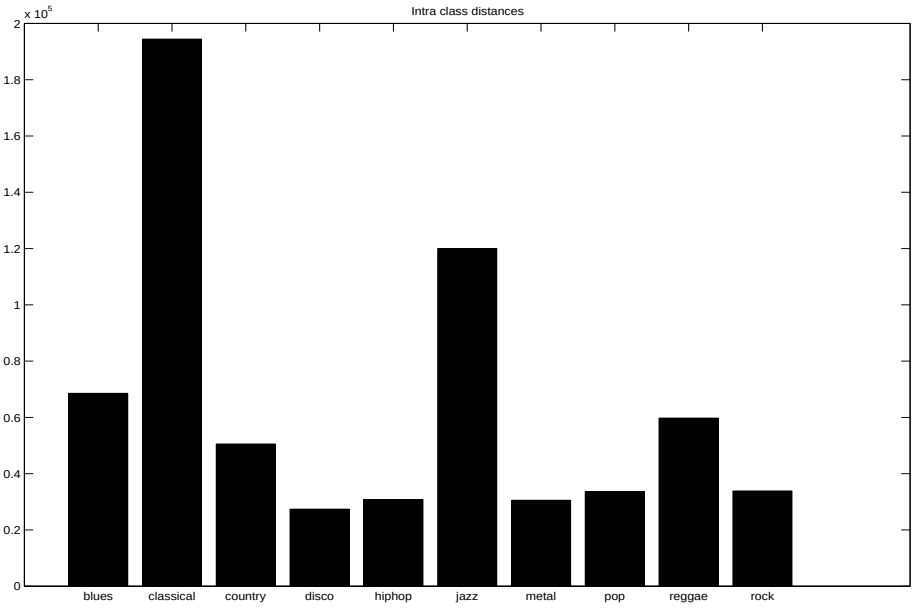

Figure 6.5. The intra class distances for a 20 component model on database 1

Genre	Classic	Electronic	Jazz	Metal/Punk	Rock	World	Mean
Accuracy / %	64	89	76	82	29	49	64.9

Table 6.3. Results of the baseline system with 20 Gaussians on the second database

In table 6.3 the classification results on database 2 are shown. The models had 20 components with full covariance matrices. Experiments with diagonal matrices did not lead to satisfying results, independent of the number of components in the mixture models. This is contradicting the usual assumption that MFCC features are uncorrelated and thus can be modelled by using diagonal covariance matrices. In most of the previous publications though it is not clear if full or diagonal matrices have been used.

6.3 System Evaluation

At first we will evaluate the utility of several blocks of the initial system and we will determine optimal parameters for the system. We tried to find if all elements take to advantages or could possibly be replaced by simpler procedures. One of the most interesting findings was that the psychoacoustical model as included into the initial concept (see figure 5.1) did not lead to any improvement. The opposite was the case, in comparison with using just a logarithmic frequency scale the results were slightly worse. An example can be seen in figure 6.6 for the system as shown in figure 5.1 using 20 Gaussians once with and once only using a logarithmic axis for the frequency (database 1 used). Even though the results are not consistent for all classes the overall performance could never be improved by using the full psychoacoustic model on neither of the two data bases. We also tried all possible combinations of the elements in the psychoacoustical model (outer ear, bark scale,

inner ear) on each of the two databases, but none of them improved the accuracy of the system in comparison with a simple logarithmic frequency axis.

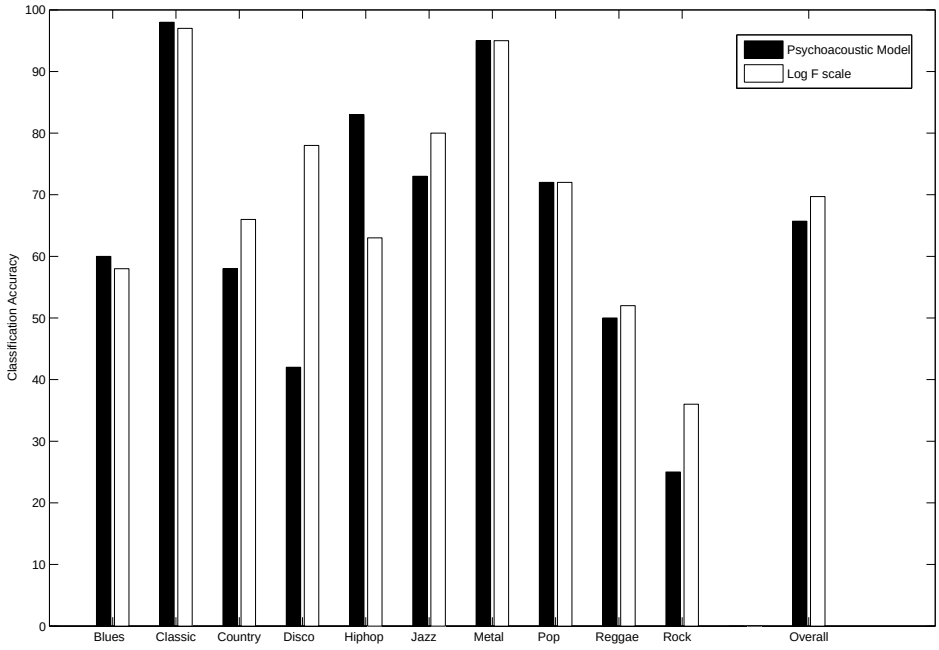

Figure 6.6. Classification results with and without using a psychoacoustic model

As shown in figure 5.1 the optimal values have to be found for a number of parameters. The parameters to determine which could influence the accuracy of our system are:

1. The number of components d
2. The length of the timbre window t_{Block}, which has influence on the number of time slices for the TFD's to be decomposed
3. The window length of the STFT
4. The number of bands per octave (N_{Bands})

As shown in figure 5.1 the approach to determine the number of components was based on a SVD of the TFD. Given a value ϕ we determined the number of components d needed to retain this information by evaluating

$$\frac{\sum_{j=1}^{d} \sigma_j}{\sum_{i=1}^{N_{Bands}} \sigma_i} \geq \phi \tag{6.1}$$

where σ_i is the $i-th$ element on the diagonal of the matrix Σ in equation 4.1. This procedure is the same as described in equation 3.2 but using SVD instead of PCA. The reason is that in a TFD we might have linear dependent columns and then the PCA does not provide a solution. Using the SVD instead we consider the row space and the columns space separately and we do not need to calculate a covariance matrix. We evaluated the influence of the number of components together with the parameter t_{Block} on a subset of data base 1 using five Gaussian mixtures and full covariance obtaining the results as shown in figure 6.7. The results show that the optimum length of the timbre window is half a second while the rising accuracies for reduced information percentages imply that further decrease should give even better results. But table 6.4 shows in detail the average number of spectral bases kept in the timbre windows for all the classes of data base 2. It can be seen that for some classes the number is already close to one and that is the point that makes further reduction infeasible also for bigger data sets because the GMM estimation runs into problems. This is because for such a small number of spectral bases retained the variables contained in some bands of the bases have a very little variance and the covariance matrices of the models are close to singularity. Taking the results from table 6.4 into account the simple approach of setting the numbers of spectral bases per timbre window to a constant number from two to four or adjusting the number of components for each class separately was tried. As a result the number of three

components lead to accuracies equal to the best possible for the tested percentage model and solved the problem of bad conditioned covariance matrices.

