Music Genre Classification Systems

- A Computational Approach

Peter Ahrendt

Kongens Lyngby 2006
IMM-PHD-2006-164

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

IMM-PHD: ISSN 0909-3192

Summary

Automatic music genre classification is the classification of a piece of music into its corresponding genre (such as jazz or rock) by a computer. It is considered to be a cornerstone of the research area Music Information Retrieval (MIR) and closely linked to the other areas in MIR. It is thought that MIR will be a key element in the processing, searching and retrieval of digital music in the near future.

This dissertation is concerned with music genre classification systems and in particular systems which use the raw audio signal as input to estimate the corresponding genre. This is in contrast to systems which use e.g. a symbolic representation or textual information about the music. The approach to music genre classification systems has here been system-oriented. In other words, all the different aspects of the systems have been considered and it is emphasized that the systems should be applicable to ordinary real-world music collections.

The considered music genre classification systems can basically be seen as a feature representation of the song followed by a classification system which predicts the genre. The feature representation is here split into a *Short-time feature extraction* part followed by *Temporal feature integration* which combines the (multivariate) time-series of short-time feature vectors into feature vectors on a larger time scale.

Several different short-time features with 10-40 ms frame sizes have been examined and ranked according to their significance in music genre classification. A *Consensus sensitivity analysis* method was proposed for feature ranking. This method has the advantage of being able to combine the sensitivities over several

resamplings into a single ranking.

The main efforts have been in temporal feature integration. Two general frameworks have been proposed; the *Dynamic Principal Component Analysis* model as well as the *Multivariate Autoregressive Model* for temporal feature integration. Especially the Multivariate Autoregressive Model was found to be successful and outperformed a selection of state-of-the-art temporal feature integration methods. For instance, an accuracy of 48% was achieved in comparison to 57% for the human performance on an 11-genre problem.

A selection of classifiers were examined and compared. We introduced *Co-occurrence models* for music genre classification. These models include the whole song within a probabilistic framework which is often an advantage compared to many traditional classifiers which only model the individual feature vectors in a song.

Resumé

Automatisk musik genre klassifikation er et forskningsområde, som fokuserer på, at klassificere musik i genrer såsom jazz og rock ved hjælp af en computer. Det betragtes som en af de vigtigste områder indenfor Music Information Retrieval (MIR). Det forventes, at MIR vil spille en afgørende rolle for f.eks. behandling og søgning i digitale musik samlinger i den nærmeste fremtid.

Denne afhandling omhandler automatisk musik genre klassifikation og specielt systemer, som kan prædiktere genre udfra det rå digitale audio signal. Modsætningen er systemer, som repræsenterer musikken i form af f.eks. symboler, som det bruges i almindelig node-notation, eller tekst-information. Tilgangen til problemet har generelt været system-orienteret, således at alle komponenter i systemet tages i betragtning. Udgangspunktet har været, at systemet skulle kunne fungere på almindelige folks musik samlinger med blandet musik.

Standard musik genre klassifikations-systemer kan generelt deles op i en feature repræsentation af musikken, som efterfølges af et klassifikationssystem til mønster genkendelse i feature rummet. I denne afhandling er feature repræsentationen delt op i hhv. *Kort-tids feature ekstraktion* og *Tidslig feature integration*, som kombinerer den (multivariate) tidsserie af kort-tids features i en enkelt feature vektor på en højere tidsskala.

I afhandlingen undersøges adskillige kort-tids features, som lever på 10-40 ms tidsskala, og de sorteres efter hvor godt de hver især kan bruges til musik genre klassifikation. Der foreslås en ny metode til dette, *Konsensus Sensitivitets Analyse*, som kombinerer sensitivitet fra adskillige resamplings til en samlet vurdering.

Hovedvægten er lagt på området tidslig feature integration. Der foreslås to nye metoder, som er *Dynamisk Principal Komponent Analyse* og en *Multivariat Autoregressiv Model* til tidslig integration. Den multivariate autoregressive model var mest lovende og den gav generelt bedre resultater end en række state-of-the-art metoder. For eksempel gav denne model en klassifikationsfejl på 48% mod 57% for mennesker i et 11-genre forsøg.

Der blev også undersøgt og sammenlignet et udvalg af klassifikationssystemer. Der foreslås desuden *Co-occurrence modeller* til musik genre klassifikation. Disse modeller har den fordel, at de er i stand til at modellere hele sangen i en probabilistisk model. Dette er i modsætning til traditionelle systemer, som kun modellerer hver feature vektor i sangen individuelt.

Preface

This dissertation was prepared at the Institute of Informatics and Mathematical Modelling, Technical University of Denmark in partial fulfillment of the requirements for acquiring the Ph.D. degree in engineering.

The dissertation investigates automatic music genre classification which is the classification of music into its corresponding music genre by a computer. The approach has been system-oriented. Still, the main efforts have been in Temporal feature integration which is the process of combining a time-series of short-time feature vectors into a single feature vector on a larger time scale.

The dissertation consists of an extensive summary report and a collection of six research papers written during the period 2003–2006.

Lyngby, February 2006

Peter Ahrendt

Papers included in the thesis

[Paper B] Ahrendt P., Meng A., and Larsen J., **Decision Time Horizon for Music Genre Classification using Short Time Features**, Proceedings of *European Signal Processing Conference (EUSIPCO)*, Vienna, Austria, September 2004.

[Paper C] Meng A., Ahrendt P., and Larsen J., **Improving Music Genre Classification by Short-Time Feature Integration**, Proceedings of *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Philadelphia, USA, March 2005.

[Paper D] Ahrendt P., Goutte C., and Larsen J., **Co-occurrence Models in Music Genre Classification**, Proceedings of *IEEE International Workshop on Machine Learning for Signal Processing* (MLSP), Mystic, Connecticut, USA, September 2005.

[Paper E] Ahrendt P. and Meng A., **Music Genre Classification using the Multivariate AR Feature Integration Model**, Audio Genre Classification contest at the *Music Information Retrieval Evaluation eXchange (MIREX)* (in connection with the annual ISMIR conference) [53], London, UK, September 2005.

[Paper F] Hansen L. K., Ahrendt P., and Larsen J., **Towards Cognitive Component Analysis**, Proceedings of *International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR)*, Espoo, Finland, June 2005.

[Paper G] Meng A., Ahrendt P., Larsen J., and Hansen L. K., **Feature Integration for Music Genre Classification**, yet unpublished article, 2006.

Acknowledgements

I would like to take this opportunity to first and foremost thank my colleague Anders Meng. We have had a very close collaboration which is a fact that the publication list certainly illustrates. However, most importantly, the collaboration has been very enjoyable and interesting. Simply having a counterpart to tear down your idea or (most often) help you build it up, has been priceless.

I would also like to thank all of the people at the Intelligent Signal Processing group for talks and chats about anything and everything. It has been interesting, funny and entertaining and contributed a lot to the good times that I've had during last three years of studies.

Last, but not least, I would like to thank my wife for help with careful proofreading of the dissertation as well as generally being very supportive.

Contents

Chapter 1

Introduction

Jazz, rock, blues, classical.. These are all music genres that people use extensively in describing music. Whether it is in the music store on the street or an online electronic store such as Apple's iTunes with more than 2 million songs, music genres are one of the most important descriptors of music.

This dissertation lies in the research area of *Automatic Music Genre Classification*[1] which focuses on computational algorithms that (ideally) can classify a song or a shorter sound clip into its corresponding music genre. This is a topic which has seen an increased interest recently as one of the cornerstones of the general area of Music Information Retrieval (MIR). Other examples in MIR are music recommendation systems, automatic playlist generation and artist identification. MIR is thought to become very important in the nearest future (and now!) in the processing, searching and retrieval of digital music.

A song can be represented in several ways. For instance, it can be represented in symbolic form as in ordinary sheet music. In this dissertation, a song is instead represented by its digital audio signal as it naturally occurs on computers and on the Internet. Figure 1.1 illustrates the different parts in a typical music genre classification system. Given the raw audio signal, the next step is to extract the essential information from the signal into a more compact form before further

[1] Throughout this dissertation, automatic music genre classification and music genre classification are often used synonymously.

processing. This information could be e.g. the rhythm or frequency content and is called the *feature representation* of the music. Note that most areas in MIR rely heavily on the feature representation. They have many of the same demands to the features which should be both compact and flexible enough to capture the essential information. Therefore, research in features for music genre classification systems is likely to be directly applicable to many other areas of MIR.

In this dissertation, the feature part is split into *Short-time Feature Extraction* and *Temporal Feature Integration*. Short-time features are extracted on a 10-40 ms time frame and therefore only capable of representing information from such a short time scale. Temporal feature integration is the process of combining the information in the (multivariate) time-series of short-time features into a single feature vector on a larger time scale (e.g. 2000 ms). This long-time feature might e.g. represent the rhythmic information.

The song is now represented by feature vectors. The ordinary procedure in music genre classification systems is to feed these values into a *classifier*. The classifier might for instance be a parametric probabilistic model of the features and their relation to the genres. A training set of songs are then used to infer the parameters of the model. Given the feature values of a new song, the classifier will then be able to estimate the corresponding genre of the song.

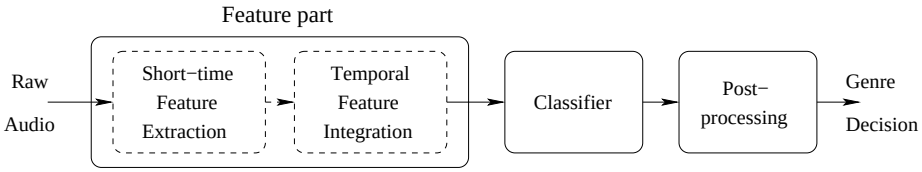


Figure 1.1: Illustration of the music genre classification systems which are given special attention in this dissertation. The model covers a large range of existing systems. Given a song, music features are first created from the raw audio signal. The feature creation is here split into two parts; Short-time feature extraction and Temporal feature integration. Short-time features represent approximately 10-40 ms of sound. Temporal feature integration uses the time series of short-time features to create features which represent larger time scales (e.g. 2000 ms). The classifier predicts the genre (or the probability of different genres) from a feature vector and post-processing is used to reach a single genre decision for the whole song or sound clip.

1.1 Scientific contributions

The objective in the current project has been to create music genre classification systems that are able to predict the music genre of a song or sound clip given the raw audio signal. The performance measure of the systems has mostly been the classification test error i.e. an estimate of the probability of predicting the wrong genre for a new song. The main efforts in this dissertation have been in the feature representation and especially in methods for temporal feature integration.

In the first part of the project, a large selection of short-time features were investigated and ranked by their significance for music genre classification. In (Paper B), the *Consensus Sensitivity Analysis* method was proposed for ranking. It has the advantage of being able to combine the sensitivities of several cross-validation or other resampling runs into a single ranking. The ranking indicated that the so-called MFCC features performed best and they were therefore used as standard short-time feature representation in the following experiments.

Several temporal feature integration methods were examined and compared to two proposed models; the *Multivariate Autoregressive model* (Papers C, G and E) for temporal feature integration and the *Dynamic Principal Component Analysis* model (Paper B). Especially the Multivariate Autoregressive model was carefully analyzed due to its good performance. It was capable of outperforming a selection of state-of-the-art methods. On an 11-genre data set, our best performing system had an accuracy of 48% in comparison with 57% for the human performance. By far the most common temporal feature integration method uses the mean and variance of the short-time features as long-time feature vector (with twice as large dimensionality as the short-time features). For comparison, this method had an accuracy of 38% on the 11-genre data set.

A selection of classifiers were examined with the main purpose of being able to generalize on the value of the different features. Additionally, novel *Co-occurrence models* (Paper D) were proposed. They have the advantage of being able to incorporate the full song into a probabilistic framework in comparison with many traditional classifiers which only model individual feature vectors in the song.

1.2 Overview of the dissertation

An overview of the dissertation is presented in the following.

Chapter 2 gives a broad introduction to the area of music genre classification as it is performed by both humans and by computers. It also discusses related areas and confine the area of research in the current dissertation.

Chapter 3 describes music features in general and *Short-time feature extraction* in particular. Furthermore, it explains about feature ranking and selection and describes the proposed *Consensus sensitivity analysis* method for feature ranking.

Chapter 4 investigates *Temporal feature integration* carefully. A selection of methods are described as well as the proposed *Dynamic Principal Component Analysis* model. The proposed *Multivariate autoregressive model* for temporal feature integration is carefully analyzed.

Chapter 5 describes classification and clustering in general. Special emphasis is given to the parametric probabilistic models that have been used in this dissertation. The proposed *Co-occurrence model* for music genre classification is carefully described. Post-processing methods with special focus on decision fusion is the topic of the last section.

Chapter 6 summarizes and discusses the main experimental results that have been achieved in this dissertation. Additionally, our performance measures are described as well as the two data sets that have been used.

Chapter 7 concludes on the results of the project as well as outline the interesting experiments that might improve future music genre classification systems.

Appendix A gives the details of a computationally cheap version of the Principal Component Analysis.

Appendix B-G contains our scientific papers which have already been published or are in the process of being published in relation to this dissertation.

Chapter 2

Music Genre Classification Systems

This chapter introduces the term *music genre classification* and explains how it is performed both by humans and by computers. Music genre classification is put into context by explaining about the structures in music and how it is perceived and analyzed by humans. The problem of defining genre is discussed and examples are given of music genre classification by computers as well as related research. In particular, the research area of music genre classification can be seen as a subtopic of *Music Information Retrieval* (MIR). The final section describes some main assumptions and choices which confine the area of research in the current dissertation.

Music genre classification is the process of assigning musical genres such as jazz, rock or acid house to a piece of music. Different pieces of music in the same genre (or subgenre) are thought to share the same "basic musical language" [84] or originate from the same cultural background or historical period.

Humans are capable of performing music genre classification with the use of the ears, the auditory processing system in the ears as well as higher-level cognitive processes in the brain. Musical genres are used among humans as a compact description which facilitates sharing of information. For instance, the statements "I like heavy metal" or "I can't stand classical music!" are often used to share

information and relies on shared knowledge about the genres and their relation to society, history and musical structure. Besides, the concept of genre is heavily used by record companies and music stores to categorize the music for search and retrieval.

Automatic music genre classification[1] is the classification of music into genres by a computer and as a research topic it mostly consists of the development of algorithms to perform this classification. This is a research area which has seen a lot of interest in the recent 5-10 years, but does not have a long history. It is very interdisciplinary and draws especially from areas such as music theory, digital signal processing, psychoacoustics and machine learning. Traditional areas of computer science and numerical analysis are also necessary since the applicability of algorithms to real world problems demand that they are somehow "reasonable" in computational space and time.

One of the first approaches to automatic music genre classification is [116] from 1996 which was thought as a commercial product. This illustrates one motivation for research in this area; commercial interests. For instance, Apple's iTunes service sell music from a database with more than 2,000,000 songs [47] and the human classification of these songs into a consistent genre taxonomy is obviously time-consuming and expensive.

Another motivation for research in automatic music genre classification is its strong relations to many of the other areas of Music Information Retrieval (MIR). The area of MIR covers most of the aspects of handling digital musical material efficiently such as managing large music databases, business issues and human-computer interaction, but also areas which are more closely related to music genre classification. These are for instance music artist identification [77], musical instrument recognition [79], tempo extraction [2], audio fingerprinting [34] and music transcription [65]. The relations are very strong since a basic representation of music (the so-called feature set) is necessary in these areas. The desire is a representation which is as compact as possible while still having enough expressive power. Hence, a good music representation for automatic music genre classification is also likely to be useful in related areas and vice versa.

[1] In the remaining parts of this dissertation, automatic music genre classification and music genre classification are used synonymously.

2.1 Human music genre classification

Humans use, among other things, their advanced auditory system to classify music into genres [18]. A simplified version of the system is illustrated in figure 2.1. The first part of the ear is the visible outer ear which is used for the vertical localization of sounds as well as magnification. This is followed by the ear canal which can be looked upon as a tube with one closed and one open end and hence gives rise to some frequency-dependency in the loudness perception near the resonance frequencies. The middle ear basically transmits the vibrations of the tympanic membrane into the fluid in the inner ear which contains the snail-shell shaped organ of hearing (the Cochlea). From a signal processing view, the inner ear can be seen as a frequency analyzer and it can be modelled as a filter bank of overlapping filters with bandwidth similar to the so-called critical bands. The following parts of the auditory system are the nerve connections from the Cochlea to the brain. At last, high-level cognitive processing in the brain is used to classify music in processes which are still far from fully understood.

The human auditory system has originally evolved to be able to e.g. localize pray or predators, communicate for mating and later speech evolved to the complex languages that exist now. Music has a history which is likely to be as long as speech and certainly goes back to prehistoric times. For instance, the first known man-made music instrument dates back to 80,000-40,000 BC and is a flute (presumably) made from the bone of a cave bear [51]. Music is also related to speech in the sense that they are both produced by humans (in most definitions of music) and therefore produced specifically for the human auditory system. Additionally, music often contains singing which is closely related to speech. Due to this relation between music and speech, research in one area could often be useful to the other. The production, perception and modelling of speech has been investigated for decades [94].

The physical production of music has traditionally been produced by human voice and instruments from three major groups (wind, string and percussion) which are distinguished by the way they produce the sound [79]. However, during the last century the definition of music has broadened considerably and modern music contains many elements which cannot be assigned to any of these three groups. Notably, "electronic sounds" should certainly be added to these groups although "electronic sounds" are often meant to resemble traditional music instruments.

The basic perceptual aspects of music are described in the area of music theory. Traditionally these aspects are those which are important in European classical music such as *melody*, *harmony*, *rhythm*, *tone color/timbre* and *form*. These aspects relate to the music piece as a whole and are closely related to the tra-

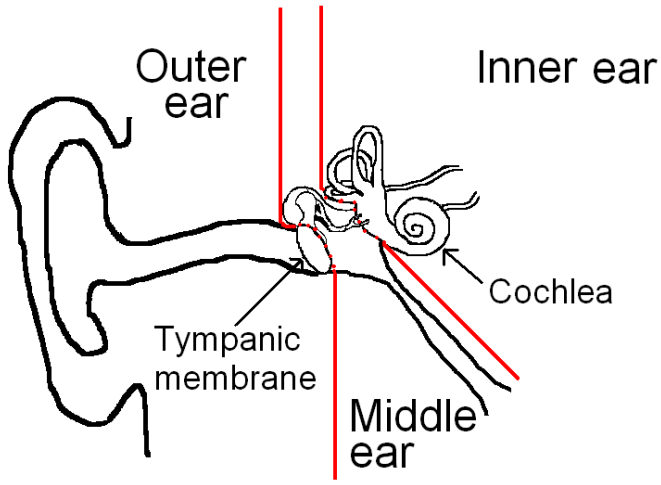


Figure 2.1: Illustration of the human auditory system. The outer ear consists of the visible ear lobe, the ear canal as well as the tympanic membrane (eardrum). The middle ear contains three small bones which transmit the vibrations of the tympanic membrane into the fluid-filled inner ear. The tube in the middle ear is the eustachian tube which connects to the pharynx. When sound is transmitted to the inner ear, it is processed in the snail-shaped Cochlea which can be seen as a frequency analyzer. Finally, nerve connections carry the electrical impulses to the brain for further processing. Note that the three semi-circular canals in the inner ear (upper part of the figure) are not related to hearing, but are instead part of the organ of balance.

ditional European music notation system. A more complete description of the aspects in music should include *loudness*, *pitch*, *timbre* and *duration* of single tones. An elaborate description of these aspects is given in e.g. [18].

Sometimes, music is also described with terms such as *texture* or *style* and these can be seen as combinations of the basic aspects. The area of musicology is, however, constantly changing and other aspects are sometimes included such as gesture and dance. This happens because the aspects of music are perceptual quantities and often based on very high-level cognitive processing.

Music genre classification by humans probably involve most of these aspects of music, although the process is far from fully understood. However, also elements which are extrinsic to the music will influence the classification. The cultural and historical background of a person will have an influence and especially the commercial companies are often mentioned as a driving force. For instance,

music is normally classified into genres in music shops, whether on the street or online, and humans are likely to be influenced by this classification.

It is therefore seen that human music genre classification happens at several levels of abstraction. However, it is unclear how important the different levels are. Especially the importance of intrinsic versus extrinsic elements of the music is relevant here i.e. elements which can be found in the actual audio signal versus the influences from culture, history, and so forth. A clue to this question comes from a recent experiment in [17]. Here, three fish (carps) are trained to classify music into blues or classical music. Their capability to hear is quite similar to human hearing. After the training, they are exposed to new music pieces in the two genres and they are found to actually be able to generalize with low test error. This result suggests that elements intrinsic to music are informative enough for classification when the genres are as different as blues and classical music since the fish are unlikely to know much about the cultural background of the music.

In [21], the abilities of humans to classify styles (subgenres) of classical music were examined. In particular, the 4 styles belong to historical periods of classical music and range from baroque to post-romantic. The experiment investigated a hypothesis about so-called "historical distance" in the sense that music which is close in time will also be similar in sound. One of the interesting points in the experiment is, that even subjects which have almost never been exposed to Western music exhibit "historical distance"-effect. Hence, the cultural background is not essential in this classification and the results in [21] suggest that the subjects use the so-called *temporal variability* in the music to discriminate. The temporal variability is a measure of the durational difference between the onsets of notes. However, the group of Western musicians and Western non-musicians performed better than the non-Westerns. Hence, simply being exposed to Western music without having formal training increases the ability to discriminate between genres, although the Western musicians performed even better.

2.1.1 The problem of defining genre

So far, a formal definition of music genre has been avoided and it has simply been assumed that a "ground-truth" exists. However, this is far from the case and even (especially?) experts on music strongly disagree about the definitions of genre. There seems to be some agreement about the broader genres such as classical music and jazz and e.g subgenres of classical music such as baroque which belongs to a certain historical period. The last century or so, however, has introduced numerous different kinds of music and, as discussed in [4], approaches to precisely define genre tend to ".. end up in circular, ungrounded projections

of fantasies" [4].

Most approaches to the creation of genre taxonomies involve a hierarchical structure as illustrated in figure 2.2 with an example of the genre tree used at Amazon.com's music store [46]. Other online music stores, however, use quite different taxonomies as seen on e.g. SonyMusicStore [55] and the iTunes music store [47]. There is some degree of consensus on the first genre level for genres such as jazz, latin and country. However, there is very little consensus on the subgenres. Note that the structure does not necessarily have to be a tree, but could be a network instead such that subgenres could belong to several genres. This usage of subgenre is sometimes referred to as the *style*.

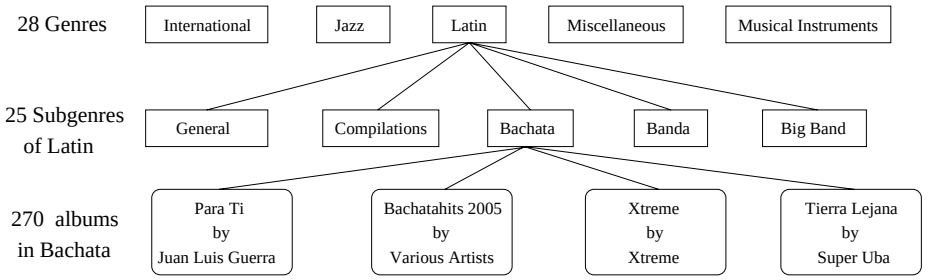


Figure 2.2: Illustration of a part of the genre taxonomy found at Amazon.com's music store [46]. Only a small part of the taxonomy is shown as can be seen from the text in the outer left part of the figure.

As mentioned previously, humans classify music based on low-level auditory and cognitive processing, but also their subjective experiences and cultural background. Hence, it cannot be expected that people will classify music similarly, but the question is how much variability there is. Is it, for instance, meaningful at all to use 500 different genres and subgenres in a genre tree if a person wants to find a certain song or artist ? Will different songs in a subgenre in such a tree sound similar to a person or is the variability among subject simply too large ? Certainly, for practical purposes of music information retrieval, it is not relevant whether music experts agree on genre labels on a song, but whether ordinary non-musicians can use the information.

Section 6.2 describes our experiment with human classification of songs into 11 (prefixed) music genres. Assume that the "true" genre of a song is given by the consensus decision among the human subjects. It is then possible to measure how much people disagree on this genre definition. The percentage of misclassifications was found to be 28% when only songs with at least 3 evaluations were considered. Assume that a person listens to one of the songs in the radio. Now, searching among these (only) 11 genres, the person will start to look for

the song in the wrong genre with a risk of 28% if the song only belongs to one genre. There might of course be methods to bring the percentage down such as using more descriptive genre names. However, in a practical application, it should still be considered whether this error is acceptable or not and especially with more genres.

Another issue is the labelling of music pieces into an existing genre hierarchy. Should a song only be given a single label or should it have multiple ?. The music information provider All Music Guide [45] normally assign a genre as well as several styles (subgenres) to a single album. The assignment on the album-level instead of song-level is very common among music providers. It is a possibility that all or most genres could be described by their proportion of a few broad "basis-genres" such as classical music, electronic music and rock. This seems plausible for especially fusion genres such as blues-rock or pop punk. Such a labelling in proportions would be particularly interesting in relation to automatic music genre classification. It could simply be found from a (large-scale) human evaluation of the music where each genre-vote for a song is used to create the distribution.

2.2 Automatic music genre classification

Automatic music genre classification only appeared as a research area in the last decade, but has seen a rapid growth of interest in that time. A typical example of an automatic music genre classification system is illustrated in figure 1.1. By comparison with figure 2.1, it is seen that the automatic system is build from components which are (more or less intentionally) analogues to the human music genre classification system. In the computer system, the microphone corresponds somehow to the role of the outer and middle ear since they both transmit the vibrations in the air to an "internal media" (electric signal and lymph, respectively). Similarly to the frequency analyzer in the inner ear, a spectral transformation is often applied as the first step in the automatic system. In humans, basic aspects in the music such as melody and rhythm are likely to be used in the classification of music and these are also often modelled in the automatic systems. The *feature part* in the automatic system is thought to capture the important aspects of music. The final human classification is top-down cognitive processing such as matching the heard sound with memories of previously heard sounds. The equivalent in the automatic system to such matching with previously heard sounds, is normally the *classifier* which is capable of learning patterns in the features from a training set of songs.

One of the earliest approaches to automatic music genre classification is found

in [116], although it is not demonstrated exactly on musical genres, but more general sound classes from animals, music instruments, speech and machines. The system first attempts to extract the loudness, pitch, brightness and bandwidth from the signal. The features are then statistics such as mean, variance and autocorrelation (with small lag) of these quantities over the whole sound clip. A gaussian classifier is used to classify the features.

Another important contribution to the field was made in [110] where three different feature sets are evaluated for music genre classification using 10 genres. The 30-dimensional feature vector represents timbral texture, rhythmic content and pitch content. The timbral representation consists of well-known short-time features such as spectral centroid, zero crossing rate and mel-frequency cepstral coefficients which are all further discussed in section 3.1. The rhythmic content, however, is derived from a novel beat histogram feature and similarly, the pitch content is derived from a novel pitch histogram feature. Experiments are made with a gaussian classifier, a gaussian mixture model and a K-nearest neighbor classifier. The best combination gave a classification accuracy of 61 %.

In [82], traditional short-time features are compared to two novel psychoacoustic feature sets for classification of five general audio classes as well as seven music genres. It was found that the psychoacoustic features outperform the traditional features. Besides, four bands of the power spectrum of the short-time features were used as features. This inclusion of the temporal evolution of the short-time features is found to improve performance (see e.g. chapter 4).

The three previously described systems focus mostly on the music representation and the classifiers have been given less interest. Many recent systems, however, use more advanced classification methods to be able to use high-dimensional feature vectors without overfitting. For instance, the best performing system of the audio (music) genre classification contest at MIREX 2005 [53] as described in [7] use a 804-dimensional (short-time) feature space which they classify with an AdaBoost.MH classifier. The contest had 10 contributions and Support Vector Machines (SVMs) were used for classification in 5 of these. SVMs are well-known for their ability to handle high-dimensional feature spaces.

Most of the proposed music genre classification systems consider a few genres in a flat hierarchy. In [12], an hierarchical genre taxonomy is suggested for 13 different music genres, three speech types and a "background" class. The genre taxonomy has four levels with 2-4 splits in each. Hence, to reach the decision of e.g. "String Quartet", the sound clip first has be classified as "Music", "Classical", "Chamber Music" and finally "String Quartet". Feature selection was used on each decision level to find the most relevant features for a given split and gaussian mixture model classifiers were trained for each of these splits.

So far, the music has been represented as an audio signal. In *symbolic music genre classification*, however, symbolic representations such as the MIDI format or ordinary music notation (sheet music) are used. This area is very closely related to "audio-based" music genre classification, but has the advantage of perfect knowledge of e.g. instrumentation and the different instruments are split into separate streams. Limitations of the symbolic representation are e.g. lack of vocal content and the use of a limited number of instruments. In [81] and [80], music genre classification is based on MIDI recordings into 38 genres with an accuracy of 57 % and 9 genres with 90 % accuracy. Although a direct comparison is not possible, these results seem better than the best audio-based results and, hence, give promises for better audio-based performance with the right features.

As explained earlier, there are elements in music genre classification which are extrinsic to the actual music. In [115], this problem is tried solved by combining musical and cultural features which are extracted from audio and text modalities, respectively. The cultural features were found from so-called community metadata [114] which were created by textual information retrieval from artist-queries to the Internet.

Automatic music genre classification is closely related to other areas in MIR. For instance, beat features from the area of music tempo extraction can be used directly as features in music genre classification. A good introduction and discussion of different methods for tempo extraction is found in [99]. Similarly, [65] presents a careful investigation of music transcription which is a difficult and still largely unsolved task in polyphonic music. Instrument recognition is examined in e.g. [79] and although exact instrument recognition as given in the MIDI format is a very difficult problem for ordinary music signals, it is possible to recognize broader instrument families. Other areas in MIR are e.g. music artist identification [77] and audio drum detection [118]. Much of the research in these areas is presented in relation to the International Conferences on Music Information Retrieval (ISMIR) [52] and the MIREX contests in relation to these conferences.

From a wider perspective, MIR and automatic music genre classification can be seen as part of a large group of overlapping topics which are concerned with the analysis of sound in general. The largest topic in this group is arguably Speech Processing if regarded as a single topic. This topic has been investigated for several decades and has several well-established subtopics such as Automatic Speech Recognition (ASR). The first speech recognition systems were actually build in the 1950s [60]. Speech processing is treated in many textbooks such as [94] and [93].

Another topic in the group is Computational Auditory Scene Analysis (CASA)

which is concerned with the analysis of sound environments in general. CASA builds on results from experimental psychology in Auditory Scene Analysis and (often quite complex) models of the human auditory system. One of the main topics in CASA is the disentanglement of different sound streams which humans perform easily. For this reason, CASA has close links to blind source separation methods. A good introduction to CASA is found in [19] and [28] as well as the seminal work by Bregman [10] where the term Auditory Scene Analysis was first introduced. Other examples in the large group are recognition of alarm sounds [29] and general sound environments [1].

2.3 Assumptions and choices

There are many considerations and assumptions in the specification of a music genre classification system as seen in the previous section. The most important assumptions and choices that have been made in the current dissertation as well as the related papers are described in the following and compared to the alternatives.

Supervised learning This requires the songs or sound clips each to have a genre label which is assumed to be the true label. It also assumes that the genre taxonomy is true. This is in contrast to unsupervised learning where the trust is often put on a similarity measure instead of the genre labels.

Flat genre hierarchy with disjoint, equidistant genres These are the traditional assumptions of genre hierarchy. It means that any song or sound clip only belong to a single genre and there are no subgenres. Equidistant genres means that any genre could be mistaken equally likely for any other genre. As seen in figure 6.6, which comes from a human evaluation of the data set, this is hardly a valid assumption. The assumptions on the genre hierarchy are build into the classifier.

Raw audio signals Only raw audio in WAV format (PCM encoding) is used. In some experiments, files with MP3 format (MPEG1-layer3 encoding) have been decompressed to WAV format. This is in contrast to e.g. the symbolic music representation or textual data.

Mono audio In contrast to 2-channel (stereo) or multi-channel sound. It is unlikely to have much influence whether the music is in mono or stereo for music genre classification. Stereo music is therefore reduced to mono by mixing the signals with equal weight.

Real-world data sets This is in contrast to specializing on only subgenres of e.g. classical music. Real-world data sets should ideally consist of all kinds of music. In practice, it should reflect the music collection of ordinary users. This is the music that people buy in the music store and listen to on the radio, TV or Internet. Hence, most of the music will be polyphonic i.e. with two or more independent melodic voices at the same time. It will also consist of a wide variety of instruments and sounds. This demands a lot of flexibility in the music features as opposed to representations of monophonic single-instrument sounds.

Chapter 3

Music features

The creation of music features is split into two separate parts in this dissertation as illustrated in figure 3.1. The first part, *Short-time feature extraction*, starts with the raw audio signal and ends with short-time feature vectors on a 10-40 ms time scale. The second part, *Temporal feature integration*, uses the (multivariate) time series of these short-time feature vectors over larger time windows to create features which exist on a larger time scale. Almost all of the existing music features can be split into two such parts. Temporal feature integration is the main topic in this dissertation and is therefore carefully analyzed in chapter 4.

The first section of the current chapter describes short-time feature extraction in general as well as introduce several of the most common methods. The methods that have been used in the current dissertation project are given special attention. Section 3.2 describes feature ranking and selection as well as the proposed *Consensus Sensitivity Analysis* method for feature ranking which we used in (Paper B).

Finding the right features to represent the music is arguably the single most important part in a music genre classification system as well as in most other music information retrieval (MIR) systems. The genre itself could even be regarded as a high-level feature of the music, but only lower-level features, that are somehow "closer" to the music, are considered here.

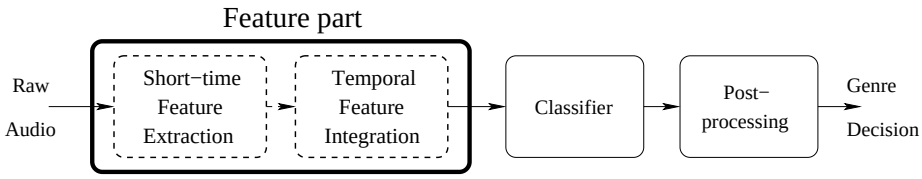


Figure 3.1: The full music genre classification system is illustrated. Special attention is given to the feature part which is here split into two separate parts; Short-time feature extraction and Temporal feature integration. Short-time features normally exist on a 10-40 ms time scale and temporal feature integration combines the information in the time series of these features to represent the music on larger time scales.

The features do not necessarily have to be meaningful to a human being, but simply a model of the music that can convey information efficiently to the classifier. Still, a lot of existing music features are meant to model perceptually meaningful quantities. This seems very reasonable in music genre classification, and even more so than e.g. in instrument recognition, since the genre classification is intrinsically subjective.

The most important demand for a good feature is that two features should be close (in some "simple" metric) in feature space if they represent somehow physically or perceptually "similar" sounds. An implication of this demand is robustness to noise or "irrelevant" sounds. In e.g. [33] and [102], different similarity measures or metrics are investigated to find "natural" clusters in the music with unsupervised clustering techniques. This builds explicitly on this "clustering assumption" of the features. In supervised learning which is investigated in the current project, the assumption is used implicitly in the classifier as explained in chapter 5.

3.1 Short-time feature extraction

In audio analysis, feature extraction is the process of extracting the vital information from a (fixed-size) time frame of the digitized audio signal. Mathematically, the feature vector $\mathbf{x}_n$ at discrete time n can be calculated with the function F on the signal s as

$$\mathbf{x}_n = F(w_0\ s_{n-(N-1)},\ \ldots,\ w_{N-1}\ s_n) \tag{3.1}$$

where w_0, w_1, ..., w_{N-1} are the coefficients of a window function and N denotes the *frame size*. The frame size is a measure of the time scale of the feature. Normally, it is not necessary to have $\mathbf{x}_n$ for every value of n and a *hop size* M is therefore used between the frames. The whole process is illustrated in figure 3.2. In signal processing terms, the use of a hop size amounts to a downsampling of the signal $\mathbf{x}_n$ which then only contains the terms $\ldots, \mathbf{x}_{n-2M}, \mathbf{x}_{n-M}, \mathbf{x}_n, \mathbf{x}_{n+M}, \mathbf{x}_{n+2M}, \ldots$.

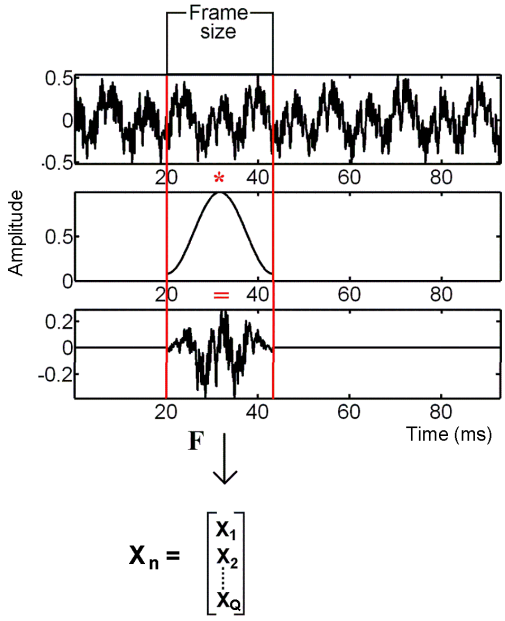


Figure 3.2: Illustration of the traditional short-time feature extraction process. The flow goes from the upper part of the figure to the lower part. The raw music signal s_n is shown in the first of the three subfigures (signals). It is shown how, at a specific time, a frame with N samples is extracted from the signal and multiplied with the window function w_n (Hamming window) in the second subfigure. The resulting signal is shown in the third subfigure. It is clearly seen that the resulting signal gradually decreases towards the sides of the frame which reduces the spectral leakage problem. Finally, F takes the resulting signal in the frame as input and returns the short-time feature vector $\mathbf{x}_n$. The function F could be e.g. the discrete Fourier transform on the signal followed by the magnitude operation on each Fourier coefficient to get the frequency spectrum.

The window function is multiplied with the signal to avoid problems due to finite frame size. The rectangular window with amplitude 1 corresponds to

calculating the features without a window, but has serious problems with the phenomenon of spectral leakage and is rarely used. The author has used the so-called *Hamming window* which has sidelobes with much lower magnitude[1], but other window functions could have been used. Figure 3.3 shows the result of a discrete Fourier transform on a signal with and without a Hamming window and the advantage of the Hamming window is easily seen. The Hamming window can be found as

$$w_n = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) \qquad n = 0, \ldots, N-1$$

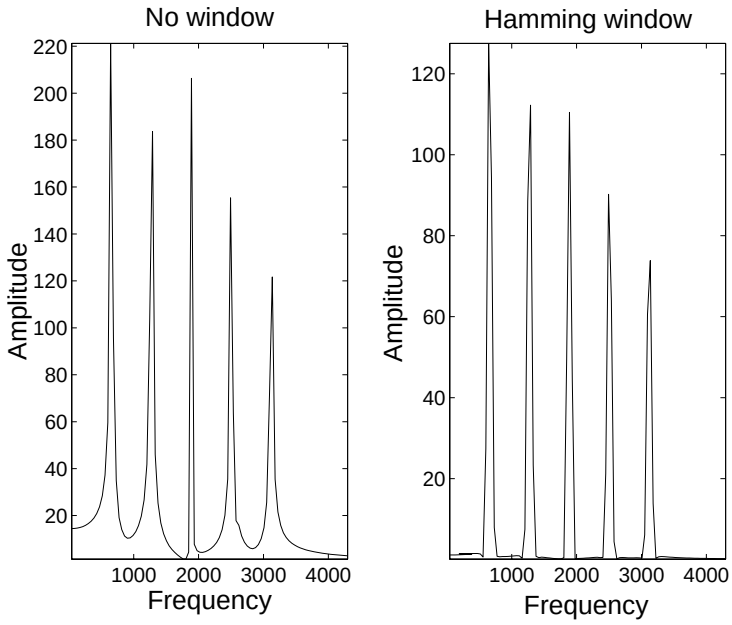


Figure 3.3: The figure illustrates the frequency spectrum of a harmonic signal with a fundamental frequency and four overtones. The signal has a sampling frequency of 22 kHz and the frame size was 512 samples. It is clearly advantageous to use a Hamming window compared to not using a window (or, in fact, a rectangular window) since it is less prone to spectral leakage.

A major part of the work in feature extraction for music and especially speech signals has focused on short-time features. They are thought to capture the

[1]The price for lower magnitudes of the sidelobes is a wider primary lobe. Although it is almost twice as wide as for the rectangular window, the Hamming window is considered much more suitable for music.

short-time aspects of music such as loudness, pitch and timbre. An "informal" definition of short-time features is, that they are extracted on a time scale of 10 to 40 ms where the signal is considered (short-time) stationary.

Numerous short-time features have been proposed in the literature. A good survey of speech features is found in e.g. [90] or [93] and many of these features have also proven useful for music. Many variations of the traditional Short-Time Fourier Transform have been proposed and they often involve a log-scaling of the frequency domain. Also many variations of cepstral coefficients have been proposed [22] [105]. However, it appears that many of these representations perform almost equally well [58] [101]. In general, the frequency representations can be sorted by their similarity with the human auditory processing system. Furthest away from the human auditory systems, we might place the discrete Fourier transform or similar representations. Closer to the human system, we find features from the area of Computational Auditory Scene Analysis (CASA) [19] [10]. For instance, Gamma-tone filterbanks [88] are often used to model the spectral analysis of the basilar membrane instead of simply summing over log-scaled frequency bands as is often done. Although the gamma-tone filterbank is more computationally demanding than a simple discrete Fourier transform, it is still designed to be a trade-off between realism and computational demands. Even more realistic, but also computationally demanding models are found in the areas of psychoacoustics and computational psychoacoustics. Short-time features quite close to the human auditory system have been applied to music genre classification in e.g. [82].

Pitch is one of the most salient basic aspects of music and sound in general. Many different approaches have been taken to estimate the pitch in music as well as speech [99] [107]. In music, pitch detection in monophonic music is largely considered as a solved problem, whereas real-world polyphonic music still offers many problems [5] [65]. Note that many pitch detection algorithms do not really fit into the short-time feature formulation since they often use larger time frames. The reason for this is, that it is important to have a high frequency resolution to distinguish between the different peaks in the spectrum. Still, they are considered as short-time features since the perceptual pitch is a short-time aspect.

In the following, a selection of short-time features will be described in more detail. These are the features which have been investigated experimentally in this dissertation. They also represent the most common features in the literature and many other short-time features can be seen as variations of these.

Mel-Frequency Cepstral Coefficients (MFCC)

Mel-Frequency Cepstral Coefficients (MFCC) originate from automatic speech recognition [93], where they have been used with great success. They were originally proposed in [22]. They have become very popular in the MIR society where they have been used successfully for music genre classification in e.g. [77] and [62] and for categorization into perceptually relevant groups such as moods and perceived complexity in [91].

The MFCCs are to some extent created according to the principles of the human auditory system [72], but also to be a compact representation of the amplitude spectrum and with considerations of the computational complexity. In [4], it is argued that they model timbre in music. [70] compare them to auditory features with more accurate (and computationally demanding) models, but still find the MFCCs superior. In (Paper B), we also find the MFCCs to perform very well compared to a variety of other short-time features and similar observations are made in [62] and [41]. For this reason, the MFCCs have been used as the standard short-time feature representation in our experiments with temporal feature integration (as described in chapter 4) and, therefore, a more careful description of these features is given in the following.

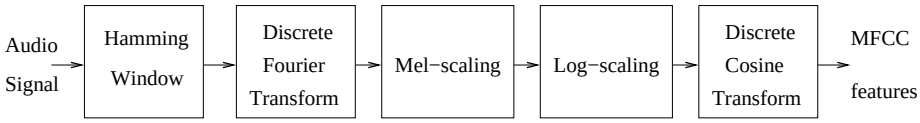


Figure 3.4: Illustration of the calculation of the Mel-Frequency Cepstral Coefficients (MFCCs). The flow chart illustrates the different steps in the calculation from raw audio signal to the final MFCC features. There exist many variations of the MFCC implementation, but nearly all of them follow this flow chart.

Figure 3.4 illustrates the construction of the MFCC features. In accordance with equation 3.1, the feature extraction can be described as a function F on a frame of the signal. After applying the Hamming window on the frame, this function contains the following 4 steps :

1. **Discrete Fourier Transform** The first step is to perform the discrete Fourier transform on the frame. For a frame size of N, this results in N (complex) Fourier coefficients. The phase is now discarded as it is thought to represent little value to human recognition of speech and music. This results in an N-dimensional spectral representation of the frame.

2. **Mel-scaling** Humans order sounds on a musical scale from low to high

with the perceptual attribute named *pitch*[2]. The pitch of a sine tone is closely related to the physical quantity of frequency and the fundamental frequency for a complex tone. However, the pitch scale is not similarly spaced as the frequency scale. The *mel-scale* is an estimate of the relation between the perceived pitch and the frequency which is found by equating 1000 mels to a 1000 Hz sine tone at 40 dB. It is used in the calculation of the MFCCs to transform the frequencies in the spectral representation into a perceptual pitch scale. Normally, the mel-scaling step has the form of a filterbank of (overlapping) triangular filters in the frequency domain and with center frequencies which are mel-spaced. A standard filterbank is illustrated in figure 3.5. Hence, this mel-scaling step is also a smoothing of the spectrum and a dimensionality reduction of the feature vector.

