Temporal Feature Integration for Music Organisation

Anders Meng

Kongens Lyngby 2006
IMM-PHD-2006-165

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

This Ph.D. thesis focuses on temporal feature integration for music organisation. Temporal feature integration is the process of combining all the feature vectors of a given time-frame into a single new feature vector in order to capture relevant information in the frame. Several existing methods for handling sequences of features are formulated in the temporal feature integration framework. Two datasets for music genre classification have been considered as valid test-beds for music organisation. Human evaluations of these, have been obtained to access the subjectivity on the datasets.

Temporal feature integration has been used for ranking various short-time features at different time-scales. This include short-time features such as the Mel frequency cepstral coefficients (MFCC), linear predicting coding coefficients (LPC) and various MPEG-7 short-time features. The 'consensus sensitivity ranking' approach is proposed for ranking the short-time features at larger time-scales according to their discriminative power in a music genre classification task.

The multivariate AR (MAR) model has been proposed for temporal feature integration. It effectively models local dynamical structure of the short-time features.

Different kernel functions such as the convolutive kernel, the product probability kernel and the symmetric Kullback Leibler divergence kernel, which measures similarity between frames of music have been investigated for aiding temporal feature integration in music organisation. A special emphasis is put on the product probability kernel for which the MAR model is derived in closed form. A thorough investigation, using robust machine learning methods, of the MAR model on two different music genre classification datasets, shows a statistical significant improvement using this model in comparison to existing temporal feature integration models. This improvement was more pronounced for the larger and more difficult dataset. Similar findings where observed using the MAR model in a product probability kernel. The MAR model clearly outperformed the other investigated density models: the multivariate Gaussian model and the Gaussian mixture model.

Resumé

Nærværende Ph.D. afhandling omhandler musik organisation ved brug af tidslig integration af "features". Tidslig integration af "features" er en proces hvor en enkelt ny "feature" vektor dannes udfra et segment med en sekvens af "feature" vektorer. Denne nye vektor indeholder information, der er nyttig i forbindelse med en efterfølgende automatiseret organisering af musikken. I denne afhandling er eksisterende metoder til håndtering af sekvenser af "feature" vektorer blevet formuleret i en generel form. To datasæt blev generet til musik genre klassifikation, og blev efterfølgende evalueret af en række individer, for at undersøge graden af subjektivitet af genre angivelserne. Begge datasæt kan betragtes som værende gode eksempler på musik organisering.
Tidslig integration af "features" er blevet anvendt i forbindelse med en undersøgelse af forskellige korttids "features" diskriminative egenskaber på længere tidsskalaer. Korttids "features", såsom: MFCC, LPC og forskellige MPEG-7 varianter, blev undersøgt ved brug af den foreslåede "consensus sensitivity ranking" til automatisk organisering af sange efter genre. En multivariabel autoregressiv model (MAR) blev foreslået til brug i forbindelse med tidslig integration af "features". Denne model er i stand til at modellere tidslige korrelationer i en sekvens af "feature" vektorer. Forskellige "kernel" funktioner såsom en "convolutive kernel", en Kullback-Leibler symmetrisk "kernel" samt en "product probability kernel" er blevet blev undersøgt i forbindelse med tidslig integration af "features". Der blev især lagt vægt på sidstnævnte "kernel" hvor et analytisk udtryk blev fundet for MAR modellen.
En grundig undersøgelse af MAR modellen blev foretaget på de ovennævtne datasæt i forbindelse med musik genreklassifikation. Undersøgelsen viste, at MAR modellen klarede sig signifikant bedre på de undersøgte datasæt i forhold til eksisterende metoder. Denne observation var især gældende for det mere komplekse af de to datasæt. Lignende resultater blev observeret ved at kombinere en "product probability kernel" med MAR modellen. Igen klarede MAR modellen sig signifikant bedre end kombinationen af førnævnte "kernel" med en multivariabel Gaussisk model samt en Gaussisk miksturmodel.