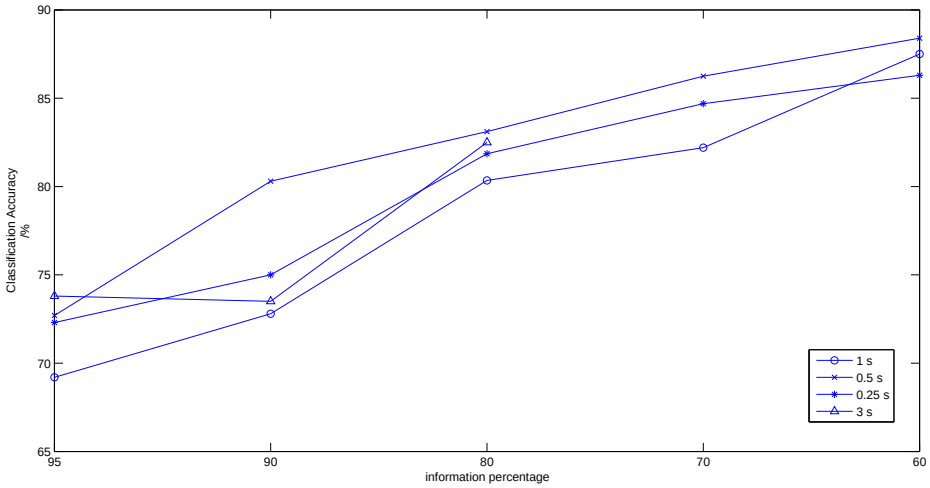

Figure 6.7. Classification accuracies for varying timbre window length and percentage of kept information

Compression	Classic	Electronic	Jazz	Metal/Punk	Rock	World
60%	1.6	2.4	1.8	3.1	2.5	1.9
70%	2.1	3.1	2.4	4.2	3.5	2.6
80%	2.88	4.3	3.3	6.0	4.8	3.5

Table 6.4. Average number of spectral bases

The number of bands per octave in the system was usually set to four, starting always from 62.5 Hz and ending at 8000 Hz giving us seven octaves. Higher resolution lead to no significant improvement but further limited the number of Gaussians. If we set the number of bands from four to eight also the dimensions of the feature space rises from 28 to 56 and this increases the number of necessary training samples in a non-linear manner. Note that the number of free parameters of a GMM with C

components in D dimensions is

$$N_{free} = C * (D + D * (D + 1)/2) + C \tag{6.2}$$

Nevertheless a way has been found to combine the use of more detailed information and a dimensionally compact representation. The number of bands per octave has been set to eight, then the NMF was applied in this space. Afterwards the logarithm of the amplitude was taken and a DCT transform was applied as shown in the diagram of the final system in figure 6.8. With a DCT matrix of twenty cosines this gives us twenty coefficients. We can consider these as Cepstral coefficients as the procedure of the inverse transform of the logarithmic vector is the same as for MFCC, but using a different frequency axis and that we do not use every Fourier transform slice. Thus the NMF can be regarded as a sampling of the frequency signal with the output a hopefully meaningful spectral base of the TFD.

The reason to set the windowlength to 40 ms is that at a sample frequency of 16000 Hz we get 320 short time Fourier coefficients and due to the non logarithmic frequency axis the values at the low end of the scale have a smaller distance. The 40 ms window gives enough information at the low frequencies as the distance of the fourier coefficients is 25 Hz.$((f_s/2)/320)$. Summing up the described results we end up with the changed system as displayed in figure 6.8.

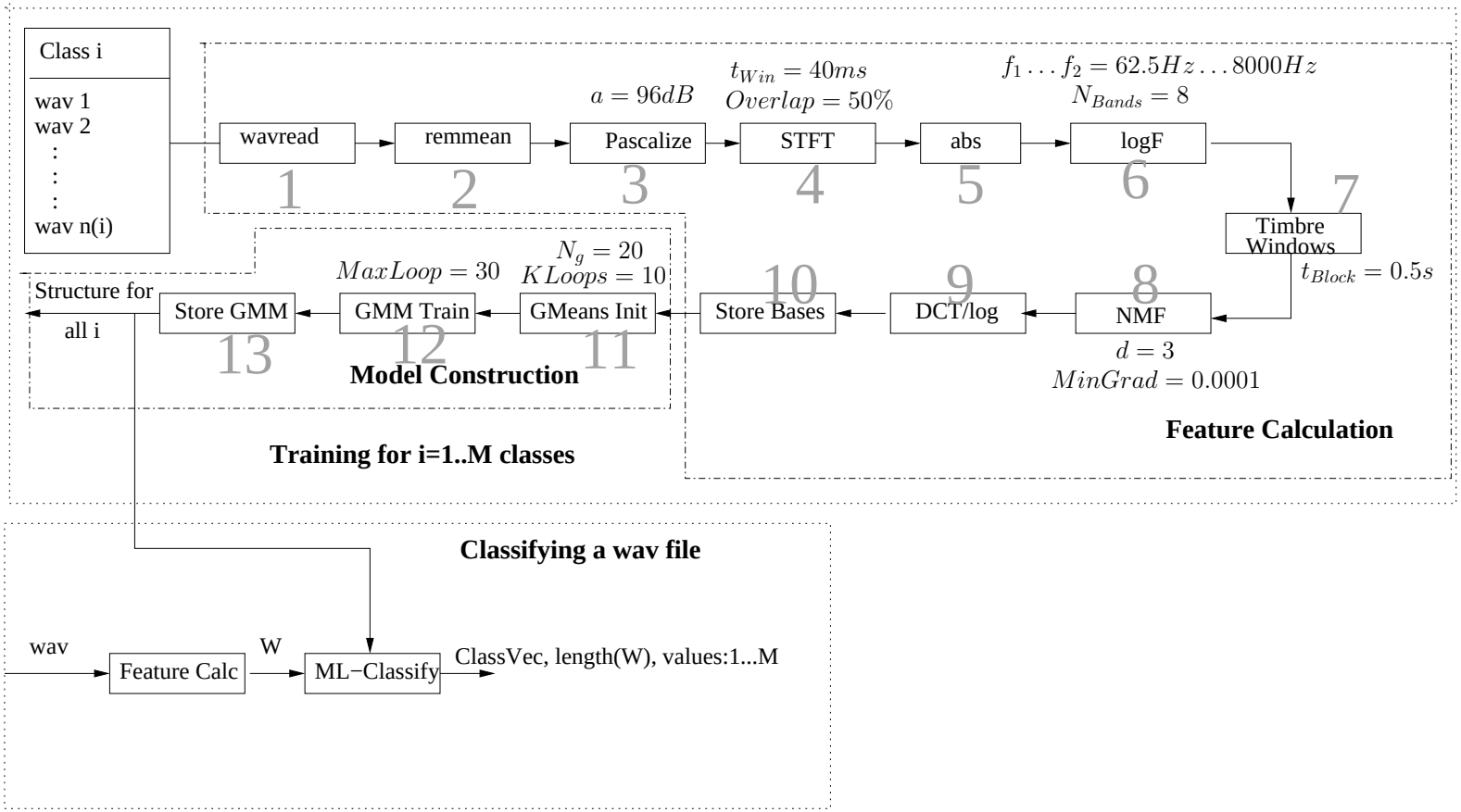

Figure 6.8. Schematic Description of the Component Based Music Similarity System

6.4 Final Results

We will now describe the experimental results of the system as shown in figure 6.8 and compare them with the results of the implemented baseline as well as with results published by other researchers. Figure 6.9 shows the classification performance on the first database along with the performance of the baseline system results that have been shown in figure 6.3 before. In order to get a valid comparison the number of Gaussians in the model is 20 for both feature sets. The parameters for the NMF based system are exactly the default parameters as shown in the system block schematic in figure 6.8. The overall performance of our system is 72.9% compared to 71.5% using MFCC features. Note that also rising the number of components of the model to 30 does not take the MFCC based system to results outperforming our system.