3. **Log-scaling** Similarly to pitch, humans order sound from soft to loud with the perceptual attribute *loudness*. Perceptual loudness corresponds quite closely to the physical measure of intensity. Although other quantities, such as frequency, bandwidth and duration, affect the perceived loudness it is common to relate loudness directly to intensity. As such, the relation is often approximated as $L \propto I^{0.3}$ where L is the loudness and I is the intensity (Stevens' power law). It is argued in e.g. [72], that the perceptual loudness can also be approximated by the logarithm of the intensity, although this is not quite similar to the previously mentioned power law. This is a perceptual motivation for the log-scaling step in the MFCC extraction. Another motivation for the log-scaling in speech analysis is that it can be used to deconvolute the slowly varying modulation and the rapid excitation with pitch period [94].

4. **Discrete Cosine Transform** As the last step, the discrete cosine transform (DCT) is used as a computationally inexpensive method to decorrelate the mel-spectral log-scaled coefficients. In [72], it is found that the basis functions of the DCT are quite similar to the eigenvectors of a PCA analysis on music. This suggests that the DCT can actually be used for the de-correlation. As illustrated in figure 4.2, the assumption of de-correlated MFCCs is, however, doubtful. Normally, only a subset of the DCT basis functions are used and the result is then an even lower dimensional feature vector of MFCCs.

It should be noted that the above procedure is the general procedure for calculating MFCCs, but other authors use variations of the above theme [35]. In our work, the Voicebox Matlab-package has been used [50].

Another note regards the zero'th MFCC which is a measure of the short-time

[2]In fact, the ANSI (1973) definition of pitch is :"..that attribute of auditory sensation in terms of which sounds may be ordered on a scale extending from high to low"

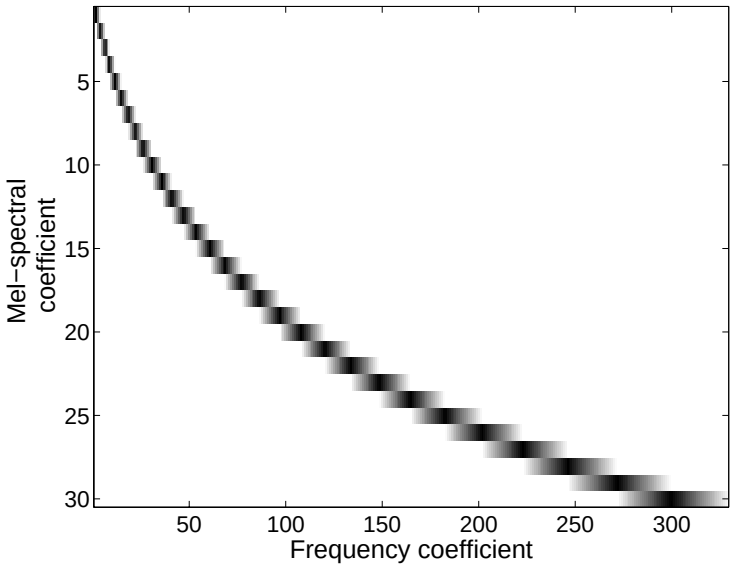


Figure 3.5: Illustration of the filterbank/matrix which is used to convert the linear frequency scale into the logarithmic mel-scale in the calculation of the Mel-Frequency Cepstral Coefficients. The filters are seen to be overlapping and have logarithmic increase in bandwidth.

energy. This value is sometimes discarded when other measures of energy are used for the total feature vector.

Linear Prediction Coefficients (LPC)

Like the MFCCs, the *Linear Prediction Coefficients* (LPC) have been used in speech analysis for many years [93]. In fact, linear prediction has an even longer history which originates in areas such as astronomy, seismology and economics. The idea behind the LPCs is to model the audio time signal with a so-called all-pole model. This model is thought to apply to the production of (non-nasal) voiced speech. In [89] the LPCs are used for recognition of general sound environments such as restaurant environment and traffic and they have been used successfully in [7] for music genre classification. Our experiments, however, suggest that the LPCs are less useful in music genre classification if the choice is between them and the MFCCs (Paper B).

The basic model in linear prediction is

$$s_n = a_1 s_{n-1} + a_2 s_{n-2} + \ldots + a_P s_{n-P} + G u_n$$

for the signal s_n and linear prediction coefficients a_i up to the model order P. Here, G is the gain factor and u_n is an error signal. Assuming the error to be a (stationary) white gaussian noise process, the LP coefficients (LPCs) a_i are found by standard least-squares minimization of the total error E_n which can be written as

$$E_n = \sum_{i=n-N+P}^{n} (s_i - \sum_{j=1}^{P} a_j s_{i-j})^2$$

for the frame n. A variety of methods can be used for the minimization such as the autocorrelation method, covariance method and the lattice method [94] which differ mostly in the computational details. In our work, the Voicebox Matlab implementation [50] has been used which uses the autocorrelation method. The LPCs are then ready to be used as a feature vector in the following classification steps. In our work, the square-root of the minimized error i.e. the estimate of the gain factor G, is added as an extra feature to the LPC feature vector.

The linear prediction model is perhaps best understood in the frequency domain. As explained in e.g. [76], the LPC captures the spectral envelope and the model order P decides the flexibility to model the envelope. In (Paper G), we have given a more careful explanation of this model to be used in the context of temporal feature integration (see chapter 4).

Delta MFCC (DMFCC) and delta LPC (DLPC)

The *delta MFCC* (DMFCC) features have been used for music genre classification in e.g. [109] and for music instrument recognition in [30]. They are derived from the MFCCs as

$$DMFCC_n^{(i)} = MFCC_n^{(i)} - MFCC_{n-1}^{(i)}$$

where i indicates the ith MFCC coefficient.

Similarly, the *delta LPC* (DLPC) features are derived from the LPCs as

$$DLPC_n^{(i)} = LPC_n^{(i)} - LPC_{n-1}^{(i)}$$

Zero-Crossing Rate (ZCR)

The *Zero-Crossing Rate* (ZCR) also has a background in speech analysis [94]. This very common short-time feature has been used for music genre classification in e.g. [67] and [117]. It is simply the number of time-domain zero-crossings in a time window. This can be formalized as

$$ZCR_n = \sum_{i=n-N+1}^{n} |\text{sgn}(s_i) - \text{sgn}(s_{i-1})|$$

where the sgn-function returns the sign of the input. For simple single-frequency tones, this is seen to be a measure of the frequency. It can also be used in speech analysis to discriminate between voiced and unvoiced speech since ZCR is much higher for unvoiced than voiced speech.

Short-Time Energy (STE)

The common *Short-Time Energy* (STE) has been used in speech and music analysis as well as many other areas. It is used to distinguish between speech and silence, but mostly useful in high signal-to-noise ratio. It is a very common short-time feature in music genre classification and has been used in one of the earliest approaches to sound classification [116] to distinguish between (among other things) different music instrument sounds. Short-Time Energy is calculated as

$$STE_n = \frac{1}{N} \sum_{i=n-N+1}^{n} s_i^2$$

for a signal s_i at time i. The loudness of a sound is closely related to the intensity of a signal and therefore the STE [94].

Basic Spectral MPEG-7 features

The MPEG (Moving Picture Experts Group [48]) is a working group of the ISO/IEC organization for standardization of audiovisual content and has had great success with MPEG-1 (1992) and MPEG-2 (1994). MPEG-7 (2002) is known as a "Multimedia Content Description Interface" and is involved with the description rather than the representation of audiovisual content.

In the following, 4 different feature sets from the MPEG-7 framework will be described. They are described in detail in [86]. Note that some degree of variation of the actual implementations and system parameters are allowed within the MPEG-7 framework and that our implementation is described in the following. In the MPEG-7 terminology the features are called the Basic Spectral low-level audio descriptors. The basis of these features is the *Audio Spectrum Envelope* (ASE) features which is the power spectrum in log-spaced frequency bands. Hence, the first step is to calculate the discrete Fourier transform (using the Hamming window again) over the 30 ms frame to estimate the power spectrum. Afterwards, a 1/4-octave spaced filterbank (of non-overlapping square filters) is applied to summarize the power in these log-spaced frequency bands. The edges are anchored at 1 kHz. The low edge is at 62.5 Hz, the high edge at 9514 Hz and two extra coefficients summarize the power below and above these edges. This spectral representation is the ASE features. It is seen that this representation is actually not very different from the first two steps of the MFCC features although the filters are neither overlapping nor triangular. The ASE features have been used in e.g. audio thumbnailing in [112] and in general sound classification in [16].

The *Audio Spectrum Centroid* (ASC) and *Audio Spectrum Spread* (ASS) features are calculated in accordance with the ASE features. The ASC feature is the normalized weighted mean (or centroid) of the log-frequency which can be formulated as

$$ASC = \frac{\sum_{i=1}^{N} \log_2(f_i/1000) P_i}{\sum_{i=1}^{p} P_i}$$

where f_i is the frequency of the i'th frequency coefficient with power P_i. The number N is the total number of frequency coefficients of the ASE feature before the log-scaling i.e. equal to the number of Fourier coefficients which is also the frame size in number of samples. This feature indicates at which frequency the dominating power lies (especially for narrow-band signals), but obviously with all the weaknesses of a simple mean value. It is thought to be the physical correlate of the perceptual concept of sharpness [86]. There exist many different

variations of the spectral centroid short-time feature, but they are basically all the weighted mean of the frequency spectrum [98] [68] [79]).

As the ASC feature can be seen as the weighted mean of the log-spaced frequency, the *Audio Spectrum Spread* can be seen as the weighted standard deviation. Mathematically, this is

$$ASS = \sqrt{\frac{\sum_{i=1}^{N}(\log_2(f_i/1000) - ASC)^2 P_i}{\sum_{i=1}^{N} P_i}}$$

with the same notation as before. The ASS feature thus measure the spread of the power about the mean and has been found to discriminate between tone-like and noise-like sounds [86].

The *Spectral Flatness Measure* (SFM) features express the deviation from a flat power spectrum of the signal in the short-time frame. Large deviation from a flat shape could indicate tonal components. The SFM feature has been used in e.g. [34] for audio fingerprinting and [12] for music genre classification. The calculation of the SFM features largely follow that of the first steps for the ASE features. Like for the ASE features, 1/4-octave frequency bands with edges f_k are used. However, to increase robustness, the bands are increased with 5% to each side in the SFM extraction. Instead of summing over the power spectrum coefficients $\tilde{P}_i$ as for the ASE features, the SFM features are found in each band k as

$$SFM_k = \frac{\sqrt[N_k]{\prod_{i=n(k)}^{n(k+1)} \tilde{P}_i}}{\frac{1}{N_k}\sum_{i=n(k)}^{n(k+1)} \tilde{P}_i}$$

where $n(k)$ is index function of the power spectrum coefficients $\tilde{P}_i$ between the edges f_k and f_{k+1} and N_k is the corresponding number of coefficients. The reader is referred to [86] for more specific details of the implementation. [26] introduces a variant of the Spectral Flatness Measure.

3.2 Feature ranking and selection

Numerous music features exist of which some have been described in the previous sections. However, for several reasons it is necessary to choose only a subset of these for our music genre classification system. For instance, there may be limitations in both computational space and time. Another important concern is the curse of dimensionality which implies that, for a given training set, adding more features (information) will actually raise the generalization error of most classifiers at a certain point. In a music genre classification task, the possible genres could be different subgenres of Classical music. However, the best features for such a system might not be the best for a system that should discriminate between subgenres of Heavy Metal. Therefore *feature selection* is often useful. Feature selection is the process of selecting the subset of features which minimizes the generalization error or another performance measure. *Feature ranking* estimates the usefulness of the features individually (in contrast to a larger subset). A wide variety of ranking and selection methods have been proposed in the literature. A good overview of these can be found in [39] and are also discussed in e.g. [8].

Note that although only feature selection is considered here, it could be viewed as just another example of *dimensionality reduction*. A multitude of different methods have been proposed for dimensionality reduction such as Principal Component Analysis, Independent Component Analysis, Partial Least Squares, Non-negative Matrix Factorization and so forth [27] [56]. However, most of these methods use a (linear) combination of all of the features (with the notable exception of *sparse* methods [43] [119]). Therefore, it is necessary to extract all of the features to apply the technique which is computationally demanding.

Feature selection has quite often been used in music genre classification. In [12], 90 different features are initially considered, but 32 of these were discarded in an initial feature selection based on robustness to added white noise and bandwidth changes. Afterwards, a sequential forward selection method was used to find subsets of increasing size that maximized a measure of class separability. [67] started out with 8 features and systematically found the classification test errors of every subset with 3 features. This brute-force feature selection method is very good if it is known that the best performance is achieved with 3 features. However, the optimal number of features is rarely known. Besides, this brute-force method becomes computationally infeasible for even quite small numbers of features since the whole system has to be trained and tested for each combination. Assuming that the optimal number of features is not known and we have 100 features, $2^{100} \approx 10^{30}$ training and testing phases are necessary to find the optimal subset with this method.

3.2.1 Consensus sensitivity analysis

In (Paper B), the author has proposed *Consensus Sensitivity Analysis* for feature ranking to estimate the usefulness of the music features individually. The method is based on the estimate of the probability $\hat{P}(C|\mathbf{z}_n)$ which is the probability of a genre conditioned on the feature vector $\mathbf{z}_n$. The idea is to quantify the change in output $\hat{P}(C|\mathbf{z})$ for a given change in the i'th feature $\mathbf{x}^{(i)}$. Here, $\mathbf{z}$ is a fixed transformation of the feature vector $\mathbf{x}$ as occurs in e.g. temporal feature integration (see chapter 4). The larger the change in output $\hat{P}(C|\mathbf{z})$, the more important the i'th feature is considered to be and this is used to rank the individual features. Mathematically, the sensitivity contribution of feature i can be found as

$$\mathbf{s}^{(i)} = \frac{1}{N\,N_c} \sum_{c=1}^{N_c} \sum_{n=1}^{N} \left| \frac{\partial \hat{P}(C=c|\mathbf{z}_n)}{\partial \mathbf{x}_n^{(i)}} \right| \tag{3.2}$$

where N is the number of frames in the training set and N_c is the number of genres. These values are named *absolute value average sensitivities* [104] [64].

The above procedure describes the creation of the sensitivities $\mathbf{s}^{(i)}$ and $\mathbf{s}$ can be seen as the values in a sensitivity map which can be used to rank the features. However, in our experiments, several cross-validation runs or other resamplings have been made which give several different rankings on the same feature set. The Consensus Sensitivity Analysis use consensus among the different runs to find a single ranking. For instance, assume that 50 resamplings are made which means that each feature $\mathbf{x}^{(i)}$ has 50 different "votes" for the ranking position. The most important ranking position (position 1) is simply found as the feature with most votes as ranking 1. This feature then "wins" this ranking position and is not considered further. To find the feature with ranking position 2, the votes to be ranked 2nd are counted, but all votes to be ranked 1 are added. Hence, all previous votes are cumulated in the competition. This procedure continues until all features are given a ranking. In the case of equal amounts of votes among several features, the ranking is random.

Chapter 4

Temporal feature integration

The topic of the current chapter is *Temporal feature integration* which is the process of combining (integrating) all the short-time feature vectors in a time frame into a new single feature vector on a larger time scale. The process is illustrated in figure 4.1. Although temporal feature integration could happen from any time scale to a larger one (e.g. 1 s to 10 s), it is most commonly applied to time series of short-time features (10-40 ms) as the ones described in section 3.1. Temporal feature integration is important since only aspects such as sound timbre or loudness is represented on the short time scale. Aspects of music such as rhythm, melody and melodic effects such as tremolo are found on larger time scales as discussed in chapter 2.

In the first part of the chapter, temporal feature integration is discussed in general terms. Then, the very commonly used *Gaussian Model* is discussed which simply uses the mean and variance (or covariance) of the short-time features as new features. Afterwards, the *Multivariate Autoregressive Model* is presented. We proposed this model in relation to the current dissertation project in (Papers C and G). The model is carefully analyzed and is considered as one of the main contributions of this dissertation. The following section discusses the *Dynamic Principal Component Analysis* model which was also proposed in relation to the current dissertation. We proposed this model in (Paper B). The remaining parts of the chapter discuss different features which were proposed by other authors, but that have been investigated for comparison in the current project. These

features are based on temporal feature integration of short-time features up to a higher time scale, but they are less general than the previously mentioned methods. For instance, the Beat Spectrum feature is meant to capture the beat explicitly and the High Zero-Crossing Rate Ratio feature is specifically meant for the Zero-Crossing Rate short-time feature.

As explained before, temporal feature integration is the process of integrating several features over a time frame into a single new feature vector as illustrated in figure 4.1. The hope is that the new feature vector will be able to capture the important temporal information as well as dependencies among the individual feature dimensions. The process can be formalized as

$$\mathbf{z}_n = T(\mathbf{x}_{n-(N-1)}, \ldots, \mathbf{x}_n) \tag{4.1}$$

where $\mathbf{z}_n$ is the new feature vector at the larger time scale, $\mathbf{x}_n$ is the time series of (short-time) features and N is the *frame size*. The transformation T performs the temporal feature integration.

The short-time features in chapter 3 are normally extracted from 10-40 ms and they are able to capture aspects which live on that time scale such as sound loudness, timbre and pitch. However, many aspects of music exist on larger time scales. For instance, the beat rate in a song normally lies in the range of 40-200 b.p.m (beats-per-minute) and therefore the time interval between successive beat pulses is in the range of 300-1500 ms. This is clearly not captured on the short time scale. In [110] it is argued that important information lives on a 1s time scale which is named a "texture window". [79] argues that e.g. note changes are important for music instrument recognition. Other phenomena in music which exist on different, longer time scales are tremolo, vibrato, auditory roughness, the melodic contour and rhythm. Although the importance of such long-term aspects is not very well known for human music genre classification, they cannot be neglected as discussed in section 2.1.

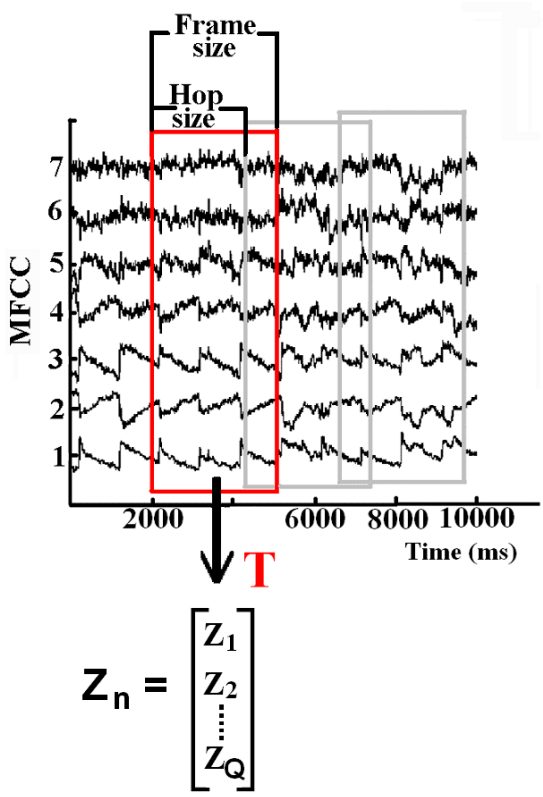


Figure 4.1: Illustration of the process of Temporal feature integration. The upper part of the figure illustrates the temporal evolution of the first 7 MFCCs which have been extracted from the middle of the song "Master of Revenge" by the band "Body Count". Hence, the x-axis shows the temporal evolution of short-time features and the y-axis shows the different dimensions of the short-time feature vector. Although MFCCs are used here, any (multivariate) time series of short-time features could be used. The feature values have been scaled for the purpose of illustration. The red box contains the information that is used for temporal feature integration. The number of short-time feature vectors which are used, is given by the frame size N and the hop size M is the distance between adjacent frames. The transformation T is the temporal feature integration transform which returns the feature vector $\mathbf{z}_n$ on the larger time scale. T might simply be to take the mean and variance over the frame size of each MFCC individually which would here result in a 14-dimensional ($Q = 14$) feature vector $\mathbf{z}_n$. Note that there appears to be structure in the signals in both time and between the short-time feature dimensions (the MFCCs). This is especially clear for the first MFCCs.

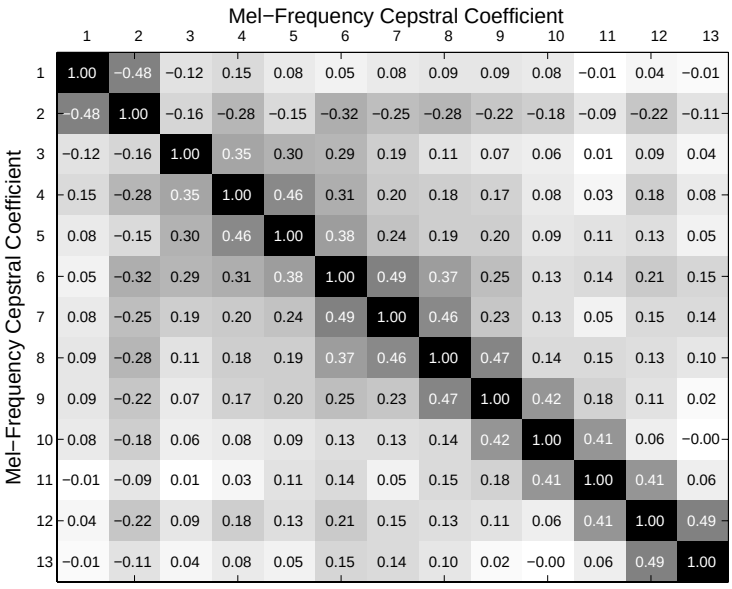


Figure 4.2: Illustration of the correlation coefficients (Pearson product-moment correlation coefficients) between the first 13 (short-time) MFCC features. The coefficients have been estimated from data set A. There appears to be (linear) dependence between some of the neighboring coefficients.

It has now been argued that humans use temporal structure in the music for genre classification. This is also quite evident when looking at the multivariate time series of MFCC coefficients in figure 4.1. There seems to be a clear pattern in the temporal structure and a good temporal feature integration method should capture this structure. However, there also seems to be correlations between the different coefficients. Figure 4.2 illustrates the correlation coefficients between the 13 first MFCCs from our data set A (see section 6.2). This indicates that some, and especially adjacent, MFCCs are correlated and a good model should consider that.

The integrated feature $\mathbf{z}_n$ in equation 4.1 normally has higher dimensionality than the $\mathbf{x}_n$ features. This is necessary to capture all the relevant information from the N frames. For instance, the common Gaussian Model uses the mean and variance of each element in $\mathbf{x}_n$ in the frame. Hence, the dimensionality of the vector $\mathbf{z}_n$ will be twice as large as for $\mathbf{x}_n$. It may therefore appear that this new representation uses twice as much space. However, as for the feature extraction, a *hop size* M is normally used between the frames and, in fact,

the temporal feature integration is normally a *data compression*. For instance, assume we start with 30s music at 22050 Hz i.e. 661500 samples. In a typical implementation, this might result in 4000 MFCCs of dimension 6 when using frame size 15 ms and hop size 7.5 ms. Using e.g. the proposed MAR features (described in section 4.2) for temporal feature integration reduces this to 70 MAR features of dimension 135 when using frame size 1200 ms and hop size 200 ms for the integration. In other words, the compression from raw audio to MFCC is approximately a factor 10 and the MAR features compress the data further with a factor 2.5. Although the main concern here is the classification performance, the usage of space for storage and handling the features is worth considering for practical applications. Additionally, our results indicate that data could be compressed much more with a fairly small loss of performance. For instance, instead of using 70 MAR feature vectors to represent a song, a single MAR feature vector could represent the whole song with approximately 10% decrease in performance.

The literature contains a variety of different temporal feature integration methods for both music, speech and sound in general. The reason is that the semantic content in sound is very important such as melodies, rhythms and lyrics in music and e.g. words or sentences in speech. In speech recognition, short-time features have traditionally been considered sufficient. However, recently, there have been signs of a paradigm-shift to consider longer time frames and with indications that temporal feature integration might be the solution [85] [40].

Common temporal feature integration methods use some simple statistics of the short-time features such as the mean, variance, skewness or autocorrelation at a small lag [38] [110] [116]. This is by far the most common methods. Another approach has been taken in [13] which model the temporal evolution in the energy contour by a polynomial function (although on quite short time frames and focusing on general sound). The temporal feature integration method in [31] focuses on music genre classification. Their technique is to use the entropy, energy ratio in frequency bands, brightness, bandwidth and silence ratio of ordinary DFT short-time features as features on a 5 s time scale.

In [108], pitch histograms were proposed to capture the short-time pitch content over a full song. The technique resembles the beat histogram procedure which is described in section 4.7. This temporal feature integration method is therefore specifically targeted at pitch short-time features although it might be possible to generalize the technique. A number of different features were extracted from the pitch histogram.

Sometimes, the line between temporal feature integration and classifier or similarity measure is thin. For instance, in the interesting contributions [78] and [83] Gaussian Mixture Models (GMM) were used to model the probability density

of short-time features. This is integrated into a Support Vector Classifier kernel and as such could be regarded as part of the classifier. However, it might also be seen as a temporal feature integration method where the parameters of the GMM are the new feature vector.

The following sections describe the most common temporal feature integration methods as well as the methods which were believed to be the most promising state-of-the-art methods. Besides, our two proposed models are introduced and carefully explained. All of the following techniques have been used in our experiments with temporal feature integration.

4.1 Gaussian Model

By far, the most common temporal feature integration method is to use the mean and variance in time over a sequence of (short-time) feature vectors. This has been used for music genre classification in e.g. [70] and [69] and to detect the mood in music in [71]. Most authors use these statistics without much notice of the implicit assumptions that are being made. In fact, it amounts to using only the mean and variance to describe the full probability density distribution $p(\mathbf{x}_{n-(N-1)}, \ldots, \mathbf{x}_n)$ of the feature vectors $\mathbf{x}_n$ at time n. Hence, the method assumes that the feature vectors $\mathbf{x}_n$ are drawn independently from a *Gaussian probability distribution with diagonal covariance matrix*. The assumption is independence both in time and among the coefficients of the feature vector. As discussed previously, this is hardly a valid assumption.

MeanVar features

The integrated feature, here named *MeanVar*, is then

$$\mathbf{z}_n = \begin{bmatrix} \hat{\mathbf{m}}_n \\ \hat{\Sigma}_{11(n)} \\ \vdots \\ \hat{\Sigma}_{dd(n)} \end{bmatrix}$$

where

$$\hat{\mathbf{m}}_n = \frac{1}{N} \sum_{i=n-(N-1)}^{n} \mathbf{x}_i$$

is the mean value estimate at time n and

$$\hat{\Sigma}_{kk(n)} = \frac{1}{N-1} \sum_{i=n-(N-1)}^{n} \left(\mathbf{x}_i^{(k)} - \hat{\mathbf{m}}_n^{(k)} \right)^2$$

is the variance estimate of feature k at time n. N is the temporal feature integration frame size.

MeanCov features

A straightforward extension of the above feature integration model, would be to allow for a full covariance matrix as has been done in (Paper G). This would capture the correlations between the individual feature dimensions. However, for a feature vector of dimension d, there are $d(d+1)/2$ (informative) elements in the full covariance matrix as opposed to only d elements in the diagonal matrix. This might be a problem for the classifier due to the "curse of dimensionality" [8].

The *MeanCov* feature is defined as

$$\mathbf{z}_n = \begin{bmatrix} \hat{\mathbf{m}}_n \\ \hat{\Sigma}_{11(n)} \\ \hat{\Sigma}_{12(n)} \\ \vdots \\ \hat{\Sigma}_{1d(n)} \\ \hat{\Sigma}_{2d(n)} \\ \vdots \\ \hat{\Sigma}_{dd(n)} \end{bmatrix}$$

where the elements are defined as for the MeanVar feature except that $\hat{\Sigma}_{ij}$ is now the covariance estimate between feature i and j instead of simply the variances.

4.2 Multivariate Autoregressive Model

As seen in the previous chapter, the ordinary MeanVar features do not model temporal dependencies or dependencies among the individual features in a feature vector (e.g. between MFCC 1 and MFCC 5). The MeanCov features were able to improve this by modelling correlations among individual features, but still not the temporal dependencies.

In relation to the current dissertation project, we have proposed and carefully investigated the *Multivariate Autoregressive Model* which models both dependencies in time and among individual features. We have examined and evaluated this temporal feature integration model in (Papers C and G). It can be seen as an improved and natural extension to the previously mentioned Gaussian Model which is in fact a special case of the multivariate autoregressive model.

In [3], the LP-TRAP model is proposed for speech recognition. This model resembles our model, but considers each feature dimension individually.

The multivariate autoregressive model is carefully explained in the following. The basic idea is to model the multivariate time series of feature vectors with an autoregressive model. Contrary to the Gaussian Model, the dynamics in the time series is then modelled. Mathematically, the model can be written in terms of the random process $\mathbf{x}_n$ as

$$\mathbf{x}_n = \sum_{p=1}^{P} \mathbf{A}_p \, \mathbf{x}_{n-p} + \mathbf{v} + \mathbf{u}_n \tag{4.2}$$

where P is the model order, the $\mathbf{A}_i$'s are (deterministic) autoregressive coefficient matrices, $\mathbf{v}$ is the so-called (deterministic) intercept term and $\mathbf{u}_n$ is the driving noise process. It is found that $\mathbf{v} = (\mathbf{I} - \sum_{p=1}^{P} \mathbf{A}_p)\boldsymbol{\mu}$ where $\boldsymbol{\mu}$ is the mean of the signal process $\mathbf{x}_n$. Hence, the intercept term $\mathbf{v}$ is included explicitly to allow a (fixed) mean value of the feature signal. The noise process $\mathbf{u}_n$ is here restricted to be white noise (ie. without temporal dependence) with zero mean and covariance matrix $\mathbf{C}$. It is immediately seen that the Gaussian Model corresponds to $P = 0$ and gaussian distributed noise $\mathbf{u}_n$.

The univariate version of this model is very common for time series modelling and has been used extensively in a wide range of areas from geosciences and astronomy to quantum physics. It is also very widely used in signal processing and it should be noted that it is in fact this model which is used to find the LPC short-time features as explained in chapter 3. The multivariate version

has, however, received less attention.

Interpretation of the model in time and frequency domain

The autoregressive model can be understood in the time domain as well as the frequency domain. In the time domain, the model can be seen as a predictor of future values. Assuming that the model parameters are known and given realizations of $\mathbf{x}_{n-1}$ to $\mathbf{x}_{n-P}$, the next feature vector can be predicted as

$$\hat{\mathbf{x}}_n = \sum_{p=1}^{P} \mathbf{A}_p \, \mathbf{x}_{n-p} + \mathbf{v} \tag{4.3}$$

which is the expectation value $E(\mathbf{x}_n | \mathbf{x}_{n-1}, \ldots, \mathbf{x}_{n-P}, \mathbf{A}_1, \ldots, \mathbf{A}_P, \mathbf{v})$. A measure of how well the model fits the signal can be found as

$$\mathbf{e}_n = \mathbf{x}_n - \hat{\mathbf{x}}_n = \mathbf{x}_n - \sum_{p=1}^{P} \mathbf{A}_p \, \mathbf{x}_{n-p} - \mathbf{v} \tag{4.4}$$

which can be seen as a (sliding) error estimate and is sometimes called the residual.

In the frequency domain, the interpretation of the multivariate autoregressive model becomes slightly more cumbersome. In the following, the interpretation of the univariate autoregressive model is discussed instead. This amounts to the assumption of diagonal matrices $\mathbf{A}_j$ and diagonal noise covariance $\mathbf{C}$.

The frequency-domain interpretation of the univariate autoregressive model can be described as *spectral matching* to the power spectrum of the signal. This capability to capture the spectral envelope of the power is illustrated in figure 4.3. To understand how this spectral matching is possible, it is useful to first consider the signal in the z-domain. The following derivations follow Makhoul [76] and starts by transforming the univariate version of equation 4.4 to

$$E(z) = \left(1 - \sum_{p=1}^{P} a_p z^{-p}\right) X(z) = A(z)X(z) \tag{4.5}$$

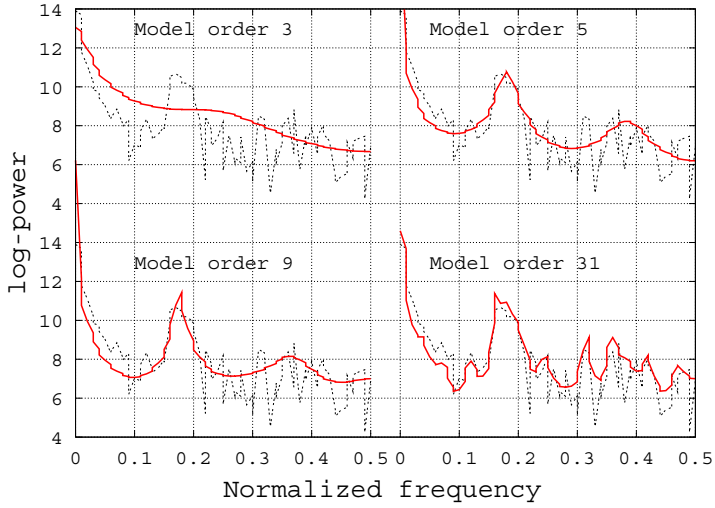


Figure 4.3: Illustration of the spectral matching capabilities of the autoregressive (AR) model. Four different subplots are illustrated which show the modelling power of four different AR model orders. The black line in each plot shows the periodogram of the time series of the first MFCC coefficient. The time series represented the sound of note $A5$ on a piano over a duration of 1.2 s. The red line illustrates the AR-model approximation for the different model orders. It is clearly seen that the AR-model approximation becomes increasingly accurate as the model order increases.

where $E(z)$ is the error or residual in the z-domain. Without loss of generality, it has been assumed that the mean value of the signal and hence $\mathbf{v}$ is zero. As explained later, the *least squares* method is being used in the current work to estimate the parameters of the model. This corresponds to an assumption of gaussian distributed noise $\mathbf{u}_n$. The parameter estimation is then found by minimization of the total error ϵ_{tot}. This can be understood in the frequency domain by the use of Parseval's Theorem as

$$\epsilon_{tot} = \sum_{i=-\infty}^{\infty} e_i^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |E(e^{j\omega})|^2 d\omega \tag{4.6}$$

where e_i is the univariate residual from equation 4.4 and $E(e^{j\omega})$ is the frequency domain version of the error in equation 4.5. Hence, minimizing ϵ_{tot} corresponds to minimizing the integrated power spectrum of $E(e^{j\omega})$. To relate E, the autoregressive model power spectrum $\hat{P}$ and the power spectrum P of the signal

x_n, it is necessary to transform the autoregressive model in 4.2 to

$$X(z) = \sum_{p=1}^{P} a_p X(z) z^{-p} + GU(z)$$

where the so-called *gain factor* G allows the noise process u_n to have unit variance or in other words that $|U(e^{j\omega})| = 1$. The system transfer function then becomes

$$H(z) \equiv \frac{X(z)}{U(z)} = \frac{G}{1 - \sum_{p=1}^{P} a_p z^{-p}}$$

and, using the substitution $z = e^{j\omega}$, the model power spectrum in the frequency domain is

$$\hat{P}(\omega) = |H(e^{j\omega})U(e^{j\omega})|^2 = \frac{G^2}{|A(e^{j\omega})|^2}$$

where A was defined in equation 4.5. Since $P(\omega) = |X(e^{j\omega})|^2$ and using the relations 4.5 and 4.6, it is seen that the total error to be minimized can be written in the frequency domain as

$$\epsilon_{tot} = \frac{G^2}{2\pi} \int_{-\pi}^{\pi} \frac{P(\omega)}{\hat{P}(\omega)} d\omega \tag{4.7}$$

Hence, minimizing ϵ_{tot} corresponds to the minimization of the integrated ratio between the signal power spectrum $P(\omega)$ and the model power spectrum $\hat{P}(e^{j\omega})$. The minimum error is found to be $\epsilon_{tot} = G^2$. After minimization, the model power spectrum can therefore be assumed to satisfy the relation

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{P(\omega)}{\hat{P}(\omega)} d\omega = 1 \tag{4.8}$$

The two relations, equations 4.7 and 4.8, has two main implications which can be stated as the *global* and *local* properties of the autoregressive model [76]. These properties describe the spectral matching capabilities of the autoregressive model.

Global property Since the contribution to the total error is determined by the ratio of the two power spectra, the spectral matching will perform uniformly over the whole frequency range irrespective of the shape of the signal power spectrum. This means that the spectrum will match just as well at frequencies with small power as well as large power. Assume for instance that the ratio $\frac{P(\omega)}{\hat{P}(\omega)} = 2$. This is independent of the power $P(\omega)$. Another kind of contribution in the form of e.g. a difference would instead give $|P(\omega) - \hat{P}(\omega)| = 0.5\ P(\omega)$ (since $P(\omega) = 2\hat{P}(\omega)$) and, hence, depend on the power $P(\omega)$.

Local property The fit of $\hat{P}(\omega)$ to $P(\omega)$ is expected to be better (on average) where $\hat{P}(\omega)$ is smaller than $P(\omega)$ than where it is larger. For instance for harmonic signals, this will imply that the peaks of the spectrum are better modelled than the area in between the peaks. The reason for this property is found in the "constraint" in equation 4.8. On average, the ratio in this equation must be 1 and therefore it will be larger in some areas and smaller in others. However, assume for instance that $P(\omega) = 10$. If $\hat{P}(\omega) = 15$, this would contribute $10/15 = 2/3$ to the integral whereas $\hat{P}(\omega) = 5$ would contribute $10/5 = 2$. The deviations from the average ratio of 1 is therefore $|1 - 2/3| = 1/3$ and $|1 - 2| = 1$, respectively, and hence, it is seen that the contribution to the error will be larger when $\hat{P}(\omega)$ is smaller than $P(\omega)$. Since the error is minimized, the signal power at such frequencies is fitted better.

Another very important result in [76] is that the model spectrum approximates the signal power spectrum closer and closer as the model order P increases and they become equal in the limit. The spectral matching results that have now been discussed, are clearly illustrated in figure 4.3.

The interpretation of the full multivariate autoregressive model in the frequency domain is more cumbersome than for the univariate model, but described in detail in [73] and [87]. The idea is basically the same, but with the main difference that also cross-spectra are estimated. This is important since it captures dependencies among the features and not just the temporal correlations of the individual features.

Parameter estimation

We now address the problem of estimating the parameters of the model. By taking the expectation value on each side of equation 4.2, the intercept term $\mathbf{v}$ is seen to capture the mean $\boldsymbol{\mu} = \mathrm{E}(\mathbf{x}_n)$. Explicitly,

$$\mathbf{v} = \left(\mathbf{I} - \sum_{p=1}^{P} \mathbf{A}_p \right) \mathrm{E}(\mathbf{x}_n)$$

where $\mathbf{I}$ is the identity matrix. Therefore, the estimated mean is simply subtracted initially from the time series $\mathbf{x}_n$ and the intercept term can be neglected in the following.

As mentioned earlier, least squares regression is used to estimate the parameters of the model. This corresponds to an assumption of gaussian distributed noise. Following the derivations in [87], the regression model can be formulated as

$$\mathbf{x}_n = \mathbf{B}\mathbf{y}_n + \mathbf{e}_n$$

where $\mathbf{e}_n$ is the error term with noise covariance $\mathbf{C}$ and

$$\mathbf{B} \equiv \begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \dots & \mathbf{A}_P \end{pmatrix}$$

and

$$\mathbf{y}_n \equiv \begin{pmatrix} \mathbf{x}_{n-1} \\ \mathbf{x}_{n-2} \\ \vdots \\ \mathbf{x}_{n-P} \end{pmatrix}$$

The least squares solution is found as the minimization of the 2-norm of the error terms and the parameter matrix $\mathbf{B}$ can be estimated as the solution to the normal equations

$$\mathbf{U}\hat{\mathbf{B}} = \mathbf{W} \tag{4.9}$$

where

$$\mathbf{U} = \sum_{i=n-(N-P-1)}^{n} \mathbf{y}_n \mathbf{y}_n^T$$

and

$$\mathbf{W} = \sum_{i=n-(N-P-1)}^{n} \mathbf{x}_n \mathbf{y}_n^T$$

where N is the frame size and P the model order. The matrices $\mathbf{U}$ and $\mathbf{W}$ are seen to be proportional to estimates of moment matrices. Since $\mathbf{U}$ is symmetric and positive semidefinite, the Cholesky decomposition has been used in (Paper G) to find $\hat{\mathbf{B}}$. The estimate of the noise covariance matrix $\mathbf{C}$ is found as

$$\begin{aligned} \hat{\mathbf{C}} &= \frac{1}{N-P} \sum_{i=n-(N-P-1)}^{n} \hat{\mathbf{e}}_n \hat{\mathbf{e}}_n^T \\ &= \frac{1}{N-P} \sum_{i=n-(N-P-1)}^{n} (\mathbf{x}_n - \hat{\mathbf{B}}\mathbf{y}_n)(\mathbf{x}_n - \hat{\mathbf{B}}\mathbf{y}_n)^T \end{aligned}$$

The order parameter P has so far been neglected. It is, however, clearly an important parameter since it determines how well the model fits the true signal. In traditional autoregressive modelling, P should ideally be found as the lowest number such that the model captures the essential structure or envelope of the spectrum. P is often chosen as the optimizer of an order selection criteria such as Akaike's Final Prediction Error or Schwarz's Bayesian Criterion [87]. Here, however, the purpose is to maximize the classification performance of the whole music genre classification system. Therefore, P has been found by optimizing the classification test error instead which has resulted in quite low P values (e.g. 3 for MAR and 5 for DAR features which are explained in the following). This clearly gives very crude representations of the power spectra and cross-spectra.

The parameters of the full multivariate autoregressive model have now been estimated. These parameters are used as the *Multivariate autoregressive* (MAR) features. The *Diagonal autoregressive* (DAR) features are created from the univariate autoregressive model instead which corresponds to diagonal coefficient matrices $\mathbf{A}_i$ and diagonal noise covariance $\mathbf{C}$. The parameter estimation is basically similar to the previously discussed, but without coupling between the individual feature dimensions. The DAR and MAR features have mainly been investigated in (Papers C and G).

MAR features

The MAR feature vectors $\mathbf{z}_n$ are created as

$$\mathbf{z}_n = \begin{pmatrix} \boldsymbol{\mu}_n \\ \text{vec}(\hat{\mathbf{B}}_n) \\ \text{vech}(\hat{\mathbf{C}}_n) \end{pmatrix}$$

where the "vec"-operator transforms a matrix into a column matrix by stacking the individual columns in the matrix. The "vech"-operator does the same, but only for the elements on and above the diagonal which is meaningful since $\hat{\mathbf{C}}_n$ is symmetric. As explained in the previous, the matrices $\hat{\mathbf{B}}_n = (\hat{\mathbf{A}}_{1n}\hat{\mathbf{A}}_{2n}\dots\hat{\mathbf{A}}_{Pn})$ and $\hat{\mathbf{C}}_n$ are the estimated model parameters and $\boldsymbol{\mu}_n$ is the estimate of the mean vector at time n. The dimensionality of the MAR feature is $(P+1/2)D^2+3D/2$ where P is the model order and D is the dimensionality of the short-time features $\mathbf{x}_n$. Assuming e.g. $D = 6$ and $P = 3$, this amounts to a 135-dimensional feature space. It is therefore necessary to use classifiers which can handle such high dimensionality or use a dimensionality reduction technique such as PCA, ICA (Paper F), PLS [103] or similar.

DAR features

The DAR feature vectors $\mathbf{z}_n$ are created similarly, but the autoregressive coefficient matrices $\mathbf{A}_i$ and the noise covariance matrix $\mathbf{C}$ are now diagonal. This leads to

$$\mathbf{z}_n = \begin{pmatrix} \boldsymbol{\mu}_n \\ \text{diag}(\hat{\mathbf{A}}_{1n}) \\ \text{diag}(\hat{\mathbf{A}}_{2n}) \\ \vdots \\ \text{diag}(\hat{\mathbf{A}}_{Pn}) \\ \text{diag}(\hat{\mathbf{C}}_n) \end{pmatrix}$$

at time n and the "diag"-operator forms a column vector from the diagonal of a matrix. Note that the diagonal matrices are not actually formed since the elements of the diagonals are found directly as the solution of D univariate models. The dimensionality of the DAR features is $(2 + P)D$. For e.g. $D = 6$ and $P = 3$, this gives a 30-dimensional feature vector.

Complexity considerations

METHOD	MULTIPLICATIONS & ADDITIONS
MeanVar	$4DN$
MeanCov	$(D+3)DN$
FC	$(4\log_2(N)+3)\,DN$
DAR	$\frac{D}{3}(P+1)^3 + ((P+6)(P+1)+3)\,DN$
MAR	$\frac{1}{3}(PD+1)^3 +$ $(P+4+\frac{2}{D})(PD+1)+(D+2)\;\;DN$

Table 4.1: Computational complexity of 5 features from temporal feature integration. The numbers in the column "Multiplications & Additions" are the estimates of the number of multiplications and additions which are necessary in the calculation of the features when standard methods are used. It assumes that the short-time features with dimension D are given. N is the temporal feature integration frame size and P is the autoregressive model order.