An even more striking improvement is the time necessary to train the systems. The MFCC based system is trained on a data set consisting of much more samples than the NMF based system. This is because before training the models we find the three characteristic spectral bases in the timbre window of half a second length using NMF. In the same time interval we have 50 MFCC coefficient vectors as we use an analysis rate of 10 ms for the baseline system. This means that we have to train a model on a dataset that is about 18 times bigger and from the experimental results we can conclude that this kind of reduction also improves classification. In our experiments the differences in training times were large. I.e. training a 20 component model on the first database took about ten minutes using our system, while training the baseline took three and a half hours. The computation of the features for NMF took longer than computing MFCC due to the gradient decent algorithm for the NMF (26 minutes compared to 11 minutes), but the overall advantage for

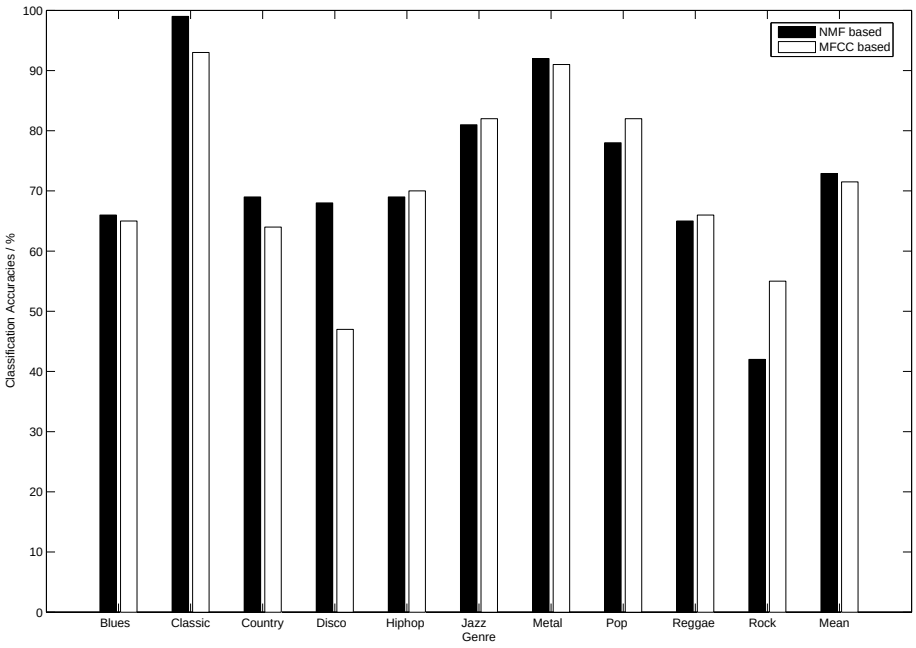

Figure 6.9. Classification accuracies on NMF and MFCC based features using a 20 component GMM

our system regarding processing time is big. A usual notebook PC has been used, all implementations are in matlab.

The classification results on the first database are slightly better than the results presented by Li and Tzanetakis in [27] where the best accuracy reported was 71%. This also means that even the baseline system shows results that are slightly better. We assume that the usage of full covariance matrices takes the MFCC baseline to these results because experiments using diagonal covariance matrices always performed worse and to our knowledge authors of [28] and [34] have used these type of matrices in their experiments. However we have to note that Bergstra and colleagues reported the best classification result on this database with 83% [4]. Their approach was outlined in the literature section and their feature set is not comparable with

our feature set as it contains about 800 dimensions. Furthermore it is an interesting thing to explore if introducing NMF based features into their system would take to further improvements. Anyhow we have to notice the training times - the authors mention them being again much higher than with a system similar to the baseline. Also their result is based just on a single run doing no cross validation so in order to get comparability the experiments would have to be conducted again using their system as well in a full cross validation.

In figure 6.10 an inter class distance matrix for again one of the cross validation runs corresponding to the distance matrix of the baseline system (figure 6.4) is shown. Similar to the MFCC based GMM model the distance between the classical music model and the metal model is the largest, even though this distance is smaller for the NMF based model (about $9 * 10^4$ compared to about $7 * 10^4$). More significantly though the smallest inter class distances for the NMF based models are larger, the smallest being $1.67 * 10^4$ for the distance between models of country and rock. Note that for the MFCC based model there were several inter class distances smaller than $1 * 10^4$.

Looking at the intra class distances in figure 6.11 and comparing it to figure 6.5 there are no big differences between these distances for the different classes. Even though the average intra class distance for our model is slightly bigger for our model ($8.5 * 10^4$ compared to $6.5 * 10^4$ in this case), in all experiments the largest intra class distance on MFCC features was always bigger than on NMF based features. All these findings have influence on the condition number calculated for the model. While the MFCC based model conditions are always smaller than one the condition for our models are always about 1.5, for example for the 20 component model considered in the examples 1.512. This gives a further proof of the superiority of this feature

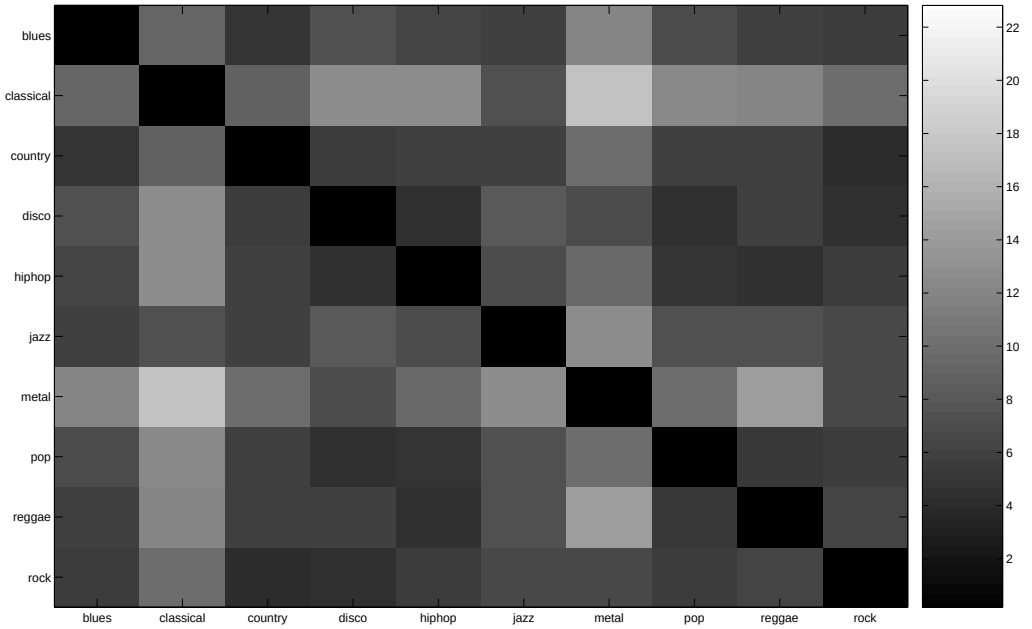

Figure 6.10. Inter Class Distance Matrix of 20 component NMF based GMM

set compared with the MFCC features. The confusion matrix of the cross validation on the data that produced the accuracies in figure 6.9 supports the influence of the model distances on the misclassifications (table 6.5). We can observe that songs from models with big distance rarely get confused. The misclassification also seem to have some musical sense with rock music getting confused with metal or country while disco music gets considered as pop frequently.

Using the second database the difference in the accuracies between the MFCC based GMM classifiers and the NMF based system gets bigger. This can be seen comparing the results in table 6.6 with the results for the baseline as shown on page 56. On the complete data set the difference is almost 9%. An interesting result that was confirmed in many tests is that the overall performance on the subset of database two (see database description at the beginning of the chapter) is always

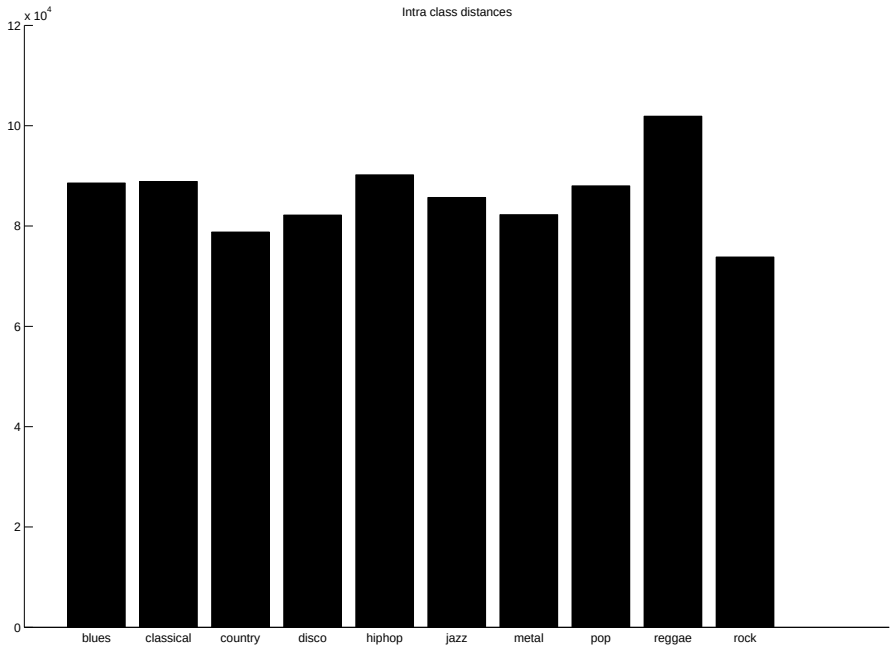

Figure 6.11. The intra class distances for a 20 component NMF based model on database 1

	Bl	Cl	Co	Di	Hi	Ja	Me	Po	Re	Ro
Blues	66	3	7	2	1	3	7	0	3	8
Classical	0	99	1	0	0	0	0	0	0	0
Country	2	1	69	10	0	3	4	5	0	6
Disco	0	1	4	68	4	0	4	15	2	2
Hiphop	0	0	0	10	69	2	5	3	8	3
Jazz	0	10	2	1	1	81	3	1	0	1
Metal	0	0	3	2	0	0	92	0	0	3
Pop	1	0	6	6	3	1	1	78	1	3
Reggae	5	1	7	2	11	2	1	5	65	1
Rock	4	2	17	10	2	3	16	3	1	42

Table 6.5. Confusion matrix for NMF based music classifier

better than on the complete database. The confusion matrix in table 6.7 shows that rock gets confused with metal and electronic music most of the time, both styles of modern popular music and as such a more musical confusion as mistaking rock for

jazz or classical music. The world music examples were always difficult to classify, an effect also present at all contributing systems in the ISMIR contest. Comparing our results with those we see that our system is second placed not too far away from the winner of the contest. The winner in genre classification was Elias Pampalk and we show the confusion matrix on the data in table 6.9 and his classification accuracies in table 6.8. This is a remarkable fact as all the systems in the contests had been using way more sophisticated classifiers or feature sets not just capable of the vertical structure of music. We have to notice though that the results published from the ISMIR contest have been produced by using a test set that has not been published so that results are not completely comparable. Furthermore the results have been achieved doing only one single test run not doing a cross validation. Classification accuracies of some cross validation iterations with our system surpassed the results of the ISMIR winner with values of about 83 per cent. Figure 6.12 gives all the accuracies of the single iterations. It shows also that the bigger amount of data in the full data base leads to less varying results.

Genre	Classic	Electro	Jazz	Metal/Punk	Rock	World	Mean
Acc./% (cpl)	96.6	84.4	88.0	77.8	60.0	37.6	74.1
Acc./% (subs)	94.1	83.0	88.0	91.4	55.0	48.6	76.7

Table 6.6. Results of the baseline system with 30 Gaussians on the second database

	Classic	Electro	Jazz	Metal/Punk	Rock	World
Classic	309	0	0	0	4	7
Electro	5	97	1	0	6	6
Jazz	0	0	22	0	3	0
Metal/Punk	0	1	0	35	7	2
Rock	3	15	0	17	60	5
World	37	24	0	1	16	47

Table 6.7. Confusion matrix for NMF based music classifier on database 2

Genre	Classic	Electro	Jazz	Metal/Punk	Rock	World	Mean
Accuracy/%	97.8	73.0	80.8	75.6	77.2	68.3	78.8

Table 6.8. Results of the ISMIR 2004 winner

	Classic	Electro	Jazz	Metal/Punk	Rock	World
Classic	309	1	0	0	3	18
Electro	1	84	0	2	10	8
Jazz	2	1	21	0	0	1
Metal/Punk	0	2	0	34	5	0
Rock	3	10	0	9	78	12
World	18	8	1	0	5	84

Table 6.9. Confusion matrix for the ISMIR 2004 winner

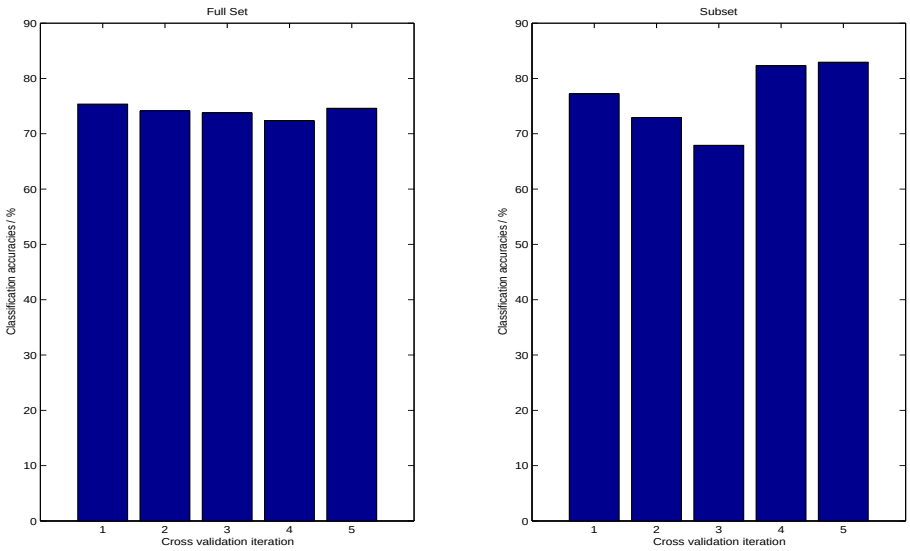

Figure 6.12. The classification accuracies on the full database 2 (left) and the subset of the database (right) that produces the results shown in table 6.6

However we have to relax our enthusiasm a bit. We could assume that further increasing the number of Gaussian components would further increase system performance. But this is not the case. Table 6.10 depicts the fact that an increase of the number of components above 30 does not improve the results.