The calculation of the MAR and DAR features have now been explained, but it is also interesting to know how computationally costly these features are. Table 4.1 compares the computational complexity of the five features DAR, MAR, MeanVar, MeanCov and FC (explained in section 4.4) which are considered as the main features in temporal feature integration. The column "Multiplications & Additions" shows an estimate of the total number of multiplications/additions necessary for temporal feature integration over a frame with N short-time features of dimension D with different methods. For the DAR and MAR models, the model order P is also included. In (Paper G), the parameters N and P were optimized with respect to the classification test error for the five different features. Using these values with the expressions in table 4.1, results in explicit estimates of the calculations necessary. Normalizing with the number of calculations for the MeanVar feature, the MeanCov, FC, DAR and MAR features required approximately 3, 16, 10 and 32 calculations. In other words, the MAR feature takes approximately 32 times as long time to calculate as the MeanVar feature whereas the FC feature only takes 10 times as long. In many situations, these differences are not very significant. However, for larger values of D and P, these ratios change. As seen from the table, the DAR feature grows like $O(P^2)$ (in units of DN) for small P when the term $\frac{D}{3}(P+1)^3$ can be neglected. The MAR feature grows as $O(DP^2)$ (in units of DN) for smaller D and P, but the

extra term $(PD+1)^3$ dominates for larger values. The ratio with the MeanVar for e.g. $D = 12$ and $P = 12$ would be roughly 600 for the MAR and 60 for the DAR.

4.3 Dynamic Principal Component Analysis

In (Paper B), we experimented with so-called *Dynamic Principal Component Analysis* (DPCA) for temporal feature integration. This is a method that has been used in chemical process monitoring [66], but to our knowledge not in music genre classification.

The idea in DPCA is to first perform a time stacking of the original signal which were short-time feature vectors in our case. This results in a high dimensional feature space. Principal component analysis (PCA) is then used to project the stacked features into a new feature space of (much) lower dimensionality. In mathematical terms, the time stacked feature vector $\mathbf{y}_n$ can be written as

$$\mathbf{y}_n = \begin{bmatrix} \mathbf{x}_{n-(N-1)} \\ \vdots \\ \mathbf{x}_n \end{bmatrix}$$

where N denotes the framesize. The final integrated feature vector $\mathbf{z}_n$ becomes

$$\mathbf{z}_n = \tilde{\mathbf{U}}^T \left(\mathbf{y}_n - \hat{\boldsymbol{\mu}}\right)$$

where the rows of $\tilde{\mathbf{U}}$ are the estimated k first eigenvectors of the covariance matrix of $\mathbf{y}_n$. The eigenvectors belong to the k largest eigenvalues and represent the directions with the greatest variance. $\hat{\boldsymbol{\mu}}$ denotes the estimate of the mean of $\mathbf{y}_n$. A good and more elaborate description of the PCA is given in [8].

The idea of using DPCA for music genre classification is to capture the strongest correlations between both the individual features and at different times. This could for example be the correlations between the 5th MFCC coefficient at time n and the 1st LPC coefficient at time $n - 10$.

It is seen that $\mathbf{z}_n$ can easily be written in the form of equation 4.1 if the parameters $\tilde{\mathbf{U}}$ and $\hat{\boldsymbol{\mu}}$ were known. However, in (Paper B), the whole training set was

used to estimate these parameters. Hence, the DPCA method of feature integration is a batch process in contrast to the other previously discussed methods which could be performed online.

Another comment regards the actual computation of the parameters and especially $\tilde{\mathbf{U}}$. In (Paper B), we had approximately 100 short-time features which were extracted with a hopsize of 10 ms. Integrating over e.g. 1 s would make the dimension of $\mathbf{y}_n$ 10,000. Having in the order of 100,000 samples, it then becomes computationally demanding in both space and time (in our case impossible) to create the covariance matrix and find its eigenvalues by traditional means. Our solution was to use a computationally cheap version of the PCA as described more carefully in appendix A.

4.4 Frequency Coefficients

As seen previously in this chapter, the time series of short-time features contain temporal information. In [82], this information is tried captured with the power spectrum of each short-time feature individually and MFCCs are used as the short-time representation. The *Frequency Coefficient* (FC)[1] features are then found by summation of power in four specific frequency bands. The first frequency band is the DC band. The second band is in the range $1-2$ Hz which is thought to capture the musical rhythm. The third band in $3-15$ Hz is on the order of speech syllabic rates and the fourth band in $20-43$ Hz should capture the perceptual roughness.

Note that these FC features are somehow related to the DAR features which were discussed in section 4.2. The DAR features implicitly model the envelope of the power spectrum whereas the FC features find the content in specific frequency bands. However, they both neglect dependencies among the short-time feature dimensions (e.g. between MFCC-2 and MFCC-5) in contrast to e.g. the MAR feature.

4.5 Low Short-Time Energy Ratio

In [74], the *Low Short-Time Energy Ratio* (LSTER) features were used for segmentation and classification of speech, music, environmental sound and silence.

[1] This name is used in the our work in (Papers C and G) for ease of reference, although the features are actually not given a name in [82]

They were also used in e.g. [98] for speech/music discrimination since the LSTER feature is higher for speech than music and in [108] for music genre classification. This feature is based specifically on the short-time feature STE as described in chapter 3 and is calculated as

$$LSTER_n = \frac{1}{2N} \sum_{i=n-N}^{n} \text{sgn}(0.5avSTE - STE_i) + 1$$

where $avSTE$ is the average of the N STE values and sgn denotes the sign-function. Hence, this feature integration method simply counts the number of short-time frames where the STE is below the average value.

4.6 High Zero-Crossing Rate Ratio

The *High Zero-Crossing Rate Ratio* (HZCRR) feature is described in [74] and is build on the ZCR feature as described in chapter 3. It is a count of the number of frames with ZCR value above 1.5 times the average ZCR value. This can be calculated as

$$HZCRR_n = \frac{1}{2N} \sum_{i=n-N}^{n} \text{sgn}(ZCR_i - 1.5avZCR) + 1$$

where $avZCR$ is the average of the ZCR feature over the frame with size N. The HZCRR feature tends to be larger for music than for speech.

4.7 Beat Histogram

The *Beat Histogram* (BH) features were originally proposed in [110] to be used in music genre classification. The BH algorithm is intended to summarize the beats over a whole song in a single beat histogram. From this histogram it should be possible to extract features such as ratio between main beat and subbeats from the largest and second largest peaks, the beat strength as the sum of peaks as well as the values of the main and subbeats in bpm (beats per minute).

The BH algorithm starts with a discrete wavelet transform which can be viewed as a transform to the frequency domain with octave spacing between the frequency bands and fixed ratio between each filters center frequency and bandwidth [111]. This is performed on frames of size 3s with 1.5s hopsize. Each band is then full-wave rectified, low pass filtered and downsampled and the mean is removed. The resulting signals are now estimates of the time domain envelope of each band. The bands are summed and the enhanced autocorrelation of the resulting signal is found as described in [107]. The enhanced autocorrelation function try to remove the (artificial) integer multiple peaks that occur naturally in the autocorrelation function. Finally, the peaks are estimated and the values of the first three peaks are added to the final beat histogram. This procedure is repeated for each frame in the song.

From the beat histogram, the BH features are extracted. In our implementation, we simply summarized the beat-content in 6 bands in the beat histogram and used these values as the BH feature vector.

4.8 Beat Spectrum

The *Beat Spectrum* (BS) features are another approach to beat estimation from the whole song which were proposed in [32].

The idea is to start out with a short-time feature representation. Then create the distance matrix with elements which are the distance between the short-time features of each frame. The distance measure is here the cosine measure, ie. the angle between two feature vectors. To create the beat spectrum, the elements on the diagonals of the distance matrix are summed. Hence, the beat spectrum can be mathematically formulated as

$$B(l) = \sum_{i \in I} D(\mathbf{x}_i, \mathbf{x}_{i+l}) = \sum_{i \in I} \frac{\mathbf{x}_i}{|\mathbf{x}_i|} \cdot \frac{\mathbf{x}_{i+l}}{|\mathbf{x}_{i+l}|}$$

where D is the distance measure and I is the index set such that $i+l$ lies within the size of the distance matrix. The peaks of this beat spectrum corresponds to beats in the music. In our implementation in relation to (Paper C), a Fourier transformation has been used to estimate the periodicity and the content in 6 bands was used as the BS features.

Chapter 5

Classifiers and Postprocessing

Given a music representation in the form of feature vectors, it is important to be able to find the patterns in feature space that belong to the different music genres. In music genre classification systems, this is the task of a *classifier*. The first four sections each describe a traditional classifier which has been used in the current dissertation. In section 5.5, a novel *Co-occurrence model* (Paper D) is proposed for music genre classification. In the last section, *Post-processing* techniques are considered with special focus on methods to combine a time-sequence of classifier outputs from a song into a single genre decision.

The music features that have been examined in chapter 3 share some common traits. Notably, they all transform the raw audio signal into a sequence of (multivariate) feature vectors with an implicit "clustering assumption". This assumption means that two feature vectors which are close (in some "simple" metric) in this feature space should represent much of the same musical content. This assumption is vital to all areas of music information retrieval. For instance in music recommendation systems [15], distance measures are used to decide which songs sound "similar" and use that to make recommendations. That is only meaningful with the previous assumption.

In music genre classification systems, a *classifier* uses the "clustering assumption" to transform the feature vectors into estimates of their corresponding music genre. The classifiers which have been investigated in this project are all

statistical classifiers. These kinds of classifier models use a so-called *training set* of genre-labelled songs to statistically infer the parameters of the model. After this training procedure, the performance of the classifier model is found by predicting genre-labels on a *test set* of songs. The performance of the classifier and the music genre classification system as a whole, is often measured by comparing the known labels of the test set songs against these predicted labels.

In some music genre classification systems such as [77] and [106], the classifier use the whole time-series of feature vectors in a song to returns an estimate of the genre for the new song. Our proposed co-occurrence model (Paper D) which is described in section 5.5 is also capable of using the whole time-series of feature vectors to reach a genre estimate for the song. However, in many systems the classifier predicts a genre, or the probability of a genre, for each feature vector $\mathbf{z}_n$ in the song. This is the case for the first four classifiers which are discussed in the first four sections of this chapter. In the *postprocessing* step, these predictions from every feature vector in the song are combined into a single genre label for the song.

In our research, we have made several assumptions about the nature of the problem as discussed in section 2.3. With these assumptions, the combined classifier and postprocessing parts can be formalized as

$$\hat{C} = g(\mathbf{z}_1, \ldots, \mathbf{z}_N) \tag{5.1}$$

where $\hat{C}$ is the estimated genre for a song that is represented by the sequence of N feature vectors $\mathbf{z}_1$ to $\mathbf{z}_N$. The classifier and postprocessing are then contained in the function g.

It should be mentioned that there exists an abundance of different classification and general pattern recognition methods in the machine learning and statistics literature. Good overviews can be found in e.g. [27], [8] and [75]. For instance, the above formulation with a training and test set lies within the realm of *supervised learning* where the genre taxonomy is restricted in advance. The *unsupervised learning* approach is not limited by such restrictions in taxonomy, but looks for patterns in feature space with special emphasis on the similarity measure [97]. This has the advantage that also new, emerging genres might be discovered, it avoids the problems of defining genre as we discussed in chapter 2 and it avoids the problems with getting reliable labels for the data. However, it relies strongly on the similarity measure. In [102], Hidden Markov Models [92] are used for unsupervised music genre classification and [95] used Self-Organizing Maps (SOMs) and a Growing Hierchical SOM variant. In (Paper F), we used Independent Component Analysis (ICA) to cluster music feature

space. Common similarity measures are the simple Euclidian or cosine distance [33] or the Kullbach-Leibler divergence [77]. Several well-known (hierarchical) clustering methods are discussed in e.g. [27] of which the K-Means method is probably the most common.

There also exist numerous supervised classification schemes. An important non-parametric method is the K-nearest neighbor (KNN) method which simply assigns a class to a new feature vector by voting among the K nearest neighbors. It has been used in music genre classification in e.g. [110] and [67]. Another important class of classifiers are the non-probabilistic *Support Vector Machines* (SVMs) [20] which have been used with great success in the recent MIREX 2005 contests in music genre classification [53]. SVMs were used in more than half of the contributions. Other examples are [117] and [83]. The Artificial Neural Network classifiers could, like the SVMs, be put in the class of discriminative non-probabilistic classifiers although they can be modified to model probability.

In the current dissertation mostly probabilistic classifiers have been considered. These can be described in a unified Graphical Model framework [57] [96]. In relation to classification, they can be split into *generative* and *discriminative* models. The generative models model $p(\mathbf{z}|C)$ which is the probability of a feature vector $\mathbf{z}$ given the class label C. Predicting the class label of a new feature vector $\tilde{\mathbf{z}}$ then requires the use of Bayes' rule to estimate $P(C|\tilde{\mathbf{z}})$. A disadvantage of the generative models is that they model each class individually without considering the other classes and hence the class-overlap may be larger than necessary. An advantage is that they directly give an estimate of the reliability of a genre prediction. For instance, if $p(\tilde{\mathbf{z}}|C = c)$ is very low for each c, this might be considered an outlier. Notable examples of generative classification models are the Gaussian Classifier and Gaussian Mixture Model which are described in the following sections as well as the (Hidden) Markov Model.

The discriminative models, in contrast to the generative ones, model the desired quantity $p(C|\mathbf{z})$ directly and they are therefore not (directly) capable of detecting outliers. However, they are often more powerful since they generally require a smaller number of parameters than the generative models for similarly flexible decision boundary and hence are less prone to overfitting. Examples of discriminative models include regression models such as the Linear Regression and Generalized Linear Model which are examined in the following sections. Another example is Gaussian Process classifiers which are closely related to SVMs.

It should also be mentioned that two different regimes exist in statistics. These are the *frequentist* and *Bayesian* approaches [75] [11]. In short, it can be said that the frequentist approach to the concept of probability solely builds on obser-

vations of events from well-defined random experiments. The relative frequency of occurrence of an event is then a measure of the probability of that event. The Bayesian approach, in contrast, is willing to assign probabilities according to the belief in a proposition. Bayesian inference then uses Bayes' theorem to update the belief in the light of new evidence.

The formalization in equation 5.1 is very restricted since it simply assigns a single genre label to a given song. All the previously mentioned supervised probabilistic classifiers instead give (using Bayes' theorem for the generative models) the probability $P(C|\tilde{\mathbf{z}})$ of a genre given the feature vector. The post-processing methods are capable of combining these probabilities into a single decision for the whole song. However, this is not necessarily the best method in a music genre classification system. As discussed in chapter 2, humans do not agree on genre and therefore it seems more natural to use the quantity $P(C = c|\tilde{s})$ directly as a measure of the degree to which the song with index $\tilde{s}$ belongs to genre c instead of using a single genre decision. Any probabilistic classifier could estimate this quantity if either the feature vector $\tilde{\mathbf{z}}$ represents the whole song or the classifier includes temporal integration such as a Hidden Markov Model or the proposed co-occurrence models in section 5.5. Although this might seem very natural, it should be noted that this would also require the labels of the training set to take the form of a distribution $P(C|\tilde{s})$.

Another restriction in equation 5.1 lies in the output of a genre taxonomy without hierarchical structure although such structure is natural in most real-world music genre classification systems. For instance [12] uses four different levels in the genre taxonomy and makes a supervised classification on each level. As mentioned before, there also exist many unsupervised hierarchical methods.

The classifiers which have been used in the current dissertation project are described in the following. First, two generative probabilistic model are described; the Gaussian Classifier (GC) and the Gaussian Mixture Model (GMM). Then the discriminative Linear regression classifier and Generalized Linear Model are discussed. Afterwards, the generative co-occurrence models which we proposed in (Paper D) are investigated and explained.

5.1 Gaussian Classifier

The *Gaussian classifier* uses the gaussian probability density function as a model for the distribution of feature vectors in each genre. The probabilistic model has been used in e.g. [70] for music genre classification due to its simplicity. The probability density function for a feature vector $\mathbf{z}_n$ in the genre with index

c is

$$p(\mathbf{z}_n|C=c) = \frac{1}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}_c|}} \exp\left(-\frac{1}{2}(\mathbf{z}_n - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c^{-1}(\mathbf{z}_n - \boldsymbol{\mu}_c)\right) \tag{5.2}$$

where $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ are the mean and covariance matrix, respectively, that belong to genre c and d is the dimensionality of the feature space. The strong assumption here is independence between the individual observations in the data set $(c_{s(j)}, \mathbf{z}_j)$ where j runs over all feature vectors in the data set and $c_{s(j)}$ is the associated genre label. The function $s(j)$ gives the song index for a feature vector with index j. Although there is much time structure in music and many features use overlapping windows, this assumption is often used (quite successfully) in practice. The assumption is used to formulate the log-likelihood as

$$L = \log p(c_{s(1)}, \ldots, c_{s(M)}, \mathbf{z}_1, \ldots, \mathbf{z}_M) = \log \prod_{j=1}^{M} p(c_{s(j)}, \mathbf{z}_j) \tag{5.3}$$

$$= \sum_{j=1}^{M} \log P(C = c_{s(j)}) + \sum_{j=1}^{M} \log p(\mathbf{z}_j|C = c_{s(j)}) \tag{5.4}$$

where M is the total number of feature vectors in the data set. The traditional maximum likelihood principle states that the model parameters can be estimated by maximizing L. This gives the well-known estimates of the mean and variance of each class. The estimate of $P(C)$ simply becomes the normalized count of occurrences in each class. After the inference of the parameters $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ of the model as well as $P(C)$, predictions of $P(C|\mathbf{z}_n)$ can be made for new feature vectors in the test set. According to Bayes' rule, the predicted probability for each genre c then becomes

$$P(C = c|\mathbf{z}_n) = \frac{P(C = c)p(\mathbf{z}_n|C = c)}{\sum_{j=1}^{N_c} P(C = j)p(\mathbf{z}_n|C = j)} \tag{5.5}$$

where N_c denotes the number of genres. This quantity is the output of our gaussian classifier model for each feature vector $\mathbf{z}_n$ to be used in the postprocessing part.

The gaussian classifier can often be used with good results when the dimensionality of the feature space is reasonably small. However, bad results are obtained

in high-dimensional spaces since the estimation of the covariance matrix becomes very unreliable. "Small" and "high" dimensional spaces are not fixed quantities, but depend on the size of the training set and were of the order of 10-30 and 80-100, respectively, in the author' experiments. This problem is a classical example of the "Curse of dimensionality" in machine learning. It should be mentioned that several solutions to this problem have been proposed. One common solution is to put restrictions on the covariance matrix such that it e.g. only has elements on the diagonal.

5.2 Gaussian Mixture Model

The *Gaussian Mixture Model* (GMM) classifier is closely related to the previously described Gaussian classifier. In fact, the Gaussian classifier is a special case of the GMM classifier with only one mixture component. It has been used in music genre classification in e.g. [12]. Instead of modelling the feature vectors in each genre with a single gaussian distribution, the GMM uses a mixture of gaussians which allows for more complex decision boundaries. This can be formalized as

$$p(\mathbf{z}_n|K=k) = \frac{1}{\sqrt{(2\pi)^d|\mathbf{\Sigma}_k|}} \exp\left(-\frac{1}{2}(\mathbf{z}_n - \boldsymbol{\mu}_k)^T \mathbf{\Sigma}_k^{-1}(\mathbf{z}_n - \boldsymbol{\mu}_k)\right) \tag{5.6}$$

where k denotes the mixture index and

$$p(\mathbf{z}_n|C=c) = \sum_{k=1}^{K} P(\mathbf{z}_n|K=k)P(K=k|C=c) \tag{5.7}$$

where K is the total number of mixture components. As for the gaussian classifier, the mixture parameters $\boldsymbol{\mu}_k$ and $\mathbf{\Sigma}_k$ as well as $P(C)$ and $P(K|C)$ are estimated with the maximum likelihood method and the strong assumption of independence between observations $(c_{s(j)}, \mathbf{z}_j)$. Unfortunately, the maxima of the log-likelihood cannot be found analytically. We have used the common solution, which is to use the *EM-algorithm* [23] [8] which iteratively searches for a maximum. After the training procedure, the predicted genre probability for a new feature vector is found as in equation 5.5. The Netlab Matlab package [54] was used for the experiments with the GMM. Note that the concerns about high-dimensional feature space are even more relevant here than for the Gaussian classifier.

5.3 Linear Regression classifier

The *Linear Regression* classifier is one of the most simple and common classifiers, but has been used under a variety of names and disguises. In (Paper B), we used this method under the name Linear Neural Network, in (Paper C) under the name of Linear Model and, to add to the confusion, I have now decided on Linear Regression classifier as it is common in the statistics literature. Another common name for it is the Perceptron model [8]. It is described in any basic statistics textbook. This linear model can be expressed as

$$\mathbf{v} = \mathbf{W}\mathbf{z} + \mathbf{b} + \mathbf{e}$$

where $\mathbf{v}$ is the regression output variable, $\mathbf{z}$ is the input variable, $\mathbf{W}$ and $\mathbf{b}$ are (deterministic) parameters and $\mathbf{e}$ is an error term. In our situation, $\mathbf{z}$ is the feature vector and $\mathbf{v}$ is a representation of the its corresponding genre label. The output variable $\mathbf{v}$ is inherently continuous in linear regression. Hence, to use linear regression as a classifier, $\mathbf{v}$ is put on so-called "1-of-c form". For instance, to signify that a feature vector belongs to genre 2 out of 5, $\mathbf{v}$ would be

$$\mathbf{v} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.8}$$

The parameters $\mathbf{W}$ and $\mathbf{b}$ can be estimated by minimizing the error $|\mathbf{e}|$ for the given set of input-output pairs $(\mathbf{v}_i, \mathbf{z}_i)_{i=1,..,M}$ in the training set. Using the method of least squares, the problem becomes to minimize

$$E(\mathbf{W}, \mathbf{b}) = \sum_{i=1}^{M} |\mathbf{e}|^2 = \sum_{i=1}^{M} (\mathbf{v}_i - \mathbf{W}\mathbf{z}_i - \mathbf{b})^2$$

with respect to $\mathbf{W}$ and $\mathbf{b}$ where M is the total number of samples in the training set. Note that it is very common to add an extra regularization term to the expression as in equation 5.14. Using the trick of adding an extra dimension to $\mathbf{z}_i$ with constant value of 1 (or indeed anything different from zero), the bias

term $\mathbf{b}$ can be contained in $\mathbf{W}$. The error function E then takes the form

$$E(\hat{\mathbf{W}}) = \sum_{i=1}^{M} (\mathbf{v}_i - \hat{\mathbf{W}}\hat{\mathbf{z}}_i)^2 = \sum_{i=1}^{M} \sum_{j=1}^{N_c} (\mathbf{v}_i^{(j)} - \mathbf{w}_j \hat{\mathbf{z}}_i)^2 \tag{5.9}$$

where $\mathbf{w}_j$ denotes the j^{th} row of $\hat{\mathbf{W}}$ and

$$\hat{\mathbf{z}}_i = \begin{bmatrix} \mathbf{z}_i \\ 1 \end{bmatrix} \tag{5.10}$$

This quadratic problem is (quite easily) solved analytically to find that E has minimum in

$$\hat{\mathbf{W}}^* = (\hat{\mathbf{Z}}\hat{\mathbf{Z}}^T)^{-1}\hat{\mathbf{Z}}\mathbf{V}^T$$

where

$$\hat{\mathbf{Z}} = \begin{bmatrix} \hat{\mathbf{z}}_1 & \hat{\mathbf{z}}_2 & \cdots & \hat{\mathbf{z}}_M \end{bmatrix} \in \mathbb{R}^{d \cdot M}$$

and

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_M \end{bmatrix} \in \mathbb{R}^{c \cdot M}$$

Given a new feature vector $\mathbf{z}_n$ from the test set, an estimate of the genre can now be given by first forming $\hat{\mathbf{z}}_n$ from equation 5.10. Next, $\mathbf{v}_n$ is found from $\mathbf{v}_n = \hat{\mathbf{W}}^* \, \hat{\mathbf{z}}_n$. Since the classifier was trained with the 1-of-c coding, the index of the largest element in $\mathbf{v}_n$ is used as the estimated index of the genre.

5.4 Generalized Linear Model

In (Paper G), we used an extended version of the previously discussed linear regression classifier which is called a *Generalized Linear Model*. This is also a model which has been used under many different names. For instance in the

artificial neural networks community [8], this model would be called a single-layer neural network with softmax activation function and cross-entropy error function. Although it has a linear discriminant function like the linear regression classifier, the generalized linear model has several advantages. For example, the outputs are now more realistic estimates of the posterior $P(C|\mathbf{z}_n)$ since they are forced to lie between 0 and 1. The basic idea is that $P(C|\mathbf{z}_n)$ is assumed to have the form

$$P(C = c|\hat{\mathbf{z}}) = \frac{\exp(\mathbf{w}_c\hat{\mathbf{z}})}{\sum_{j=1}^{N_c}\exp(\mathbf{w}_j\hat{\mathbf{z}})} \tag{5.11}$$

where $\hat{\mathbf{z}}$ is here the extended feature vector defined in equation 5.10. This assumption holds for a wide variety of distributions of $P(\hat{\mathbf{z}}|C)$. In fact, it has been shown in e.g. [59] that for any function $P(\hat{\mathbf{z}}|C)$ in the so-called exponential family of distributions, the relation 5.11 will hold. This family of distributions contains e.g. models where $P(\hat{\mathbf{z}}|C)$ is gaussian distributed with same covariance matrices, but different means for the different classes. Using the strong assumption of independence in equation 5.3, the conditional data log-likelihood can be formulated as

$$E = \sum_{i=1}^{M}\sum_{j=1}^{N_c}\mathbf{v}_i^{(j)}\log\frac{\exp(\mathbf{w}_j\hat{\mathbf{z}}_i)}{\sum_{j=1}^{N_c}\exp(\mathbf{w}_j\hat{\mathbf{z}}_i)} \tag{5.12}$$

$$= \sum_{i=1}^{M}\sum_{j=1}^{N_c}-\mathbf{v}_i^{(j)}\log(1+\sum_{j\neq i}\exp(\mathbf{w}_j\hat{\mathbf{z}}_i)) \tag{5.13}$$

where $\mathbf{v}_i$ denotes the label of sample i with the 1-of-c coding as in equation 5.8. In (Paper C), we have additionally added a regularization term on the weights $\mathbf{w}_j$. In the probabilistic setting, this corresponds to a gaussian prior on the weights and results in the final conditional log-likelihood

$$E = \sum_{i=1}^{M}\sum_{j=1}^{N_c}-\mathbf{v}_i^{(j)}\log(1+\sum_{j\neq i}\exp(\mathbf{w}_j\hat{\mathbf{z}}_i)) + \alpha|\mathbf{w}_j|^2 \tag{5.14}$$

The variance of the prior on the weights, $1/\alpha$, was simply found by cross-validation in (Paper G). Unfortunately, in contrast with the linear regression classifier, the minimization of 5.14 for the generalized linear classifier cannot be

made analytically. The details of the minimization has been explained in [8]. Given a new feature vector $\mathbf{z}_n$, equation 5.11 is used to estimate the posterior which is used by the postprocessing methods to reach a final song genre label. The Netlab Matlab package [54] was used in the experiments.

5.5 Co-occurrence models

In the previous classifiers, $P(C|\mathbf{z}_n)$ is estimated for each individual feature vector $\mathbf{z}_n$ in a song. These values are then combined with the more or less heuristic postprocessing methods, such as majority voting, to give the song a genre label. However, it would often be useful to have the probability of the genre conditioned on the whole song, $P(C|s)$, explicitly. In (Paper D), we proposed two so-called *co-occurrence models* that include the song directly in the model and therefore can be used to find an estimate of $P(C|s)$. The two novel models are named the *Aspect Gaussian Classifier* and the *Aspect Gaussian Mixture Model* which are extensions of the previously described gaussian classifier and gaussian mixture model, respectively. Aspect models have also been proposed in [61] and [9] for the Bernoulli and Hidden Markov Models, respectively.

Figure 5.1: The left part of the figure compares the Probabilistic Graphical Models of the Gaussian Classifier (GC) and the proposed Aspect Gaussian Classifier (AGC). Consult e.g. [57] or [49] for an introduction to Graphical Models. C denotes the genre class random variable, S denotes the song index and X is the feature vector. Round circles represent continuous variables, while squares represent discrete variables. The right part of the figure compares the graphical models of the Gaussian Mixture Model (GMM) and the proposed Aspect Gaussian Mixture Model (AGMM).

The idea in the co-occurrence model is to represent a song by a set of independent co-occurrences $(s(j), \mathbf{z}_j)$ between the song with index $s(j)$ (as a function of the feature vector index j) and its constituent feature vectors $\mathbf{z}_j$. Let $c_{s(j)}$ be the genre index of the song with index $s(j)$. The previous assumption of independence between the observations $(c_{s(j)}, \mathbf{z}_j)$, then becomes independence between observations $(s(j), c_{s(j)}, \mathbf{z}_j)$. The graphical models of the two proposed

co-occurrence models are illustrated along with their traditional counterparts in figure 5.1. In similarity with equation 5.3, the co-occurrence log-likelihood becomes

$$L = \sum_{j=1}^{M} \log P(S = s(j)|C = c_{s(j)})\ P(C = c_{s(j)})\ p(\mathbf{z}_j|C = c_{s(j)}) \tag{5.15}$$

where M is the total number of feature vectors $\mathbf{z}_j$ in the training set and $s(j)$ is the index of the song which contains feature vector $\mathbf{z}_j$. The log-likelihood L is now maximized and it is seen that $P(S|C)$ simply becomes an independent additive term. Hence, the estimates of the parameters in $p(\mathbf{z}_j|C = c_{s(j)})$ and $P(C = c_{s(j)})$ become the same as without the extra term. In addition, the estimate of $P(S|C)$ simply becomes

$$\hat{P}(S = s|C = c) = \begin{cases} \frac{N_s}{N_c} & \text{if song } s \text{ has genre label } c \\ 0 & \text{else} \end{cases}$$

where N_s and N_c are the number of feature vectors (j's) in song s and genre c, respectively. To predict the genre of a new song, it is necessary to add an extra index $\tilde{s}$ to the range of S. The desired quantity is now $P(S = \tilde{s}|C = c)$ which can be seen as an extra row (or column) vector added to the matrix $P(S|C)$. Adding this extra vector is called *Folding-in* and the *Folding-in method*, as described in [42], is used to find this vector. The idea is to consider $\tilde{s}$ as a latent variable and this results in the log-likelihood

$$L(\tilde{s}) = \sum_{j=1}^{N_{\tilde{s}}} \log \left(\sum_{c=1}^{N_c} P(S = \tilde{s}|C = c)\hat{P}(C = c)\hat{p}(\mathbf{z}_j|C = c) \right) \tag{5.16}$$

where $N_{\tilde{s}}$ is the number of feature vectors in the new song $\tilde{s}$ and N_c is the total number of genres. Except for $P(S = \tilde{s}|C)$, all model parameters are assumed to be known (from the training phase) and kept constant. As in [36], the EM algorithm is used to estimate $P(S = \tilde{s}|C)$ with the two iterative update equations

$$P^{(t)}(c|\mathbf{z}_j, \tilde{s}) = \frac{P^{(t)}(\tilde{s}|c)\ \hat{P}(c)\ \hat{p}(\mathbf{z}_j|c)}{\sum_{c=1}^{N_c} P^{(t)}(\tilde{s}|c)\ \hat{P}(c)\ \hat{p}(\mathbf{z}_j|c)} \tag{5.17}$$

$$P^{(t+1)}(\tilde{s}|c) = \frac{\sum_{j=1}^{N_{\tilde{s}}}\ P^{(t)}(c|\mathbf{z}_j, \tilde{s})}{C_c + \sum_{j=1}^{N_{\tilde{s}}}\ P^{(t)}(c|\mathbf{z}_j, \tilde{s})} \tag{5.18}$$

for each genre c and C_c is the total number of feature vectors in genre c (in the training set). The stochastic variables S and C have been left out to simplify the notation and (t) denotes the iteration index. As starting condition, it can be assumed that $P^{(0)}(c|\tilde{s})$ is uniformly distributed from which $P^{(0)}(\tilde{s}|c)$ can be found with Bayes' rule. Having estimated $P(S = \tilde{s}|C = c)$ for all c, Bayes' rule is simply used with the previously estimated parameters $\hat{P}(C)$ to find $P(C = c|S = \tilde{s})$ for the new song $\tilde{s}$ as

$$P(C = c|S = \tilde{s}) = \frac{P(S = \tilde{s}|C = c)\ \hat{P}(C = c)}{\sum_{c=1}^{N_c} P(\tilde{s}|C = c)\ \hat{P}(C = c)}$$

In the proposed *Aspect Gaussian Classifier* model, $p(\mathbf{z}_j|C = c_{s(j)})$ in equation 5.15 is an ordinary gaussian distribution and the parameter estimation for $p(\mathbf{z}|C)$ and $P(C)$ proceeds as explained in section 5.1 for the traditional gaussian classifier. Afterwards, the Folding-in method as explained above is used to infer the genre of a new song.

Regarding the *Aspect Gaussian Mixture Model*, $p(\mathbf{z}_j|C = c_{s(j)})$ is a mixture of gaussians as described in equation 5.7 and the parameter estimation in the training phase is again the same as for the ordinary GMM model. Again, the Folding-in method is applied for testing.

It is seen that the additional parameter $P(S|C)$ is not used in the testing phase in any of the models. Hence, the only practical difference between the co-occurrence models and the ordinary models lies in the testing phase with the use of the Folding-in method. As explained in the end of section 5.6, this method has a close relation to the postprocessing method called the sum-rule.

5.6 Postprocessing

The last part of a music genre classification system is *postprocessing* [8] [27] and can consist of many different things. The emphasis here is on postprocessing in the form of *information fusion* between the classifier outputs $P(C|\mathbf{z}_n)$ for each feature vector $\mathbf{z}_n$ in a song. This particular kind of information fusion has been discussed in e.g. [24] and [63]. It is used to reach a final genre label prediction for the song. The author has essentially experimented with two different methods; *Majority Voting* and the *Sum Rule*. The Sum Rule was found to perform slightly better experimentally than Majority Voting. This is in agreement with the findings in e.g. [63].

5.6.1 Majority Voting

The well-known Majority Voting method takes a vote Δ_n for each feature vector $\mathbf{z}_n$ in the song by

$$\Delta_n = \arg\max_c P(C = c|\mathbf{z}_n)$$

and the genre with the most votes is assigned to the song.

5.6.2 Sum Rule

The Sum Rule instead makes a prediction of the genre as

$$\hat{C} = \arg\max_c \sum_{j=1}^{N_s} P(C = c|\mathbf{z}_j) \tag{5.19}$$

where $\hat{C}$ is the predicted genre label for the new song and N_s is the number of feature vectors in the song. In contrast to Majority Voting, the Sum Rule method can be seen as a "soft" assignment since it sums the quantities $P(C|\mathbf{z}_j)$ instead of using "hard" 0/1-decisions. This implies that parts of the song with large uncertainty, where $P(C|\mathbf{z}_j)$ is nearly uniform, will have less influence than parts where the classifier is very certain about the genre. Majority Voting will give just as much influence to these uncertain parts.

In section 5.5, co-occurrence models were proposed for classification which used a particular Folding-in method in the test phase. An advantage of the co-occurrence models is that they give the probability $P(C = c|S = \tilde{s})$ of genre c given the song with index $\tilde{s}$ directly and, hence, postprocessing is not necessary. In fact, analyzing the Folding-in method reveals a relation to the Sum-rule postprocessing method. Assume first that the genre distribution $P(C)$ is uniform which implies that the starting condition in equation 5.17, $P^{(0)}(\tilde{s}|C = c)$, is the same for all genres c. This is a reasonable assumption. The right side of equation 5.17 then reduces to $P(C = c|\mathbf{z}_j)$ and the sums on the right side of equation 5.18 are seen to be similar to the sum in the Sum-rule method (equation 5.19). Hence, with the mentioned assumptions, it is seen that *the decisions from the Sum-rule method are exactly the same as the decisions from the first iteration of the Folding-in method*. The Sum-rule method may therefore be seen as an approximation to the full probabilistic model with the Folding-in method.

Chapter 6

Experimental results

The previous chapters explain the different parts of music genre classification systems in theory and discuss several different algorithms for short-time feature extraction, temporal feature integration, classification and post-processing. However, the usefulness of the algorithms in practice has not been demonstrated. This chapter presents results which have been obtained with the different algorithms and illustrates their practical value.

The first section discusses different evaluation methods which have been used to compare different music genre classification systems or evaluate uncertainty on a performance measure. Afterwards, our two personal music data sets with 5 and 11 genres and 100 and 1210 songs, respectively, are presented. The results of a human genre classification experiment with these data sets are also discussed. The last three sections of the chapter presents the main experimental results which have been achieved in this dissertation project. The first of the three sections gives results from an evaluation of short-time features and the ranking of these as described in chapter 3. The next section regards temporal feature integration which is the area that has received the most attention. Several results are given and e.g. the results for the proposed DPCA, DAR and MAR features are compared with the other features from chapter 4. The last section concerns the comparison between the proposed co-occurrence models in section 5.5 and their traditional counterparts in music genre classification.

6.1 Evaluation methods

One of the most important parts in the process of developing new music genre classification systems is the evaluation methods. Given a (labelled) set of songs, we should be able to tell how well the system performs and to compare the performances of different systems reliably. Many different performance measures have been used in the literature [8] [27]. A good introduction to resampling methods is given in [37].

A very important measure of performance for a music genre classification system is the *generalization error* which is the probability of giving the wrong label to a new song. For supervised learning classifiers, the term "new song" means a song which has not been used in the training procedure and should ideally be completely unrelated to the training set. Hence, an estimate of the generalization error could simply be found by splitting the labelled data set into two disjoint sets, use the first for training (the training set) and predict the genre labels of the second set (the test set). The proportion of test data where the true label is different from the predicted label is then an estimate of the generalization error and is called the *classification test error*. However, a problem with this estimate is that the amount of labelled data is often not very large and it is therefore desirable to use as much of the data set for testing as possible. Note that we sometimes use the term *classification test accuracy* instead which is simply the opposite of the classification test error ie. "accuracy = 1− error".

In our experiments, the *k-fold Cross-validation* method has been used extensively to find an average classification test error. This method effectively uses the whole data set for testing and, besides, can be used to estimate the uncertainty on the estimate of the generalization error. The k-fold cross-validation method is very common in the machine learning society and has been examined in e.g. [8] [37]. The first step of the method is to split the data set. A set with M songs is split randomly into k disjoint sets of equal size (ie. with $\frac{M}{k}$ songs in each). Next, the music genre classification system is trained k times and each time one of the k sets is used as test set and the other $k-1$ sets are used for training. In this way, k classification test error values are found from independent test sets and the average over these values is a reasonable estimate of the true generalization error. The uncertainty on this estimate can be found as e.g. the standard deviation of the mean classification test error.

In some of the experiments (Papers B and C), we used a variation of the above method. This resampling method keeps the test set fixed and use random subsets of the training set for training. However, the above k-fold cross-validation method is considered to give a more accurate evaluation of the classification test error.

We have used the average classification test error in the comparisons between different features and between different classifiers. Besides, the minimization of this term has been used to estimate the optimal values for almost all parameters. This includes parameters such as frame- and hop-sizes of all features, choosing optimal number of features, finding classifier parameters such as the number of mixture components in a GMM and deciding which pre- and post-processing methods to use.

In the important comparisons between e.g. two different feature sets, it is valuable to know whether one feature can be trusted to perform better than the other or not. The author has often used the statistical *McNemar test* for this purpose. As described in e.g. [25], this test starts with the traditional split of the data set into a training set and a test set and the two systems are then trained and tested on these sets. Next, the contingency table with the elements n_{00}, n_{01}, n_{10} and n_{11} is created. The elements are the counts of true and false classifications for each classier. Hence, element n_{01} is the number of times where the prediction from classifier 1 was false, but the prediction from classifier 2 was true. The null hypothesis is that the two classifiers have the same classification test error and hence $n_{01} = n_{10}$. The McNemar test then builds on the test statistic

$$\frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}}$$

which is approximately distributed as a χ^2 distribution with 1 degree of freedom under the null hypothesis. Testing on e.g. a 5 % significance level, the hypothesis should then be rejected if the test statistic is above approximately 3.84.

The *k-fold cross-validated t-test* [25] is especially useful when cross-validation is used to estimate the generalization error. The basic assumption in this test is that the difference between the classification test errors p_A and p_B of two different classifiers A and B is drawn independently from a normal distribution. Assume that 10-fold cross-validation is used which results in classification test errors $p_A^{(i)}$ and $p_B^{(i)}$ for each of the $i = 1, .., 10$ cross-validation runs. Then, the assumption states that the values $p^{(i)} = p_A^{(i)} - p_B^{(i)}$ for $i = 1, .., 10$ are independent and drawn from a gaussian distribution. With this assumption, the Student's t-test can be applied which gives the test statistic

$$\frac{\overline{p}\,\sqrt{k}}{\sqrt{\frac{1}{k-1}\sum_{i=1}^{k}(p^{(i)} - \overline{p})^2}}$$

where k is the number of cross-validation runs and $\overline{p} = \frac{1}{k}\sum_{i=1}^{k} p^{(i)}$ is the average of the differences. This test statistic has a t-distribution with $k-1$ degrees of freedom under the null-hypothesis of equal classification test error.

The cross-validation estimate of generalization error is useful in e.g. parameter estimation and model selection. However, it does not give the full picture of a music genre classification system. For instance, it is also useful to know whether the system mistakes heavy metal with rock or with classical. The latter case would probably often be considered as worse than the first. The *confusion matrix* is a quantity which illustrates this "confusion" in the predictions between the classes. The elements in the confusion matrix are

$$Q_{ji} = P(\hat{C} = i | C = j)$$

where C is the true genre, $\hat{C}$ is the predicted genre from our system and $P(\hat{C} = i|C = j)$ is the estimated probability of predicting a (random) song to belong to genre i when it actually belongs to genre j. P is simply found by counting the different classifications of the test set and normalizing over each class. An example of a confusion matrix is shown in figure 6.6.

Note that slightly other definitions of the confusion matrix exist. For instance, it is sometimes presented simply as the counts of the different (prediction, true label)-pairs or normalized to $P(\hat{C}, C)$. In the current dissertation only balanced classes have been used ie. the estimate of $P(C)$ is uniform. Therefore $P(\hat{C}|C) = \frac{P(\hat{C},C)}{P(C)}$ and $P(\hat{C}, C)$ are equally informative. However, the former is considered more intuitive since each row sums to 100% and the predictions in the columns can easily be compared to that.

6.2 The data sets

One of the most basic elements in the training of a music genre classification system is the data set. Ideally, the data set should cover the whole (user-defined) "music universe", have the true proportions between the number of songs in each genre and the labels should be "ground-truth". This is never the case, but should still be the goal. Music data are fairly easy to obtain in comparison with many other kinds of data. However, there are legal issues concerning the copyright laws of music in sharing music data sets. Hence, only a limited amount of data sets are made publicly available which makes it difficult to compare the algorithms of researchers. One solution to this has been to only share meta-data such as

features. This was done in e.g. the 2004 ISMIR genre classification contest [44] [6]. In the MIREX 2005 contests [53], the algorithms of several different researchers were compared on a common data set by having the researchers submit the actual implementations to an evaluation committee. Recently, as the area of MIR has matured, repositories such as the MTG-database [14] has become available.

The ground-truth of labels of songs are another concern. As discussed in chapter 2 and e.g. [4], a universal ground-truth does not exist. Even getting reliable labels for the data is often a serious practical problem that researchers has to consider. In [68] and [115], the All Music Guide [45] was used to estimate the ground-truth similarities between artists and songs. The All Music Guide has one of the most extensive collections of evaluations of music in many different genres.

In the following, two different data sets are described. These were the most heavily used data sets in the current dissertation project and human evaluations were made of both. In (Paper C), we used a data set from the "Free Download section" at Amazon.com [46], but we did not perform any investigations of the human performance on this data set and it is therefore not described in the following.

6.2.1 Data set A

The data set A consists of 100 songs which are evenly distributed among the five genres Classical music, Jazz, Pop, Rock and Techno. In relation to (Paper B), the first co-author and I labelled the songs. They were chosen to be (somehow) characteristic of the specific genre and therefore the songs in a certain genre were quite similar. Additionally, the genres were chosen to be as different as possible. Hence, this data set was created to give only little variability in the human genre classifications. All songs were ripped from personal CDs with a sampling frequency of 22050 Hz.

Human Evaluation

A classification experiment with human subjects was made to evaluate the data set A. The 22 test subjects (mostly younger people between 25 and 35 years old from the signal processing department without any specific knowledge of music) were asked to log in to a website (at different times) which was build for the purpose. They were then asked each to classify 100 different sound samples of length 740 ms and 30 sounds of length 10 s in two different experiment rounds. The choice of genre was restricted to the five possible (forced-choice) and no prior

information was given apart from the genre names. Both the 740 ms and 10 s samples were taken randomly from the test set in (Paper B) which consisted of 25 songs. The experiment with 740 ms samples had to be completed first before proceeding to the 10 s experiment. This was done to avoid too much correlation between the answers in the two experiments due to recognition of the songs. The subjects could listen to the sound samples repeatedly before deciding, if desired.

It was found that the individual human classification test accuracy in the 10s experiment was 98 % with 95%-confidence interval limits at 97 % and 99 % under the assumption of binomially distributed number of errors. This is in agreement with the desired property of the data set; that it should be a data set with only a small amount of variability on the human classification and with reliable labels. For the 740 ms experiment, the accuracy was 92 % with 95%-confidence interval between 91 and 93 %.