Number of components	20	30	40
Condition number	1.31	1.273	1.16
Accuracy / %	73.1	74.1	73.2

Table 6.10. Saturation effect when increasing the number of Gaussians

The data seem not to give further meaningful detail to model and only the distances between the models get smaller. Due to the small size of the first database we were not able to confirm this result on this dataset but here increasing the number of components from 20 to 25 did not improve the classification results.

CHAPTER 7
CONCLUSIONS

This thesis succeeded in developing a new feature set for the description of the vertical structure of music. We were able to show its superiority to description with MFCC's as the standard features for describing the vertical dimension of sounds. In addition it has the advantage of fast feature calculation and low training times of the mixture models. This is due to the reduction of the amount of data and the fact that increasing the number of components in a model does not improve the performance beyond a certain limit.

The better performance of NMF compared to ICA on musical signals as well as on speech signals has been shown, putting in question the assumption of independency of components in these types of signals. Also the usage of HMM to model temporal development did not show promising results in our context.

The initial system concept that was based on findings documented in literature and on our findings considering component decompositions has been evaluated and simplified. Especially including physiological models for inner and outer ear did not lead to improvemed results. This leads to the conclusion that even though the system gives good results it does not model the way humans do a differentiation between different styles of music. On the other hand, it seems that there is a high degree of information about the spectral space of the musical genres contained in the computed spectral bases.

Astonishingly, we were able to produce very good genre recognition results with our features when modeling them by GMM's and classifying new samples by a maximum

likelihood approach. At first, this is surprising because the feature set aims to capture only the vertical structure of music. It describes some spectral characteristics and neglects rhythm or melodic developments completely. And second, the used classification method is simple compared to techniques like SVM's included in some other approaches.

Some final words about the data used: There is still only little data available for experiments due to legal limitations. But this clearly limits the possibility to compare results between different publications. Even our database 2 which was used in the ISMIR contest is rather small and contains only 6 classes. A bigger and more versatile collection would enable researchers to develop better systems or give the possibilities to reproduce results easily.

CHAPTER 8
FUTURE WORK

As mentioned in the previous chapters the feature set developed here is capable of the vertical structure of music. So obviously the next goal would have to be to get descriptors for the horizonal dimension. This means we would then try to capture characteristic modulations and rhythmic structure of music. There has been work presented on this as mentioned in the literature review by the authors of [44] and [37], while recently in [26] the necessity of rhythmic descriptors for genre classification tasks has been put into question. We will try to highlight these contradictions by incorporating a new approach for computing features for the horizontal structure. Approaches to compute such descriptors have been evaluated in [17] and a feature set capable of a high recognition rate was presented. Cepstrum like coefficients had been computed from an inter-onset interval histogram (IOIH) getting a 15 dimensional feature set. The usage of IOIH has been criticized later on by [38] and its replacement by a multiplicative combination of a DFT and the autocorelation function has been proposed. A necessary step before using these methods is an onset estimation. Here we will check the usability of the rows of the matrix $\mathbf{H}$ in $\mathbf{X} \approx \mathbf{WH}$ (equation 2.10). While the columns of $\mathbf{W}$ represent the spectral space, the rows of $\mathbf{H}$ can be interpreted as the weights of these spectral bases along the time axis. As they are constrained to be non-negative we can use them for estimating onset times. An overview of this concept for getting a feature set for the horizontal structure is shown in figure 8.1. In [38] the length of the window for calculation of the transform and the autocorelation is given as six seconds and the hop size is 0.5 seconds. This means we would have to calculate NMF decomposition with this step parameters, giving

us a set of d MFCC like vectors each 0.5 seconds, which means that we have the same feature rate as for our vertical feature set. For the construction of the models and classification we can reuse the same methods we use for the vertical feature set. But until now we have only been using the unconstrained version of the NMF.

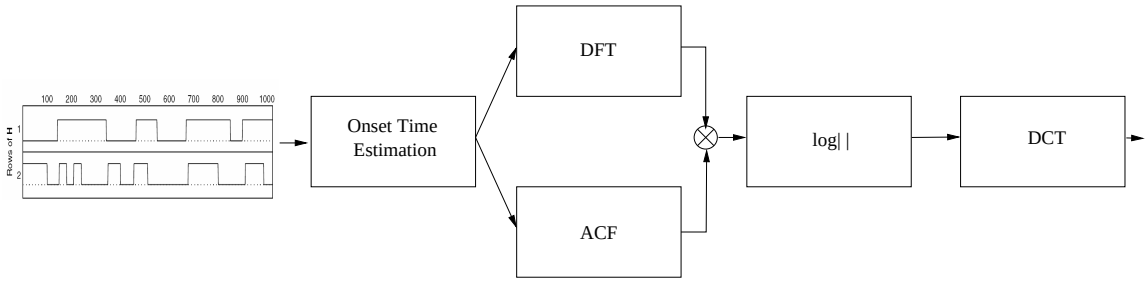

Figure 8.1. A concept for extraction of features for horizontal structure from NMF

As mentioned in the literature reviews, there have been many tries to constrain the problem in order to get more meaningful results. In general it has been tried to increase sparsity in the rows of $\mathbf{H}$. Sparsity and its opposite diversity of an n element vector $\mathbf{x}$ is defined by

$$sparsity \quad : \quad \#\{\mathbf{x}[i] = 0\} \tag{8.1}$$

$$diversity \quad : \quad \#\{\mathbf{x}[i] \neq 0\} = n - sparsity \tag{8.2}$$

Finding a way to introduce this quality into $\mathbf{H}$ can be interpreted as approximating the observed signal by a minimum number of active spectral base vectors at every time instance. This can help in estimating onset times from $\mathbf{H}$ because the signals in its rows are less noisy but it also gives rise to the question if by this means a auditory signal could be segmented into basic components. In [24] an algorithmic framework for introducing sparsity in dictionary learning algorithms is presented. It contains assumptions about the distributions of the variables that seem to be reasonable in

case of speech and music signals. More precisely the unknown sources **h** have to follow the probability density function

$$P_p(h) = \frac{e^{-\gamma_p \sum_{k=1}^{n} |h[k]|^p}}{\int e^{-\gamma_p \sum_{k=1}^{n} |h[k]|^p} dh} \tag{8.3}$$

with parameter vector $p \leq 1$ which leads to a subclass of supergaussian priors and γ being the variance to be estimated. If this model is applicable to speech signal it would be of interest to see what kind of dictionaries can be learned applying this framework and to investigate its connections to the phonetic structure of speech.