Note that the "individual" human accuracy was found by considering all of the classifications of the songs as coming from a single classifier and comparing these classifications with our "ground-truth" labelling (from the co-author and I). Since the human subjects were not involved in the labelling of the data set, it is interesting to compare their consensus labelling on the data set with our "ground-truth" labelling. This is a measure of the validity of our "ground-truth". All of the songs are considered to be properly labelled since they were each given 20 "votes" for a genre label. To find the consensus labelling of a song, we simply use majority voting i.e. choosing the genre which has the most votes. Comparing this consensus labelling of the data set with our "ground-truth" gave 100 % classification accuracy. In other words, the consensus decision among the human subjects was completely similar to our "ground-truth". This confirms our belief that this is indeed a simple data set and that our "ground-truth" labelling is valid.

6.2.2 Data set B

The data set B contains 1210 songs in 11 genres with 110 in each i.e. the songs are evenly distributed. The 11 genres are Alternative, Country, Easy Listening, Electronica, Jazz, Latin, Pop&Dance, Rap&HipHop, R&B Soul, Reggae and Rock. The songs were originally in the MP3-format (MPEG1-Layer 3 encoding) with a bit-rate of at least 128 kBit, but were converted to mono PCM format with a sampling frequency of 22050 Hz. A preliminary experiment indicated that the decompression from MP3-format does not have a significant influence on the classification test error when the bit-rate is as large as here.

The labels came from a reliable external source, but only the labels were given and a human evaluation of the genre confusion is therefore desirable.

Human evaluation

Data set B was classified in a similar website-based setup as for data set A and with a comparable group of 25 persons. They were now asked to classify 33 music samples of 30s length into the 11 classes. The samples were taken randomly from a subset with 220 songs from data set B.

The individual human classification test accuracy was estimated to 57 % with a 95 %-confidence interval from 54 to 61 % under the assumption of binomially distributed number of errors. This accuracy was found by considering each "vote" for a genre label on a song as the outcome of a single "individual human"-classifier. The corresponding individual human confusion matrix is shown in figure 6.6 where it is compared to the performance of our best performing system (MAR features with the GLM classifier).

As discussed in relation to the human evaluation in subsection 6.2.1, the human consensus labelling can be used to evaluate the "ground-truth" labelling. A procedure to find this human consensus labelling of songs is also given in subsection 6.2.1. The same procedure is used here i.e. majority voting among the "votes" on a song is used to find the human consensus genre for the song. Since the number of evaluations (825) is here quite small compared to the number of songs (220), only a few number of votes were given to each song. It was (heuristically) decided that each song should be given at least 3 votes to be included in the comparison and 172 of the 220 songs fulfill this criterion. The human consensus classification test accuracy was found to be 68 % when compared to the "ground-truth" labelling. Ideally, this accuracy should have been 100%. The quite large discrepancy (32%) is mainly thought to originate in a lack of knowledge about music genres among the human test subject. An indicator of this is, that a few particular subjects with a background in music had very high test accuracy. The corresponding human consensus confusion matrix is illustrated in figure 6.1. It clearly illustrates that the human subjects do not agree with our "ground-truth" on especially Alternative and Easy-listening which are probably not as easily defined as e.g. Country. It should also be noted that there are other sources of uncertainty. Certainly, the number of votes for each song was not very large. 94% of the considered songs had between 3 and 6 votes and 33% only had 3 votes. This is arguably a large source of uncertainty on the consensus labelling.

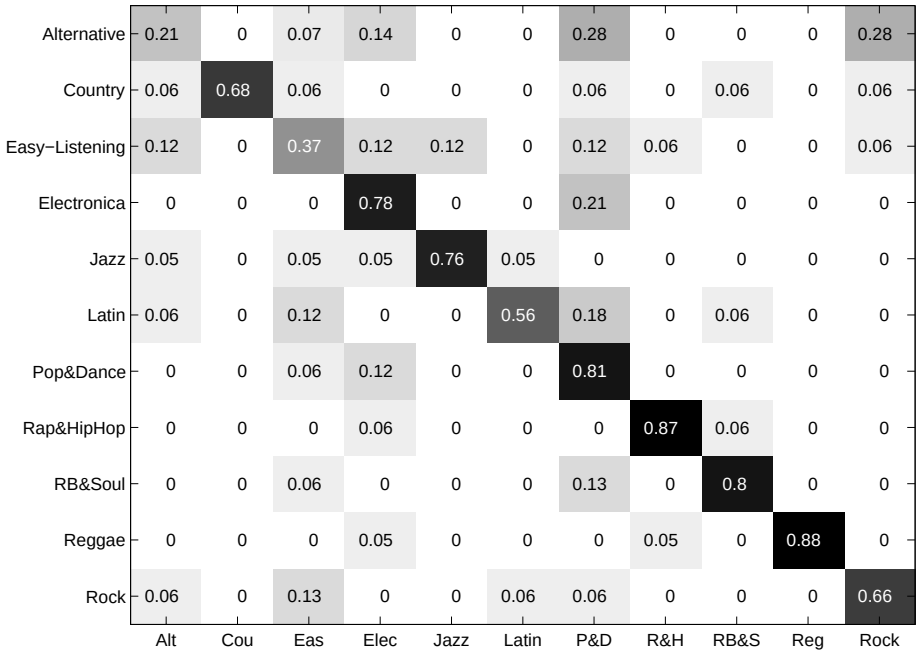


Figure 6.1: The figure illustrates the human consensus confusion matrix from the human evaluation of data set B. The human consensus on a song is found by majority voting among all the human evaluations of the song. The genres in the rows are the "ground-truth" labels and the columns are the human consensus genres. It is seen that e.g. the music with "ground-truth" label Alternative is mostly classified into Pop&Dance or Rock in our human evaluation.

6.3 Ranking of short-time features

The first part of this disseration project was mainly exploratory and several different short-time features were investigated in the process. The main results are described in (Paper B). The features were ranked using the consensus sensitivity analysis method as described in subsection 3.2.1. Figure 6.2 illustrates the ranking of the MFCC, DMFCC, LPC, DLPC, STE, ZCR, ASE, ASC, ASS and SFM features which were described in section 3.1. Data set A was used for the ranking and it is seen that the MFCC (**A** in the figure), LPC (**C**) and ZCR features appear to be the most relevant. In contrast, the derived DMFCC (**B**) and DLPC (**D**) features show least importance. The relevance of the MPEG-7 features is less consistent. For this reason, we decided to use the MFCCs as the short-time feature representation to be used in the temporal feature integration experiments.

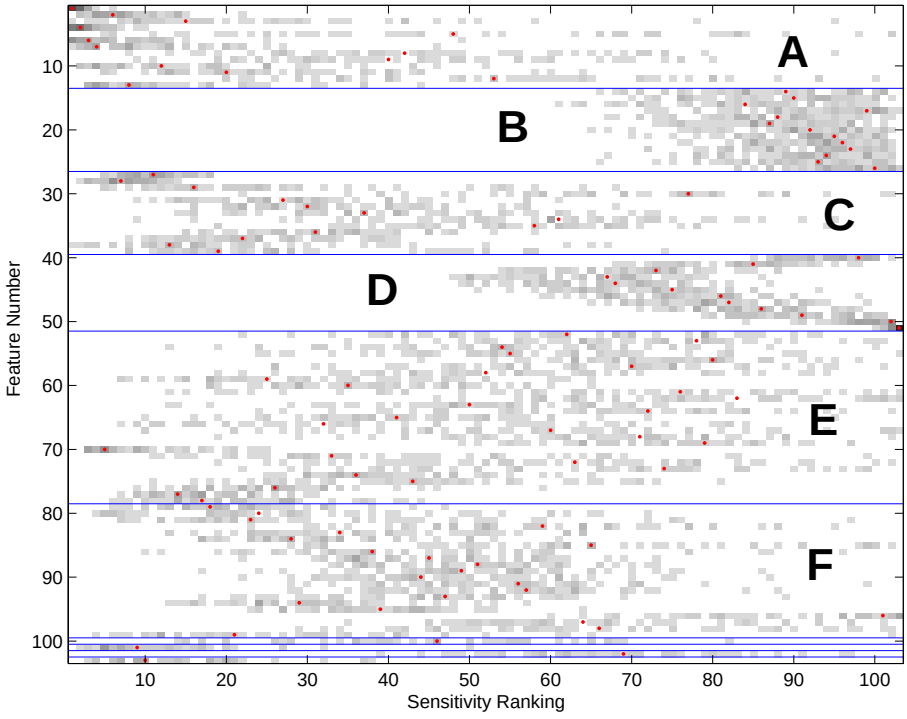


Figure 6.2: The figure is the result of Consensus sensitivity analysis for feature ranking with the Dynamic Principal Component Analysis (DPCA) method and using 50 resamplings. The frame size was $N = 100$ which corresponds to 1s frames. The y-axis shows the feature number out of a total of 103 and they are, from above, MFCC(**A**), DMFCC(**B**), LPC(**C**), DLPC(**D**), ASE(**E**), SFM(**F**) and the single features ASC, ASS, STE and ZCR. These short-time features are all described in section 3.1. The x-axis gives the ranking number of the features. The red dots indicate the final ranking of each feature with the use of the Consensus sensitivity analysis method. The ten best features in decreasing order are found to be $\{1, 4, 6, 7, 70, 2, 28, 13, 101, 103\}$. The grey colors indicate the total number of "votes" to a feature for a given ranking in all resamplings. The darker the color, the more "votes" has been given to a given ranking number. It is seen that e.g. the DMFCCs and DLPCs are quite consistently ranked low whereas the results are less clear for the ASE features. The MFCCs and LPCs are generally seen to rank high.

6.4 Temporal feature integration methods

Temporal feature integration has been the major topic in this project and several experiments were made. Concerning the DPCA method and results with this method, the reader is referred to (Paper B) since the method seemed less promising. The results were not better than classifying each short-time feature vector in the song individually and using majority voting for the post-processing. Results for the proposed DAR and MAR features from section 4.2 will be treated more carefully in the following and compared to other temporal feature integration methods.

In (Paper C), we examined several different combinations of temporal feature integration to different timescales with the MFCCs as short-time feature representation as always and 6 MFCCs (the first 6) were found to be optimal with the chosen classifiers. The resampling method as explained in section 6.1 was used to estimate the classification test error. The temporal feature integration methods that have been used were all described in chapter 4. The results are illustrated for data set A in figure 6.3 for the Linear Regression and Gaussian classifiers from chapter 5 and discussed in the following.

The part of the y-axis named "Long time feature integration" illustrates different combinations of feature integration to the long time scale. In the context, the long time scale is 10s, the medium time scale is 740 ms and the short time scale (of the MFCCs) is 30 ms. For instance, the "MeanVar23d" feature is therefore the combination of first finding the DAR features from temporal feature integration to the medium time scale. The MeanVar temporal feature integration method is then applied on these medium time scale DAR features (signified by the "d" in the feature name) up to the long time scale. "23" signifies the integration between the medium and long time scale. In contrast, the "DAR13" features are found by applying the DAR feature integration directly from the short time scale up to the long time scale. Although many different combinations of temporal feature integration were examined in this part, the results were not as good as in the "Medium to Long Sum Rule" part. One reason for this might be that it was necessary to apply PCA for dimensionality reduction on the "DAR23m", "DAR23d" and "MeanVar23d" methods due to problems with overfitting in the classifiers. These methods might therefore have given better results with classifiers that can better handle high-dimensional features or with a larger data set.

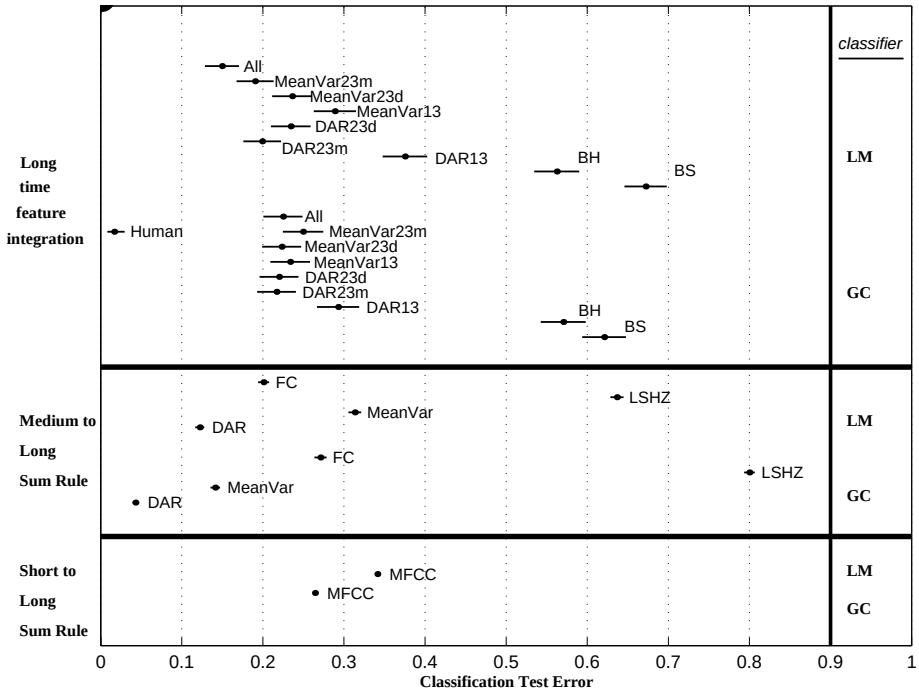


Figure 6.3: The average classification test error on data set A is illustrated for several temporal feature integration combinations as well as the human performance. The figure consists of three parts which indicate whether temporal feature integration or the sum-rule postprocessing method has been used to achieve a decision on the long time scale (10s). For instance, in the "Medium to Long Sum Rule" part, integration has been used from the short time scale (30 ms) up to the medium time scale (740 ms) and then the sum rule method has been used from the medium to long time scale. It should be noted that the MFCCs have been used as the common short-time representation. Since all of the results are classifications of the whole song (10s), they can be compared directly. The feature names are explained in the text. Results are given for both the Gaussian Classifier (GC) and the Linear Regression classifier (LM). The error bar on the human performance ("Human") indicates the 95 % confidence intervals under the assumption of binomially distributed errors. The error bars on the features are the estimated standard deviation on the average classification test error on each side. Note that the Low Short-Time Energy Ratio (LSTER) and High Zero-Crossing Rate Ratio (HZCRR) features are used together under the label "LSHZ".

In the "Medium to Long Sum Rule" part, the temporal feature integration is solely applied from the short to the medium time scale. Hence, each 10 s (long time scale) sound clip is represented by a time series of feature vectors instead of a single feature vector as in the "Long time feature integration"-part. The result for e.g. the "DAR" feature is therefore the application of DAR features from short to medium time and succeeded by the sum-rule postprocessing method to achieve a decision on the long time scale. This 3-step procedure of first extracting short-time features, then performing temporal feature integration up to an intermediate time-scale and finally applying post-processing of classifier decisions gave the best results. This indicates that certain important aspects of the music exist on this intermediate level and are captured by the DAR features.

It is seen that the Low Short-Time Energy Ratio (LSTER) and High Zero-Crossing Rate Ratio (HZCRR) features (used together in a single 2-dimensional feature vector with the name "LSHZ") perform much worse than the best features. However, the comparison is not really fair since these are of much lower dimensionality. Hence, they cannot stand alone, but might be very useful as supplementary features and, besides, they were created for audio signals in general. Similarly, the Beat Histogram (BH) and Beat Spectrum (BS) features are likely to be very useful as supplementary features, but their individual performance is low in the comparison. Hence, these four features were not considered further in (Paper G).

The Frequency Coefficient (FC) and MeanVar features were quite successful, but still less than the DAR features. This hypothesis was supported with a McNemar test on a 1% significance level.

In the part named "Short to Long Sum Rule", no temporal feature integration methods are used, but instead the sum-rule method is used directly on the classifier outputs from the short-time MFCCs to reach a decision on the long time scale.

The DAR, FC and MeanVar features were investigated further in (Paper G) with the inclusion of the proposed MAR features and the MeanCov features. Figure 6.4 illustrates the average classification test errors of these features on data set A and B using four different classifiers.

The figure is the result of numerous experiments and optimizations to get a fair comparison between the temporal feature integration methods. The use of four different classifiers increases the generalisability of the results and the MFCCs have again been used as short-time feature representation. In the optimization phase as well as in general, the performance was evaluated with the average classification test error from k-fold cross-validation.

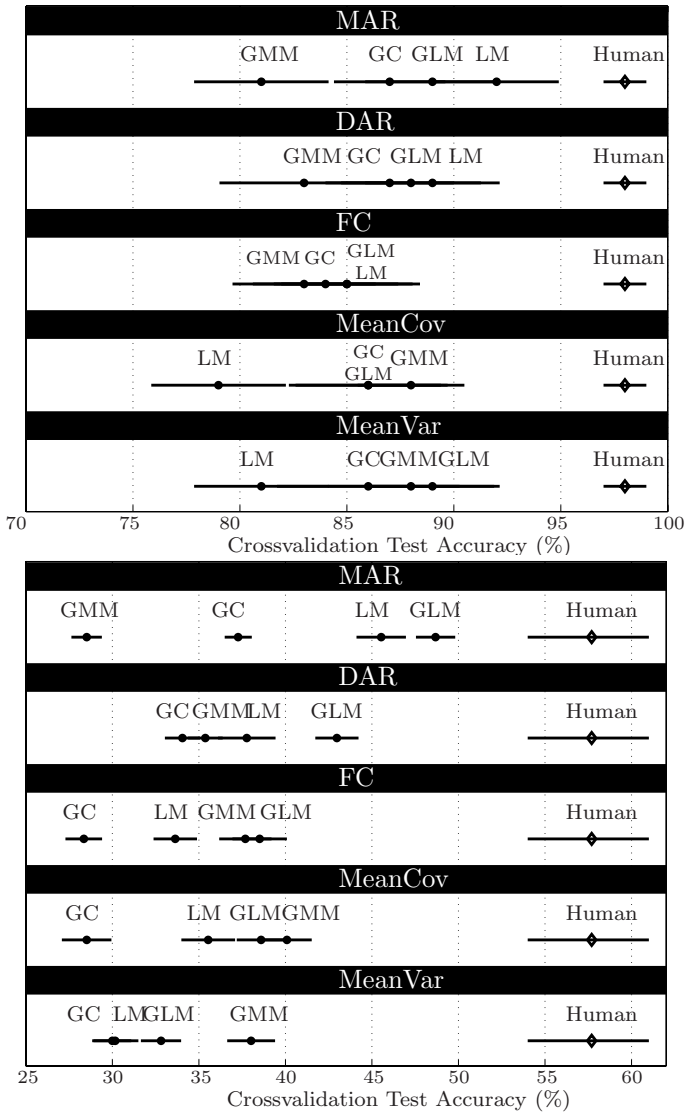


Figure 6.4: The average classification test accuracies are illustrated for 5 different features from temporal feature integration. The upper part shows the results from data set A and the lower from data set B. The MeanVar, MeanCov and FC features are compared to the proposed DAR and MAR features (see chapter 4). To increase the generalisability of the results, 4 different classifiers have been used (Gaussian classifier (GC), Gaussian Mixture Model (GMM), Linear Regression model (LM) and Generalized Linear Model (GLM)). The MFCCs were used as short-time feature representation. The individual human classification accuracy from the human evaluations of the data sets is also shown for comparison. The error bars on the human performance are the 95% confidence interval under assumption of binomially distributed number of errors. The error bars on the features are one standard deviation of the average classification test error on each side.

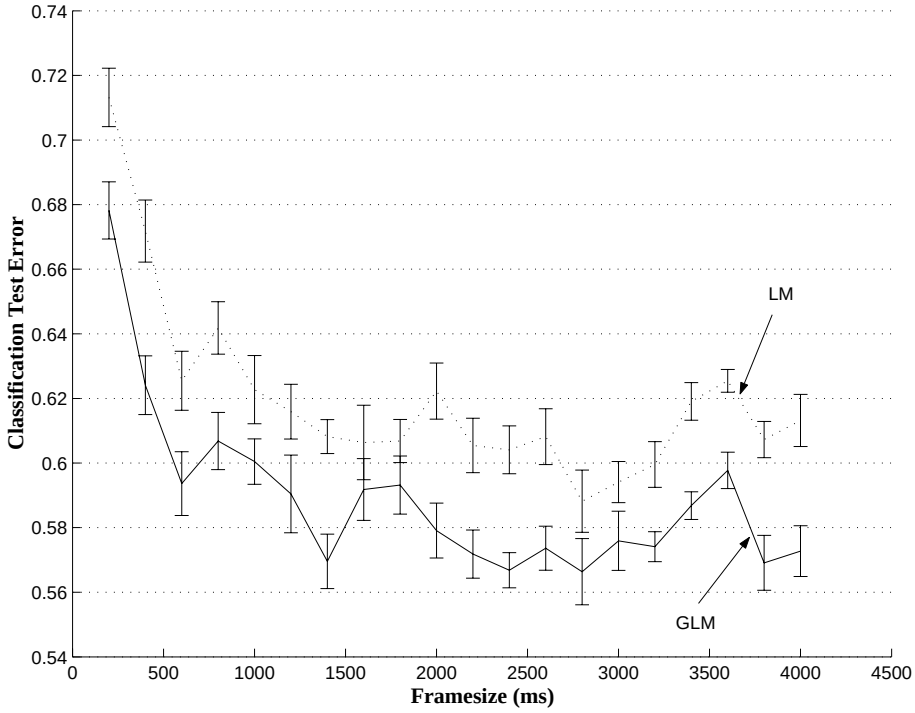


Figure 6.5: The figure illustrates the average classification test error for the DAR feature as a function of frame size on data set B. Results are shown for both the Linear Regression classifier (LM) and the Generalized Linear Model classifier (GLM). The error bars are the standard deviation on the average over 10 cross-validation runs. There is clearly a large variation over the different frame sizes which shows that the frame size is an important parameter in temporal feature integration.

The optimization of parameters such as the number of MFCCs, hop- and frame-sizes in both short-time features as well as temporal feature integration for each feature, DAR and MAR model order parameters, classifier parameters, etc., will clearly be suboptimal since the parameter space is vast. Here, some preliminary experiments were made to find "acceptable" system parameters. Afterwards, the parameters were further optimized sequentially and following the flow in the classification system. In other words, first the feature extraction related parameters were optimized. Next, the temporal feature integration parameters were optimized and so forth. The optimal number of MFCCs were 6 and with optimal hop-size 7.5 ms and frame-size 15 ms. As seen in figure 6.5, the optimization of especially the frame-size of the temporal feature integration seems to be important. The optimal frame-sizes were found to be 1400 ms, 2000 ms, 2400 ms, 2200 ms and 1200 ms for the MeanVar, MeanCov, FC, DAR and MAR features, respectively. The optimal model order P was found to be 5 for the DAR model and 3 for the MAR model. Note that experiments were also made with single MAR feature vectors to describe the whole 30s sound clip i.e. choosing a 30s frame-size. The performance with this frame-size was not as good (44% accuracy) as for the combination of the 1200 ms frame-size and sum-rule postprocessing up to 30s. However, this results still illustrates that a lot of the information in a 30s sound clip can be represented in a single (135-dimensional) feature vector. Such a feature vector could be used directly in similarity measures for music recommendation or unsupervised clustering.

Returning to figure 6.4, there are several things to note. The MAR feature seems to outperform the other features on data set B when the best classifiers are used for each feature. This result was supported with a 10-fold cross-validated t-test on a 2.5 % significance level.

The performance on data set A is less clear, but it should also be remembered that data set A was chosen specifically to have clearly (artificially) separated genres and this probably explains the good performance of all of the systems.

It is seen that the human performance is better than the systems on both data sets. The human performance is here measured by considering the human evaluations as individual classifications i.e. the systems are compared to the average human performance (as discussed in chapter 2).

The DAR feature appears to perform better than the MeanVar, MeanCov and FC features on data set B, but this could only be supported for the MeanVar and FC features with the cross-validated t-test on the 2.5% significance level.

Another interesting detail is the differences between the classifiers. There is the tendency that the discriminative classifiers LM and GLM perform better on the high-dimensional features DAR (42-dim.) and MAR (135-dim.) whereas the

generative GC and GMM classifiers were better with the FC (24-dim.), MeanCov (27-dim.) and MeanVar (12-dim.) features. Although our learning curves did not show clear evidence of overfitting (unless for the MAR features), this tendency is still thought to be related to the curse of dimensionality. Note that it was necessary to use Principal Component Analysis (PCA) for dimensionality reduction on the MAR features to be able to use the GMM classifier due to overfitting problems. This is a likely explanation for the poor performance of the MAR features with the GMM classifier.

Figure 6.6 compares the confusion matrices for the best performing system (MAR features with the GLM classifier) with the individual human confusions between genres on data set B. Overall, there seems to be some agreement about the easy and difficult genres. Notably, the three genres that a human would classify correctly most often (Country, Rap&HipHop and Reggae) are similar to the three genres that our system is best at.

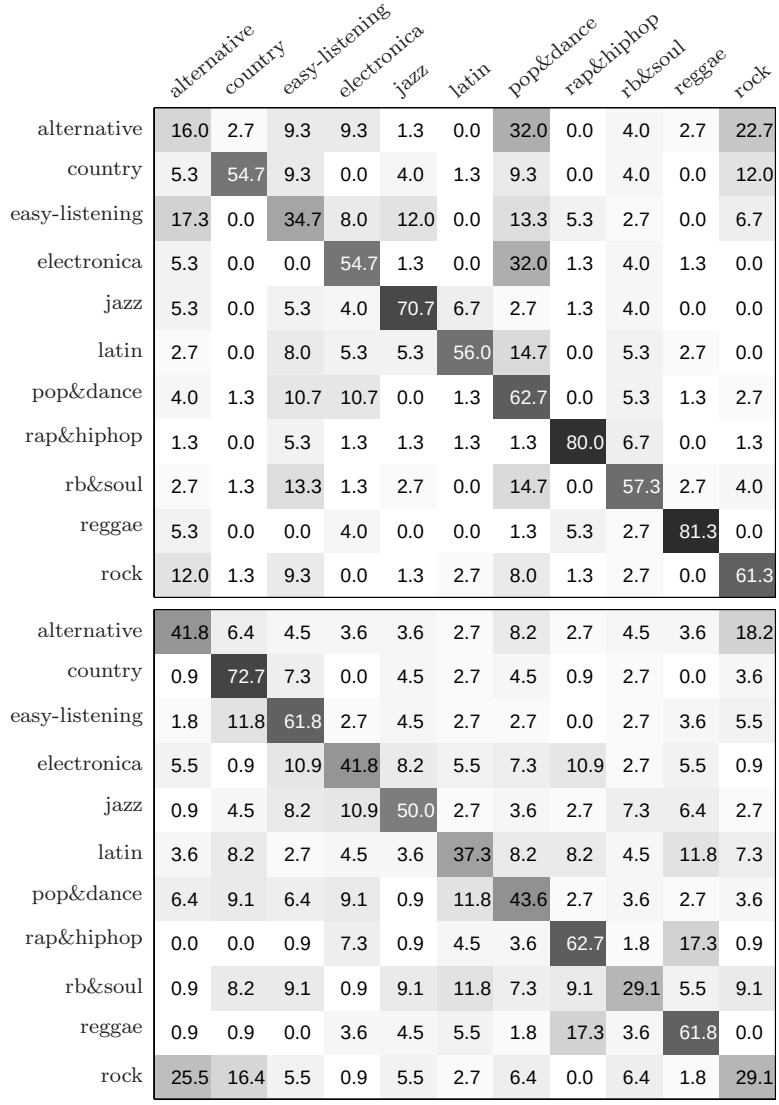


Figure 6.6: Confusion matrices for our best performing music genre classification system as well as the individual human confusion on data set B. The upper figure corresponds to the human evaluation and the lower to the system which used MAR features on MFCCs with the Generalized Linear Model classifier. The "true" genres are shown as the rows and sum to 100% whereas the predicted genres are in the columns. Hence, the diagonal illustrates the accuracy of each genre separately.

6.5 Co-occurrence models

In (Paper D), we proposed co-occurrence models for music genre classification which resulted in the Aspect Gaussian Classifier (AGC) and the Aspect Gaussian Mixture Model (AGMM) classifier. These classifiers were compared to their classical counterparts, the Gaussian classifier (GC) and Gaussian Mixture Model (GMM), and the results are illustrated in figure 6.7. The 30-dimensional DAR features (as in (Paper C)) on data set B were used in the experiments. It is seen that the AGMM performs slightly better than the classical GMM model while the AGC performs comparably to the GC. However, the real force of the AGMM and AGC models is considered to be the probabilistic modelling of the whole song instead of just individual frames. Mathematically, this corresponds to modelling the genre probability $p(C|S = s)$ for a song s instead of modelling $p(C|\mathbf{z}_n)$ where $\mathbf{z}_n$ is a feature vector. The figure also illustrates the result of the "Discrete Model" which uses a vector quantization of the feature space into a discrete codebook before using the "discrete" version of the AGC. The reader is referred to (Paper D) for the details.

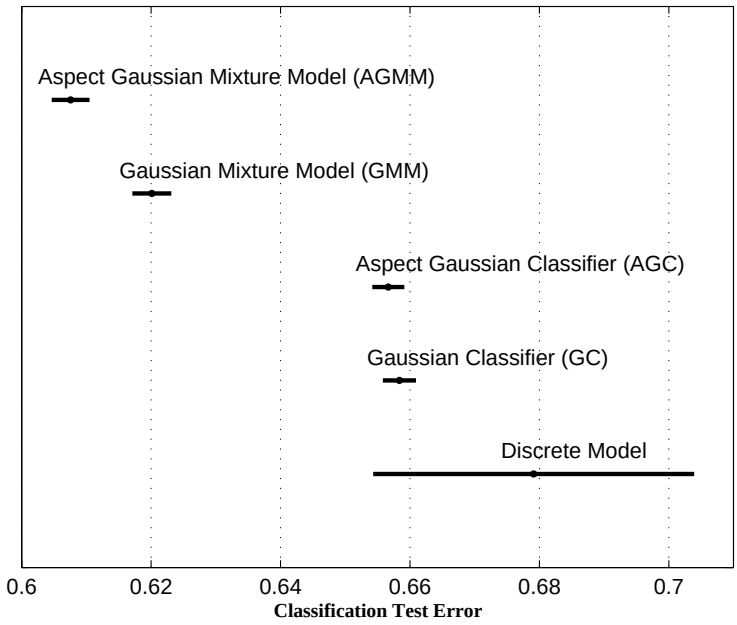


Figure 6.7: Classification test errors for the Discrete Model, the Aspect Gaussian Classifier, the Aspect Gaussian Mixture Model and the two baseline methods Gaussian Classifier and Gaussian Mixture Model on data set B. The results are the mean values using resampling (only 5-fold for the Discrete Model due to computational constraints and 50-fold for the rest) and the error bars are the standard deviations on the means. 7 mixture components were used for the GMM and AGMM.

Chapter 7

Discussion and Conclusion

Music genre classification systems have been the primary topic of this dissertation. The emphasis has been on systems which use music in form of raw audio as input and return an estimate of the corresponding genre of the music as output. The main goal has been to create systems with as low error as possible on the genre-predictions of new songs. Briefly, our best performing music genre classification system is capable of classifying into 5 genres with an accuracy of 92% compared to a human accuracy of 98%. For 11 genres, the accuracy was 48% compared to 57% for humans. The full music genre classification procedure on a song is possible in real-time on an ordinary PC. These results illustrate the overall perspectives in our state-of-the-art music genre classification system.

Although the focus has been on music genre classification, most of the results are directly applicable to other areas of Music Information Retrieval such as music artist identification and in music recommendation systems. These areas also need a compact, expressive feature representation of the music. Our main investigations of features on larger time scales (in the order of several seconds) might also be relevant in Speech Analysis as suggested in [85]. The proposed ranking and classification methods have an even wider audience.

Generally, our approach to the music genre classification problem has been system-oriented i.e. all the different parts of the system have to be taken into consideration. The main parts of a music genre classification system are

traditionally the feature representation and the classifier. However, there are many other concerns such as optimization of hop- and frame-sizes, normalization aspects, post-processing methods, considerations about data sets, validity of labels, performance measures and many others. This dissertation try to give an overview of the challenges in building real-world music genre classification systems.

Although system-oriented, special focus has been given to the feature representation which is here split into *Short-time feature extraction* and *Temporal feature integration* (see e.g. figure 1.1 for an overview of a system). Briefly, the class of short-time features are extracted on a time-scale of 10-40 ms. This class contains numerous different features and a selection of these have been investigated and ranked by their significance in music genre classification. We proposed the *Consensus sensitivity analysis* method for ranking in (Paper B) which has the advantage of being able to combine the sensitivities over several cross-validation or other resampling runs into a single ranking.

Temporal feature integration is the process of combining the information in a (multivariate) time series of short-time features. The main contributions of the dissertation have been made in this area where two new methods have been proposed; *Dynamic Principal Component Analysis* (Paper B) and the *Multivariate Autoregressive Model* (Papers C and G) for integration. Especially the Multivariate Autoregressive Model showed promising results. Two novel features, the DAR and MAR features, were extracted from this model. They were compared to state-of-the-art temporal feature integration methods and found to generally outperform those. Our best performing system with MAR features was compared to the most common integrated features which use mean and variance of the short-time features. Our system achieved 48% accuracy compared to 38% for these features on an 11-genre problem.

Besides, the proposed Multivariate Autoregressive Model is a general flexible framework. Hence, it may be included in e.g. probabilistic models or kernels for Support Vector Machines [83]. The DAR and MAR features contain the model order as a parameter and are hence quite flexible. These parameters should be optimized to the specific problem.

The classification part should not be neglected and although given less emphasis than the feature representation, several classifiers have been examined in the experiments. In (Paper D), we proposed novel *Co-occurrence models* for music genre classification. Although they did not give large improvements in classification test accuracy, they have other advantages. For instance, they are capable of explicitly modelling the whole song in the probabilistic framework. This is in contrast to most of the classifiers which have traditionally been used in music genre classification.

Summary and discussion

The early phases of the project involved a variety of investigations of different short-time features as described in (Paper B). However, the main result from these investigations is considered to be the ranking of the features and here, the Mel-Frequency Cepstral Coefficients (MFCCs) appeared to be the highest ranked set of features. This was the motivation to use the MFCCs as the short-time representation in all of the following experiments with temporal feature integration. The proposed Consensus sensitivity analysis method was used for the ranking. This method is an extended version of an ordinary sensitivity analysis method. The advantage is that it is able to combine the sensitivities of the features from several cross-validation or other resampling runs into a single ranking. One disadvantage of the method is that it measures sensitivity by changing each feature individually. However, it is quite possible that the combination of several low-ranked features perform better than a combination of the same size, but with high-ranked features. This is a motivation to use incremental feature selection techniques instead of ranking. Still, the choice of the MFCCs as short-time representation appears to have been reasonable since many others have also had good results with these short-time features [77] [110].

As mentioned before, temporal feature integration has been the main topic in this dissertation. Several methods from the literature have been examined and compared to the novel Dynamic Principal Component Analysis (DPCA) and Multivariate Autoregressive Models. The most common temporal feature integration method in the literature is simply to take the mean and variance of the short-time features in the larger time frame (e.g. 2000 ms) and use these statistics as an integrated feature vector. This feature is so common that it is considered the baseline against which we have compared our own methods. We named it the MeanVar features for reference.

The DPCA feature is created by first stacking the short-time features in the frame into a single (high-dimensional) feature vector and then use Principal Component Analysis for dimensionality reduction. This feature captures the correlations in both time and among short-time feature dimensions. In (Paper B), we compared it against a simple approach without temporal feature integration which instead used Majority Voting on the short-time decisions. The results with these two approaches were fairly similar and since the DPCA feature was more computationally demanding, it was not considered further.

The idea of the Multivariate Autoregressive Model for temporal feature integration is, as the name suggests, to model the multivariate time series of short-time feature vectors with a multivariate autoregressive model. In the frequency domain, the autoregressive model can be seen as "spectral matching" of the power

cross-spectra of the short-time features. The parameters of the model are used as the features. We examined two different kinds of features from this model; the *Diagonal Autoregressive* (DAR) features and the *Multivariate Autoregressive* (MAR) features. The MAR features use the parameters of the full multivariate model, whereas the DAR features consider each short-time feature dimension individually which corresponds to diagonal autoregressive coefficient and noise matrices in the model. Hence, where the MAR features are capable of modelling both temporal dependencies as well as among feature dimensions, the DAR features only model the temporal information. Note that the MeanVar features do not model any of these dependencies.

Both the DAR and MAR features were found to outperform the baseline MeanVar features on our difficult data set B. The MeanVar, DAR and MAR features had classification test accuracies of 38%, 43% and 48%, respectively. In comparison, the estimate of the human accuracy on this data set was 57%. We also made an investigation of the computational complexity of the methods. With our choices of model order and MFCC feature dimension, the DAR and MAR features were about an order of magnitude more computationally demanding in time than the MeanVar features. This suggests that the DAR and MAR features are good replacements for the MeanVar features in many applications where this difference in computation time is not critical. It might be argued that the DAR feature is less useful than the MAR, but note that the MAR features have much higher dimensionality. For instance, in our experiments, the DAR features are 42-dimensional whereas the MAR features are 135-dimensional. In some situations, this would make the DAR features more attractive.

Another advantage of the DAR and MAR features are their flexibility. Since they are build from the autoregressive model, it is possible to adjust the model order to the given problem. In fact, the MeanVar feature can be seen as a special case of the DAR features with model order 0. However, note that the computational demands are closely related to the model order and the number of short-time features. Choosing for instance model order 12 for 12 short-time features would make the calculation of the MAR feature approximately 600 times slower than the MeanVar and the DAR feature 60 times slower. Fortunately, our results were obtained with model order 3 and 5 for the DAR and MAR features, respectively, and using only 6 MFCCs.

An interesting aspect of temporal feature integration is the frame size since it gives the natural time scale of the features. We believe the frame size to be related to certain elements of the music. For instance, we found optimal frame sizes to be 1200 ms and 2200 ms, respectively, for the MAR and DAR features. Although it is not known, it is likely that the DAR and MAR features capture dynamics on those time scales like the rhythm. Certainly, it is found that the frame size in temporal feature integration is an important parameter. This is

in agreement with e.g. [108], [7] and [113].

In (Paper E), we describe our MAR features in relation to the MIREX 2005 music genre classification contest [53] which we participated in. Such contests are very informative since they allow researchers to compare their algorithms in a common framework on similar data sets, with similar performance measures and so forth. Our system had an overall accuracy estimate of 72% compared to the winning system with 82 % accuracy. One may argue that our features are not interesting after such an evaluation. However, this would not be the right conclusion to draw. The reason is that even with the mentioned advantages of a common testing framework, there are many differences among the submitted systems. For instance, very different classifiers have been used in the contest which might explain a 10% difference in performance. This is an illustration of the difficulties in comparing full systems due to their complexity ("the devil is in the detail"). As discussed in the following section, it is indeed likely that the combination of elements from the different systems may give the best result.

In (Paper D), we investigated co-occurrence modelling for classification of music into genres. We proposed two different classifiers which are based on the co-occurrence model; the *Aspect Gaussian Classifier* and the *Aspect Gaussian Mixture Model*. These names were given since they can be seen as extensions of the Gaussian Classifier (GC) and the Gaussian Mixture Model (GMM), respectively. Many traditional classifiers (such as the GC and GMM) first model each feature vector individually. Afterwards, they need to apply post-processing methods such as majority voting to combine the decisions from each of the feature vectors in the sequence. This is used to reach a single genre decision for the whole song. In contrast, the co-occurrence models have the advantage of being able to include the whole song in the probabilistic model. In other words, the probability $P(s|C)$ of a song s given the genre (which is transformed to the desired quantity $P(C|s)$ with Bayes' rule) is modelled directly instead of modelling $P(\mathbf{z}_n|C)$ where $\mathbf{z}_n$ is one of the feature vectors in the song s.

Future work

The current project has investigated many different elements and problems in music genre classification. In the progress many new ideas were fostered, but only a few made it into the "large-scale investigation" step. The following part discusses the ideas which are believed to be the most promising.

More powerful classifiers on high-dimensional features One of the main results, in my view, from the MIREX 2005 music genre contest [53] (Paper

E) is the importance of the classifier. In our experiments, we mostly experimented with different features and the classifiers were fairly simple. It would be very interesting to experiment with high-dimensional DAR and MAR features (e.g. 1000-dimensional) with more powerful classifiers such as Adaboost-methods [7], SVMs, Gaussian Process classifiers or similar. Recall that we used only six MFCCs in the short-time feature representation whereas e.g. [77] used 20 MFCCs. Hence, DAR or MAR features with a larger number of MFCCs might increase performance. It would also be possible to increase the model order of the autoregressive model.

Effective dimensionality reduction on high-dimensional features It is generally desirable to have as low-dimensional feature vectors as possible. This is contradictory to the previous idea of experiments with high-dimensional features, but there the motivation was only the specific task of assigning a genre to a piece of music. In other tasks, it is convenient to have the generative probabilistic model of the song which normally requires low-dimensional features. This could e.g. be used to detect outliers which might indicate the emergence of a new genre. The generative model might, for instance, be the proposed Aspect Gaussian Mixture Model to include the full song in the model. It would be interesting to experiment with different dimensionality reduction techniques on high-dimensional DAR and MAR features. Our experiments indicate that the PCA method is insufficient for this purpose. However, methods such as ICA (Paper F), sparse methods or supervised methods might be useful.

Enforcing genre relations Most music genre classification systems consider the genres as equidistant and in a flat hierarchy (a notable exception is [12]). This is clearly not correct. For instance, soft rock songs are much closer to the genre pop than to traditional classical music. The genre relations could be enforced by many different methods. For instance, with a hierarchy. Another possibility would be to train the system with multi-labelled songs or ideally a full genre-distribution as discussed in chapter 2, but this would require such a data set which is likely to be a problem. Another interesting solution would be to simply apply a utility function [75] on the classifier (also called a loss function [8]). Here, it should be a matrix which signifies the relations between genres. Hence, assume that a song would have been classified as 40% classical, 38% rock and 22% pop. The utility matrix will then increase the probability of rock due to the large probability of pop and the song would be classified as rock.

Appendix A

Computationally cheap Principal Component Analysis

We need to find the first, say, $l = 50$ eigenvectors of the covariance-matrix from the training set. The training set matrix is called X_{train} with form [m dimensions x n samples]. Since time stacking is used in the DPCA feature, m can be around 10000 and n 100000. This gives computational problems in both time and space in the creation of the covariance matrix (or even forming the X_{train} matrix). A computationally cheap method is used as described in [100] where only k samples are taken from X_{train}, e.g. k equal 1000 or 1500. The samples are taken randomly. Note, that the mean should be subtracted first to get the covariance matrix eigenvectors instead of just the second moment eigenvectors. Then form the $\widetilde{X}$ [m dimension x k samples] containing the k columns from X_{train}. Since $\widetilde{X} = \widetilde{U}\widetilde{S}\widetilde{V}^T$ with dimensions [m x k], [k x k] and [k x k], respectively (this is the so-called "thin" Singular Value Decomposition), it is possible to form $\widetilde{X}^T\widetilde{X} = \widetilde{V}\widetilde{S}^2\widetilde{V}^T$ [k x k]. Note that normally the matrix $\widetilde{\Sigma} = \widetilde{X}\widetilde{X}^T$ (here, the covariance matrix) would be created instead since its eigenvectors is the PCA projection vectors directly. However, in this case it would be [m x m] (e.g. [10000 x 10000]) which would be hard to handle computationally.

It is now simple to find $\widetilde{V}$ and $\widetilde{S}$ from an eigen-decomposition of $\widetilde{X}^T\widetilde{X}$. Afterwards, since $\widetilde{U} = \widetilde{X}\widetilde{V}\widetilde{S}^{-1}$, it is easy to calculate $\widetilde{U}$ [m x k]. Finally, only l (in this case 50) eigenvectors are taken from $\widetilde{U}$ to get $\widehat{U}$ (by taking l columns of

$\widetilde{U}$). To transform the test data in X_{test} [m dims x p samps] into the cheap PCA basis, simply use $\widehat{X}_{test} = \widehat{U}^T X_{test}$ [l dims x p samps].

Appendix B

Decision Time Horizon for Music Genre Classification using Short-Time Features

Ahrendt P., Meng A. and Larsen J., **Decision Time Horizon for Music Genre Classification using Short Time Features**, Proceedings of *European Signal Processing Conference (EUSIPCO)*, Vienna, Austria, September 2004.

DECISION TIME HORIZON FOR MUSIC GENRE CLASSIFICATION USING SHORT TIME FEATURES

Peter Ahrendt, Anders Meng and Jan Larsen

Informatics and Mathematical Modelling, Technical University of Denmark
Richard Petersens Plads, Building 321, DK-2800 Kongens Lyngby, Denmark
phone: (+45) 4525 3888,3891,3923, fax: (+45) 4587 2599, email: pa,am,jl@imm.dtu.dk, web: http://isp.imm.dtu.dk

ABSTRACT

In this paper music genre classification has been explored with special emphasis on the decision time horizon and ranking of tapped-delay-line short-time features. Late information fusion as e.g. majority voting is compared with techniques of early information fusion[1] such as dynamic PCA (DPCA). The most frequently suggested features in the literature were employed including mel-frequency cepstral coefficients (MFCC), linear prediction coefficients (LPC), zero-crossing rate (ZCR), and MPEG-7 features. To rank the importance of the short time features *consensus sensitivity analysis* is applied. A Gaussian classifier (GC) with full covariance structure and a linear neural network (NN) classifier are used.

1. INTRODUCTION

In the recent years, the demand for computational methods to organize and search in digital music has grown with the increasing availability of large music databases as well as the growing access through the Internet. Current applications are limited, but this seems very likely to change in the near future as media integration is a high focus area for consumer electronics [6]. Moreover, radio and TV broadcasting are now entering the digital age and the big record companies are starting to sell music on-line on the web. An example is the popular product iTunes by Apple Computer, which currently has access to a library of more than 500,000 song tracks. The user can then directly search and download individual songs through a website for use with a portable or stationary computer.

A few researchers have attended the specific problem of music genre classification, whereas related areas have received more attention. An example is the early work of Scheirer and Slaney [17] which focused on speech/music discrimination. Thirteen different features including *zero-crossing rate* (ZCR), *spectral centroid* and *spectral roll-off point* were examined together using both Gaussian, GMM and KNN classifiers. Interestingly, choosing a subset of only three of the features resulted in just as good a classification as with the whole range of features. In another early work Wold *et al.* [22] suggested a scheme for audio retrieval and classification. Perceptually inspired features such as pitch, loudness, brightness and timbre were used to describe the audio. This work is one of the first in the area of content-based audio analysis, which is often a supplement to the classification and retrieval of multimodal data such as video. In [12], Li *et al.* approached segment classification of audio streams from TV into seven general audio classes. They find that *mel-frequency cepstral coefficients* (MFCCs) and *linear prediction coefficients* (LPCs) perform better than features such as ZCR and *short-time energy* (STE).

The genre is probably the most important descriptor of music in everyday life. It is, however, not an intrinsic property of music such as e.g. tempo and makes it somewhat more difficult to grasp with computational methods. Aucouturier *et al.* [2] examined the inherent problems of music genre classification and gave an overview of some previous attempts. An example of a recent computational method is Xu *et al.* [23], where support vector machines were used in a multi-layer classifier with features such as MFCCs, ZCR and LPC-derived cepstral coefficients. In [13], Li *et al.* introduced DWCHs (Daubechies wavelet coefficient histograms) as novel features and compared these to previous features using four different classifiers. Lambrou *et al.* [11] examined different wavelet transforms for classification with a minimum distance classifier and a least-squares minimum distance classifier to classify into rock, jazz and piano. The state-of-art percentage correct performance is around 60% considering 10 genres, and 90% considering 3 genres.

In the MPEG-7 standard [8] audio has several *descriptors* and are meant for general sound, but in particular speech and music. Casey [5] introduced some of these descriptors, such as the *audio spectrum envelope* (ASE) to successfully classify eight musical genres with a hidden markov model classifier.

McKinney *et al.* [15] approached audio and music genre classification with emphasis on the features. Two new feature sets based on perceptual models were introduced and compared to previously proposed features with the use of Gaussian-based quadratic discriminant analysis. It was found that the perceptually based features performed better than the traditional features. To include temporal behavior of the short-time features (23 ms frames), four summarized values of the power spectrum of each feature is found over a longer time frame (743 ms). In this manner, it is argued that temporal descriptors such as beat is included.

Tzanetakis and Cook [20] examined several features such as spectral centroid, MFCCs as well as a novel beat-histogram. Gaussian, GMM and KNN classifiers were used to classify music on different hierarchical levels such as e.g. classical music into choir, orchestra, piano and string quartet.

In the last two mentioned works, some effort was put into the examination of the time-scales of features and the decision time-horizon for classification. However, this generally seems to be a neglected area and has been the motivation for the current paper. How much time is, for instance, needed to make a sufficiently accurate decision about the musical genre? This might be important in e.g. hearing aids and streaming media. Often, some kind of early information fusion of the short-time features is achieved by e.g. taking the mean or another statistics over a larger window. Are the best features then the same on all time-scales or does it depend on the decision time horizon? Is there an advantage of early information fusion as compared to late information fusion such as e.g. majority voting among short-time classifications, see further e.g., [9]. These are the main questions to be addressed in the following.

In section 2 the examined features will be described. Section 3 deals with the methods for extracting information about the time scale behavior of the features, and in section 4 the results are presented. Finally, section 5 state the main conclusions.

2. FEATURE EXTRACTION

Feature extraction is the process of capturing the complex structure in a signal using as few features as possible. In the case of timbral textual features a frame size, in which the signal statistics are assumed stationary is analyzed and features are extracted. All

[1] This term refers to the decision making, i.e., early information fusion is an operation on the features *before* classification (and decision making). This is opposed to late information fusion (decision fusion) that assembles the information on the basis of the decisions.

features described below are derived from short-time 30 ms audio signal frames with a hop-size of 10 ms.

One of the main challenges when designing music information retrieval systems is to find the most descriptive features of the system. If good features are selected one can relax on the classification methodology for fixed performance criteria.

2.1 Spectral signal features

The spectral features have all been calculated using a Hamming window for the *short time Fourier transform* (STFT) to minimize the side-lobes of the spectrum.

MFCC and LPC. The MFCC and LPC both originate from the field of automatic speech recognition, which has been a major research area through several decades. They are carefully described in this context in the textbook by Rabiner and Juang [16]. Additionally, the usability of MFCCs in music modeling has been examined in the work of Logan [14]. The idea of MFCCs is to capture the short-time spectrum in accordance with human perception. The coefficients are found by first taking the logarithm of the STFT and then performing a mel-scaling which is supposed to group and smooth the coefficients according to perception. At last, the coefficients are decorrelated with the discrete cosine transform which can be seen as a computationally cheap PCA. LPCs are a short-time measure where the coefficients are found from modeling the sound signal with an all-pole filter. The coefficients minimizes a least-square measure and the LPC gain is the residual of this minimization. In this project, the autocorrelation method was used. The delta MFCC (DMFCC $\equiv$ MFCC$_n$ - MFCC$_{n-1}$) and delta LPC (DLPC $\equiv$ LPC$_n$ - LPC$_{n-1}$) coefficients are further included in the investigations.

MPEG-7 audio spectrum envelope (ASE). The *audio spectrum envelope* is a description of the power contents in log-spaced frequency bands of the audio signal. The log-spacing is done as to resemble the human auditorial system. The ASE have been used in e.g. audio thumbnailing and classification, see [21] and [5]. The frequency bands are determined using an $1/4$-octave between a lower frequency of 125 Hz, which is the "low edge" and a high frequency of 9514 Hz.

MPEG-7 audio spectrum centroid (ASC). The *audio spectrum centroid* describes the center of gravity of the log-frequency power spectrum. The descriptor indicates whether the power spectrum is dominated by low or high frequencies. The centroid is correlated with the perceptual dimension of timbre named *sharpness*.

MPEG-7 audio spectrum spread (ASS) . The *audio spectrum spread* describes the second moment of the log-frequency power spectrum. It indicates if the power is concentrated near the centroid, or if it is spread out in the spectrum. It is able to differentiate between tone-like and noise-like sounds [8].

MPEG-7 spectral flatness measure (SFM). The *audio spectrum flatness measure* describes the flatness properties of the spectrum of an audio signal within a number of frequency bands. The SFM feature expresses the deviation of a signal's power spectrum over frequency from a flat shape (noise-like or impulse-like signals). A high deviation from a flat shape might indicate the presence of tonal components. The spectral flatness analysis is calculated for the same number of frequency bands as for the ASE, except that the low-edge frequency is 250 Hz. The SFM seem to be very robust towards distortions in the audio signal, such as MPEG-$1/2$ layer 3 compression, cropping and dynamic range compression [1]. In [4] the centroid, spread and SFM have been evaluated in a classification setup.

All MPEG-7 features have been extracted in accordance with the MPEG-7 audio standard [8].

2.2 Temporal signal features

The temporal features have been calculated on the same frame basis as the spectral features.

Zero crossing rate (ZCR). ZCR measures the number of time domain zero-crossings in the frame. It can be seen as a descriptor of the dominant frequency of music and to find silent frames.

Short time energy (STE). This is simply the mean square power in the frame.

3. FEATURE RANKING - SENSITIVITY MAPS

3.1 Time stacking and dynamic PCA

To investigate the importance of the features at different time scales a tapped-delay line of time stacking features is used. Define an extended feature vector as

$$\mathbf{z}_n = \left[\mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{x}_{n-2}, \ldots, \mathbf{x}_{n-L}\right]^T,$$

where L is the lag-parameter and $\mathbf{x}_n$ is the row feature vector at frame n. Since the extended vector increases in size as a function of L, the data is projected into a lower dimension using PCA. The above procedure is also known as dynamic PCA (DPCA) [10] and reveals if there is any linear relationship between e.g. $\mathbf{x}_n$ and $\mathbf{x}_{n-1}$; thus not only correlations but also cross-correlations between features. The decorrelation performed by the PCA will also include a decorrelation of the time information, e.g. is MFFC-1 at time n correlated with LPC-1 at time $n-5$?

At $L = 100$ the number of features will be 10403 which makes the PCA computational intractable due to memory and speed. A "simple" PCA have been used where only 1500 of the total of 10403 largest eigenvectors is calculated by random selection of training data, see e.g. [19]. To investigate the validity of the method 200 eigenvectors was used at $L = 50$ and the number of random selected data points was varied between $200 - 1500$. The variation in classification error was less than a percent, thus indicating that this is a robust method. Due to memory problems originating from the time stacking, the largest used lag time is $L = 100$, which corresponds to one second of the signal.

3.2 Feature ranking

One of the goals of this project is to investigate which features are relevant to the classification of music genres at different time scales. Selection of single best method for feature ranking is not possible, since several methods exists each with their advantages and disadvantages. An introduction to feature selection can be found in [7], which also explains some of the problems using different ranking schemes. Due to the nature of our problem a method known as the *sensitivity map* is used, see e.g. [18]. The influence of each feature on the classification bounds is found by computing the gradient of the posterior class probability $P(C_k|\mathbf{x})$ w.r.t. all the features. Here C_k denotes the k'th genre. One way of computing a sensitivity map for a given system is the *absolute value average sensitivities* [18]

$$\mathbf{s} = \frac{1}{NK}\sum_{k=1}^{K}\sum_{n=1}^{N}\left|\frac{\partial P(C_k|\tilde{\mathbf{x}}_n)}{\partial \mathbf{x}_n}\right|, \qquad (1)$$

where $\mathbf{x}_n$ is the n'th time frame of a test-set and $\tilde{\mathbf{x}}_n$ is the n'th time frame of the same test-set projected into the M largest eigenvectors of the training-set. Both $\mathbf{s}$ and $\mathbf{x}_n$ are vectors of length D - the number of features. N is the total number of test frames and K is the number of genres. Averaging is performed over the different classes as to achieve an overall ranking independent of the class. It should be noted that the sensitivity map expresses the importance of each feature individually - correlations are thus neglected.

For the linear neural network an estimate of the posterior distribution is needed to use the sensitivity measure. This is achieved using the softmax-function, see e.g. [18].

4. RESULTS

Two different classifiers were used in the experiments: a Gaussian classifier with full covariance matrix and a simple single-layer neural network which was trained with sum-of-squares error function to facilitate the training procedure. These classifiers are quite similar, but they differ in the discriminant functions which are quadratic

and linear, respectively. Furthermore the NN is inherently trained discriminatively. They are also quite simple, but after experimentation with more advanced methods, like the Gaussian mixture models and HMMs, this became a necessity in order to carry out the vast amount of training operations needed. Further, the purpose of this study is not to obtain optimal performance rather to investigate the relevance of relevant short-time features.

The data set was split into training, validation and test sets. The validation set was used only to select the number of DPCA-components. The best classification was found with 50 components at both $L = 50$ and $L = 100$. The data was split with 50, 25 and 25 sound files in each set, respectively, and each of these were distributed evenly into five music genres: Pop, Classical, Rock, Jazz, Techno. All sound files have a duration of 10 s and with a hop-size of 10 ms. This resulted in 1000 30 ms frames per sound file. The used sampling frequency is 22050 Hz. The size of the training set as well as duration of the sound files was determined from learning curves[2] (results not shown). After the feature extraction, the features were normalized to zero mean and unit variance to make them comparable.

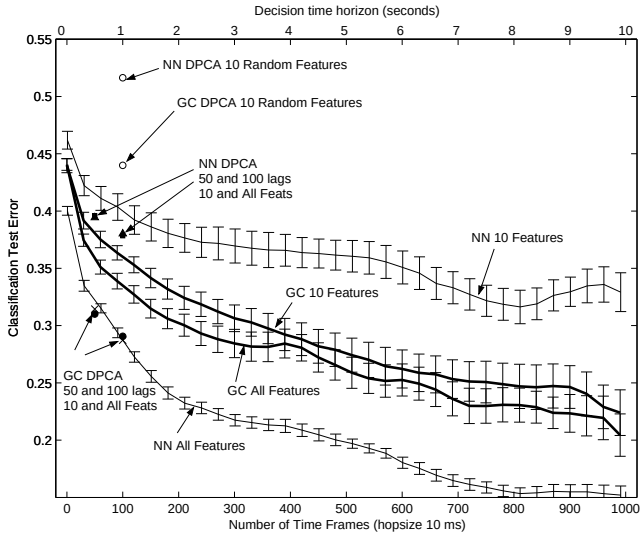


Figure 1: Classification error as a function of the lag of the GC and NN using DPCA and majority voting, respectively.

Figure 1 summarizes the examination of the decision time horizon as well as the comparison between early and late information fusion using DPCA and majority voting, respectively. It is seen from the figure that there is not an obvious advantage of using the DPCA transform instead of the computationally much cheaper majority voting. However, it can be seen from table 1 and 2 that the methods' performance depends on the genre. The tables show test classification error for each genre with error-bars obtained by repeating the experiment 50 times on randomly selected training data. The number in parenthesis shows the percentage relative to lag $L = 0$ of the classifier. For instance, it is seen that the DPCA gives remarkably better classification of jazz than majority voting. This might be used constructively to create a better classifier.

Figure 1 also shows the results after choosing the 10 features with the best sensitivity consensus ranks (see below). There is a small deviation for the GC and a large deviation for the NN between the 10 best features and the full feature set when majority voting is used. This might be connected to the differences in the number of variables in the two classifiers which implies that the curve for the NN with 10 features is dominated by bias since the number of variables is only $5 \cdot 11 = 55$. Thus, 10 features is not really enough for this classifier. In contrast, the GC with 103 features has more than 25000 different variables and might be dominated by variance which increases the test error. However, the sensitivity ranking still seems reasonable when compared to the full feature sets and when comparisons are made with the classification error from a set of 10 random features (illustrated in the figure).

Another examination of early information fusion was also carried out by using the mean values of the short-time features over increasing time frames (from 1 to 1000 frames). The classification results are not illustrated, however, since approximately the same classification rate as without the time information (lag $L = 0$) was achieved at all time scales, though with a lot of fluctuations.

Full Feature Set	**Pop**	**Classic**	**Rock**	**Jazz**	**Techno**
NN (L=0)	36% ± 0.8%	27% ± 2%	29% ± 1.1%	67% ± 1.1%	41% ± 0.7%
Maj.Vote (L=100)	17%(−19)	19%(−8)	26%(−3)	63%(−4)	29%(−12)
Time Stacking (L=100)	21%(−15)	22%(−5)	21%(−8)	45%(−22)	34%(−7)
GC (L=0)	50% ± 0.2%	39% ± 0.5%	27% ± 0.2%	71% ± 0.5%	31% ± 0.3%
Maj.Vote (L=100)	32%(−18)	28%(−11)	22%(−5)	68%(−3)	17%(−14)
Time Stacking (L=100)	28%(−22)	29%(−10)	21%(−6)	39%(−32)	26%(−5)

Table 1: Test error classificstion rates of Gaussian Classifier (GC) and Neural Network (NN) using the full feature set.

Best 10 Feat.	**Pop**	**Classic**	**Rock**	**Jazz**	**Techno**
NN (L=0)	38% ± 1.4%	30% ± 2.5%	40% ± 2.1%	86% ± 1.4%	37% ± 0.96%
Maj.Vote (L=100)	27%(−11)	23%(−7)	38%(−2)	88%(+2)	25%(−12)
Time Stacking (L=100)	21%(−17)	23%(−7)	45%(+5)	65%(−21)	37%(0)
GC (L=0)	34% ± 0.6%	35% ± 1.5%	38% ± 1.4%	65% ± 1.2%	47% ± 0.8%
Maj.Vote (L=100)	22%(−12)	26%(−9)	32%(−6)	62%(−3)	39%(−8)
Time Stacking (L=100)	36%(+2)	32%(−3)	22%(−16)	43%(−22)	12%(−35)

Table 2: Test error classification rates of Gaussian Classifier (GC) and Neural Network (NN) using the 10 best features.

The training of the models has been repeated 50 times on different song clips, and the sensitivities have been calculated and ranked. It is now possible to obtain a consensus ranking from the cumulated sensitivity histograms of the 103 features, which is shown in figure 2. Each row shows the cumulated sensitivity histogram where dark color corresponds to large probability. For $L = 0$ the number of features is $D = 103$, but for $L = 100$ the amount of features is $D = 10403$ due to the time stacking. A similar plot could be generated at $L = 100$ but the histograms of each feature would not be easy to see and interpret. To rank the features, at e.g. $L = 100$, the mean value of the sensitivity over time of each feature is applied, which results in only 103 time-averaged features in figure 2. The mean value is applied since only low frequency variation in sensitivity over lag-parameters are present (below 5 Hz). To provide the consensus features, the feature which has the highest cumulated histogram frequency in each column is selected.

Experiments with ranking of the features at $L = \{0, 50, 100\}$ clearly indicates that delta features generally ranks lower at higher lag time, see also area **B** and **D** in figure 2 for $L = 100$. The MFCC(**A**) and LPC(**C**) generally rank better than e.g. the ASE(**E**) and SFM(**F**) coefficients. However, the high frequency components of both the ASE and SFM also show relevance, which is an indicator of "noise-like" parts in the music. The 10 best consensus features for $L = \{0, 50, 100\}$ are shown in table 3. A sanity check of the sen-

[2] Classification error or log-likelihood as a function of the size of the training set.

sitivity map was performed using the Optimal Brain Damage [3] for $L = 0$ and showed similar results.

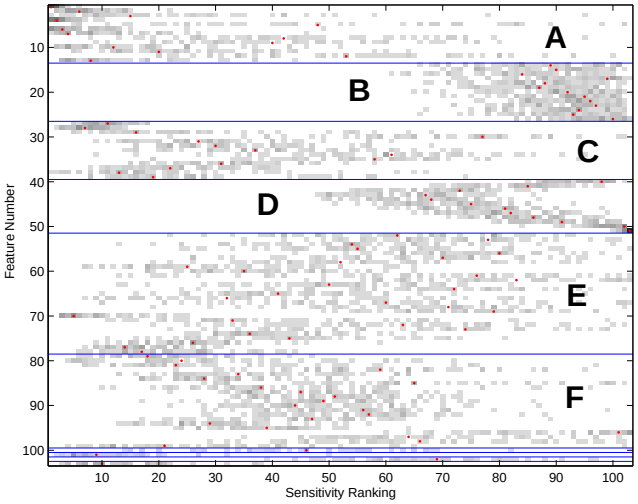


Figure 2: Consensus feature ranking of individual feature at $L = 100$. See text for interpretation. The features are MFCC(**A**), DMFCC(**B**), LPC(**C**), DLPC(**D**), ASE(**E**), SFM(**F**) and the single features ASC, ASS, STE and ZCR. The ten best features in decreasing order are: $\{1, 4, 6, 7, 70, 2, 28, 13, 101, 103\}$.

L=0 (1 to 5)	LPC2	LPC1	MFCC2	LPC3	MFCC4
L=50 (1 to 5)	MFCC1	MFCC4	MFCC6	MFCC2	LPC2
L=100 (1 to 5)	MFCC1	MFCC4	MFCC6	MFCC7	ASE19
L=0 (6 to 10)	LPC4	LPC5	GAIN	MFCC1	MFCC3
L=50 (6 to 10)	MFCC7	ASE19	LPC1	ASS	MFCC10
L=100 (6 to 10)	MFCC2	LPC2	MFCC13	ASS	ZCR

Table 3: The 10 best consensus features of the NN classifier as a function of the time stack lag, L. The DPCA transform was employed.

5. CONCLUSION

Music genre classification has been explored with special emphasis on the decision time horizon and ranking of tapped-delay line short-time features. A linear neural network and a Gaussian classifier were used for classification. Information fusion showed increasing performance with time horizon, thus state-of-art 80% correct classification rate is obtained within 5 s decision time horizon. Early and late information fusion showed similar results, thus we recommend the computational efficient majority decision voting. However, investigation of individual genres showed that e.g. jazz is better classified using DPCA. Consensus ranking of feature sensitivities enabled the selection and interpretation of the most salient features. MFCC, LPC and ZCR showed to be most relevant, whereas MPEG-7 features showed less consistent relevance. DMFCC and DLPC showed to be least important for the classification. With only the 10 best features, 70% classification accuracy was obtained using a 5 s decision time horizon.

Acknowledgment

The work is supported by the European Commission through the sixth framework IST Network of Excellence: Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL), contract no. 506778.

REFERENCES

[1] E. Allamanche, J. Herre, O. Helmuth, B. Frba, T. Kasten, and M. Cremer, "Content-Based Identification of Audio Material Using MPEG-7 Low Level Description," in *Proc. of the ISMIR*, Indiana University, USA, Oct. 2001.

[2] J.-J. Aucouturier and F. Pachet, "Representing musical genre: A state of the art," *Journal of New Music Research*, vol. 32, pp. 83–93, Jan. 2003.

[3] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press, 1995.

[4] J.J. Burred and A. Lerch, "A Hierarchical Approach to automatic musical genre classification," in *Proc. 6th Int. Conf. on Digital Audio Effects '03*, London, Great Britain, Sept. 2003.

[5] M. Casey, "Sound Classification and Similarity Tools," in B.S. Manjunath, P. Salembier and T. Sikora (eds), *Introduction to MPEG-7: Multimedia Content Description Language*, J.Wiley, 2001.

[6] 2004 International Consumer Electronics Show, Las Vegas, Nevada, Jan. 8–11, 2004, www.cesweb.org

[7] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, Mar. 2003.

[8] *Information technology Multimedia content description interface - Part 4: Audio*, ISO/IEC FDIS 15938-4:2002(E) Retrieval (ISMIR 2003), Baltimore, Oct. 2003, www.chiariglione.org/mpeg/.

[9] J. Kittler, M. Hatef, R.P.W. Duin and J. Matas, "On Combining Classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 226–239, Mar. 1998.

[10] W. Ku, R.H. Storer, C. Georgakis, "Disturbance Detection and Isolation by Dynamic Principal Component Analysis", *Chemometrics and Intell Lab Sys.*, pp. 179–196, Sept. 1995.

[11] T. Lambrou *et al.*, "Classification of Audio Signals using Statistical Features on Time and Wavelet Transform Domains," in *Proc. ICASSP '98* , Seattle, USA, May 1998, pp. 3621–3624.

[12] D. Li *et al.*, "Classification of General Audio Data for Content-Based Retrieval," *Pattern Recognition Letters*, vol. 22, pp. 533–544, Apr. 2001.

[13] T. Li and M. Ogihara and Q. Li, "A comparative study on content-based music genre classification," in *Proc. ACM SIGIR '03*, Toronto, Canada, July 2003, pp. 282–289.

[14] B. Logan, "Mel Frequency Cepstral Coefficients for Music Modeling," in *Proc. of the International Symposium on Music Information Retrieval 2000*, Plymouth, USA, Oct. 2000.

[15] M.F. McKinney and J. Breebaart, "Features for Audio and Music Classification," in *4th International Conference on Music Information*, http://ismir2003.ismir.net/papers/McKinney.PDF

[16] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[17] E. Scheirer and M. Slaney, "Construction and Evaluation of a Robust Multifeature Speech/Music Discriminator," in *Proc. ICASSP '97*, Munich, Germany, 1997, pp. 1331–1334.

[18] S. Sigurdsson *et al.*, "Detection of Skin Cancer by Classification of Raman Spectra," accepted for *IEEE Transactions on Biomedical Engineering, 2003*.

[19] H. Schweitzer, "A Distributed Algorithm for Content Based Indexing of Images by Projections on Ritz Primary Images," *Data Mining and Knowledge Discovery 1*, pp. 375-390, 1997.

[20] G. Tzanetakis and P. Cook, "Musical Genre Classification of Audio Signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, pp. 293–302, July 2002.

[21] J. Wellhausen and M. Höynck, "Audio Thumbnailing Using MPEG-7 Low Level Audio Descriptors," in *Proc. ITCom '03*, Orlando, USA , Sept. 2003.

[22] E. Wold, T. Blum, D. Keislar and J. Wheaton, "Content-based classification, search and retrieval of audio," *IEEE Multimedia Mag.*, vol. 3, pp. 27–36, July 1996.

[23] C. Xu *et al.*, "Musical Genre Classification using Support Vector Machines," in *Proc. ICASSP '03*, Hong Kong, China, Apr. 2003, pp. 429–432.

Appendix C

Improving Music Genre Classification by Short-Time Feature Integration

---

Meng A., Ahrendt P. and Larsen J., **Improving Music Genre Classification by Short-Time Feature Integration**, Proceedings of *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Philadelphia, USA, March 2005.

IMPROVING MUSIC GENRE CLASSIFICATION BY SHORT-TIME FEATURE INTEGRATION

Anders Meng, Peter Ahrendt and Jan Larsen

Informatics and Mathematical Modelling, Technical University of Denmark
Richard Petersens Plads, Building 321, DK-2800 Kongens Lyngby, Denmark
phone: (+45) 4525 3891,3888,3923, fax: (+45) 4587 2599, email: am,pa,jl@imm.dtu.dk, web: http://isp.imm.dtu.dk

ABSTRACT

Many different short-time features, using time windows in the size of 10-30 ms, have been proposed for music segmentation, retrieval and genre classification. However, often the available time frame of the music to make the actual decision or comparison (the decision time horizon) is in the range of seconds instead of milliseconds. The problem of making new features on the larger time scale from the short-time features (*feature integration*) has only received little attention. This paper investigates different methods for feature integration and late information fusion[1] for music genre classification. A new feature integration technique, the *AR* model, is proposed and seemingly outperforms the commonly used mean-variance features.

1. INTRODUCTION

Classification, segmentation and retrieval of music (and audio in general) are topics that have attracted quite some attention lately from both academic and commercial societies. These applications share the common need for features which effectively represent the music. The features ideally contain the information of the original signal, but compressed to such a degree that relatively low-dimensional classifiers or similarity metrics can be applied. Most efforts have been put in short-time features, which extract the information from a small sized window (often $10-30$ ms). However, often the decision time horizon is in the range of seconds and it is then necessary either to find features directly on this time scale or somehow integrate the information from the time series of short-time features over the larger time window. Additionally, it should be noted that in classification problems, the information fusion could also be placed after the actual classifications. Such late fusion could e.g. be majority voting between the classifications of each short-time feature.

In [1] and [2], features are calculated directly on the large time-scale (long-time features). They try to capture the perceptual beats in the music, which makes them intuitive and easy to test against a music corpora. In contrast, short-time features can only be tested indirectly through e.g. their performance in a classification task.

Feature integration is most often performed by taking the mean and variance of the short-time features over the decision time horizon (examples are [3], [4] and [5]). Computationally, the mean and variance features are cheap, but the question is how much of the relevant feature dynamics they are able to capture. As an attempt to capture the dynamics of the short-time features, [6] uses a spectral decomposition of the Mel-Frequency Cepstral Coefficients (*MFCCs*) into 4 different frequency bands. Another approach, by [7], takes the ratio of values above and below a constant times the mean as the long-time feature. Their short-time features are Zero-Crossing Rate and Short-Time Energy.

In a previous investigation [8], the authors examined feature integration by dynamic PCA where the idea is to stack short-time features over the decision time horizon and then use PCA to reduce the dimensionality (finding correlations both across time and features). Dynamic PCA was compared with late fusion in the form of majority voting, but the results did not strongly favor any of the methods.

Altogether, the idea of short-time feature integration seems scarcely investigated, although several researchers (necessarily) make use of it. This has been the main motivation for the current work, together with methods for late information fusion.

In Section 2, the investigated features and feature integration techniques are described. Section 3 concerns the employed classifiers and late information fusion schemes. In section 4, the results are analyzed and, finally, section 5 concludes on the results.

2. FEATURE MODEL

In this article the selected features exist either on a short, medium or long time scale. The timescales used can be seen from table 1. Short time only consider the immediate frequencies, and do not contain long structural temporal information. Medium time features can contain temporal information such as e.g. modulation (instrumentation) and long time features can contain structural information such as beat. Classification at short time only provide reasonable results using a computer, since human decision time horizons typically are 250ms or above for a moderate error [5].

Time scale	Frame size	Perceptual meaning
Short time	30ms	timbre (instant frequency)
Medium time	740ms	modulation (instrumentation)
Long time	9.62s	beat, mood vocal etc.

Table 1. The different time levels with corresponding perceptual interpretation.

[1] Late information fusion assemble the probabilistic output or decisions from a classifier over the short-time features (an example is majority voting). In early information fusion (which includes feature integration) the information is integrated before or in the classifier.

Depending on the decision time horizon, the performance at short time might not be adequate, in which more time is needed. There are several possibilities to increase the decision time horizon, either using the classifier in an early/late information fusion setting, which will be elaborated in section 3, or to use features derived at these time horizons. Figure 1 show the investigated features for the music genre setup and their relationships.

2.1. Short time features (1)

The short time features have been derived using a hop- and frame size of 10 and 30ms, respectively. Typically the frame size is selected such that the in-frame signal is approximately stationary.

Mel Frequency Cepstral Coefficients were originally developed for automatic speech recognition systems [9, 10], but have lately been used with success in various audio information retrieval tasks. Recent studies [8, 11] indicate that they outperform other features existing at a similar time level. From the previous investigations [8], good performance was achieved, hence, these are the only features considered at this decision time horizon. It was found that the first 6 *MFCCs* were adequate for the music genre classification task, in line with [5].

2.2. Medium time features (2)

The medium time features are based on a frame size of 740ms similar to [6] and a hop size of 370ms.

Mean and variance (*MV*) of the *MFCCs*. Mean and variance is a simple way to perform feature integration and the most commonly used, see e.g. [1, 3, 4].

Filterbank Coefficients (*FC*) is another method of feature integration. This method was proposed in [6] and suggests to calculate the power spectrum for each *MFCC* on a frame size of 740ms. The power is summarized in four frequency bands: 1) 0 Hz average of *MFCCs*, 2) $1-2$ Hz modulation energy of the *MFCCs*, 3) 3-15Hz and 4) 20-50 Hz (50Hz is half the sampling rate of the *MFCCs*). Experiments suggested that better performance could be achieved using more than 4 bins, which seems reasonable since these features was originally developed for general sound recognition.

Autoregressive model (*AR*) is a well-known technique for time series regression. Due to its simplicity and good performance in time-series modelling, see e.g. [12], this model is suggested for feature integration of the *MFCCs*. The *AR* method and *FC* approach resembles each other since the integrated ratio of the signal spectrum to the estimated spectrum is minimized in the *AR* method [13]. This suggests that the power spectrum of each *MFCC* is modelled. The *AR* parameters have been calculated using the windowed autocorrelation method, using a rectangular window. To the authors knowledge an *AR*-model has not previously been used for music feature integration. In all of the AR-related features, the mean and gain are always included along with a number of AR-coefficients. This number is given by the model order, which is found by minimizing validation classification error on the data set.

High Zero-Crossing Rate Ratio (*HZCRR*) is defined as the ratio of the number of frames whose time zero crossing rates (No. of times the audio signal crosses 0) are above 1.5 times the average.

Low Short-Time energy ratio (*LSTER*) is defined as the ratio of the number of frames whose short time energy is less than 0.5 times the average.

Both the *LSTER* and *HZCRR* features are explained further in [7]. They are derived directly from the audio signal, which makes them computationally cheap. It should be mentioned that the *HZCRR* and *LSTER* were originally meant for speech/music segmentation. In the experiments, they were combined into the feature *LSHZ* to improve their performance.

2.3. Long time features (3)

All the long time features have a hop- and frame size of 4.81 and 9.62 seconds, respectively. Many of the features at this decision time have been derived from features at an earlier timescale (feature integration), e.g. AR_{23a} is integrated from medium time to long time using an *AR* model on each of the *AR* medium time features. The different combinations applied can be seen from figure 1, where the arrows indicate which features are integrated to a longer time scale. Additionally, all the long-time features have been combined into the feature, *All*, and PCA was used for dimensionality reduction.

Beat spectrum (*BS*) has been proposed by [2] as a method to determine the perceptual beat. The *MFCCs* are used in the beat spectrum calculation. To calculate the frame similarity matrix, the cosine measure has been applied. The beat spectrum displays peaks when the audio has repetitions. In the implementation the discrete fourier transform is applied to the beat spectrum in order to extract the main beat and sub beats. The power spectrum is then aggregated in 6 discriminating bins wrt. music genre.

Beat histogram (*BH*) was proposed in [1] as a method for calculating the main beat as well as sub-beats. The implementation details can be found in [1]. In our implementation the discrete wavelet transform is not utilized, but instead an octave frequency spacing has been used. The resulting beat histogram is aggregated in 6 discriminating bins.

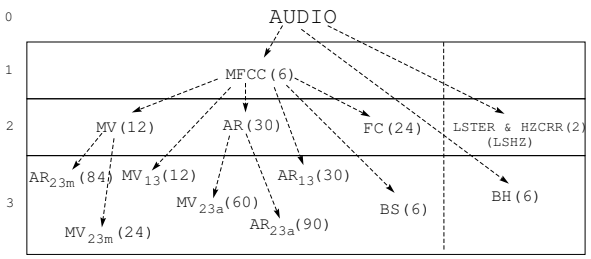


Fig. 1. Short(1), medium(2) and long(3) time features and their relationships. The arrow from e.g. medium time *MV* to the long time feature AR_{23m} indicate feature integration. Thus, for each of the 12 time-series of *MV* coefficients, 7 *AR* features have been found, resulting in a $7 \cdot 12 = 84$ dimensional feature vector AR_{23m}. The optimal feature dimension (shown in parenthesis) for the various features have been determined from a validation set, hence selecting the dimension which minimizes the validation error.

3. CLASSIFIERS AND COMBINATION SCHEMES

For classification purposes two classifiers were considered: 1) A simple single-layer neural network (LNN) trained with sum-of -squares error function to facilitate the training procedure and 2) A gaussian classifier (GC) with full covariance matrix. The two

classifiers differ in their discriminant functions which are linear and quadratic, respectively. Furthermore the LNN is inherently trained discriminatively. More sophisticated methods could have been used for classification, however, the main topic of this research was to investigate methods of information fusion in which the proposed classifiers will suffice.

The two fusion schemes considered were early and late information fusion. In early information fusion the complex interactions that exist between features in time is modelled in or before the statistical classification model. The feature integration techniques previously mentioned (such as the *AR*, *FC*, AR_{23a} and MV_{13} features) can be considered as early fusion. Late information fusion is the method of combining results provided from the classifier. There exists several combination schemes for late information fusion, see e.g. [14]. In the present work, the majority vote rule, sum rule and the median rule were investigated. In the majority vote rule, the votes received from the classifier are counted and the class with the largest amount of votes is selected, hereby performing consensus decision. In sum-rule the posterior probabilities calculated from each example are summed and a decision is based on this result. The median rule is like the sum rule except being the median instead of the sum. During the initial studies it was found that the sum rule outperformed the majority voting and median rule, consistent with [14], and therefore preferred for late information fusion in all of the experiments.

4. RESULTS AND DISCUSSION

Experiments were carried out on two different data sets. The purpose was not so much to find the actual test error on the data sets, but to compare the relative performances of the features.

For some of the features, dimensionality reduction by PCA was performed. Learning curves, which are plots of the test error as a function of the size of the training set, were made for all features. From these curves, it was found necessary to use PCA on AR_{23a}, AR_{23m}, MV_{23a} and the combined long-time features set (denoted All). It was found that approximately 20 principal components gave optimal results.

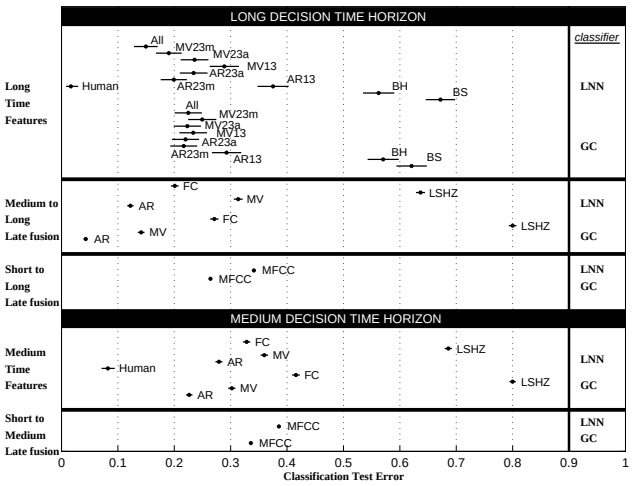
The classification test errors are shown in figure 2 for both of the data sets and both the medium time and long time classification problems.

4.1. Data set 1

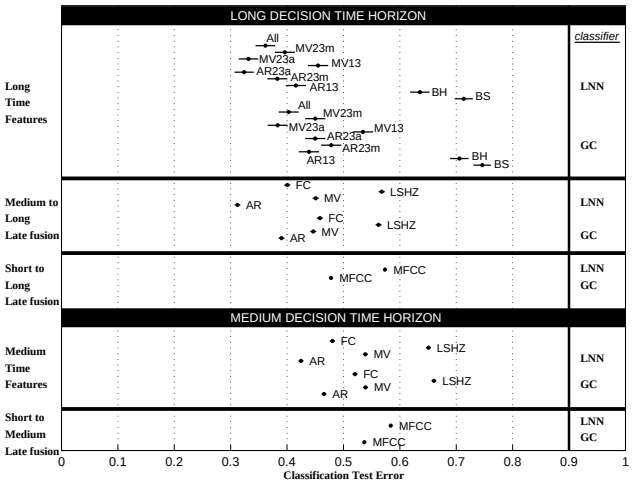
The data set consisted of the same 100 songs, that were also used in [8]. The songs were distributed evenly among classical, (hard) rock, jazz, pop and techno. The test set was fixed with 5 songs from each genre and using 30 seconds from the middle of the songs. The training set consisted of three pieces each of 30 seconds from each song, resulting in 45 pieces. For cross-validation, 35 of these pieces were picked randomly for each of the 10 training runs.

4.1.1. Human classification

To test the integrity of the music database, a human classification experiment was carried out on the data set. 22 persons were asked each to classify (by forced-choice) 100 of the 740 ms and 30 of 10 s samples from the test set. The average classification rate across people and across samples was 98% for the 10 s test and 92% for the 740 ms test. The lower/upper 95% confidence limits were



(a) Experiment on data set 1



(b) Experiment on data set 2

Fig. 2. The figure illustrates the classification test errors for data set 1 in the upper part and data set 2 in the lower. Each part contains test errors from both the long decision time horizon (10 s) and the medium decision time horizon (740 ms). Thus, the block "Medium to Long Late Fusion" under "Long Decision Time Horizon" include all the medium-time features, such as *AR* and *FC* features, where the sum rule has been used to fuse information from the medium to long time scale. The results for the same medium-time features without any late fusion, would then be placed in "Medium Time Features" under "Medium Decision Time Horizon". The results from both classifiers on the same features are placed in the same block (GC is Gaussian Classifier, LNN is Linear Neural Network). All the abbreviations of the features are explained in section 2. The 95%- confidence intervals have been shown for all features.

97/99% and 91/93%, respectively. This suggests that the genre labels, that the authors used, are in good agreement with the common genre definition.

4.2. Data set 2

The data set consisted of 354 music samples each of length 30 seconds from the "Amazon.com Free-Downloads" database [15]. The songs were classified evenly into the six genres classical, country, jazz, rap, rock and techno and the samples were split into 49 for training and 10 for testing. From the training samples, 45 were randomly chosen in each of the 10 cross-validation runs. The authors found it much harder to classify the samples in this data set than in the previous, but it is also considered as a much more realistic representation of an individuals personal music collection.

4.3. Discussion

Notably, as seen in figure 2, the feature *LSHZ*, *BS* and *BH* perform worse than the rest of the features on both data sets. This may not be surprising since they were developed for other problems than music classification and/or they were meant as only part of a larger set of features. The *FC* did not do as well as the *AR* features. A small investigation indicated that *FCs* have the potential to perform better by changing the number of frequency bins, though still not as good as *ARs*.

A careful analysis of the *MV* and *AR* features, and the feature integration combinations of these, has been made. By comparing the early fusion combinations of these, as seen in figure 2 (in the part "Long-time features"), it is quite unclear which of these perform the best. When the late fusion method is used (in the part "Medium to long late fusion"), the results are more clear and it seems that the *AR* feature performs better than the *MV* and *FC* features. This view is supported by the results in the "Medium-time features" part. Using the McNemar-test, it was additionally found that the results from the *AR* feature differ from the *MV* and *FC* features on a 1% significance level.

The late fusion of the *MFCC* features directly did not perform very well compared to the *MV* and *AR* features. This indicates the necessity of feature integration up to at least a certain time scale before applying a late fusion method.

5. CONCLUSION

The problem of music genre classification addresses many problems and one of these being the identification of useful features. Many short-time features have been proposed in the literature, but only few features have been proposed for longer time scales.

In the current paper, a careful analysis of feature integration and late information fusion has been made with the purpose of music genre classification on longer decision time horizons. Two different data sets were used in combinations with two different classifiers. Additionally, one of the data sets were manually classified in a listening test involving 22 test persons to test the integrity of the data set.

A new feature integration technique, the *AR* model, has been proposed as an alternative to the dominating mean-variance feature integration. Different combinations of the *AR* model and the mean-variance model have been tested, both based on the *MFCC* features. The *AR* model is slightly more computationally demanding, but performs significantly better on the tested data sets. A particularly good result was found with the three-step information fusion of first calculating *MFCC* features, then integrating with the *AR* model and finally using the late fusion technique *sum rule*. This combination gave a classification test error of only 5% on data set 1, as compared to the human classification error of 3%.

6. ACKNOWLEDGEMENTS

The work is supported by the European Commission through the sixth framework IST Network of Excellence: Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL), contract no. 506778.

7. REFERENCES

[1] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, July 2002.

[2] J. Foote and S. Uchihashi, "The beat spectrum: A new approach to rhythm analysis," *Proc. International Conference on Multimedia and Expo (ICME)*, pp. 1088–1091, 2001.

[3] S. H. Srinivasan and M. Kankanhalli, "Harmonicity and dynamics-based features for audio," in *IEEE Proc. of ICASSP*, May 2004, vol. 4, pp. 321–324.

[4] Y. Zhang and J. Zhou, "Audio segmentation based on multi-scale audio classification," in *IEEE Proc. of ICASSP*, May 2004, pp. 349–352.

[5] G. Tzanetakis, *Manipulation, Analysis and Retrieval Systems for Audio Signals*, Ph.D. thesis, Faculty of Princeton University, Department of Computer Science, 2002.

[6] M. F. McKinney and J. Breebaart, "Features for audio and music classification," in *Proc. of ISMIR*, 2003, pp. 151–158.

[7] L. Lu, H-J. Zhang, and H. Jiang, "Content analysis for audio classification and segmentation," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 7, pp. 504–516, October 2002.

[8] P. Ahrendt, A. Meng, and J. Larsen, "Decision time horizon for music genre classification using short-time features," in *Proc. of EUSIPCO*, 2004, pp. 1293–1296.

[9] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP, no. 28, pp. 357–366, August 1980.

[10] C. R. Jankowski, H.-D. Vo, and R. P. Lippmann, "A comparison of signal processing front ends for automatic word recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 3(4), pp. 286–293, 1995.

[11] Kim H.-Gook. and T. Sikora, "Audio spectrum projection based on several basis decomposition algorithms applied to general sound recognition and audio segmentation," in *Proc. of EUSIPCO*, 2004, pp. 1047–1050.

[12] A. C. Harvey, *Forecasting, structural time series models and the Kalman filter*, Cambridge University Press, 1994.

[13] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, 1975.

[14] J. Kittler, M. Hatef, Robert P.W. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[15] *www.amazon.com*, "Free-downloads section," 2004.

Appendix D

Co-occurrence Models in Music Genre Classification

Ahrendt P., Goutte C., Larsen J., **Co-occurrence Models in Music Genre Classification**, Proceedings of *IEEE International Workshop on Machine Learning for Signal Processing* (MLSP), Mystic, Connecticut, USA, September 2005.

CO-OCCURRENCE MODELS IN MUSIC GENRE CLASSIFICATION

Peter Ahrendt[*]*, Jan Larsen*

Informatics and Mathematical Modelling
Technical University of Denmark
2800 Kongens Lyngby, Denmark
pa,jl@imm.dtu.dk

Cyril Goutte

Xerox Research Centre Europe
6 ch. de Maupertuis
F-38240 Meylan, France
cyril.goutte@xrce.xerox.com

ABSTRACT

Music genre classification has been investigated using many different methods, but most of them build on probabilistic models of feature vectors x_r which only represent the short time segment with index r of the song. Here, three different co-occurrence models are proposed which instead consider the whole song as an integrated part of the probabilistic model. This was achieved by considering a song as a set of independent co-occurrences (s, x_r) (s is the song index) instead of just a set of independent (x_r)'s. The models were tested against two baseline classification methods on a difficult 11 genre data set with a variety of modern music. The basis was a so-called AR feature representation of the music. Besides the benefit of having proper probabilistic models of the whole song, the lowest classification test errors were found using one of the proposed models.