Finally after the construction of a genre classification based on a horizontal and on a vertical feature set separately we are left with the most challenging question: How can we gain in combining these two directions into a classification system that respects both dimensions of music? We have conducted some experiments using SVM's that could give a hint in this direction. The models for our system have been trained as described above. Instead of using the likelihoods of the test data on the models in order to get to a decision, we calculated the likelihoods for all of the training data on the trained models. For C classes, we get a C dimensional likelihood vector for each of the spectral base vectors. On these vectors we trained an SVM classifier with a one against the rest method as the SVM is only capable of two class decision function. This is a typical way in order to get a system for classifying into more than two classes when using SVM. The approach of the DAG-SVM [39] has also been applied with the same results. In order to classify a piece of music the likelihoods of each of its spectral base vectors are calculated in the same way as for the training data before and the data is classified using the SVM. For both of the databases we were able to improve the classification results; for database one we reached the score of 75% of correct classification and for database two the score of

75.8% (we remind that the scores obtained before are 72.9% and 74.1% for the two databases). In order to combine vertical and horizontal structure based classifiers, we could also use a SVM system working on the combined output likelihoods, figure 8.2 clarifies the principle. We expect to get a more discriminant classification function by combining the likelihoods into a set of meta features. In parallel the classification approach implemented by [4] based on ADABOOST could give further hints on the class specific relevance of the features.

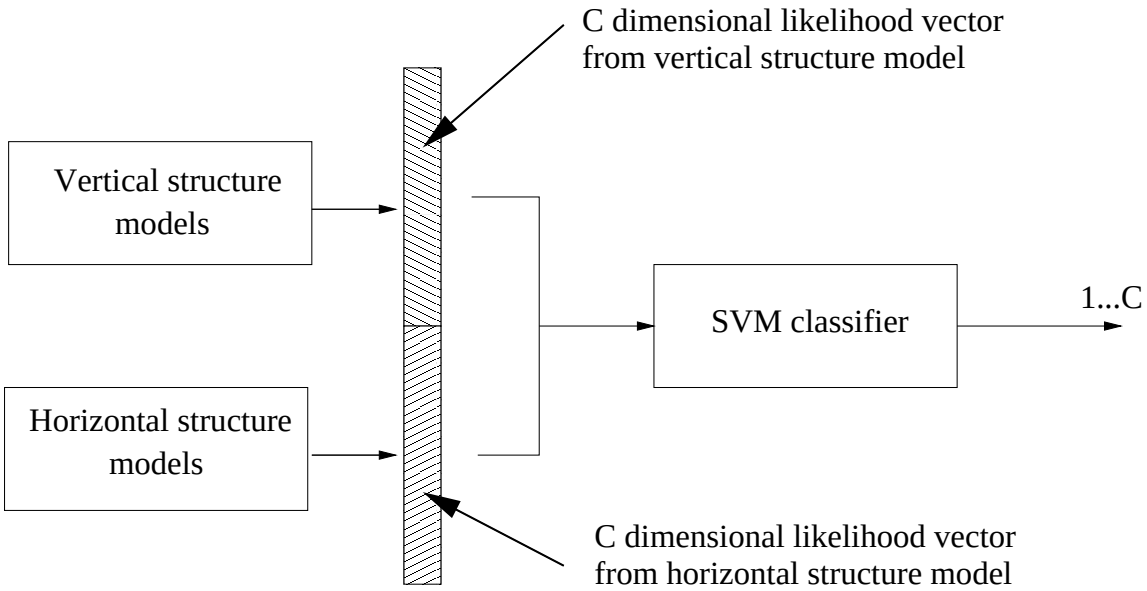

Figure 8.2. A concept for a meta classifier on combined vertical and horizontal feature set

Bibliography

[1] American Standards Association. *Acoustical Terminology*. American Standards Association, 1960.

[2] A. Berenzweig, B. Logan, D. Ellis, and B. Whitman. A large scale evaluation of acoustic and subjective music similarity measures. In *International Symposium on Music Information Retrieval*, 2003.

[3] J. Bergstra, N. Casagrande, and D. Eck. Two algorithms for timbre- and rhythm-based multi-resolution audio classification. Technical report, University of Montreal, Canada, 2005.

[4] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kégl. Aggregate features and adaboost for music classification. Technical report, Kluwer Academic Publishers, 2006.

[5] J. A. Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, International Computer Science Institute, Berkeley, 1998.

[6] R. Bladon and B. Lindblom. Modeling the judgement of vowel quality differences. *Journal of the Acoustical Society of America*, 69(5):1414–1422, 1981.

[7] B. Bogert, M. Healy, and J. Tukey. The quefrency analysis of time series for echos: Cepstrum, pseudo-autocovariance, cross-cepstrum, and saphe cracking. In *Time Series Analysis*, chapter 15, pages 209–243. M. Rosenblatt, Ed., 1963.

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

[9] M. Casey. General sound classification and similarity in mpeg-7. *Organized Sound*, 6(2):153–164, 2001.

[10] M. A. Casey. *Auditory Group Theory with Applications to Statistical Basis Methods for Structured Audio*. PhD thesis, Massachusetts Institute of Technology, 1998.

[11] M. A. Casey. Separation of mixed audio sources by independent subspace analysis. Technical report, Mitsubishi Electric Research Laboratory, 2001.

[12] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36:287–314, 1994.

[13] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustic, Speech and Signal Processing*, ASSP-28(4):357–366, August 1980.

[14] D. Donoho and V. Stodden. When does non-negative matrix factorization give a correct decomposition into parts? Technical report, Department of Statistics, Stanford University, 2004.

[15] A. Eronen. Musical instrument recognition using ica-based transform of features and discriminatively trained hmms. Technical report, Tampere University of Technology, 2003.

[16] Y. Feng and G. Hamerly. Learning the number of clusters in a data set. Technical report, Computer Science Department, Baylor University, Waco, Texas, 2006.

[17] F. Gouyon, S. Dixon, E. Pampalk, and G. Widmer. Evaluating rhythmic descriptors for musical genre clasification. In *AES 25th International Conference*, 2004.

[18] G. Hamerly and C. Elkan. Learning the k in kmeans. Technical report, Department of Computer Science and Engineering, University of California, San Diego, 2003.

[19] P. Herrera-Boyer, G. Peeters, and S. Dubnov. Automatic classification of musical instrument sounds. *Journal of New Music Research*, 32(1):3–21, 2003.

[20] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

[21] D. Huron. What is a musical feature? forte's analysis of brahms's opus 51, no. 1, revisited. *Music Theory Online*, 7(4), 2001.

[22] A. Hyvärinen and E. Oja. Independent component analysis by general non-linear hebbian-like learning rules. *Signal Processing*, 64(3):???–???, 1998.

[23] T. Kohonen. Self-organizing formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, 1982.

[24] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski. Dictionary learning algorithms for sparse representations. *Neural Computation*, 15:349–396, 2003.

[25] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems*, 13:556–562, 2001.

[26] T. Li and M. Ogihara. Toward intelligent music information retrieval. *IEEE Transactions on Multimedia*, 8(3), 2006.

[27] T. Li and G. Tzanetakis. Factors in automatic musical genre classification of audio signals. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2003.

[28] M. I. Mandel and D. P. Ellis. Song-level features and support vector machines for music classification. Technical report, Columbia University, NY, 2005.

[29] B. S. Manjunath, P. Salembier, and T. Sikora. *Introduction to MPEG-7*. John Wiley & sons, 2002.

[30] P. J. Moreno, P. P. Ho, and N. Vasconcelos. A kullback-leibler divergence based

kernel for SVM classification in multimedia applications. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing*, chapter 16. MIT Press, Cambridge, 2004.

[31] M.R.Schroeder, B.S.Atal, and J.L.Hall. Optimizing digital speech coders by exploiting masking properties of the human ear. *Journal of the Acoustical Society of America*, 66(6):1647–1652, 1979.

[32] A. V. Oppenheim and R. W. Schafer. From frequency to quefrency: A history of the cepstrum. *IEEE Signal Processing Magazine*, pages 95–99, September 2004.

[33] P. Paalanen, J. Kamarainen, and J. Ilonen. Gmmbayes toolbox. http://www.it.lut.fi/project/gmmbayes/.

[34] F. Pachet and J.-J. Aucouturier. Improving timbre similarity: How high is the sky? *Journal of negative results in speech and audio sciences*, 1(1), 2004.

[35] E. Pampalk. A matlab toolbox to compute music similarity from audio. In *5th International ISMIR 2004 Conference*, 2004.

[36] E. Pampalk, S. Dixon, and G. Widmer. Exploring music collections by browsing different views. In *ISMIR*, 2003.

[37] E. Pampalk, S. Dixon, and G. Widmer. On the evaluation of perceptual similarity measures for music. In *6. Int. Conference on Digital Audio Effects*, 2003.

[38] G. Peeters. Rhythm classification using spectral rhythm patterns. Technical report, IRCAM - Sound analysis/synthesis team, 2005.

[39] J. C. Platt, N. Christrianini, and J. Shawe-Taylor. Large margin dags for multiclass classification. *Advances in Neural Information Processing Systems*, 12:547–553, 2000.

[40] M. N. Schmidt and M. Morup. Non negative matrix factor 2-d deconvolution for

blind single channel source separation. Technical report, Technical University of Denmark, 2006.

[41] P. Smaragdis. Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs. In *Proc. 5th International Conference on Independent Component Analysis and Blind Signal Separation*, 2004.

[42] P. Smaragdis. From learning music to learning to separate. Technical report, Mitsubishi Electric Research Laboratories, 2005.

[43] E. Terhardt. Calculating vitual pitch. *Hearing Research*, 1:155–182, 1979.

[44] G. Tzanetakis and P. Cook. Music genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[45] S. Vembu and S. Baumann. Separation of vocals from polyphonic audio recordings. 2005.

[46] E. Vincent and M. D. Plumbley. Single-channel mixture decomposition using bayesian harmonic models. Technical report, Queen Mary, University of London, 2006.

[47] T. Virtanen. Separation of sound sources by convolutive sparse coding. In *Proc. Workshop on Statistical and Perceptual Audio Processing*, 2004.

[48] D. L. Wallace. Asymptotic approximations to distributions. *Ann. Math. Statist.*, 29:635–654, 1958.

[49] E. Zwicker and H. Fastl. *Psychoacoustics - Facts and Models*. Springer, 1990.

[50] E. Zwicker and E. Terhardt. Analytical expressions for critical-band rate and critical bandwidth as a function of frequency. *Journal of the Acoustical Society of America*, 68(5):1523–1526, 1980.

APPENDICES

APPENDIX A
MAXIMUM LIKELIHOOD

When a data set $\mathcal{X}$ of size M is given that has been drawn from a density $p(\mathbf{x}|\mathbf{\Theta})$ we call the vectors $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M\}$ independent and identically distributed and the likelihood function is

$$\mathcal{L}(\mathbf{\Theta}|\mathcal{X}) = p(\mathcal{X}|\Theta) = \prod_{i=1}^{M} p(\mathbf{x}_i|\Theta) \tag{A.1}$$

This function is now considered as a function of the parameter $\mathbf{\Theta}$ while the data is fixed. The maximum likelihood estimation is now defined by calculating the parameter $\mathbf{\Theta_{max}}$:

$$\mathbf{\Theta}_{max} = \arg\max_{\mathbf{\Theta}} \mathcal{L}(\mathbf{\Theta}|\mathcal{X}) \tag{A.2}$$

In practice usually the logarithm of the likelihood is maximized which leads to the same parameter maximum. Depending on the type of parameters this problem can be easy or hard. In an easy case the solution can be found analytically by setting the derivative of the likelihood to zero. In the case of a GMM the parameters are the means, variances and weights of all distribution. In this case the parameters have to be estimated and one possible and the most popular one is the EM algorithm.

APPENDIX B
EM ALGORITHM

B.1 General form

In general the EM algorithm can be applied for the estimation of distribution parameters on a data set with missing values. We can assume that the given data set $\mathcal{X}$ is incomplete and the complete data set is given by $\mathcal{Z} = (\mathcal{X}, \mathcal{Y})$. We then assume the joint density of the data set

$$p(\mathbf{z}|\mathbf{\Theta}) = p(\mathbf{x}, \mathbf{y}|\mathbf{\Theta}) = p(\mathbf{y}|\mathbf{x}, \mathbf{\Theta})p(\mathbf{x}|\mathbf{\Theta}) \tag{B.1}$$

The joint density comes from the marginal density $p(\mathbf{x}|\mathbf{\Theta})$ and the assumption of hidden parameters $\mathbf{y}$ which in the case of the GMM describe the parameter guesses for the distributions and by which of the M distributions an instance of $\mathcal{X}$ has been created. We refer to $\mathcal{L}(\mathbf{\Theta}|\mathcal{X}, \mathcal{Y}) = p(\mathcal{X}, \mathcal{Y}|\mathbf{\Theta})$ as the complete-data likelihood which is a function if the random variable $\mathcal{Y}$ for $\mathcal{X}, \mathbf{\Theta}$ fixed. The likelihood $\mathcal{L}(\mathbf{\Theta}|\mathcal{X})$ is referred to as the incomplete-data likelihood.
The EM algorithm consists of two steps; the expectation step which computes the expected value of the complete-data likelihood (E-step):

$$\begin{aligned} Q(\mathbf{\Theta}, \mathbf{\Theta}^{(i-1)}) &= \mathcal{E}\left[\log p(\mathcal{X}, \mathcal{Y}|\mathbf{\Theta})|\mathcal{X}, \mathbf{\Theta}^{(i-1)}\right] && \text{(B.2)} \\ &= \int_{\mathbf{y}\in\mathcal{Y}} \log p(\mathcal{X}, \mathbf{y}|\mathbf{\Theta}) f(\mathbf{y}|\mathcal{X}, \mathbf{\Theta}^{(i-1)}) d\mathbf{y} && \text{(B.3)} \end{aligned}$$

The expression $f(\mathbf{y}|\mathcal{X}, \mathbf{\Theta}^{(i-1)})$ is the marginal distribution of the unobserved data and is dependent on both the observed data $\mathcal{X}$ and on the current parameters of the distribution $\mathbf{\Theta}^{(i-1)}$. The parameter $\mathbf{\Theta}$ stands for the set of parameters that are to be optimized.
The maximization step is the second of the two steps of the EM algorithm and it maximizes the expectation computed in the first step:

$$\mathbf{\Theta}^{(i)} = \arg\max_{\Theta} Q(\mathbf{\Theta}, \mathbf{\Theta}^{(i-1)}) \tag{B.4}$$

This is the general form of the EM algorithm. We will now derive its form for the ML estimation of the parameters of a GMM model.