1. INTRODUCTION

In these years, the growth in digital music on the Internet is tremendous. Several companies now offer music for on-line sale, such as iTunes with more than 800,000 song tracks available. Besides, radio channels and TV broadcasting companies have started offering their services and the demand for efficient information retrieval in these streams is obvious. An important part in this is *music genre classification*, which will be addressed here. The general idea is to extract features from (most often) short frames of the digitized sound signal. A classifier then use this time sequence of features to classify the song[1] into genres such as jazz, pop and blues. Several researchers have contributed to this field, such as [1], [2], and [3].

In the current work, music genre classification will be addressed with a *co-occurrence model*. In this novel view, a song is seen as a set of co-occurrences between a song and its constituent *sound elements* which represent segments in time of the song. The inspiration to use this model came from the area of information retrieval, where the method of Probabilistic Latent Semantic Analysis (PLSA) [4] has shown to be very powerful in e.g. automated document indexing. In PLSA, the co-occurrences are between a document and words in the document where the words are elements of a discrete, finite vocabulary.

Analogies between music and textual language can be found on many levels and both can be seen to contain a notion of grammar, syntax and semantics [5]. [6] shows that the frequencies of the usage of different notes in musical compositions follow Zipf's law, which is also known to apply to word frequencies in documents. Zipf's law is said to apply if $f = k \cdot r^{-b}$, where f is the frequency, k is some constant, r is the rank (of the frequencies) and b is a constant that should be close to 1. These previous findings support the usage of the co-occurrence model in music genre classification, but extracting the note and instrument composition directly and correctly from general digitized music is still an open problem.

For this reason, experiments have been made with several different approaches to represent the equivalents of words in music (the sound elements). Section 2 discuss the so-called *AR features*, which give the basic (30 dimensional) feature space representation of the music. This feature space is seen as the ground on which to build different word equivalents. Section 3 first gives the formalism and theory of the co-occurrence model and PLSA. Afterwards, discrete and continuous vocabulary models are described. Section 4 presents the results using these models and section 5 concludes and outlines future perspectives.

2. MUSIC FEATURES

Many different features have been proposed to represent music, however, this work only use the so-called AR features due to the good results in music genre classification as reported in [7], where they were first proposed. Calcu-

[*]The author performed the work while at Xerox Research Centre Europe.

[1]The quantity to classify is often a whole song, but could be a sound clip of varying length. In the following, the quantity will simply be called a song.

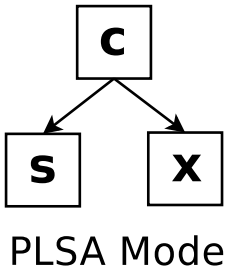


Fig. 1. The graphical model used in Probabilistic Latent Semantic Analysis (PLSA). This is also called the Aspect Model. Squares represent discrete variables.

lating the AR features is a 2-step procedure. First, the mel-frequency cepstral coefficients (MFCC) are calculated on small sound segments (here 30 ms). The MFCC features are very well-known in both speech and music processing, however, they represent only very short sound segments. Thus, the next step is to model the time sequence of the MFCC features individually as AR (auto-regressive) processes and use the AR coefficients as features. Together with the residuals from the AR models and the mean of the MFCCs, this gives the AR features which can now represent much larger sound segments than the MFCCs (here 760 ms).

3. CO-OCCURRENCE MODELS

Co-occurrence models regard a song as a set of co-occurrences (s, x_r) where s denotes the song label and x_r is some feature x at index r in the song. This implies that the song can be modelled directly into the probabilistic model as opposed to previous music genre classification methods.

One advantage of this framework is that a probabilistic measure of $p(c|s)$ can be found, where c denotes the genre label and s is the song index of the new song to be classified. Traditional approaches ([2], [8]) only model $p(c|x_r)$ or $p(x_r|c)$ and combine this information to take a decision for the entire song s. Combination techniques include Majority Voting and the Sum-rule method. With Majority Voting the quantity of interest would be the vote

$$\Delta_r = \arg\max_c p(c|x_r) \qquad r = 1, \ldots, N_r \tag{1}$$

for each of the N_r time frames in the new song and the genre label of the whole song is chosen as the genre with the most votes. The Sum-rule use the quantity

$$\hat{C} = \arg\max_c \sum_r p(c|x_r) \qquad r = 1, \ldots, N_r \tag{2}$$

directly as the estimate of the genre label.

3.1. PLSA and Folding-in

The graphical model used in Probabilistic Latent Semantic Analysis (PLSA) is illustrated in fig. 1 and the original formulation will be described in the following. This model is also called the *Aspect Model*. The idea is that a topic c is first chosen with probability $p(c)$. Then a word x_r is generated with probability $p(x_r|c)$ and the document with index s is generated with probability $p(s|c)$. Note that all the variables are discrete and finite and the topic c is seen as a hidden variable. Assuming that co-occurrences are independent, the log-likelihood function for a given training set then becomes :

$$L = \sum_r \log p(x_r, s_{n(r)}) \tag{3}$$

$$= \sum_r \log \sum_c p(s_{n(r)}|c)p(c)p(x_r|c) \tag{4}$$

where r runs over all samples/words in all documents and $n(r)$ is a function that assigns the words to the document which they belong to. In the supervised version where the topics of the training set are known, this simply becomes :

$$L = \sum_r \log p(s_{n(r)}|c_{n(r)})p(c_{n(r)})p(x_r|c_{n(r)}) \tag{5}$$

Note that the document index s is in the range $1, \ldots, N_s$, where N_s is the total number of training documents. Thus, to predict the topic of a new document, a new index $N_s + 1$ is used and $p(c|\tilde{s}) \equiv p(c|s = N_s + 1)$ is found by the so-called *Folding-in method*[2] as described in [4]. The idea is to consider $\tilde{s}$ as a hidden variable, which results in the following log-likelihood function :

$$L(\tilde{s}) = \sum_{r=1}^{N_r} \log \left(\sum_{c=1}^{N_c} p(\tilde{s}|c)p(c)p(x_r|c) \right) \tag{6}$$

where N_r is the number of words in the new document and N_c is the number of topics. All probabilities apart from $p(\tilde{s}|c)$ were estimated in the training phase and are now kept constant. Using the EM algorithm to infer $p(\tilde{s}|c)$, as in [9], results in the following update equations :

$$p^{(t)}(c|x_r, \tilde{s}) = \frac{p^{(t)}(\tilde{s}|c)\ p(c)\ p(x_r|c)}{\sum_{c=1}^{N_c} p^{(t)}(\tilde{s}|c)\ p(c)\ p(x_r|c)} \tag{7}$$

$$p^{(t+1)}(\tilde{s}|c) = \frac{\sum_{r=1}^{N_r}\ p^{(t)}(c|x_r, \tilde{s})}{C_c + \sum_{r=1}^{N_r}\ p^{(t)}(c|x_r, \tilde{s})} \tag{8}$$

where C_c is the total number of words in all documents from class c. The quantity $p(c|\tilde{s})$ can now be found using Bayes' rule.

[2]Folding-in refers to folding in the new document into the existing collection of documents

3.2. Discrete vocabulary model

In the discrete word model, a vector quantization was first performed on the AR feature space. This is a method that has been quite successful in e.g. speech recognition together with (discrete) hidden Markov models. Using the training set, a finite code book of code vectors was obtained in analogy to the vocabulary of words for a set of documents. A standard vector quantization method was used, where the code vectors were initially chosen randomly from the training set. Then, iteratively, each vector in the training set was assigned to the cluster with the closest (in Euclidean distance) code vector and the new code vectors were found as the means in each cluster. The stopping criteria was a sufficiently small change in the total MSE distortion measure. Finally, each vector in the test set was given the label (word) of the closest code vector in the code book. Now, having mapped the original multi-dimensional, continuous AR feature space into a finite, discrete vocabulary of sound elements, the supervised version of PLSA model can be applied.

The motivation for the discretisation of the feature space was the analogies between music and language. However, the vocabulary of sound elements has a very different distribution from the distribution of words in documents, which usually follows Zipf's law. This is illustrated in figure 2. Several explanations for this could of course be hypothesized, such as the tendency of vector quantization to cluster vectors evenly, but note also that contrary to e.g. [6], the analyzed music spans a large range of genres. The right mapping of such multifaceted music to a finite vocabulary is a problem that is far from being solved. Adding the fact that the AR feature space is continuous in nature, motivated the development of *continuous vocabulary models*.

3.3. Continuous vocabulary models

These models can be seen as the natural generalization of discrete co-occurrence models like PLSA into the limit where the words become continuous, multidimensional feature vectors. Besides, they can be seen as extensions of well-known probabilistic models to include co-occurrence. Two generative, probabilistic models with considerable success in music genre classification, the Gaussian Classifier (GC) and the Gaussian Mixture Model (GMM), have been augmented to co-occurrence models, which will be named *Aspect Gaussian Classifier* (AGC) and *Aspect Gaussian Mixture Model* (AGMM), respectively[3]. Note that similar ideas are proposed in [11], where a so-called Aspect Hidden Markov Model was developed, and in [12] where an Aspect Bernoulli model was proposed.

[3]The word aspect is used with reference to [10], although only supervised training is considered here.

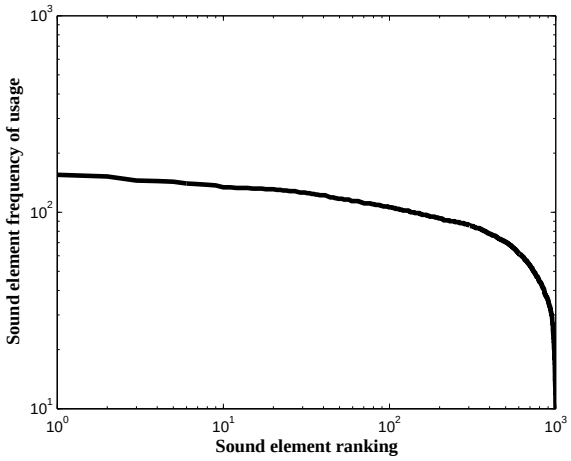


Fig. 2. Frequency of usage of sound elements in the music training set vs. rank of the sound elements (sorted in descending order). The vocabulary of sound elements was found by vector quantization of the AR feature space using 1000 code vectors. A log-log plot is used to test whether Zipf's law applies to the sound elements, in which case the graph should have resembled a straight line with slope approximately -1.

Aspect Gaussian Classifier (AGC)

In figure 3, the graphical models of both the GC and the AGC are illustrated. The log-likelihood function of the AGC becomes :

$$L = \sum_r \log p(s_{n(r)}|c_{n(r)})p(c_{n(r)})p(x_r|c_{n(r)})$$

which seems to be identical to the PLSA equation 5. Note, however, that x_r is now a continuous variable and $p(x|c)$ is a gaussian probability distribution $\mathrm{N}_x(\mu_c, \Sigma_c)$. x_r is the feature vector from time frame r which belongs to the song with index $n(r)$. Additionally, notice that the only difference to the log-likelihood function of the GC is the additional term $p(s_{n(r)}|c_{n(r)})$. Following the maximum likelihood paradigm of parameter inference, the log-likelihood can be maximized directly without resorting to methods like the EM algorithm and the parameter estimates are :

$$\hat{\mu}_c = \frac{1}{N_c}\sum_{r\in C} x_r \quad (9)$$

$$\hat{\Sigma}_c = \frac{1}{N_c}\sum_{r\in C}(x_r - \hat{\mu_c})(x_r - \hat{\mu_c})^T \quad (10)$$

$$\hat{p}(c) = \frac{1}{N_c} \quad (11)$$

$$\hat{p}(s|c) = \frac{N_s}{N_c}\ (\mathrm{if} s \in C, 0\ \mathrm{otherwise}) \quad (12)$$

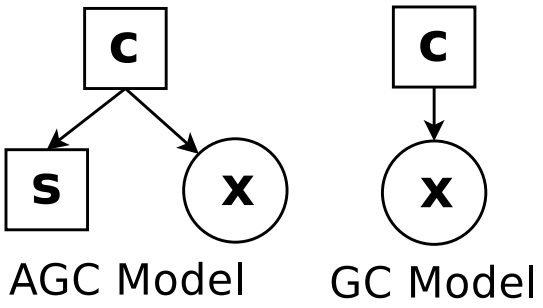


Fig. 3. The graphical models of the Gaussian Classifier (GC) and Aspect Gaussian Classifier (AGC). Round circles represent continuous variables, while squares represent discrete variables.

where N_s and N_c are the total number of time frames in song s and in class c, respectively, and C is the set of time frames from the songs in class c. These estimates are exactly the same as ordinary GC, with the addition of the song probability $p(s|c)$. Given a new song in the testing phase, now requires using the Folding-in method to estimate the probability $p(c|\tilde{s})$, where $\tilde{s}$ is the index of the new song to be folded in. This is done using the update equations in 7 and 8.

Aspect Gaussian Mixture Model (AGMM)

The graphical models of the GMM and the AGMM are shown in figure 4. Now, the log-likelihood function of the AGMM is again similar to the one of the GMM, but with an additional co-occurrence term :

$$L = \sum_r \log \left(p(s_{n(r)}|c_{n(r)}) \sum_{k=1}^{K} p(c_{n(r)}) p(x_r|k) p(k|c_{n(r)}) \right) \tag{13}$$

K denotes the number of components in the model. As for the AGC model, all the parameter estimation equations become the same as in the original GMM model where now the EM algorithm will be used due to the hidden variable k. The probability $p(s_{n(r)}|c)$ again becomes a count of the number of songs in each genre in the training set as in equation 12. The equivalent of equation 6 for the Folding-in procedure, now becomes :

$$L(\tilde{s}) = \sum_{r=1}^{N_r} \log \left(\sum_{c=1}^{N_c} p(\tilde{s}|c) \sum_{k=1}^{K} p(c) p(x_r|k) p(k|c) \right)$$

with update equations :

$$p^{(t)}(c|x_r, \tilde{s}) = \frac{p^{(t)}(\tilde{s}|c) \sum_{k=1}^{K} p(c)\ p(x_r|k) p(k|c)}{\sum_{c=1}^{C_c} p^{(t)}(\tilde{s}|c) \sum_{k=1}^{K} p(c) p(x_r|k) p(k|c)}$$

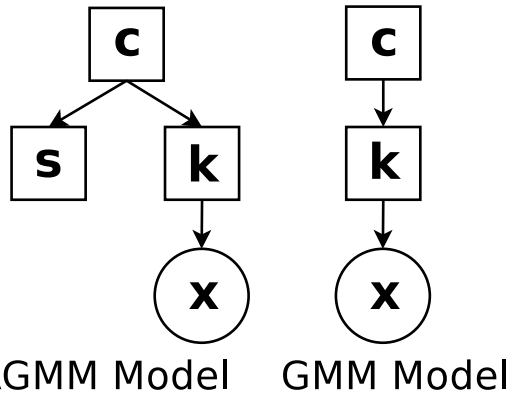


Fig. 4. The graphical models of the Gaussian Mixture Model (GMM) and Aspect Gaussian Mixture Model (AGMM). Round circles represent continuous variables, while squares represent discrete variables.

and

$$p^{(t+1)}(\tilde{s}|c) = \frac{\sum_{r=1}^{N_r} p^{(t)}(c|x_r, \tilde{s})}{C_c + \sum_{r=1}^{N_r} p^{(t)}(c|x_r, \tilde{s})}$$

Note that the only necessary quantity in the E-step is simply the estimate of $p(x_r, c)$ from the training phase for both the AGC and AGMM models. Thus, standard software packages can be used for training both GC and GMM and calculating the estimates of $p(x_r, c)$ for the new song. The Folding-in procedure then becomes a simple extension to this.

Comparing Folding-in and Sum-rule

Looking more carefully at the Folding-in method as described in the last part of section 3.1 reveals a relation to the Sum-rule method in equation 2. It is assumed that the initial guess of $p^{(0)}(\tilde{s}|c)$ in equation 7 is uniform over the classes c and that $p(c)$ is also uniform over classes. This is obviously often not the case, however, in the current music genre classification problem these are reasonable assumptions. It is now seen that the right side of equation 7 simply reduces to the probability $p(c|x_r)$ and the sums on the right side of equation 8 are seen to be simply equal to the sum used in the Sum-rule. Thus, with the mentioned assumptions *the decisions from the Sum-rule are the same as from the first iteration of the Folding-in method*. In this view, the Sum-rule may be seen as an approximation to the full probabilistic model with the Folding-in method.

4. RESULTS AND DISCUSSION

A series of experiments were made to compare the three proposed models (the Discrete Model, the AGC and the

AGMM models) with the GC and GMM models. These two models combined with the Sum-rule method (equation 2) can be seen as good baseline methods [7]. The choice of using the Sum-rule instead of Majority Voting (equation 1), is based on experimental results which show that the Sum-rule consistently performs slightly better than Majority Voting. This is in agreement with the findings in [13].

Data set

The music data set that was used in the experiments consisted of $115 * 11 = 1265$ songs evenly distributed among 11 genres which were "Alternative", "Country", "Easy Listening", "Electronica", "Jazz", "Latin", "Pop and Dance", "Rap and HipHop", "R&B and Soul", "Reggae" and "Rock". The songs had a sampling frequency of 22050 Hz. From each song, 30 seconds were used from the middle part of the song. The data set is considered difficult to classify with overlap between genres, since a small-scale human evaluation involving 10 people gave a classification error rate with mean 48 % and standard deviation on the mean of 1.6 %. The evaluation involved each person classifying 30 of the sound clips (randomly chosen) on a forced-choice basis.

Feature extraction

The AR features were extracted from the data set along the lines described in section 2. 6 MFCC features were calculated from each frame of size 30 ms and the hopsize between frames was 10 ms. For each of the MFCC features, 3 AR coefficients were found along with the residual and the mean, thus resulting in $6 * 5 = 30$ dimensional AR features. The AR framesize was 760 ms and with a hopsize of 390 ms. Thus, each 30 second song was represented by 80 30-dimensional AR features.

Classification

At first, methods for preprocessing were examined such as whitening and dimension reduction by PCA. However, the classification performance was not significantly affected by the preprocessing. It was decided to normalize each feature dimension individually to avoid numerical problems in the covariance matrix calculations.

The results for all the examined models are shown in figure 5, calculated as described in section 3. The results were found by cross-validation using 80 songs in the training set and 20 in the testing set from each genre. Parameters in the model structure, such as the number of components in the GMM and AGMM models were also found by cross-validation as shown in figure 6. For the continuous models, experiments were made with both diagonal and full covariance matrices in $p(x_r|c)$ and $p(x_r|k)$. Best results were obtained with fairly small numbers of full covariance matrices.

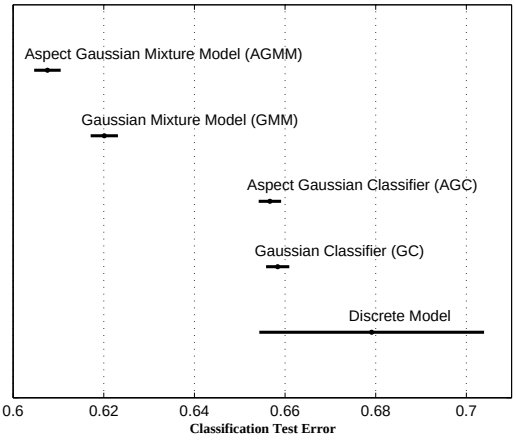


Fig. 5. Classification test error results for the Discrete Model, the Aspect Gaussian Classifier, the Aspect Gaussian Mixture Model and the two baseline methods Gaussian Classifier and Gaussian Mixture Model. The results are the mean values using cross-validation (5-fold for the Discrete Model and 50-fold for the rest) and the error bars are the standard deviations on the means. 7 components were used for the GMM and AGMM.

Note that only similar numbers of mixtures were chosen to represent each genre as seen in figure 6. Better results could possibly be obtained using different numbers of mixtures for the different genres, however, the main focus in the current work has been the comparisons between the baselines and their extensions more than optimizing for performance.

A practical complication was the choice of the vocabulary size in the Discrete Model, since the code book generation was computationally demanding in both space and time due to the large vocabulary size. Experiments were made with sizes in the range of 25 to 2000 code vectors and the test error minimum was found to be around 1000 code vectors.

Discussion

Figure 5 shows that the Discrete Model performs within the range of the GC/AGC models, but it has the added computational processing in the vocabulary creation and mapping parts in the training and test phases, respectively. The testing parts of the AGC and AGMM models are much less computationally demanding which makes them more useful in practical applications. Both of the proposed continuous vocabulary aspect models do better than their baseline counterparts, although it is almost negligible in the case of the AGC as compared to the GC.

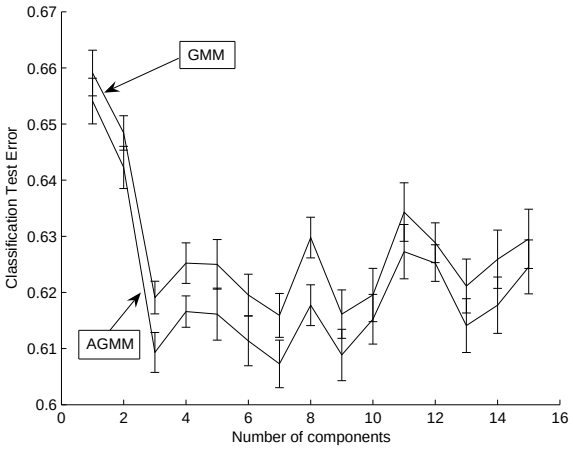


Fig. 6. Classification test error is shown as a function of the number of components in the Gaussian Mixture Model and the Aspect Gaussian Mixture Model. The line illustrates the mean value over 20-fold cross-validation and the error bars show the standard deviation on the mean.

5. CONCLUSION

Three co-occurrence models have been proposed and tested in this work. The first model was the Discrete Model, which was fully based on the PLSA model and used vector quantization to transform the continuous feature space into a finite, discrete vocabulary of sound elements. The two other models, the Aspect Gaussian Classifier and the Aspect Gaussian Mixture Model, were modifications of well-known probabilistic models into co-occurrence models.

The proposed models all have the benefit of modelling the class-conditional probability $p(\tilde{s}|c)$ of the whole song $\tilde{s}$ instead of just modelling short time frames $p(x_r|c)$ as is often the case. This feature of the models could be useful in e.g. music recommendation systems, where only the songs with the highest $p(c|\tilde{s})$ are recommended.

The Discrete Model gave classification test errors in a range comparable to the GC/AGC models, but suffers from the drawback of being demanding in computational time and space due to the vector quantization. The AGC and AGMM models performed slightly better than their baseline counterparts in combination with the Sum-rule method and with a fairly modest increase in computational time.

6. ACKNOWLEDGEMENTS

The work is supported by the European Commission through the sixth framework IST Network of Excellence: Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL), contract no. 506778.

7. REFERENCES

[1] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, July 2002.

[2] M. F. McKinney and J. Breebaart, "Features for audio and music classification," in *Proceedings of ISMIR*, 2003.

[3] C. Xu, N. C. Maddage, X. Shao, F. Cao, and Q. Tian, "Musical genre classification using support vector machines," in *Proceedings of ICASSP*, Hong Kong, China, Apr. 2003, pp. 429–432.

[4] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of SIGIR*, Berkeley, CA, 1999, pp. 35–44.

[5] A. D. Patel, "Language, music, syntax and the brain," *Nature Neuroscience*, vol. 6, no. 7, pp. 674–681, July 2003.

[6] D. H. Zanette, "Zipf's law and the creation of musical context," *Musicae Scientiae*, 2005, In Press.

[7] A. Meng, P. Ahrendt, and J. Larsen, "Improving music genre classification using short-time feature integration," in *Proceedings of ICASSP*, 2005.

[8] P. Ahrendt, A. Meng, and J. Larsen, "Decision time horizon for music genre classification using short-time features," in *Proceedings of EUSIPCO*, 2004.

[9] E. Gaussier, C. Goutte, K. Popat, and F. Chen, "Hierarchical model for clustering and categorising documents," in *Proceedings of the 24th BCS-IRSG European Colloqium on IR Research (ECIR-02)*, 2002.

[10] T. Hofmann and J. Puzicha, "Unsupervised learning from dyadic data," Tech. Rep. TR-98-042, International Computer Science Institute, Berkeley, CA, December 1998.

[11] D. Blei and P. Moreno, "Topic segmentation with an aspect hidden markov model.," in *Proceedings of the 24th international ACM SIGIR conference.*, 2001, pp. 343–348.

[12] A. Kaban, E. Bingham, and T. Hirsimki, "Learning to read between the lines: The aspect bernoulli model," in *Proceedings of the 4th SIAM International Conference on Data Mining*, Lake Buena Vista, Florida, April 2004, pp. 462–466.

[13] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

Appendix E

Music Genre Classification using the Multivariate AR Feature Integration Model

---

Ahrendt P. and Meng A., **Music Genre Classification using the Multivariate AR Feature Integration Model**, Audio Genre Classification contest at the *Music Information Retrieval Evaluation eXchange (MIREX)* (in connection with the annual ISMIR conference) [53], London, UK, September 2005.

Music Genre Classification using the multivariate AR feature integration model

Peter Ahrendt
Technical University of Denmark (IMM)
Building 321, office 120, 2800 Kgs. Lyngby
Denmark
pa@imm.dtu.dk

Anders Meng
Technical University of Denmark (IMM)
Building 321, office 105, 2800 Kgs. Lyngby
Denmark
am@imm.dtu.dk

Keywords: Feature Integration, Multivariate AR, Generalized Linear Classifier

1 INTRODUCTION

Music genre classification systems are normally build as a feature extraction module followed by a classifier. The features are often short-time features with time frames of 10-30ms, although several characteristics of music require larger time scales. Thus, larger time frames are needed to take informative decisions about musical genre. For the MIREX music genre contest several authors derive long time features based either on statistical moments and/or temporal structure in the short time features. In our contribution we model a segment (1.2 s) of short time features (texture) using a multivariate autoregressive model. Other authors have applied simpler statistical models such as the mean-variance model, which also has been included in several of this years MIREX submissions, see e.g. Tzanetakis (2005); Burred (2005); Bergstra et al. (2005); Lidy and Rauber (2005).

2 FEATURES & FEATURE INTEGRATION

The system is designed to handle 22.5kHz mono signals, but could easily be extended to arbitrary sample-rate of the audio signal. Each song is represented by a 30s music snippet taken from the middle of the song. From the raw audio signal the first 6 Mel Frequency Cepstral Coefficients (MFCC) are extracted (including the 0th order coefficient) using a hop- and framesize of 7.5ms and 15ms, respectively. Thus, each song is now represented by a 6 dimensional multivariate time-series. The time series typically display dependency among feature dimensions as well as temporal correlations. Simple statistical moments can be used to characterize important information of the short time features or more elaborate models can be applied. Statistical models which include correlations among feature dimensions as well as time correlations is e.g. the multivariate autoregressive model. Assume that $\mathbf{x}_n$ for $n = 1, \ldots, N$ is the time series of short time features then the multivariate AR model (MAR) can be written as

$$\mathbf{x}_n = \sum_{p=1}^{P} \mathbf{A}_p \mathbf{x}_{n-p} + \mathbf{v} + \mathbf{u}_n, \tag{1}$$

where the noise term $\mathbf{u}_n$ is assumed i.i.d. with zero mean and finite covariance matrix $\mathbf{C}$. The 6 dimensional parameter vector $\mathbf{v}$ is a vector of intercept terms related to the mean of the time series. The $\mathbf{A}_p$'s are the autoregressive coefficient matrices and P denotes the model order. The parameters of the model are estimated using ordinary least squares method and the new feature now consists of elements of $\mathbf{v}$, $\mathbf{C}$ (diagonal + upper triangular part) and $\mathbf{A}_p$ for $p = 1, \ldots, P$. In the actual setup a hopsize of 400ms, framesize of 1200ms and a model order of $P = 3$ results in 72 medium time feature vectors each of dimension 135 ($\mathbf{v} \sim 6, \mathbf{C} \sim 15$ and $A_{1,2,3} \sim 36 * 3 = 108$) for each music snippet. The hopsize, framesize as well as the model order of $P = 3$ have been selected from earlier experiments on other data sets (a-priori information). Thus, not tuned specifically to the unknown data sets in contest. To avoid numerical problems in the classifier each feature dimension of the MAR features is normalized to unit variance and zero mean. The normalization constants for each dimension are calculated from the training set.

3 CLASSIFIER

A generalized linear model (GLM), Bishop (1995), with softmax activation function is trained on all the MAR-feature vectors from all the songs. This classifier is simply an extension of a logistic regression classifier to more than two classes. It has the advantage of being discriminative, which makes it more robust to non-equal classes. Furthermore, since it is a linear model it is less prone to overfitting (as compared to a generative model). Each frame of size 1200ms is classified as belonging to one of c classes, where c is the total number of music genres. In the actual implementation the *Netlab* package was used, see `http://www.ncrg.aston.ac.uk/netlab/` for more details.

3.1 Late information fusion

To reach a final decision for a 30s music clip the sum-rule, Kittler et al. (1998), is used over all the frames in the

music clip. The sum-rule assigns a class as

$$\hat{c} = \arg\max_{c} \sum_{r=1}^{n_f} P(c|\mathbf{x}_r) \quad (2)$$

where r and n_f is the frame index and number of frames of the music clip, respectively, and $P(c|\mathbf{x}_r)$ is the estimated posterior probability of class c given the MAR feature vector $\mathbf{x}_r$. As mentioned earlier $n_f = 72$ frames for each music clip.

Figure 1 shows the full system setup of the music genre classification task from the raw audio to a decision on genre of each music snippet.

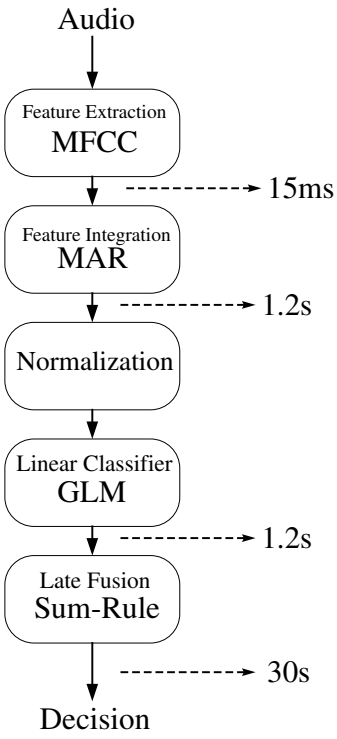


Figure 1: Overview of system from audio to a genre decision at 30s. The time scale at each step is indicated to the right.

4 CONTEST RESULTS

This years *Audio Genre Classification* contest consisted of two audio databases

- *USPop* (single level genre), `http://www.ee.columbia.edu/~dpwe/research/musicsim/uspop2002.html`
- *Magnatune* (hierarchical genre taxonomy) `www.magnatune.com`

from which two independent data sets were compiled. Originally, a third database, *Epitonic* (`http://www.epitonic.com`), was proposed, but due to lack of time only the first two databases were investigated.

The first data set was generated from the USPop database and consisted of a training set of 940 music files distributed un-evenly among 6 genres (Country, Electronica/Dance, Newage, Rap/Hiphop, Reggae and Rock) and a test set of 474 music files. The second data set was generated from the Magnatune database with a training/test set of $1005/510$ music files distributed un-evenly among the 10 genres: Ambient, Blues, Classical, Electronic, Ethnic, Folk, Jazz, Newage, Punk and Rock.

4.1 Parameter optimization

The various parameters of both the feature extraction and integration step as well as nuisance parameters for the GLM classifier were preselected, and therefore not tuned to the specific data sets. Cross-validation or an approximative approach could have been utilized in order to optimize the values of the classifier and feature extraction/integration step.

4.2 Results & Discussion

Figure 2 shows the raw mean classification accuracy of both data sets of the methods, which completed within the 24 hour time limit (8th of September). A 95% binomial confidence interval was applied on each method to illustrate the possible variation in mean value. Our al-

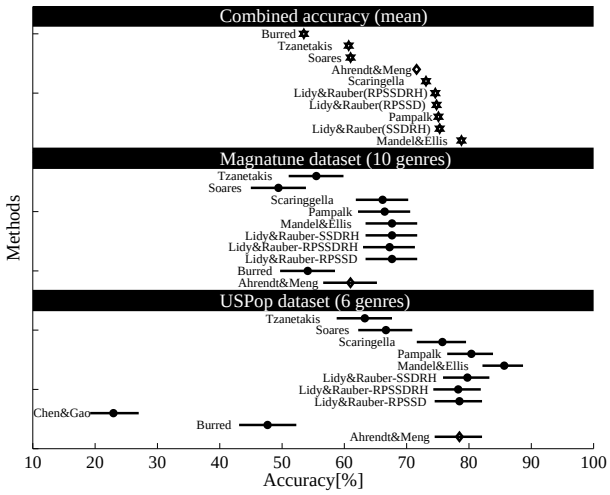


Figure 2: Mean accuracy on both USPop and Magnatune data sets illustrated with a 95% binomial confidence interval. The "Combined accuracy" is the mean accuracy on the two data sets.

gorithm, denoted as *Ahrendt&Meng*, shows a mean accuracy of 60.98% for uncorrected classes on the Magnatune data set and a mean accuracy of 78.48% on the USpop data set. Our method showed a mean accuracy of 71.55% when averaging across data sets compared with the best performing method of 78.8% by *Mandel&Ellis*. There is several observations, which can be made from this years contest. Our model is solely based on the first 6 MFCCs, which subsequently are modelled by a multivariate autoregressive model, hence, the temporal structure is modelled. The best performing method in this years contest is by Mandel and Ellis (2005) (8th of September), see

figure 2). Their approach consist of extracting the first 20 MFCCs and then model the MFCCs of the entire song by a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. This model is then used in a modified KL-divergence kernel, from which a support vector classifier can be applied. Since the mean and covariance are static components no temporal information is modelled in this approach, however, good results were observed. Even better results might have been achieved by using models, which include temporal information.

In order to make a proper statistical comparison of the different methods the raw classifications should have been known.

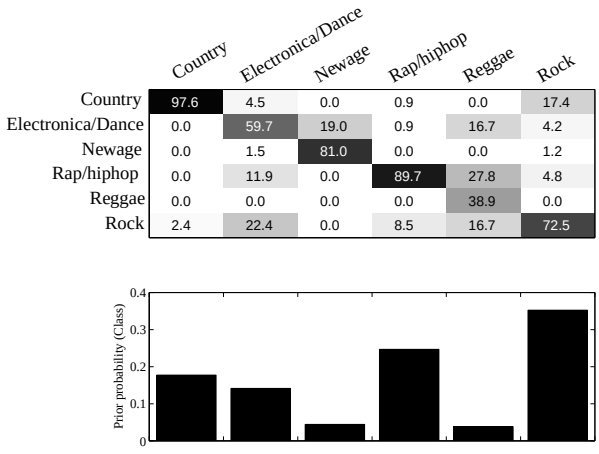


Figure 3: *Upper:* Confusion matrix (accuracy) of proposed method on the USPop data set. *Lower:* The prior probabilities of the genres.

The upper figure of figure 3 and 4 shows the confusion matrix of our method on the USPop and Magnatune data set, respectively. The lower figures shows the prior probability on the genres calculated from the test sets. The true genre is shown along the horizontal axis. The confusion matrix on the Magnatune data set illustrates that our method provides reasonable predictive power of *Punk, Classical and Blues*, whereas *Newage* is actually below a random guessing of 2.9%.

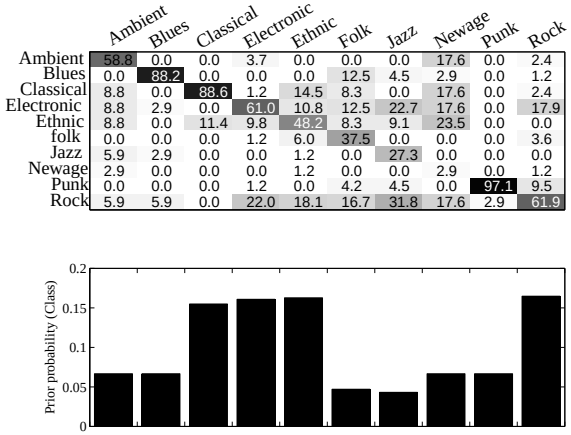


Figure 4: *Upper:* Confusion matrix (accuracy) of proposed method on the Magnatune data set. *Lower:* The prior probabilities of the genres.

5 CONCLUSION & DISCUSSION

A mean accuracy over the two data sets of 71.6% was achieved using only the first 6 MFCCs as compared to a mean accuracy of 78.8% by Mandel and Ellis (2005) (8th of September) using the first 20 MFCCs. A further performance increase could have been achieved by optimizing nuisance parameters of the classifier and by correcting for uneven classes. Furthermore, the model order of the multivariate autoregressive model could have been optimized using cross-validation on the training set. Future perspectives would be to use a support vector classifier, which would alleviate problems of overfitting. The approach presented in this extended abstract could easily have been applied in the *Audio Artist Identification* contest as well.

References

J. Bergstra, N. Casagrande, and D. Eck. Music genre classification, mirex contests, 2005.

C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

Juan Jose Burred. Music genre classification, mirex contests, 2005.

J. Kittler, M. Hatef, Robert P.W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

T. Lidy and A. Rauber. Music genre classification, mirex contests, 2005.

Michael Mandel and Daniel Ellis. Music genre classification, mirex contests, 2005.

George Tzanetakis. Music genre classification, mirex contests, 2005.

Appendix F

Towards Cognitive Component Analysis

Hansen L. K., Ahrendt P. and Larsen J., **Towards Cognitive Component Analysis**, Proceedings of *International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR)*, Espoo, Finland, June 2005.

TOWARDS COGNITIVE COMPONENT ANALYSIS

Lars Kai Hansen, Peter Ahrendt, and Jan Larsen

Intelligent Signal Processing,
Informatics and Mathematical Modelling,
Technical University of Denmark B321,
DK-2800 Kgs. Lyngby, Denmark

ABSTRACT

Cognitive component analysis (COCA) is here defined as the process of unsupervised grouping of data such that the ensuing group structure is well-aligned with that resulting from human cognitive activity. We have earlier demonstrated that independent components analysis is relevant for representing semantics, not only in text, but also in dynamic text (chat), images, and combinations of text and images. Here we further expand on the relevance of the ICA model for representing context, including two new analyzes of abstract data: social networks and musical features.

1. INTRODUCTION

In this paper our aim is to discuss the generality of the so-called *independent component hypothesis*. It is well documented that human perceptional systems can model complex multi-agent scenery. Human cognition uses a broad spectrum of cues for analyzing perceptual input and separate individual signal producing agents, such as speakers, gestures, affections etc. Unsupervised signal separation has also been achieved in computers using a variety of independent component analysis algorithms [1]. It is an intriguing fact that representations are found in human and animal perceptual systems which closely resembles the information theoretically optimal representations obtained by independent component analysis, see e.g., [2] on visual contrast detection, [3] on visual features involved in color and stereo processing, and [4] on representations of sound features. Here we go one step further and ask: *Are such optimal representation rooted in independence also relevant in higher cognitive functions*? Our presentation is largely qualitative and will mainly be based on simple visualizations of data and avoid unnecessary algebraic complication.

Brittanica online defines cognition as the 'act or process of knowing', and continues:

> Cognition includes every mental process that may be described as an experience of knowing (including perceiving, recognizing, conceiving, and reasoning), as distinguished from an experience of feeling or of willing.

Wagensberg has recently argued the importance of being able to recognize independence for successful 'life forms' [5]

> A living individual is part of the world with some identity that tends to become independent of the uncertainty of the rest of the world

Thus natural selection favors innovations that increase independence of the agent in the face of environmental uncertainty, while maximizing the gain from the predictable aspects of the niche. This view represents a precision of the classical Darwinian formulation that natural selection simply favors adaptation to given conditions. Wagensberg points out that recent biological innovations, such as nervous systems and brains are means to decrease the sensitivity to un-predictable fluctuations. Furthermore, by creating alliances, agents can in Wagensberg's picture give up independence for the benefit of a group, which in turns may increase independence for the group as an entity. Both in its simple one-agent form and in the more tentative analysis of the group model, Wagensberg's theory points to the crucial importance of *statistical independence* for evolution of perception, semantics and indeed cognition.

While cognition may be hard to quantify, its direct consequence, human behavior, has a rich phenomenology which is becoming increasingly accessible to modeling. The digitalization of everyday life as reflected, say, in telecommunication, commerce, and media usage allows quantification and modeling of human patterns of activity, often at the level of individuals.

Grouping of events or objects in categories is fundamental to human cognition. In machine learning, classification is a rather well-understood task when based on *labelled* examples [6]. In this case classification belongs to the class of *supervised* learning problems. Clustering is a closely related *unsupervised* learning problem, in which we use general statistical rules to group objects, without a priori providing a set of labelled examples. It is a fascinating finding in many real world data sets that the label structure discovered by unsupervised learning closely coincides with labels obtained by letting a human or a group of humans perform classification, labels derived from human cognition. *Here we will define cognitive component analysis (COCA) as the process of unsupervised group-*

ing of data such that the ensuing group structure is well-aligned with that resulting from human cognitive activity. Without directly using the phrase 'cognitive component analysis', the concept of cognitive components appears frequently in the context of Factor analysis of behavioral studies, see e.g., [7, 8].

We have pursued grouping by independent component analysis in several abstract data types including text, dynamic text (chat), images, and combinations hereof, see e.g., [9, 10, 11, 12, 13]. In this presentation we will briefly review our analysis of text data and add visualizations of two new types of abstract data, namely co-worker networks and music, that further underlines the broad relevance of the independent component hypothesis.

2. COGNITIVE COMPONENT ANALYSIS

In 1999 Lee and Seung introduced the method of non-negative matrix factorization (NMF) [14] as a scheme for parts-based object recognition. They argued that the factorization of an observation matrix in terms of a relatively small set of cognitive components, each consisting of a feature vector and a loading vector (both non-negative) lead to a parts based object representation. They demonstrated this for objects in images and in text representations. More recently, in 2002, it was shown that very similar parts-based decompositions were obtained in a latent variable model based on positive linear mixture of positive *independent* source signals [15]. Holistic, but parts-based, recognition of objects is frequently reported in perception studies across multiple modalities and increasingly in abstract data, where object recognition is a cognitive process. Together these findings are often referred to as instances of the more general *Gestalt laws*.

2.1. Latent semantic indexing (LSI)

Salton proposed the so-called vector space representation for statistical modeling of text data, for a review see [16]. A term set is chosen and a document is represented by the vector of term frequencies. A document database then forms a so-called term-document matrix. The vector space representation can be used for classification and retrieval by noting that similar documents are somehow expected to be 'close' in the vector space. A metric can be based on the simple Euclidean distance if document vectors are properly normalized, otherwise angular distance may be useful. This approach is principled, fast, and language independent. Deerwester and co-workers developed the concept of latent semantics based on principal component analysis of the term-document matrix [17]. The fundamental observation behind the latent semantic indexing (LSI) approach is that similar documents are using similar vocabularies, hence, the vectors of a given topic could appear as produced by a stochastic process with highly correlated term-entries. By projecting the term-frequency vectors on a relatively low dimensional subspace, say determined by the maximal amount of variance one would be able to filter out the inevitable 'noise'. Noise should here be thought of as individual document differences in term usage within a specific context. For well-defined topis, one could simply hope that a given context would have a stable core term set that would come out as a 'direction' in the term vector space. Below we will explain why this is likely not to happen in general document databases, and LSI is therefore often used as a dimensional reduction tool, which is then post-processed to reveal cognitive components, e.g., by interactive visualization schemes [18].

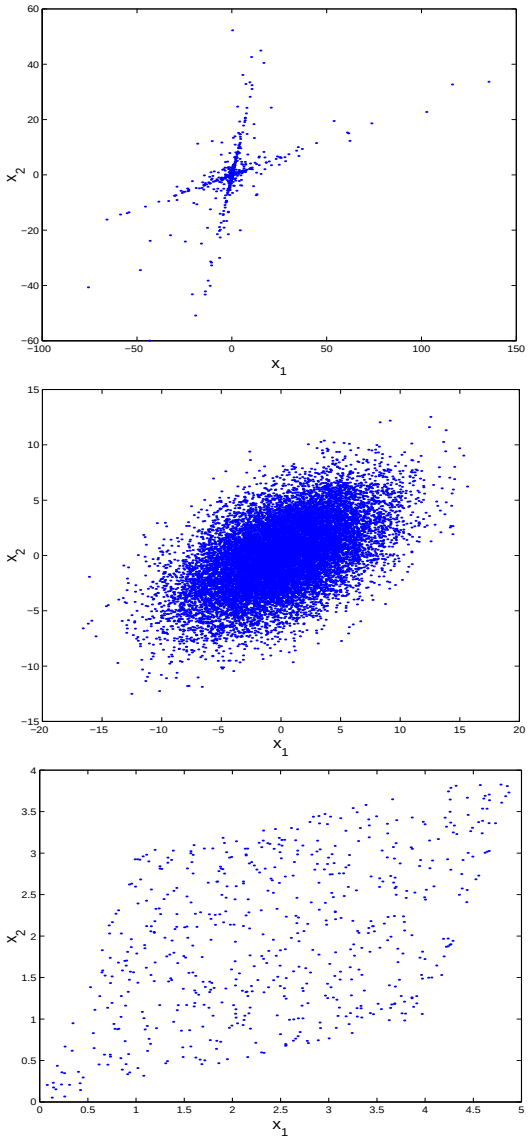


Figure 1. Prototypical feature distributions produced by a linear mixture, based on sparse (top), normal (middle), or dense source signals (bottom), respectively. The characteristic of the sparse signal is that it consists of relatively few large magnitude samples on a background of small signals.