B.2 EM for GMM parameter estimation

In the case of the GMM we have N_g component densities mixed together weighted by the coefficients α, see equation 2.16. The log-likelihood for this density given the data set $\mathcal{X}$ is

$$\log \mathcal{L}(\Theta|\mathcal{X}) = \log \prod_{i=1}^{M} p(\mathbf{x}_i|\Theta) = \sum_{i=1}^{M} \log \left(\sum_{j=1}^{N_g} \alpha_j p_j(\mathbf{x}_i|\theta_j) \right) \tag{B.5}$$

This is not easy to maximize because in the equation we have the logarithm of a sum. However if we assume our dataset $\mathcal{X}$ as incomplete and the missing part $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1}^{M}$ consists of labels $y_i = 1, ..., N_g$ which assign a mixture component to each data sample. If these labels are known then we can express the complete data log-likelihood as

$$\begin{aligned} \log \mathcal{L}(\Theta|\mathcal{X}, \mathcal{Y}) &= \log \left(P(\mathcal{X}, \mathcal{Y}|\Theta) \right) \\ = \sum_{i=1}^{M} \log \left(P(\mathbf{x}_i|y_i) P(y_i) \right) &= \sum_{i=1}^{M} \log \left(\alpha_{y_i} p_{y_i}(\mathbf{x}_i|\theta_{y_i}) \right) \end{aligned} \tag{B.6}$$

But in our case we do not know in advance to which gaussian component a data vector belongs to. So we consider $\mathcal{Y}$ being a random variable and describe its distribution given the observations and a guess of the parameters $\mathbf{\Theta}^g$. Then from Bayes' rule follows

$$p(y_i|x_i, \mathbf{\Theta}^g) = \frac{\alpha_{y_i}^g p_{y_i}(x_i|\theta_{y_i}^g)}{p(x_i|\mathbf{\Theta}^g)} = \frac{\alpha_{y_i}^g p_{y_i}(x_i|\theta_{y_i}^g)}{\sum_{k=1}^{N_g} \alpha_k^g p_k(x_i|\theta_k^g)} \tag{B.7}$$

and

$$p(\mathbf{y}|\mathcal{X}, \mathbf{\Theta}^g) = \prod_{i=1}^{M} p(y_i|x_i, \mathbf{\Theta}^g) \tag{B.8}$$

Now we have derived the marginal distribution of the unobserved data in equation B.3 by guessing the parameters and assuming the existence of hidden variables. Taking

this term from equation B.8 and putting it into equation B.3 gives after simplifying

$$Q(\mathbf{\Theta}, \mathbf{\Theta}^g) = \sum_{l=1}^{N_g} \sum_{i=1}^{M} \log(\alpha_l)\, p(l|x_i, \mathbf{\Theta}^g) + \sum_{l=1}^{N_g} \sum_{i=1}^{M} \log\left(p_l(x_i|\theta_l)\right) p(l|x_i, \mathbf{\Theta}^g) \tag{B.9}$$

In equation B.9 we recognize two terms, one depending on α and the other depending on the parameters of the distribution. So in the case of estimating parameters of a mixture of distributions we can optimize for the distribution priors α and the parameters of the distributions themselves independently.
For the distribution priors we include a Lagrange multiplier λ into the left term of equation B.9 which constrains the problem to solutions for $\sum_l \alpha_l = 1$ and set its derivative to zero:

$$\frac{\partial}{\partial \alpha_l}\left[\sum_{l=1}^{N_g} \sum_{i=1}^{M} \log(\alpha_l)\, p(l|x_i, \mathbf{\Theta}^g) + \lambda(\sum_l \alpha_l - 1)\right] = 0 \tag{B.10}$$

for a specific l:

$$\sum_{i=1}^{M} \frac{1}{\alpha_l} p(l|x_i, \mathbf{\Theta}^g) + \lambda = 0 \tag{B.11}$$

from this we get $\lambda = -M$ and

$$\alpha_l = \frac{1}{M} \sum_{i=1}^{M} p(l|x_i, \mathbf{\Theta}^g) \tag{B.12}$$

In the case of the underlying distributions of the model being Gaussians as is the case in GMM we can also derive an analytic solution for minimizing the left term in equation B.9 for the logarithmic distribution. Recall the form of a d-dimensional Gaussian distribution:

$$p_l(\mathbf{x}|\mu_l, \mathbf{\Sigma}_l) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}_l|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_l)^T \mathbf{\Sigma}_l^{-1}(\mathbf{x} - \mu_l)\right) \tag{B.13}$$

Taking the logarithm, ignoring constant terms and substituting into the right term of equation B.9 we get

$$\sum_{l=1}^{N_g}\sum_{i=1}^{M}\log\left(p_l(x_i|\mu_l,\mathbf{\Sigma}_l)\right)p(l|x_i,\mathbf{\Theta}^g) \tag{B.14}$$

$$= \sum_{l=1}^{N_g}\sum_{i=1}^{M}\left(-\frac{1}{2}\log(|\mathbf{\Sigma}_l|)-\frac{1}{2}(x_i-\mu_l)^T\mathbf{\Sigma}_l^{-1}(x_i-\mu_l)\right)p(l|x_i,\mathbf{\Theta}^g) \tag{B.15}$$

Taking the partial derivative with respect to μ we obtain

$$\mu_l = \frac{\sum_{i=1}^{M} x_i p(l|x_i,\mathbf{\Theta}^g)}{\sum_{i=1}^{M} p(l|x_i,\mathbf{\Theta}^g)} \tag{B.16}$$

and after some matrix algebra by setting the derivative with respect to $\mathbf{\Sigma}$ to zero we get

$$\mathbf{\Sigma}_l = \frac{\sum_{i=1}^{M} p(l|x_i,\mathbf{\Theta}^g)(\mathbf{x}_i-\mu_l)(\mathbf{x}_i-\mu_l)^T}{\sum_{i=1}^{M} p(l|x_i,\mathbf{\Theta}^g)} \tag{B.17}$$

These updates for $\mathbf{\Theta}^g$ perform the expectation and the maximization step simultaneously and they are repeated until the changes in the complete-data log likelihood are beneath a specified threshold. Summarizing we obtained the three parameter updates:

$$\alpha_l^* = \frac{1}{M}\sum_{i=1}^{M} p(l|\mathbf{x}_i,\mathbf{\Theta}^g) \tag{B.18}$$

$$\mu_l^* = \frac{\sum_{i=1}^{M}\mathbf{x}_i p(l|\mathbf{x}_i,\mathbf{\Theta}^g)}{\sum_{i=1}^{M} p(l|\mathbf{x}_i,\mathbf{\Theta}^g)} \tag{B.19}$$

$$\mathbf{\Sigma}_l^* = \frac{\sum_{i=1}^{M} p(l|\mathbf{x}_i,\mathbf{\Theta}^g)(\mathbf{x}_i-\mu_l)(\mathbf{x}_i-\mu_l)^T}{\sum_{i=1}^{M} p(l|\mathbf{x}_i,\mathbf{\Theta}^g)} \tag{B.20}$$

where the asterisk denotes the new parameter to be used in the next computation step. For a more detailed derivation of the parameter computation see [5]. In the implementation of this work the GMMBayes toolbox for matlab has been used [33].