2.2. Non-negative matrix factorization (NMF)

Noting that many basic feature sets are naturally positive and that a non-negative decomposition could lead to a parts-based decomposition, Lee and Seung analyzed several data sets using the NMF decomposition technique

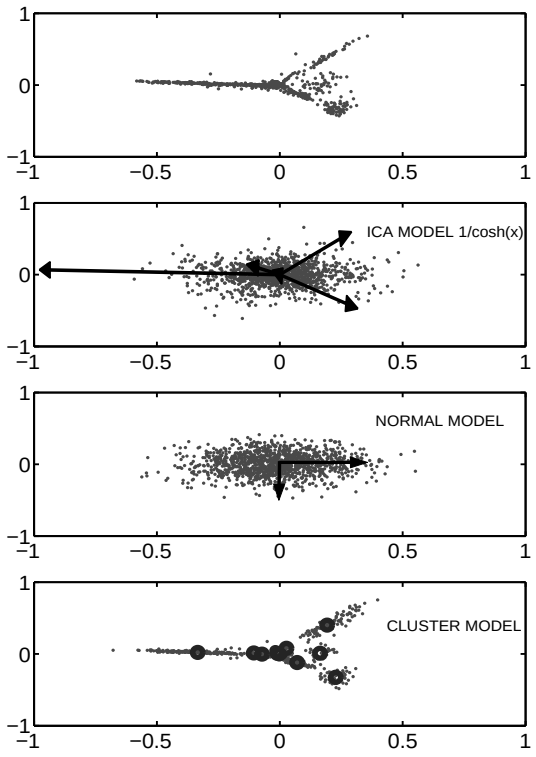


Figure 2. Latent semantic analysis of text, based on Salton's vector space representation reveals that the semantic components are very sparse, as indicated in the scatter plot of two components (top). We spot the signature of a sparse linear mixture: 'rays' emanating from $(0, 0)$. Performing a five component ICA on the subspace spanned by the five most significant latent vectors, provide the mixing matrix with column vector as shown in the second panel. Using a simple classification scheme (magnitude of the source signal) yields a classifier with less then 10% error rate using the document labels manually assigned by a human editor. Below, in the third plot, we indicate the corresponding normal model, with axis aligned latent vectors. Finally, we show in the bottom plot the results of an alternative unsupervised analysis, based on clustering, using a Gaussian mixture model. While the mixture model do capture the density well, the ensuing components are not related in a simple way to content.

[14]. A basic difficulty of the approach is the possible non-uniqueness of the components. This issue has been discussed in detail by Donoho and Stodden [19]. A possible route to more unique solutions, hence, potentially more interpretable and relevant components is to add a priori knowledge, e.g., in form of independence assumptions. An algorithm for decomposing independent positive components from a positive mixture is discussed in [15].

2.3. Independent component analysis (ICA)

Blind signal separation is the general problem of recovering source signals from an unknown mixture. This aim is in general not feasible without additional information. If we assume that the unknown mixture is linear, i.e., that the mixture is a linear combination of the sources, and furthermore assume that the sources are statistically independent processes it is often possible to recover sources and mixing, using a variety of independent component analysis techniques [1]. Here we will discuss some basic characteristics of mixtures and the possible recovery of sources.

First, we note that LSI/PCA can not do the job in general. Let the mixture be given as

$$\mathbf{X} = \mathbf{AS}, \;\; X_{j,t} = \sum_{k=1}^{K} A_{j,k} S_{k,t}, \tag{1}$$

where $X_{j,t}$ is the value of j'th feature in the t'th measurement, $A_{j,k}$ is the mixture coefficient linking feature j with the component k, while $S_{k,t}$ is the level of activity in the k'th source. In a text instance a feature is a term and the measurements are documents, the components are best thought as topical contexts. The k'th column $A_{j,k}$ holds the relative frequencies of term occurrence in documents within context k. The source matrix element $S_{k,t}$ quantifies the level of expression of context k in document t.

As a linear mixture is invariant to an invertible linear transformation we need define a normalization of one of the matrices $\mathbf{A}, \mathbf{S}$. We will do this by assuming that the sources are unit variance. As they are assumed independent the covariance will be trivial,

$$\Sigma_S = \lim_{T\to\infty} \frac{1}{T} \mathbf{S}\mathbf{S}^\top = \mathbf{I}. \tag{2}$$

LSI, hence PCA, of the measurement matrix is based on analysis of the covariance

$$\Sigma_X = \lim_{T\to\infty} \frac{1}{T} \mathbf{X}\mathbf{X}^\top = \mathbf{A}\mathbf{A}^\top. \tag{3}$$

Clearly the information in $\mathbf{A}\mathbf{A}^\top$ is not enough to uniquely identify $\mathbf{A}$, since if a solution $\mathbf{A}$ is found, any (row) rotated matrix $\tilde{\mathbf{A}} = \mathbf{AU}, \mathbf{U}\mathbf{U}^\top = \mathbf{I}$ is also a solution, because $\tilde{\mathbf{A}}$ has the same outer product as $\mathbf{A}$.

This is a potential problem for LSI based analysis. If the document database can be modelled as in eq. (1) then the original characteristic context histograms will not be found by LSI. The field of independent component analysis has on the other hand devised many algorithms that use more informed statistics to locate $\mathbf{A}$ and thus $\mathbf{S}$, see [1] for a recent review.

The histogram of a source signal can roughly be described as sparse, normal, or dense. Scatter plots of projections of mixtures drawn from source distributions with one of these three characteristics are shown in Figure 1. In the upper panel of Figure 1 we show the typical appearance of a sparse source mixture. The sparse signal consists of relatively few large magnitude samples in a background of a large number of small signals. When mixing such independent sparse signals as in Eq. (1), we obtain a set of rays emanating from origo. The directions of the rays are directly given by the column vectors of the $\mathbf{A}$-matrix.

If the sources are truly normal distributed like in the middle panel of Figure 1, there is no additional information but the covariance matrix. Hence, in some sense this is a singular worst case for separation. Because we work from finite samples an ICA method, which assumes some non-normality, will in fact often find good approximations to the mixing matrix, simply because a finite normal sample will have non-normal oddities. But fortunately, many, many interesting real world data sets are not anywhere near normal, rather they are typically very sparse, hence, more similar to the upper panel of Figure 1.

3. COGNITIVE COMPONENTS FROM UNSUPERVISED DATA ANALYSIS

Having argued that tools are available for recovering, relatively uniquely, the underlying components in a mixture we now turn to some illustrative examples. In a text analysis example we show that an ICA based analysis indeed finds a small set of semantic components that very well aligned with human assigned labels that were not used in the analysis.

3.1. Text analysis

In Figure 2 (top) we indicate the corresponding scatter plots of a small text database. The database consists of documents with overlapping vocabulary but five different (high level cognitive) labels [20]. The 'ray'-structure is evident. In the second panel we show the directions identified by ICA. If we use a simple projection based classification rule, and associate a ray with a topic, the classification error rate is less than 10% [20]. If an ICA is performed with less components, the topics with close content are merged.

This rather striking alignment between human and machine classification in abstract features like those of vector space text analysis, is a primary motivation for the present work. In this example we also estimated an alternative unsupervised model based on document clustering using a gaussian mixture model. This model provides the representation shown in bottom panel of Figure 2, in this case the clusters are not specific enough to have a simple one-to-one correspondence, however, with a limited amount of supervision it will be possible to convert this cluster based representation into a classifier with similar performance as the ICA model.

3.2. Social networks

The ability to navigate social networks is a hallmark of successful cognition. Is it possible that the simple unsupervised scheme for identification of independent components, whose relevance we have established above for perceptual tasks, for context grouping in different media, could play a role in this human capacity? To investigate this issue we have initiated an analysis of a well-known social network of some practical importance. The so-called *actor network* is a quantitative representation of the co-participation of actors in movies, for a discussion of this network, see e.g., [21]. The observation model for the network is not too different from that of text. Each movie is represented by the *cast*, i.e., the list of actors. We have converted the table of the about $T = 128.000$ movies with a total of $J = 382.000$ individual actors, to a sparse $J \times T$ matrix $\mathbf{X}$. For visualization we have projected the data onto principal components (LSI) of the actor-actor covariance matrix. The eigenvectors of this matrix are called 'eigen casts' and represent characteristic communities of actors that tend to co-appear in movies. The sparsity and magnitude of the network means that the components are dominated by communities with very small intersections, however, a closer look at such scatter plots reveals detail suggesting that a simple linear mixture model indeed provides a reasonable representation of the (small) coupling between these relative trivial disjunct subsets, see Figure 3.

Such insight may be used for computer assisted navigation of collaborative, peer-to-peer networks, for example in the context of search and retrieval.

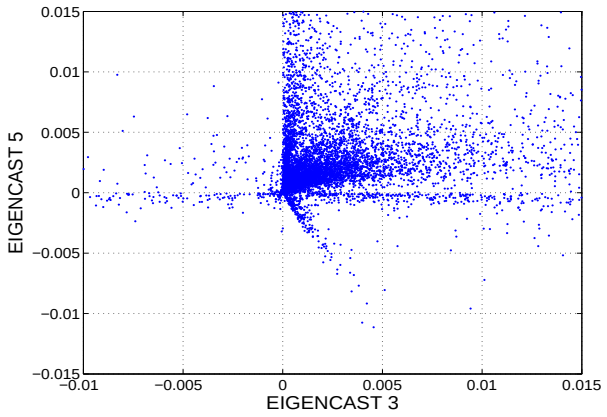


Figure 3. The so-called actor network quantifies the collaborative pattern of 382.000 actors participating in almost 128.000 movies. For visualization we have projected the data onto principal components (LSI) of the actor-actor co-variance matrix. The eigenvectors of this matrix are called 'eigencasts' and they represent characteristic communities of actors that tend to co-appear in movies. The network is extremely sparse, so the most prominent variance components are related to near-disjunct subcommunities of actors with many common movies. However, a close up of the coupling between two latent semantic components (the region $\sim (0, 0)$) reveals the ubiquitous signature of a sparse linear mixture: A pronounced 'ray' structure emanating from (0,0). We speculate that the cognitive machinery developed for handling of independent events can also be used to locate independent subcommunities, hence, navigate complex social networks, a hallmark of successful cognition.

3.3. Musical genre

The growing market for digital music and intelligent music services creates an increasing interest in modeling of music data. It is now feasible to estimate consensus musi-

cal genre by *supervised* from rather short music segments, say 10-30 seconds, see e.g., [22], thus enabling computerized handling of music request at a high cognitive complexity level. To understand the possibilities and limitations for unsupervised modeling of music data we here visualize a small music sample using the latent semantic analysis framework. The intended use is for a music search engine function, hence, we envision that a largely text based query has resulted in a few music entries, and the algorithm is going to find the group structure inherent in the retrieval for the user. We represent three tunes (with human labels: `heavy, jazz, classical`) by their spectral content in overlapping small time frames ($w = 30$msec, with an overlap of 10msec, see [22], for details). To make the visualization relatively independent of 'pitch', we use the so-called mel-cepstral representation (MFCC, $K = 13$ coefficients pr. frame). To reduce noise in the visualization we have 'sparsified' the amplitudes. This was achieved simply by retaining only coefficients that belonged to the upper 5% magnitude fractile. The total number of frames in the analysis was $F = 10^5$. PCA provided unsupervised latent semantic dimensions and a scatter plot of the data on the subspace spanned ny two such dimensions is shown in Figure 4. For interpretation we have coded the data points with signatures of the three genres involved. The ICA ray-structure is striking, however, we note that the situation is not one-to-one as in the small text databases. A component quantifies a characteristic 'theme' at the temporal level of a frame (30msec), it is an issue for further research whether genre *recognition* can be done from the salient themes, or we need to combine more than one theme to reach the classification performance obtained in [22] for $10-30$ second un-structured frame sets.

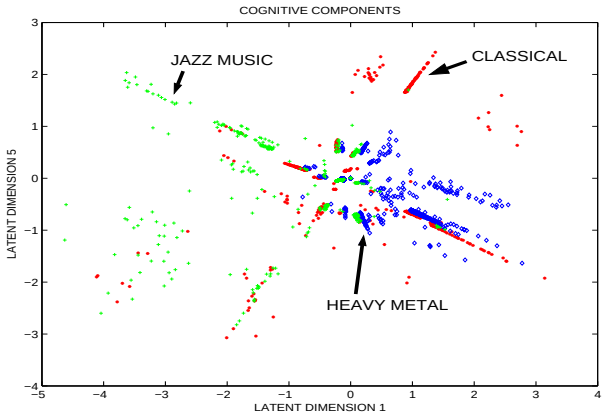


Figure 4. We represent three music tunes (with labels: `heavy metal, jazz, classical`) by their spectral content in overlapping small time frames ($w = 30$msec, with an overlap of 10msec, see [22], for details). To make the visualization relatively independent of 'pitch', we use the so-called mel-cepstral representation (MFCC, $K = 13$ coefficients pr. frame). To reduce noise in the visualization we have 'sparsified' the amplitudes. This was achieved simple by keeping coefficients that belonged to the upper 5% magnitude fractile. The total number of frames in the analysis was $F = 10^5$. Latent semantic analysis provided unsupervised subspaces with maximal variance for a given dimension. We show the scatter plot of the data on a 2D subspace within an original 5D PCA. For interpretation we have coded the data points with signatures of the three genres involved: classical ($*$), heavy metal (diamond), jazz ($+$). The ICA ray-structure is striking, however, note that the situation is not one-to-one (ray to genre) as in the small text databases. A component (ray) quantifies a characteristic musical 'theme' at the temporal level of a frame (30msec), i.e., an entity similar to the 'phoneme' in speech.

4. CONCLUSION

Cognitive component analysis (COCA) was defined as the process of unsupervised grouping of data such that the ensuing group structure is well-aligned with that resulting from human cognitive activity. It is well-established that information theoretically optimal representations, similar to those found by ICA, are in use in several information processing tasks in human and animal perception. By visualization of data using latent semantic analysis-like plots, we have shown that independent components analysis is also relevant for representing semantic structure, in text and also in other abstract data such as social networks, and musical features. We therefore speculate that the cognitive machinery developed for analyzing complex perceptual signals from multi-agent environments may also be used in higher brain function, such as understanding music or navigation of complex social networks, a hallmark of successful cognition. Hence, independent component analysis given the right representation may be a quite generic tool for COCA.

5. ACKNOWLEDGMENTS

This work is supported by the Danish Technical Research Council, through the framework project 'Intelligent Sound', www.intelligentsound.org (STVF No. 26-04-0092).

6. REFERENCES

[1] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley & Sons, 2001.

[2] Anthony J. Bell and Terrence J. Sejnowski, "The 'independent components' of natural scenes are edge filters," *Vision Research*, vol. 37, no. 23, pp. 3327–3338, 1997.

[3] Patrik Hoyer and Aapo Hyvrinen, "Independent component analysis applied to feature extraction from colour and stereo images.," *Network: Comput. Neural Syst.*, vol. 11, no. 3, pp. 191–210, 2000.

[4] M.S. Lewicki, "Efficient coding of natural sounds,"

Nature Neuroscience, vol. 5, no. 4, pp. 356–363, 2002.

[5] Jorge Wagensberg, "Complexity versus uncertainty: The question of staying alife," *Biology and philosophy*, vol. 15, pp. 493–508, 2000.

[6] C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, 1995.

[7] David Rawlings, Neus Barrantes-Vidal, Gordon Claridge, Charles McCreery, and Georgina Galanos, "A factor analytic study of the hypomanic personality scale in british, spanish and australian samples.," *Personality and Individual Differences*, vol. 28, pp. 73–84, 2000.

[8] C.H. Waechtler, E.A. Zillmer, M.J. Chelder, and B. Holde, "Neuropsychological patterns of component factor scores from the positive and negative syndrome scale (panss) in schizophrenics.," *Archives of Clinical Neuropsychology (Abstracts)*, vol. 10, pp. 400, 1995.

[9] L. K. Hansen, J. Larsen, and T. Kolenda, "On independent component analysis for multimedia signals," in *Multimedia Image and Video Processing*, pp. 175–199. CRC Press, sep 2000.

[10] L. K. Hansen, J. Larsen, and T. Kolenda, "Blind detection of independent dynamic components," in *IEEE International Conference on Acoustics, Speech, and Signal Processing 2001*, 2001, vol. 5, pp. 3197–3200.

[11] T. Kolenda, L. K. Hansen, and J. Larsen, "Signal detection using ICA: Application to chat room topic spotting," in *Third International Conference on Independent Component Analysis and Blind Source Separation*, 2001, pp. 540–545.

[12] T. Kolenda, L. K. Hansen, J. Larsen, and O. Winther, "Independent component analysis for understanding multimedia content," in *Proceedings of IEEE Workshop on Neural Networks for Signal Processing XII*, H. Bourlard et al. Ed., Piscataway, New Jersey, 2002, pp. 757–766, IEEE Press, Martigny, Valais, Switzerland, Sept. 4-6, 2002.

[13] J. Larsen, L.K. Hansen, T. Kolenda, and F.AA. Nielsen, "Independent component analysis in multimedia modeling," in *Fourth International Symposion on Independent Component Analysis and Blind Source Separation*, Shun ichi Amari et al. Ed., Nara, Japan, apr 2003, pp. 687–696, Invited Paper.

[14] D.D. Lee and H.S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[15] Pedro A. D. F. R. Højen-Sørensen, Ole Winther, and Lars Kai Hansen, "Mean-field approaches to independent component analysis," *Neural Comput.*, vol. 14, no. 4, pp. 889–918, 2002.

[16] Gerard Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.*, Addison-Wesley, 1989.

[17] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman, "Indexing by latent semantic analysis.," *JASIS*, vol. 41, no. 6, pp. 391–407, 1990.

[18] T.K̃. Landauer, D. Laham, and M. Derr, "From paragraph to graph: latent semantic analysis for information visualization.," *Proc Natl Acad Sci*, vol. 101, no. Sup. 1, pp. 5214–5219, 2004.

[19] David L. Donoho and Victoria Stodden, "When does non-negative matrix factorization give a correct decomposition into parts?," in *NIPS*, 2003.

[20] T. Kolenda, L. K. Hansen, and S. Sigurdsson, "Indepedent components in text," in *Advances in Independent Component Analysis*, pp. 229–250. Springer-Verlag, 2000.

[21] A.-L. Barabasi and R. Albert, "Emergence of scaling in random networks.," *Science*, vol. 286, pp. 509–512, 1999.

[22] P. Ahrendt, A. Meng, and J. Larsen, "Decision Time Horizon For Music Genre Classification Using Short Time Features," in *EUSIPCO*, Vienna, Austria, sep 2004, pp. 1293–1296.

Appendix G

Feature Integration for Music Genre Classification

Meng A., Ahrendt P., Larsen J. and Hansen L. K., **Feature Integration for Music Genre Classification**, yet unpublished article, 2006.

Feature Integration for Music Genre Classification

Anders Meng, Peter Ahrendt, Jan Larsen and Lars Kai Hansen,
(am,pa,jl,lkh@imm.dtu.dk)
Technical University of Denmark, Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark

September 23, 2005

Abstract. Feature integration is the process of combining all the feature vectors in a time frame into a single feature vector in order to captures the relevant information in the frame. The mean and variance along the temporal dimension are often used for feature integration, but captures neither the temporal dynamics nor dependencies among the individual feature dimensions. Here, a multivariate autoregressive feature model is proposed to solve this problem for music genre classification. This model gives two different feature sets, the DAR and MAR features, which are compared against the baseline mean-variance as well as two other feature integration techniques. Reproducibility in performance ranking of feature integration methods were demonstrated using two data sets with five and eleven music genres, and by using four different classification schemes. The methods were further compared to human performance. The proposed MAR features perform significantly better than the other features without much increase in computational complexity.

Keywords: Feature integration, autoregressive model, music genre classification

1. Introduction

In recent years, there has been an increasing interest in the research area of Music Information Retrieval (MIR). This is spawned by the new possibilities on the Internet such as on-line music stores like Apple's iTunes and the enhanced capabilities of ordinary computers. The related topic of music genre classification can be defined as computer-assigned genre labelling of pieces of music. It has received much attention in its own right, but it is also often used as a good test-bench for music features in related areas where the labels are harder to obtain than the musical genres. An example of this is (Gouyon et al., 2004), where rhythm features are assessed in a music genre classification task.

Music genre classification systems normally consist of feature extraction from the digitized music, followed by a classifier that uses features to estimate the genre. In this work we focus on identifying features integration methods, which give consistent good performance over different data sets and choices of classifier.

In several feature extraction models, perceptual characteristics such as the beat (Foote and Uchihashi, 2001) or pitch (Tzanetakis, 2002) are modelled directly. This has the clear advantage of giving features which

can be examined directly without the need of a classifier. However, most of the previous research has concentrated on short-time features e.g. Audio Spectrum Envelope and the Zero-Crossing Rate (Ahrendt et al., 2004)) which are extracted from $20-40$ ms frames of the song. Such features are thought to represent perceptually relevant characteristics such as e.g. music roughness or timbre. They have to be evaluated as part of a full classification system. A song or sound clip is thus represented by a multivariate time series of these features and different methods exist to fuse this information into a single genre label for the whole song. An example is (Soltau et al., 1998), based on a hidden Markov model of the time series of the cepstral coefficient features.

Feature integration is another approach to information fusion. It uses a sequence of short-time feature vectors to create a single new feature vector at a larger time scale. It assumes that the short-time features describe all (or most) of the important information for music genre classification. Feature integration is a very common technique. Often basic statistic estimates like the mean and variance of the short-time features have been used (Srinivasan and Kankanhalli, 2004; Zhang and Zhou, 2004; Tzanetakis, 2002). Another similar feature is the mean-covariance feature which simply uses the upper triangular part of the covariance matrix instead of the diagonal.

Here, a new multivariate autoregressive feature integration model is proposed as an alternative to the mean-variance feature set. The main advantage of the autoregressive model is its ability to model temporal dynamics as well as dependencies among the short-time feature dimensions. In fact, the model is a natural generalization of the mean-variance feature integration model.

Figure 1 illustrates the full music genre classification system which was used for evaluating the feature integration methods.

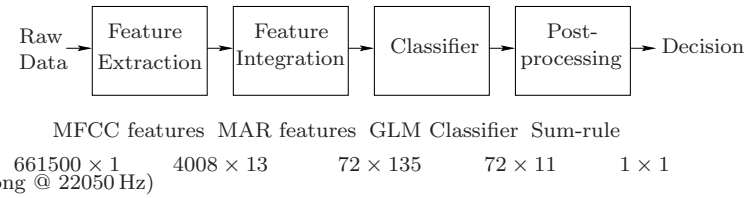


Figure 1. The full music genre classification system. The flow-chart illustrates the different parts of the system, whereas the names just below the chart are the specific choices that gives the best performing system. The numbers in the bottom part of the figure illustrates the (large) dimensionality reduction that takes place in such a system (the number of genres are 11).

Section 2 describes common feature extraction and integration methods, while section 3 gives a detailed explanation of the proposed multivariate autoregressive feature model. Section 4 reports and discusses the results of experiments that compare the newly proposed features with the best of the existing feature integration methods. Finally, section 5 concludes on the results.

2. Feature extraction and integration

Several different features have been suggested in music genre classification. The general idea is to process fixed-size time windows of the digitized audio signal with an algorithm which can extract the most vital information in the audio segment. The size of the windows gives the time scale of the feature. The features are often thought to represent aspects of the music such as the pitch, instrumentation, harmonicity or rhythm.

The following subsections explain popular feature extraction methods. They are listed on the basis of their time scale. The process of feature integration is explained in detail in the end of the section.

2.1. Short-time features

Most of the features that have been proposed in the literature are short-time features which usually employ window sizes of $20-40$ ms. They are often based on a transformation to the spectral domain using techniques such as the Short-Time Fourier Transform. The assumption in these spectral representations is (short-time) stationarity of the signal which means that the window size has to be small.

In (Ahrendt et al., 2004), we found the so-called *Mel-Frequency Cepstral Coefficient* (MFCC) to be very successful. Similar findings were observed in (H.-Gook. and Sikora, 2004) and (Herrera et al., 2002). They were originally developed for speech processing (Rabiner and Juang, 1993). The details of the MFCC feature extraction are shown in figure 2. It should be mentioned, however, that other slightly different MFCC feature extraction schemes exist.

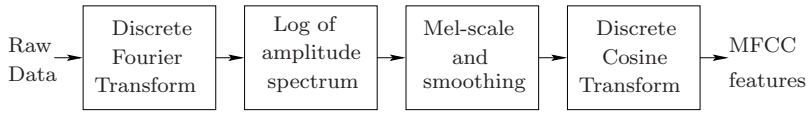


Figure 2. MFCC feature extraction as described in (Logan, 2000).

According to (Aucouturier and Pachet, 2003), short-time representations of the full time-frequency domain, such as the MFCC features, can be seen as models of the music timbre.

2.2. Medium-time features

Medium-time features are here defined as features which are extracted on time scales around $1000-2000$ ms. (Tzanetakis, 2002) uses the term *Texture window* for this time scale where important aspects of the music lives such as note changes and tremolo (Martin, 1999). Examples of features for this time scale are the Low Short-Time Energy Ratio (LSTER) and High Zero-Crossing Rate Ratio (HZCRR) (Lu et al., 2002).

2.3. Long-time features

Long-time features describe important statistics of e.g. a full song or a larger sound clip. An example is the beat histogram feature (Tzanetakis and Cook, 2002), which summarize the beat content in a sound clip.

2.4. Feature Integration

Feature integration is the process of combining all the feature vectors in a time frame into a single feature vector which captures the information of this frame. The new features generated do not necessarily capture any explicit perceptual meaning such as perceptual beat or mood, but captures implicit perceptual information which are useful for the subsequent classifier. In (Foote and Uchihashi, 2001) the "beat-spectrum" is used for music retrieval by rhythmic similarity. The beat-spectrum can be derived from short-time features such as STFT or MFCCs as noted in (Foote and Uchihashi, 2001). This clearly indicates that short-time features carry important perceptual information across time, which is one of the reasons for modelling the temporal behavior of short-time features. Figure 3 shows the first six MFCCs of a ten second excerpt of the music piece "Masters of Revenge" by "Body Count". This example shows a clear repetitive structure in the short-time features. Another important property of feature integration is data reduction. Consider a four minute piece of music represented as short-time features (using the first 6 MFCCs). With a hop- and framesize of 10 ms and 20 ms, respectively, this results in approximately 288 kB of data using a 16 bit representation of the features. The hopsize is defined as the framesize minus the amount of overlap between frames and specifies the "effective sampling rate" of the features. This is a rather good compression compared to the original size of the music (3.84 MB, *MPEG1-layer* 3

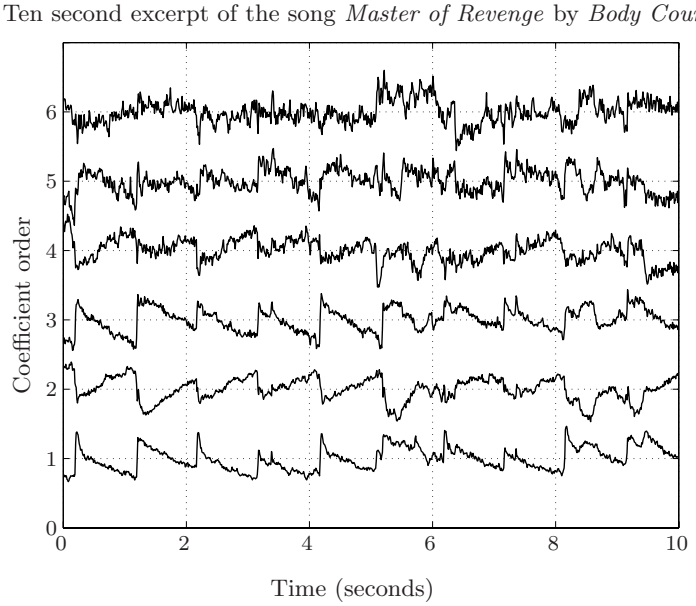


Figure 3. The first six normalized MFCCs of a ten second snippet of "Body Count - Masters of Revenge". The temporal correlations is very clear from this piece of music as well as the cross-correlations among the feature dimensions. This suggests that relevant information is present and could be extracted by selecting a proper feature integration model.

@ 128 kBit). However, if the relevant information can be summarized more efficiently in less space, this must be preferred.

The idea of feature integration can be expressed more rigorously by observing a sequence of consecutive short-time features, $\mathbf{x}_i \in \mathcal{R}^D$ where i represents the i'th short time feature and D is the feature dimension. These are integrated into a new feature $\mathbf{z}_k \in \mathcal{R}^M$

$$\mathbf{z}_k = \mathbf{f}(\mathbf{x}_{(k-1)H_s+1}, \ldots, \mathbf{x}_{(k-1)H_s+F_s}), \tag{1}$$

where H_s is the *hopsize* and F_s *framesize* (both defined in number of samples) and $k = 1, 2, \ldots$ is the discrete time index of the larger time scale. There exists a lot of different models, here denoted by $\mathbf{f}(.)$ which maps a sequence of short-time features into a new feature vector.

In the following the *MeanVar*, *MeanCov* and *Filterbank Coefficients* will be discussed. These methods have been suggested for feature integration in the literature.

2.4.1. *Gaussian model*

A very simple model for feature integration is the so-called *MeanVar* model, which has been used in work related to music genre classification, see e.g. (Tzanetakis and Cook, 2002; Meng et al., 2005). This

model implicitly assumes that consecutive samples of short-time features are independent and Gaussian distributed and, furthermore, that each feature dimension is independent. Using maximum-likelihood the parameters for this model are estimated as

$$\begin{aligned} \mathbf{m}_k &= \frac{1}{F_s}\sum_{n=1}^{F_s}\mathbf{x}_{(k-1)H_s+n} \\ c_{k,i} &= \frac{1}{F_s}\sum_{n=1}^{F_s}\left(x_{(k-1)H_s+n,i} - m_{k,i}\right)^2 \end{aligned}$$

for $i = 1, \ldots, D$, which results in the following feature at the new time scale

$$\mathbf{z}_k = \mathbf{f}(\mathbf{x}_{(k-1)H_s+1}, \ldots, \mathbf{x}_{(k-1)H_s+F_s}) = \begin{bmatrix} \mathbf{m}_k \\ \mathbf{c}_k \end{bmatrix}, \tag{2}$$

where $\mathbf{z}_k \in \mathcal{R}^{2D}$. As seen in figure 3, the assumption that each feature dimension is independent is not correct. A more reasonable feature integration model is the multivariate Gaussian model, denoted in the experimental section as MeanCov, where correlations among features are modelled. This model of the short-time features can be formulated as $\mathbf{x} \sim \mathcal{N}(\mathbf{m}, \mathbf{C})$, where the mean and covariance are calculated over the given feature integration window. Thus, the diagonal of $\mathbf{C}$ contains the variance features from MeanVar. The mean vector and covariance matrix are stacked into a new feature vector $\mathbf{z}_k$ of dimension $\frac{D}{2}(3+D)$.

$$\mathbf{z}_k = \begin{bmatrix} \mathbf{m}_k \\ \text{vech}(\mathbf{C}_k) \end{bmatrix}, \tag{3}$$

where vech($\mathbf{C}$) refers to stacking the upper triangular part of the matrix including the diagonal.

One of the drawbacks of the Gaussian model, whether this is the simple (MeanVar) or the multivariate model (MeanCov), is that the temporal dependence of the data is not modelled.

2.4.2. *Filter-bank coefficients (FC)*
The filter-bank approach was considered in (McKinney and Breebaart, 2003) aims at capturing some of the dynamics in the sequence of short-time features. They investigated the method in a general audio and music genre classification task. The idea is to extract a summarized power of each feature dimension independently in four specified frequency bands. The feature integration function $\mathbf{f}(.)$ for the filter bank approach can be written compactly as

$$\mathbf{z}_k = \text{vec}\left(\mathbf{P}_k\mathbf{W}\right), \tag{4}$$

where $\mathbf{W}$ is a filter matrix of dimension $N \times 4$ and $\mathbf{P}_k$ contains the estimated power spectrum of each short-time feature and has dimension $D \times N$, where $N = F_s/2$ when F_s is even and $N = (F_s - 1)/2$ for odd values.

The four frequency bands in which the power is summarized are specified in the matrix $\mathbf{W}$. In (McKinney and Breebaart, 2003) the four filters applied to handle the short-time features are: 1) a DC-filter, 2) $1 - 2$ Hz modulation energy, 3) $3 - 15$ Hz modulation energy and 4) $20 - 43$ Hz modulation energy.

The advantage of this method is that the temporal structure of the short-time features is taken into account, however, correlations among feature dimensions are not modelled. In order to model these, cross-correlation spectra would be required.

3. Multivariate Autoregressive Model for feature integration

The simple mean-variance model does not model temporal feature correlations, however, these features have shown to perform remarkably well in various areas of music information retrieval, see e.g. (Tzanetakis and Cook, 2002; Ellis and Lee, 2004). The dependencies among features could be modelled using the MeanCov model, but still do not model the temporal correlations. The filterbank coefficient (FC) approach includes temporal information in the integrated features, but the correlations among features are neglected.

This section will focus on the multivariate autoregressive model (*MAR*) for feature integration, since it has the potential of modelling both temporal correlations and dependencies among features.

For simplicity we will first study the diagonal multivariate autoregressive model (DAR). The DAR model assumes independence among feature dimensions similar to the MeanVar and FC feature integration approaches. The full multivariate autoregressive model (MAR) in considered in section 3.2.

3.1. Diagonal multivariate autoregressive model (*DAR*)

The DAR model was investigated in (Meng et al., 2005) where different feature integration methods were tested and showed improved performance compared to the MeanVar and FC approaches, however, the theory behind the model was not fully covered. For completeness we will present a more detailed description of the model.

Assuming independence among feature dimensions the P'th order causal autoregressive model for each feature dimension can be written

as

$$x_n = \sum_{p=1}^{P} a_p x_{n-p} + G u_n, \tag{5}$$

where a_p, for $p = 1, .., P$ is the autoregressive coefficients, u_n is the noise term, assumed i.i.d. with unit variance and mean value v. Note that the mean value of the noise process v is related to the mean m of the time series by $m = (1 - \sum_{p=1}^{P} a_p)^{-1} v$.

Equation 5 expresses the "output" x_n as a linear function of past outputs and present inputs u_n. There are several methods for estimating the parameters of the autoregressive model, either in the frequency domain (Makhoul, 1975) or directly in time-domain (Lütkepohl, 1993). The most obvious and well-known method is the ordinary least squares method, where the mean squared error is minimized. Other methods suggested are the generalized (or weighted) least squares where the noise process is allowed to be colored. In our case the noise process is assumed white, therefore the least squares method is applied and described in the following. The prediction of a new sample based on estimated parameters, a_p, becomes

$$\tilde{x}_n = \sum_{p=1}^{P} a_p x_{n-p}, \tag{6}$$

and the error signal e_n measured between $\tilde{x}_n$ and x_n is

$$e_n = x_n - \tilde{x}_n = x_n - \sum_{p=1}^{P} a_p x_{n-p}, \tag{7}$$

where e_n is known as the residual. Taking the z-transformation on both sides of equation 7, the error can now be written as

$$E(z) = \left(1 - \sum_{p=1}^{P} a_p z^{-p} \right) X(z) = A(z) X(z). \tag{8}$$

In the following we will switch to frequency representation $z = e^{j\omega}$ and in functions use $X(\omega)$ for representing $X(e^{j\omega})$. Assuming a finite energy signal, x_n, the total error to be minimized in the ordinary least squares method, $\mathcal{E}_{tot}$, is then according to Parseval's theorem given by

$$\mathcal{E}_{tot} = \sum_{n=0}^{F_s} e_n^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |E(\omega)|^2 d\omega. \tag{9}$$

To understand why this model is worthwhile to consider, we will now explain the spectral matching capabilities of the model. First, we

look at the model from equation 5 in the z-transformed domain which can now be described as

$$X(z) = \sum_{p=1}^{P} a_p X(z) z^{-p} + GU(z), \tag{10}$$

where $v = 0$ is assumed without loss of generalizability. The gain factor G sets the scale. The system transfer function becomes

$$H(z) \equiv \frac{X(z)}{U(z)} = \frac{G}{1 - \sum_{p=1}^{P} a_p z^{-p}}, \tag{11}$$

and its corresponding model power spectrum

$$\hat{P}(\omega) = |H(\omega)U(\omega)|^2 = |H(\omega)|^2 = \frac{G^2}{|A(\omega)|^2}. \tag{12}$$

Combining the information in equations 8, 9, 12 and the fact that $P(\omega) = |X(\omega)|^2$, the total error to be minimized can be written as

$$\mathcal{E}_{tot} = \frac{G^2}{2\pi} \int_{-\pi}^{\pi} \frac{P(\omega)}{\hat{P}(\omega)} d\omega. \tag{13}$$

The first observation is that trying to minimize the total error $\mathcal{E}_{tot}$ is equivalent to minimization of the integrated ratio of the signal spectrum $P(\omega)$ and its estimated spectrum $\hat{P}(\omega)$. Furthermore, at minimum error $\mathcal{E}_{tot} = G^2$ the following relation holds

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{P(\omega)}{\hat{P}(\omega)} d\omega = 1. \tag{14}$$

The two equations 13 and 14 result in two major properties, a *global* and *local* property (Makhoul, 1975):

- The global property states that since the contribution to the total error $\mathcal{E}_{tot}$ is determined as a ratio of the two spectra, the matching process should perform uniformly over the whole frequency range, irrespective of the shaping of the spectrum. This means that the spectrum match at frequencies with small energy is just as good as frequencies with high energy.

- The local property deals with the matching of the spectrum in each small region of the spectrum. (Makhoul, 1975) basically concludes that a better fit of $\hat{P}(\omega)$ to $P(\omega)$ will be obtained at frequencies where $P(\omega)$ is larger than $\hat{P}(\omega)$, than at frequencies where $P(\omega)$ is smaller. Thus, for harmonic signals the peaks will be better approximated than the area in between the harmonics.

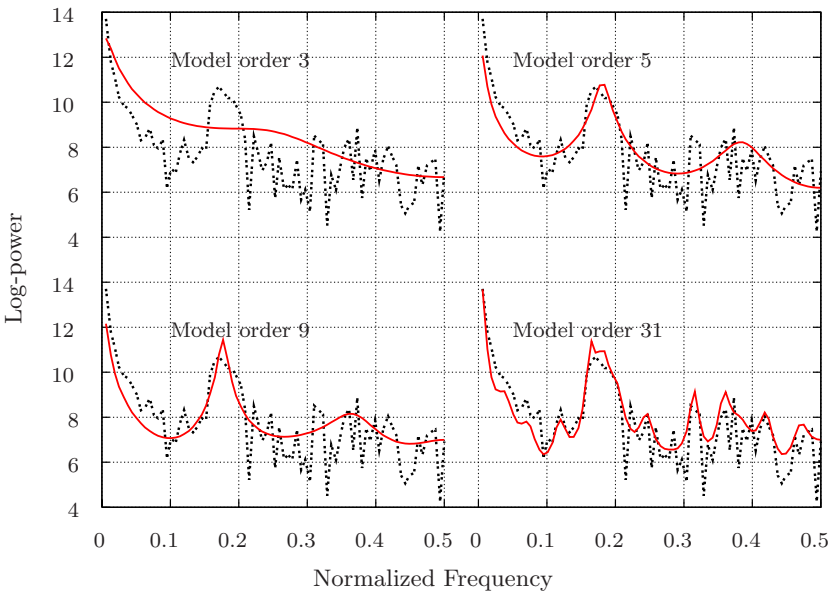


Figure 4. Power density of a first order MFCC of a piano note $A5$ played for a duration of 1.2 s. The four figures show the periodogram as well as the AR-model power spectrum estimates of orders $3, 5, 9$ and 31, respectively.

It is now seen that there is a clear relationship between the AR-model and the FC approach since in the latter method, the power spectrum is summarized in four frequency bands. With the AR-model approach selection of proper frequency bands is unnecessary since the power spectrum is modelled directly.

Figure 4 shows the periodogram of the first order MFCC coefficient of the piano note $A5$ corresponding to the frequency 880 Hz recorded over a duration of 1.2 seconds as well as the AR-model approximation for four different model orders, $3, 5, 9$ and 31. The hopsize of the MFCCs were 7.5 ms corresponding to a samplerate of 133.33 Hz. As expected, the model power spectrum becomes more detailed as the model order increases.

3.2. Multivariate autoregressive model (MAR)

In order to include both temporal and among feature correlations the multivariate AR model with full matrices is applied instead of only considering the diagonal of the matrices as in the DAR model. A full treatment of the MAR models are given in (Lütkepohl, 1993) and (Neumaier and Schneider, 2001).

For a stationary time series of state vectors $\mathbf{x}_n$ the multivariate AR model is defined by

$$\mathbf{x}_n = \sum_{p=0}^{P} \mathbf{A}_p \mathbf{x}_{n-p} + \mathbf{u}_n \tag{15}$$

where the noise term $\mathbf{u}_n$ is assumed i.i.d. with mean $\mathbf{v}$ and finite covariance matrix $\mathbf{C}$. Note that the mean value of the noise process $\mathbf{v}$ is related to the mean $\mathbf{m}$ of the time series by $\mathbf{m} = (\mathbf{I} - \sum_{p=1}^{P} \mathbf{A}_p)^{-1}\mathbf{v}$.

The matrices $\mathbf{A}_p$ for $p = 1, \ldots, P$ are the coefficient matrices of the P'th order multivariate autoregressive model. They encode how much of the previous information in $\{\mathbf{x}_{n-1}, \mathbf{x}_{n-2}, \ldots, \mathbf{x}_{n-P}\}$ is present in $\mathbf{x}_n$.

A frequency interpretation of the vector autoregressive model can, as for the univariate case, be established for the multivariate case. The main difference is that all cross spectra are modelled by the MAR model. In e.g. (Bach and Jordan, 2004), a frequency domain approach is used for explaining the multivariate autoregressive model by introducing the *autocovariance function*, which contains all cross covariances for the multivariate case. The power spectral matrix can be defined from the autocovariance function as

$$\mathbf{f}(\omega) = \sum_{h=-Fs+1}^{Fs-1} \mathbf{\Gamma}(h) e^{-ih\omega}, \tag{16}$$

where the autocovariance function $\mathbf{\Gamma}(h)$ is a positive function and fulfills $\sum_{h=-\infty}^{\infty} ||\mathbf{\Gamma}(h)||_2 < \infty$, under stationarity.

As with the DAR model the ordinary least squares approach has been used in estimating the parameters of the MAR model, see e.g. (Lütkepohl, 1993) for detailed explanation of parameter estimation.

The parameters which are extracted from the least squares approach for both the DAR and MAR models are the AR-matrices: $\{\mathbf{A}_1, \ldots, \mathbf{A}_P\}$, the intercept term $\mathbf{v}$ and the noise covariance $\mathbf{C}$. The feature integrated vector of frame k then becomes

$$\mathbf{z}_k = [\mathrm{vec}\,(\mathbf{B}_k)^T\, \mathbf{v}_k^T \mathrm{vech}\,(\mathbf{C}_k)^T]^T, \tag{17}$$

where $\mathbf{B} = [\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_P] \in \mathcal{R}^{D \times PD}$ and $\mathbf{z}_k \in \mathcal{R}^{(P+1/2)D^2+(3/2)D}$. Note that for the DAR model, only the diagonals of the $\mathbf{A}_p$ matrices are used as well as only the diagonal of $\mathbf{C}$.

3.2.1. *Issues on stability*

Until now we have assumed that the time-series under investigation is stationary over the given feature integration frame. The frame-size,

however, is optimized to the given learning problem which means that we are not guaranteed that the time-series is stationary within each frame. This could e.g. be in transitions from silence to audio, where the time-series might locally look non-stationary. In some applications, this is not a problem, since reasonable parameter estimates are obtained anyhow. In the considered music genre setup, the classifier seems to handle the non-stationary estimates reasonably. In other areas of music information retrieval, the power-spectrum estimate provided through the AR-model might be more critical, hence, in such cases it would be relevant to investigate the influence of non-stationary frames.

3.2.2. *Selection of optimal length*

There exists multiple order selection criteria. Examples are BIC (Bayesian Information Criterion) and AIC (Akaike Information Criterion), see e.g. (Neumaier and Schneider, 2001). The order selection methods are traditionally applied on a single time series, however, in the music genre setup, we are interested in finding one single optimal model order for a large set of time-series. Additionally, there is a tradeoff between model order and feature space dimensionality and, hence, problems with overfitting of the subsequent classifier, see figure 1. Therefore, the optimal order of the time-series alone is normally not the same as the optimal order for the vector time-series.

3.3. Complexity considerations

Table I shows the complete number of multiplications and additions for a frame of all the examined feature integration methods. The column "multiplications & additions" shows the number of calculated multiplications / additions of the particular method. D is the dimensionality of the feature space, P is the DAR/MAR model order, and F_s is the framesize in number of short-time feature samples. In the calculations the effect of overlapping frames have not been exploited. Figure 5 shows the computational complexity in our actual music genre setup.

4. Experiments

Quite a few simulations were made to compare the baseline MeanVar features with the newly proposed DAR and MAR features. Additionally, the FC features and MeanCov features were included in the comparisons. The FC features performed very well in (Meng et al., 2005) and the MeanCov features were included for the sake of completeness.

Table I. Computational complexity of algorithms of a frame of short-time features

METHOD	MULTIPLICATIONS & ADDITIONS
MeanVar	$4DF_s$
MeanCov	$(D+3)DF_s$
FC	$(4\log_2(F_s)+3)\,DF_s$
DAR	$\frac{D}{3}(P+1)^3+((P+6)(P+1)+3)\,DF_s$
MAR	$\frac{1}{3}(PD+1)^3+$ $\left((P+4+\frac{2}{D})(PD+1)+(D+2)\right)DF_s$

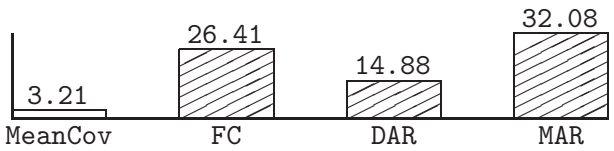


Figure 5. Computational complexity of the music genre setup using the optimized values from the experimental section, hence $P = 3$, $D = 6$ and $F_s = 188, 268, 322, 188, 162$ for the MeanVar, MeanCov, FC, DAR and MAR, respectively. Note that the complexity values are scaled such that the MeanVar has complexity 1.

The features were tested on two different data sets and four different classifiers to make the conclusions generalizable. In all of the experiments, 10-fold cross-validation was used to estimate the mean and standard deviation of the mean classification test accuracy, which was used as the performance measure. Figure 1 in section 1 illustrates the complete classification system. The optimization of the system follows the data stream, which means that the MFCC features were optimized first (choosing number of coefficients to use, whether to use normalization etc.). Afterwards, the feature integration part was optimized and so forth.

4.1. Preliminary investigations

Several investigations of preprocessing both before and after the feature integration were made. Dimensionality reduction of the high-dimensional MAR and DAR features by PCA did not prove beneficial[1], and neither did whitening (making the feature vector representation zero-mean and unit covariance matrix) or normalization (making each feature component zero-mean and unit variance individually) for any of the features. To avoid numerical problems, however, they were all normalized. Preprocessing, in terms of normalization of the short-time MFCC features didn't seem to have an effect either.

4.2. Features

To ensure a fair comparison between the features, their optimal hop- and framesizes were examined individually since especially framesize seems important with respect to classification accuracy. An example of the importance of the framesize is illustrated in figure 6.

For the short-time MFCC features, optimal hop- and framesizes were found to be 7.5 ms and 15 ms, respectively. The optimal hopsize was 400 ms for the DAR, MAR, MeanVar and MeanCov features and 500 ms for the FC features. The framesizes were 1200 ms for the MAR features, 2200 ms for the DAR features, 1400 ms for the MeanVar, 2000 ms for the MeanCov and 2400 ms for the FC features.

An important parameter in the DAR and MAR feature models is the model order parameter P. The optimal values for this parameter were found to be 5 and 3 for the DAR and MAR features, respectively. This optimization was based on the large data set B, see section 4.6. Using these parameters, the resulting dimensions of the feature spaces become : MAR - 135, DAR - 42, FC - 24, MeanCov - 27 and MeanVar - 12.

4.3. Classification and Post-processing

Several classifiers have been tested such as a linear model trained by minimizing least squares error (LM), Gaussian classifier with full covariance matrix (GC), Gaussian mixture model (GMM) classifier with full covariance matrices and a Generalized Linear Model (GLM) classifier (Nabney and Bishop, 1995). Due to robust behavior, the LM and GLM classifiers have been used in all of the initial feature investigations.

The LM classifier is simply a linear regression classifier, but has the advantage of being fast and non-iterative since the training essentially

[1] This is only true for the standard GLM and LM classifiers, that does not have significant overfitting problems.

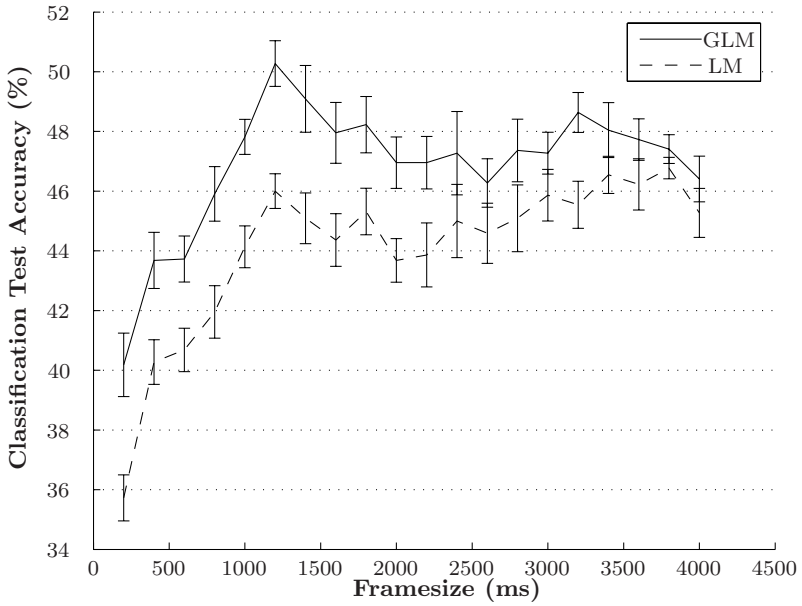


Figure 6. Classification test accuracy is plotted against framesize for the MAR features using the LM and GLM classifiers. The hopsize was 200 ms in these experiments and data set B, section 4.6, was used. The importance of the framesize is clearly seen. The baseline classification accuracy by random guessing is $\sim 9.1\%$.

amounts to finding the pseudo-inverse of the feature-matrix. The GLM classifier is the extension of a logistic regression classifier to more than two classes. It can also be seen as an extension of the LM classifier, but with inclusion of a regularisation term (prior) on the weights and a cross-entropy error measure to account for the discrete classes. They are both discriminative, which could explain their robust behavior in the fairly high-dimensional feature space. 10-fold cross validation was used to set the prior of the GLM classifier.

4.3.1. *Post-processing*

Majority voting and sum-rule were examined to integrate the c classifier outputs of all the medium-time frames into 30 s (the size of the song clips). Whereas majority voting counts the hard decisions $\arg\max_c P(c|\mathbf{z}_k)$ for $k = 1, \ldots, K$ of the classifier outputs, the sum-rule sums over the "soft" probability densities $P(c|\mathbf{z}_k)$ for $k = 1, \ldots, K$. The sum-rule was found to perform slightly better than majority voting.

4.4. Human evaluation

The level of performance in the music genre setups using various algorithms and methods only shows their relative differences. However, by estimating the human performance on the same data sets the quality of automatic genre classification systems can be assessed.

Listening tests have been conducted on both the small data set (A) and the larger data set (B) consisting of 5 and 11 music genres, respectively. At first, subsets of the full databases were picked randomly with equal amounts from each genre (25 of 100 and 220 of 1210) and these subsets are believed to represent the full databases. A group of people (22 specialists and non-specialists) were kindly asked to listen to 30 different snippets of length 10 s (randomly selected) from data set A and classify each music piece into one of the genres on a forced-choice basis. A similar setup was used for the larger data set B, but now 25 persons were asked to classify 33 music snippets of length 30 s. No prior information except the genre names were given to the test persons. The average human accuracy on data set A to lies in a 95%-confidence interval $[0.97; 0.99]$, and for data set B it is $[0.54; 0.61]$. Another interesting measure is the confusion between genres, which will be compared to the automatic music classifier in figure 8.

4.5. Data set A

The data set consists of 5 music genres distributed evenly among the categories: *Rock, Classical, Pop, Jazz* and *Techno*. It consists of 100 music snippets each of length 30 s. Each of the music snippets are recorded in mono PCM format at a sampling frequency of 22050 Hz.

4.6. Data set B

The data set consists of 11 music genres distributed evenly among the categories: *Alternative, Country, Easy Listening, Electronica, Jazz, Latin, Pop&Dance, Rap&HipHop, R&B Soul, Reggae* and *Rock*. It consists of 1210 music snippets each of length 30 s. The music snippets are *MPEG1-layer* 3 encoded music with a bit-rate of 128 kBit which were converted to mono PCM format with a sampling frequency of 22050 Hz.

4.7. Results and discussion

The main classification results are illustrated in figure 7 for both the small and the large data set. The figure compares the classification test accuracies of the FC and MeanCov features and the baseline MeanVar with the newly proposed DAR and MAR features. It is difficult to see

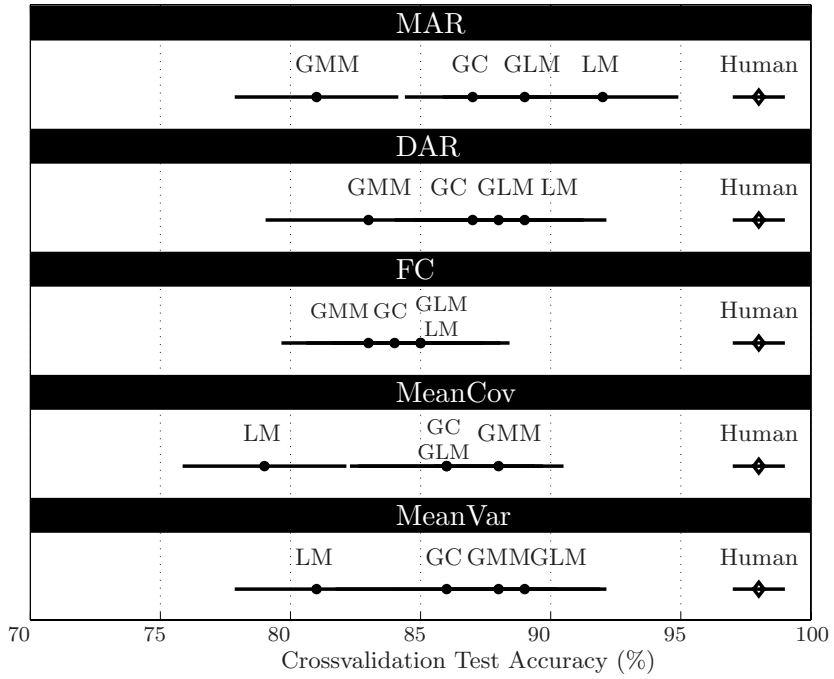
much difference in performance between the features for the small data set A, but note that it was created to have only slightly overlapping genres which could explain why all the features perform so well compared to the random guess of only 20% accuracy. The accuracies are all quite close to the average human classification accuracy of 98%.

The results from the more difficult, large data set B are shown on the lower part of figure 7. Here, the MAR features are seen to clearly outperform the conventional MeanVar features when the LM or GLM classifiers are used. Similarly, they outperform the MeanCov and DAR features. The DAR features only performed slightly better than the three reference features, but in a feature space of much lower dimensionality than the MAR features. The GMM classifier is the best for the low-dimensional MeanVar features, but gradually loses to the discriminative classifiers as the feature space dimensionality rises. This overfitting problem was obviously worst for the 135-dimensional MAR features and dimensionality reduction was necessary. However, a PCA subspace projection was not able to capture enough information to make the GMM classifier competitive for the MAR features. Improved accuracy of the GMM classifier on the MAR features was achieved by projecting the features into a subspace spanned by the $c-1$ weight directions of the partial least squares (PLS) (Shawe-Taylor and Cristianini, 2004), where c refers to the no. of genres. The classification accuracy, however, did not exceed the accuracy of the GLM classifier on the MAR features.

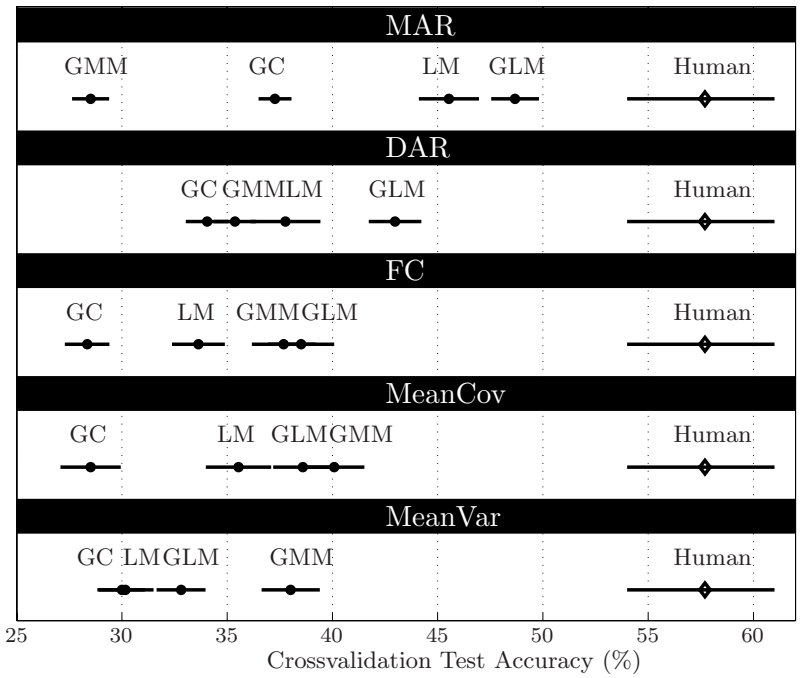
It is seen that the MAR features perform almost as well as humans which have an average classification test accuracy of 57%. Note that the random classification accuracy is only 9%.

The cross-validation paired t-test (Dietterich, 1998) was made on both data sets to test whether the best performances of the DAR and MAR features differed significantly from the best performances of the other features. Comparing the MAR features against the other four features gave t-statistics estimates all above 3.90; well above the 0.975 percentile critical value of $t_{9,0.975} = 2.26$ for 10-fold cross-validation. Thus, the null hypothesis of similar performance can be rejected. The comparison between the DAR features and the three reference features gave t-statistics estimates of 2.67 and 2.83 for the FC and MeanVar features, but only 1.56 for the MeanCov features which means that the null hypothesis cannot be rejected for the MeanCov.

As described in section 4.2, the framesizes were carefully investigated and the best results were found using framesizes in the range of 1200 ms to 2400 ms, followed by the sum-rule on the classifier decisions up to 30 s. However, in e.g. music retrieval and regarding computational speed and storage, it would be advantageous to model the whole 30 s music



(a) Experiment on data set A



(b) Experiment on data set B

Figure 7. The figures show the music genre classification test accuracies for the GC, GMM, LM and GLM classifiers on the five different integrated features. The results for the small data set A is shown in the upper panel of the figure and the results for the larger data set B in the lower panel. The mean accuracy of 10-fold cross-validation is shown along with error bars which are one ± standard deviation of the mean to each side. 95% binomial confidence intervals have been shown for the human accuracy.

snippet with a single feature vector. Hence, experiments were made with the MAR features with a framesize of 30 s, i.e. modelling the full song with a single MAR model. The best mean classification test accuracies on data set B were 44% and 40% for the LM and GLM classifiers, respectively, using a MAR model order of 3. In our view, this indicates that these MAR features could be used with success in e.g. song similarity tasks. Additional experiments with a Support Vector Machine (SVM) classifier (Meng and Shawe-Taylor, 2005) using a RBF kernel even improved the accuracy to 46%. The SVM classifier was used since it is less prone to overfitting. This is especially important when each song is represented by only one feature vector, which means that our training set only consists of $11 \cdot 99 = 1089$ samples in each cross-validation run.

Besides the classification test accuracy, an interesting measure of performance is the confusion matrix. Figure 8 illustrates the confusion matrix of the MAR system with highest classification test accuracy and shows the relation to the human genre confusion matrix on the large data set. It is worth noting that the three genres that humans classify correctly most often, i.e., Country, Rap&HipHop and Reggae, are also the three genres that our classification system typically classifies correctly.

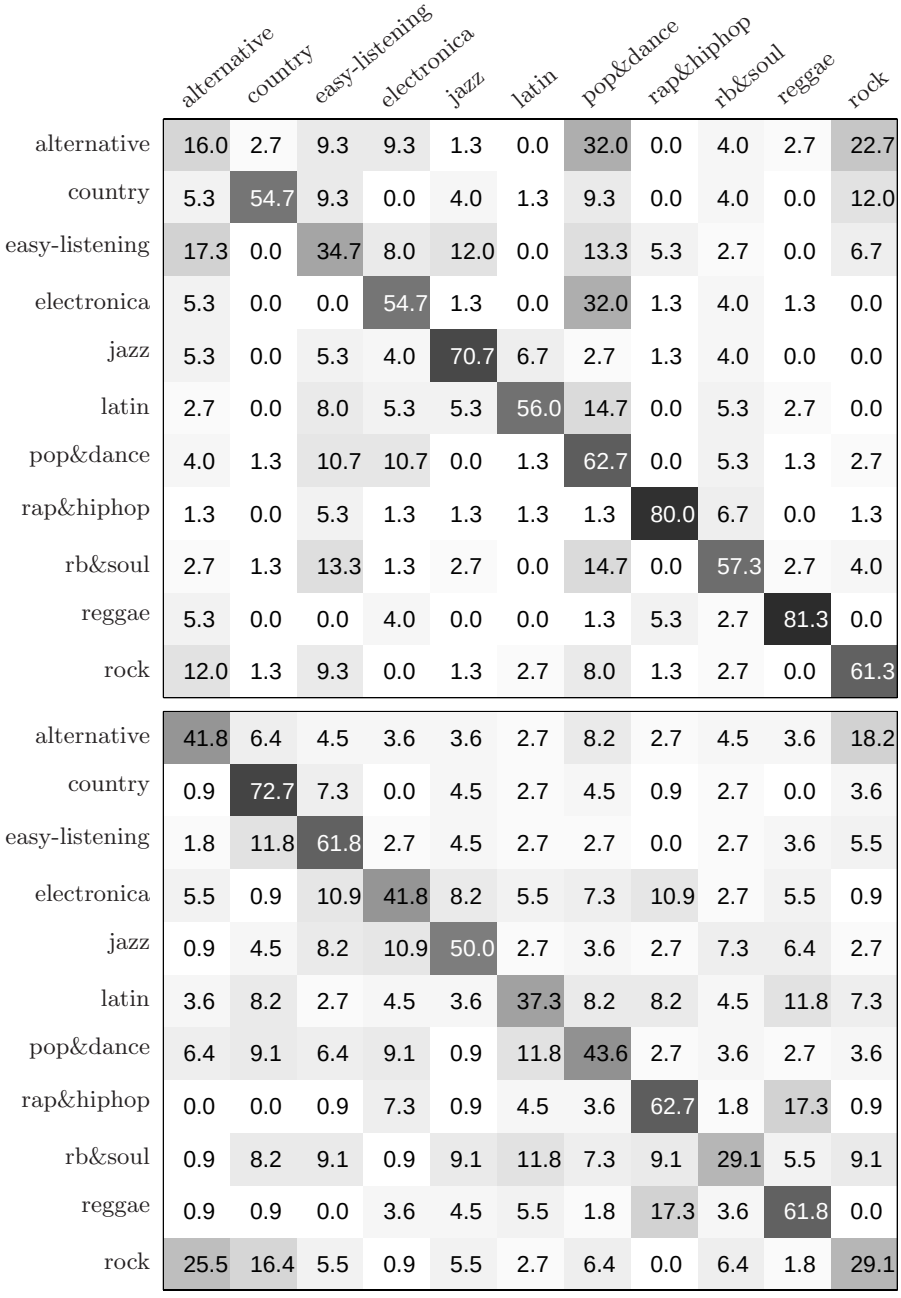


Figure 8. The above confusion matrices were created from data set B. The upper figure shows the confusion matrix from evaluations of the 25 people, and the lower figure shows the average of the confusion matrices over the 10 cross-validation runs of the best performing combination (MAR features with the GLM classifier). The "true" genres are shown as the rows which each sum to 100%. The predicted genres are then represented in the columns. The diagonal illustrates the accuracy of each genre separately.

5. Conclusion

In this paper, we have investigated feature integration of short-time features in a music genre classification task and a novel multivariate autoregressive feature integration scheme was proposed to incorporate dependencies among the feature dimensions and correlations in the temporal domain. This scheme gave rise to two new features, the DAR and MAR, which were carefully described and compared to features from existing feature integration schemes. They were tested on two different data sets with four different classifiers and the successful MFCC features were used as the short-time feature representation. The framework is generalizable to other types of short-time features. Especially the MAR features were found to perform significantly better than existing features, but also the DAR features performed better than the FC and baseline MeanVar features on the large data set and in a much lower dimensional feature space than the MAR.

Human genre classification experiments were made on both data sets and we found that the mean human test accuracy was less than 10% above our best performing MAR features approach.

A direction for future research is to investigate the robustness of the MAR feature integration model to various compressions such as *MPEG1-layer* 3 and other perceptually inspired compression techniques.

Acknowledgements

The work is partly supported by the European Commission through the sixth framework IST Network of Excellence: Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL), contract no. 506778 and by the Danish Technical Research Council project No. 26-04-0092 Intelligent Sound (www.intelligentsound.org).

References

Ahrendt, P., A. Meng, and J. Larsen: 2004, 'Decision Time Horizon for Music Genre Classification using Short-Time Features'. In: *Proc. of EUSIPCO*. Vienna, pp. 1293–1296.

Aucouturier, J.-J. and F. Pachet: 2003, 'Representing Music Genre: A State of the Art'. *Journal of New Music Research* **32**(1), 83–93.

Bach, F. R. and M. I. Jordan: 2004, 'Learning Graphical Models for Stationary Time Series'. *IEEE Transactions on Signal Processing* **52**(8), 2189–2199.

Dietterich, T. G.: 1998, 'Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms'. *Neural Computation* **10**(7), 1895–1923.

Ellis, D. and K. Lee: 2004, 'Features for Segmenting and Classifying Long-Duration Recordings of Personal Audio'. In: *ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing SAPA-04*. Jeju, Korea, pp. 1–6.

Foote, J. and S. Uchihashi: 2001, 'The Beat Spectrum: A New Approach to Rhythm Analysis'. *Proc. International Conference on Multimedia and Expo (ICME)* pp. 1088–1091.

Gouyon, F., S. Dixon, E. Pampalk, and G. Widmer: 2004, 'Evaluating rhytmic descriptors for musical genre classification'. In: *Proceedings of 25th International AES Conference*. London, UK.

H.-Gook., K. and T. Sikora: 2004, 'Audio Spectrum Projection based on Several Basis Decomposition Algorithms Applied to General Sound Recognition and Audio Segmentation'. In: *Proc. of EUSIPCO*. pp. 1047–1050.

Herrera, P., A. Yeterian, and F. Gouyon: 2002, 'Automatic classification of drum sounds: A comparison of feature selection and classification techniques'. In: *Proc. of Second International Conference on Music and Artificial Intelligence*. pp. 79–91.

Logan, B.: 2000, 'Mel Frequency Cepstral Coefficients for Music Modeling'. In: *Proceedings of International Symposium on Music Information Retrieval*. Massachusetts, USA.

Lu, L., H.-J. Zhang, and H. Jiang: 2002, 'Content Analysis for Audio Classification and Segmentation'. *IEEE Transactions on Speech and Audio Processing* **10**(7), 504–516.

Lütkepohl, H.: 1993, *Introduction to Multiple Time Series Analysis*. Springer, 2nd edition.

Makhoul, J.: 1975, 'Linear Prediction: A Tutorial Review'. *Proceedings of the IEEE* **63**(4), 561–580.

Martin, K.: 1999, 'Sound-Source Recognition: A Theory and Computational Model'. Ph.D. thesis, Massachusetts Institute of Technology.

McKinney, M. F. and J. Breebaart: 2003, 'Features for Audio and Music Classification'. In: *Proc. of ISMIR*. pp. 151–158.

Meng, A., P. Ahrendt, and J. Larsen: 2005, 'Improving Music Genre Classification using Short-Time Feature Integration'. In: *Proceedings of ICASSP*. pp. 497–500.

Meng, A. and J. Shawe-Taylor: 2005, 'An Investigation of Feature Models for Music Genre Classification using the Support Vector Classifier'. In: *International Conference on Music Information Retrieval*. pp. 604–609.

Nabney, I. and C. Bishop: 1995, 'NETLAB package'. *http://www.ncrg.aston.ac.uk/netlab/index.php*.

Neumaier, A. and T. Schneider: 2001, 'Estimation of Parameters and Eigenmodes of Multivariate Autoregressive Models'. *ACM Trans. on Mathematical Software* **27**(1), 27–57.

Rabiner, L. R. and B. Juang: 1993, *Fundamental of Speech Recognition*. Prentice Hall.

Shawe-Taylor, J. and N. Cristianini: 2004, *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Soltau, H., T. Schultz, M. Westphal, and A. Waibel: 1998, 'Recognition of Music Types'. In: *Proceedings of ICASSP*, Vol. 2. Seattle, USA, pp. 1137–1140.

Srinivasan, S. H. and M. Kankanhalli: 2004, 'Harmonicity and Dynamics-Based Features for Audio'. In: *ICASSP*. pp. 321–324.

Tzanetakis, G.: 2002, 'Manipulation, Analysis and Retrieval Systems for Audio Signals'. Ph.D. thesis, Faculty of Princeton University, Department of Computer Science.

Tzanetakis, G. and P. Cook: 2002, 'Musical Genre Classification of Audio Signals'. *IEEE Transactions on Speech and Audio Processing* **10**(5).
Zhang, Y. and J. Zhou: 2004, 'Audio Segmentation based on Multi-Scale Audio Classification'. In: *IEEE Proc. of ICASSP*. pp. 349–352.

Bibliography

[1] Allegro, S., Buchler, M., and Launer, S. Automatic sound classification inspired by auditory scene analysis. In *CRAC Workshop* (Aalborg, Denmark, September 2001).

[2] Alonso, M., David, B., and Richard, G. Tempo and beat estimation of musical signals. In *Int. Symp. on Music Information Retrieval* (Barcelona, Spain, Oct. 2004).

[3] Athineos, M., Hermansky, H., and Ellis, D. Lp-trap: Linear predictive temporal patterns. In *Int. Conf. on Spoken Language Processing* (Jeju, Korea, Oct. 2004), pp. 949–952.

[4] Aucouturier, J.-J., and Pachet, F. Representing music genre : A state of the art. *Journal of New Music Research 32*, 1 (Jan. 2003), 83–93.

[5] Bello, J. P. *Towards the Automated Analysis of Simple Polyphonic Music: A Knowledge-based Approach*. PhD thesis, University of London, 2003.

[6] Berenzweig, A., Logan, B., Ellis, D., and Whitman, B. A large-scale evaluation of acoustic and subjective music-similarity measures. *Computer Music Journal 28*, 2 (june 2004), 63–76.

[7] Bergstra, J., Casagrande, N., and Eck, D. Two algorithms for timbre- and rhythm-based multi-resolution audio classification. In *Music Information Retrieval Information Exchange (MIREX)* (London, UK, Sept. 2005).

[8] Bishop, C. M. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[9] Blei, D., and Moreno, P. Topic segmentation with an aspect hidden markov model. In *Proceedings of the 24th international ACM SIGIR conference.* (2001), pp. 343–348.

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

[11] Bretthorst, L. G. *Bayesian Spectrum Analysis and Parameter Estimation.* No. 48 in Lecture Notes in Statistics. Springer-Verlag, New York, 1988.

[12] Burred, J. J., and Lerch, A. A hierarchical approach to automatic musical genre classification. In *International Conference on Digital Audio Effects (DAFx-03)* (London, UK, Sept. 2003).

[13] Cai, R., Lu, L., Zhang, H.-J., and Cai, L.-H. Improve audio representation by using feature structure patterns. In *IEEE Proc. of ICASSP* (2004).

[14] Cano, P., Koppenberger, M., Ferradans, S., Martinez, A., Gouyon, F., Sandvold, V., Tarasov, V., and Wack, N. Mtg-db: A repository for music audio processing. In *Int. Conf. on Web Delivering of Music* (Barcelona, Spain, 2004).

[15] Cano, P., Koppenberger, M., Wack, N. G., Mahedero, J., Aussenac, T., Marxer, R., Masip, J., Celma, O., Garcia, D., Gómez, E., Gouyon, F., Guaus, E., Herrera, P., Massaguer, J., Ong, B., Ramírez, M., Streich, S., and Serra, X. Content-based music audio recommendation. In *ACM Multimedia* (Singapore, 2005).

[16] Casey, M. Mpeg-7 sound recognition tools. *IEEE Transactions on Curcuits and Systems for Video Technology 11*, 6 (June 2001).

[17] Chase, A. R. Music discriminations by carp (cyprinus carpio). *Animal Learning & Behavior 29*, 4 (2001), 336–353.

[18] Cook, P. R. *Music, Cognition, and Computerized Sound: An Introduction to Psychoacoustics.* The MIT Press, 1999.

[19] Cooke, M., and Ellis, D. The auditory organization of speech and other sources in listeners and computational models. *Speech Communication 35* (2001), 141–177.

[20] Cristianini, N., and Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.* Cambridge University Press, 2000.

[21] Dalla Bella, S., and I., P. Differentiation of classical music requires little learning but rhythm. *Cognition 96*, 2 (2005), B65–B78.

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

[23] Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. *J. of the Royal Statistical Society 39*, 1 (1977), 1–38.

[24] Dietrich, C., Schwenker, F., and Palm, G. Classification of time series utilizing temporal and decision fusion. In *Multiple Classifier Systems* (2001), J. Kittler and F. Roli, Eds., Springer, pp. 378–387.

[25] Dietterich, T. G. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation 10*, 7 (Oct. 1998), 1895–1923.

[26] Dubnov, S. Non-gaussian source-filter and independent components generalizations of spectral flatness measure. In *Int. Conf. on Independent Component Analysis* (2003), pp. 143–148.

[27] Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification.* Wiley-Interscience, 2000.

[28] Ellis, D. P. W. *Prediction-Driven Computational Auditory Scene Analysis.* PhD thesis, Massachusetts Institute of Technology, June 1996.

[29] Ellis, D. P. W. Detecting alarm sounds. In *CRAC Workshop* (Aalborg, Denmark, September 2001).

[30] Eronen, A. Comparison of features for musical instrument recognition. In *Consistent & Reliable Acoustic Cues Workshop* (Aalborg, Denmark, Sept. 2001).

[31] Esmaili, S., Krishnan, S., and Raahemifar, K. Content based audio classification and retrieval using joint time-frequency analysis. In *IEEE Int.l Conf. on Acoustics, Speech and Signal Processing* (Montreal, Canada, May 2004).

[32] Foote, J., and Uchihashi, S. The beat spectrum: A new approach to rhythm analysis. *Proc. International Conference on Multimedia and Expo (ICME)* (2001), 1088–1091.

[33] Foote, J. T. Content-based retrieval of music and audio. In *Multimedia Storage and Archiving Systems II, Proc. of SPIE* (1997), C.-C. J. e. a. Kuo, Ed., vol. 3229, pp. 138–147.

[34] Fröba, B., Allamanche, E., Herre, J., Kastner, T., Hellmuth, O., and Cremer, M. Content-based identification of audio material using mpeg-7 low level description. In *Int. Symp. on Music Information Retrieval* (Bloomington, USA, Sept. 2001).

[35] Ganchev, T., Fakotakis, N., and Kokkinakis, G. Comparative evaluation of various mfcc implementations on the speaker verification task. In *10th International Conference on Speech and Computer, SPECOM* (Patras, Greece, 2005), vol. 1, pp. 191–194.

[36] Gaussier, E., Goutte, C., Popat, K., and Chen, F. Hierarchical model for clustering and categorising documents. In *Proceedings of the 24th BCS-IRSG European Colloqium on IR Research (ECIR-02)* (2002).

[37] Good, P. I. *Resampling Methods : A Practical Guide to Data Analysis*. Birkhäuser, 1999.

[38] Gouyon, F., Dixon, S., Pampalk, E., and Widmer, G. Evaluating rhytmic descriptors for musical genre classification. In *Proceedings of 25th International AES Conference* (London, UK, 2004).

[39] Guyon, I., and Elisseeff, A. An introduction to variable and feature selection. *Journal of Machine Learning Research 3* (March 2003), 1157–1182.

[40] Hermansky, H. Exploring temporal domain for robustness in speech recognition. In *Int. Congress on Acoustics* (Trondheim, Norway, June 1995), vol. 2, pp. 61–64.

[41] Herrera, P., Yeterian, A., and Gouyon, F. Automatic classification of drum sounds : A comparison of feature selection methods and classification techniques. In *Proceedings of Second International Conference on Music and Artificial Intelligence* (Edinburgh, Scotland, 2002), pp. 69–80.

[42] Hofmann, T. Probabilistic latent semantic indexing. In *Proceedings of SIGIR* (Berkeley, CA, 1999), pp. 35–44.

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

[44] http://ismir2004.ismir.net/genre_contest/index.htm. Ismir 2004 music genre classification contest, 2004. (took place during the ISMIR 2004 conference).

[45] http://www.allmusic.com. The all music guide.

[46] http://www.amazon.com. Free-downloads section, 2006.

[47] HTTP://WWW.APPLE.COM/ITUNES. itunes music store, 2006.

[48] HTTP://WWW.CHIARIGLIONE.ORG/MPEG. Moving pictures expert group (mpeg).

[49] HTTP://WWW.CS.UBC.CA/ MURPHYK/BAYES/BNINTRO.HTML. Murphy k. - "a brief introduction to graphical models and bayesian networks".

[50] HTTP://WWW.EE.IC.AC.UK/HP/STAFF/DMB/VOICEBOX/VOICEBOX.HTML. Voicebox matlab toolbox, 2006.

[51] HTTP://WWW.GREENWYCH.CA/FL COMPL.HTM. Neanderthal flute.

[52] HTTP://WWW.ISMIR.NET/. The international conferences on music information retrieval.

[53] HTTP://WWW.MUSIC IR.ORG/MIREX2005/INDEX.PHP/. Music information retrieval evaluation exchange (mirex), 2005. (took place during the ISMIR 2005 conference).

[54] HTTP://WWW.NCRG.ASTON.AC.UK/NETLAB/INDEX.PHP. Netlab matlab package.

[55] HTTP://WWW.SONYMUSICSTORE.COM. Sonymusicstore, 2006.

[56] HYVÄRINEN, A., AND OJA, E. Independent component analysis: Algorithms and applications. *Neural Networks 13*, 4-5 (2000), 411–430.

[57] JENSEN, F. V. *Bayesian Networks and Decision Graphs*. Springer, 2001.

[58] JENSEN, J. H., CHRISTENSEN, M. G., MURTHI, M., AND JENSEN, S. H. Evaluation of mfcc estimation techniques for music similarity. In *European Signal Processing Conference* (2006).

[59] JORDAN, M. I. Why the logistic function? a tutorial discussion on probabilities and neural networks. Tech. rep., MIT, August 1995. Computational Cognitive Science Report 9503.

[60] JURAFSKY, D., AND MARTIN, J. H. *Speech and Language Processing : An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice-Hall, 2000.

[61] KABAN, A., BINGHAM, E., AND HIRSIMÄKI, T. Learning to read between the lines: The aspect bernoulli model. In *Proceedings of the 4th SIAM International Conference on Data Mining* (Lake Buena Vista, Florida, April 2004), pp. 462–466.

[62] KIM, H.-G., AND SIKORA, T. Audio spectrum projection based on several basis decomposition algorithms applied to general sound recognition and audio segmentation. In *Proc. of EUSIPCO* (2004), pp. 1047–1050.

[63] Kittler, J., Hatef, M., Duin, R. P., and Matas, J. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 3 (1998), 226–239.

[64] Kjems, U., Hansen, L., Anderson, J., Frutiger, S., Muley, S., Sidtis, J., Rottenberg, D., and Strother, S. C. The quantitative evaluation of functional neuroimaging experiments: Mutual information learning curves. *NeuroImage 15*, 4 (2002), 772–786.

[65] Klapuri, A. *Signal Processing Methods for the Automatic Transcription of Music*. PhD thesis, Tampere University of Technology, 2004.

[66] Ku, W., Storer, R. H., and Georgakis, C. Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems 30* (1995), 179–196.

[67] Lambrou, T., Kudumakis, P., Sandler, M., Speller, R., and Linney, A. Classification of audio signals using statistical features on time and wavelet transform domains. In *IEEE ICASSP* (Seattle, USA, May 1998).

[68] Li, T., and Ogihara, M. Music artist style identification by semi-supervised learning from both lyrics and contents. In *ACM Multimedia* (2004).

[69] Li, T., Ogihara, M., and Li, Q. A comparative study on content-based music genre classification. In *ACM SIGIR* (2003).

[70] Lippens, S., Martens, J. P., De Mulder, T., and Tzanetakis, G. A comparison of human and automatic musical genre classification. In *IEEE Int. Conf. on Audio, Speech and Signal Processing* (Montreal, Canada, 2004).

[71] Liu, D., Lu, L., and Zhang, H.-J. Automatic music mood detection from acoustic music data. In *International Symposium on Music Information Retrieval* (Baltimore, USA, Oct. 2003), pp. 81–87.

[72] Logan, B. Mel frequency cepstral coefficients for music modeling. In *Proceedings of International Symposium on Music Information Retrieval* (Massachusetts, USA, Oct. 2000).

[73] Lütkepohl, H. *Introduction to Multiple Time Series Analysis*, 2nd ed. Springer, 1993.

[74] Lu, L., Zhang, H.-J., and Jiang, H. Content analysis for audio classification and segmentation. *IEEE Transactions on Speech and Audio Processing 10*, 7 (October 2002), 504–516.

[75] MacKay, D. J. C. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, 2002.

[76] Makhoul, J. Linear prediction: A tutorial review. *Proceedings of the IEEE 63*, 4 (1975), 561–580.

[77] Mandel, M., and Ellis, D. Song-level features and support vector machines for music classification. In *Int. Conf. on Music Information Retrieval ISMIR-05* (London, UK, Sept. 2005).

[78] Mandel, M., Poliner, G., and Ellis, D. Support vector machine active learning for music retrieval. *ACM Multimedia Systems Journal* (2006), 10 pp. (Accepted for publication).

[79] Martin, K. D. *Sound-Source Recognition : A Theory and Computational Model*. PhD thesis, Massachusetts Institute of Technology, June 1999.

[80] McKay, C. Automatic genre classification of midi recordings. Master's thesis, McGill University, Canada, 2004.

[81] McKay, C., and Fujinaga, I. Automatic genre classification using large high-level musical feature sets. In *Int. Conf. on Music Information Retrieval* (Barcelona, Spain, Oct. 2004), pp. 525–530.

[82] McKinney, M. F., and Breebaart, J. Features for audio and music classification. In *ISMIR* (2003).

[83] Meng, A., and Shawe-Taylor, J. An investigation of feature models for music genre classification using the support vector classifier. In *International Conference on Music Information Retrieval* (2005), pp. 604–609.

[84] Merwe, P. v. d. *Origins of the Popular Style - The Antecedents of Twentieth-Century Popular Music*. Oxford University Press, 1989.

[85] Morgan, N., Zhu, Q., Stolcke, A., Sonmez, K., Sivadas, S., Shinozaki, T., Ostendorf, M., Jain, P., Hermansky, H., Ellis, D., Doddington, G., Chen, B., Cetin, O., Bourlard, H., and Athineos, M. Pushing the envelope – aside. *IEEE Signal Processing Magazine 22*, 5 (Sept. 2005), 81–88.

[86] MPEG-7. Information technology - multimedia content description interface, part 4 : Audio, 2003. ISO/IEC FDIS 15938-4:2002(E).

[87] Neumaier, A., and Schneider, T. Estimation of parameters and eigenmodes of multivariate autoregressive models. *ACM Trans. on Mathematical Software 27*, 1 (Mar. 2001), 27–57.

[88] Patterson, R. The sound of a sinusoid: Spectral models. *J. Acoust. Soc. Am. 96* (1993), 1409–1418.

[89] Peltonen, V., Tuomi, J., Klapuri, A., Huopaniemi, J., and Sorsa, T. Computational auditory scene recognition. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing* (2002).

[90] Picone, J. Signal modeling techniques in speech recognition. *IEEE Proceedings 81*, 9 (Sept. 1993), 1215–1247.

[91] Pohle, T., Pampalk, E., and Widmer, G. Evaluation of frequently used audio features for classification of music into perceptual categories. In *Int. Workshop on Content-Based Multimedia Indexing (CBMI)* (Riga, Latvia, 2005).

[92] Rabiner, L. R. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE 77*, 2 (1989), 257–286.

[93] Rabiner, L. R., and Juang, B. *Fundamental of Speech Recognition*. Prentice Hall, 1993.

[94] Rabiner, L. R., and Schafer, R. W. *Digital Processing of Speech Signals*, 1st ed. Prentice-Hall, 1978.

[95] Rauber, A., Pampalk, E., and Merkl, D. Using psycho-acoustic models and self-organizing maps to create a hierarchical structuring of music by musical styles. In *Int. Conf. on Music Information Retrieval* (Paris, France, Oct. 2002), pp. 71–80.

[96] Roweis, S., and Ghahramani, Z. A unifying review of linear gaussian models. *Neural Computation 11*, 2 (1999), 305–345.

[97] Scaringella, N., Zoia, G., and Mlynek, D. Automatic genre classification of music content. (submitted to IEEE Signal Processing Magazine : Special Issue on Semantic Retrieval of Multimedia, March 2006).

[98] Scheirer, E., and Slaney, M. Construction and evaluation of a robust multi-feature speech/music discriminator. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing* (1997), vol. 2, pp. 1331–1334.

[99] Scheirer, E. D. *Music-Listening Systems*. PhD thesis, Massachusetts Institute of Technology, 2000.

[100] Schweitzer, H. A distributed algorithm for content based indexing of images by projections on ritz primary images. *Data Mining and Knowledge Discovery 1*, 4 (1997), 375–390.

[101] Shannon, B., and Paliwal, K. K. A comparative study of filter bank spacing for speech recognition. In *Microelectronic Engineering Research Conference* (Brisbane, Australia, Nov. 2003).

[102] Shao, X., Xu, C., and Kankanhalli, M. S. Unsupervised classification of music genre using hidden markov model. In *IEEE Int. Conf. of Multimedia Explore* (Taiwan, 2004).

[103] Shawe-Taylor, J., and Cristianini, N. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[104] Sigurdsson, S., Philipsen, P. A., Hansen, L. K., Larsen, J., Gniadecka, M., and Wulf, H. C. Detection of skin cancer by classification of raman spectra. *IEEE Transactions on Biomedical Engineering 51*, 10 (2004), 1784–1793.

[105] Skowronski, M. D., and Harris, J. G. Exploiting independent filter bandwidth of human factor cepstral coefficients in automatic speech recognition. *J. Acoustical Society of America 116*, 3 (Sept. 2004), 1774–1780.

[106] Soltau, H., Schultz, T., Westphal, M., and Waibel, A. Recognition of music types. In *Proceedings of ICASSP* (Seattle, USA, May 1998).

[107] Tolonen, T., and Karjalainen, M. A computationally efficient multipitch analysis model. *IEEE Transactions on Speech and Audio Processing 8*, 6 (Nov. 2000), 708–716.

[108] Tzanetakis, G. *Manipulation, Analysis and Retrieval Systems for Audio Signals*. PhD thesis, Faculty of Princeton University, Department of Computer Science, 2002.

[109] Tzanetakis, G., and Cook, P. Audio information retrieval (air) tools. In *Int. Symposium on Music Information Retrieval (ISMIR)* (Plymouth, Massachussets, 2000).

[110] Tzanetakis, G., and Cook, P. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing 10*, 5 (July 2002).

[111] Tzanetakis, G., Essl, G., and Cook, P. Audio analysis using the discrete wavelet transform. In *WSES Int. Conf. Acoustics and Music: Theory and Applications (AMTA 2001)* (Skiathos, Greece, 2001).

[112] Wellhausen, J., and Höynck, M. Audio thumbnailing using mpeg-7 low level audio descriptors. In *SPIE Int. Symposium on ITCom 2003 - Internet Multimedia Management Systems IV* (2003).

[113] West, K. Mirex audio genre classification. In *Music Information Retrival Evaluation eXchange (MIREX)* (London, UK, Sept. 2005).

[114] Whitman, B., and Lawrence, S. Inferring descriptions and similarity for music from community metadata. In *Int. Computer Music Conference* (Göteborg, Sweden, Sept. 2002).

[115] Whitman, B., and Smaragdis, P. Combining musical and cultural features for intelligent style detection. In *Int. Conf. on Music Information Retrieval* (Paris, France, Oct. 2002).

[116] Wold, E., Blum, T., Keislar, D., and Wheaton, J. Content-based classification, search and retrieval of audio. *IEEE Multimedia 3*, 3 (1996), 27–36.

[117] Xu, C., Maddage, N. C., Shao, X., Cao, F., and Tian, Q. Musical genre classification using support vector machines. In *Proc. of ICASSP* (Hong Kong, China, Apr. 2003), pp. 429–432.

[118] Yoshii, K., Goto, M., and Okuno, H. G. Automatic drum sound description for real-world music using template adaptation and matching methods. In *Int. Conf. on Music Information Retrieval* (Oct. 2004), pp. 184–191.

[119] Zou, H., T. Hastie, T., and Tibshirani, R. Sparse principal component analysis. Tech. rep., Statistics department, Stanford University, 2004.