Preface

This thesis was prepared at Informatics Mathematical Modelling, the Technical University of Denmark in partial fulfilment of the requirements for acquiring the Ph.D. degree in engineering.
The work is funded partly by DTU and by an UMTS-grant. Furthermore, the work has been supported by the European Commission through the sixth framework IST Network of Excellence: Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL), contract no. 506778.
The project commenced in April 2003 and was completed in April 2006. Throughout the period, the project was supervised by associate professor Jan Larsen with co-supervision by professor Lars Kai Hansen. The thesis reflects the studies done during the Ph.D. project and concerns machine learning approaches for music organisation. During the thesis period I have had collaboration with Peter Ahrendt, a fellow researcher, who recently finished his Ph.D. thesis [1]. Having browsed the index of his thesis some overlap is expected in sections concerning: feature extraction, temporal feature integration and some of the experimental work.
Various homepages have been cited in this thesis. A snapshot of these, as of March 2006 has been shown in Appendix C.
The thesis is printed by IMM, Technical University of Denmark and available as softcopy from `http://www.imm.dtu.dk`.

Lyngby, 2006

Anders Meng

Anders Meng

Publication Note

Parts of the work presented in this thesis have previously been published at conferences and contests. Furthermore, an unpublished journal paper has been submitted recently. The following papers have been produced during the thesis period.

JOURNAL PAPER

Appendix H Anders Meng, Peter Ahrendt, Jan Larsen and Lars Kai Hansen, "Temporal Feature Integration for Music Genre Classification", *submitted to IEEE Trans. on Signal Processing*, 2006

CONFERENCES

Appendix D Peter Ahrendt, Anders Meng and Jan Larsen, "Decision Time Horizon for Music Genre Classification Using Short Time Features", *In Proceedings of EUSIPCO*, pp. 1293-1296, Vienna, Austria, Sept. 2004.

Appendix E Anders Meng, Peter Ahrendt and Jan Larsen, "Improving Music Genre Classification by Short-Time Feature Integration", *In Proceedings of ICASSP*, pp. 497-500, Philadelphia, March 18-23, 2005.

Appendix F Anders Meng and John Shawe-Taylor, "An Investigation of Feature Models for Music Genre Classification using the Support Vector Classifier", *In Proceedings of ISMIR*, pp. 504-509, London, Sept. 11-15, 2005.

COMPETITIONS

Appendix G Peter Ahrendt and Anders Meng, "Music Genre Classification using the multivariate AR feature integration model", *Music Information Retrieval Evaluation eXchange*, London, Sept. 11-15, 2005.

Nomenclature

Standard symbols and operators are used consistently throughout the thesis. Symbols and operators are introduced as they are needed. In general, matrices are presented in uppercase bold letters e.g. $\mathbf{X}$, while vectors are shown in lowercase bold letters, e.g. $\mathbf{x}$. Letters not bolded are scalars, e.g. x. The vectors are assumed to be column vectors if nothing else is specified.

Acknowledgements

I would like to thank Jan Larsen and Lars Kai Hansen for giving me the opportunity to do a Ph.D. under their supervision. A special thanks to Peter Ahrendt, a fellow Ph.D. student, who I have collaborated with during my studies. Our many discussions have been invaluable for me. Furthermore, I would like to thank the people at the Intelligent Signal Processing group, and a special thank to Kaare Brandt Petersen, Sigurdur Sigurdsson, Rasmus Elsborg Madsen, Jacob Schack Larsen, David Puttick and Jerónimo Arenas-Garcia for proofreading parts of my thesis. Furthermore, a special thank to the department secretary Ulla Nørhave for all practical details. Also, a big thank to my office mate Jerónimo Arenas-Garcia, who have kept me company during the late hours of the final period of writing.

I am grateful to professor John Shawe-Taylor and his machine learning group at Southampton University in Great Britain, which I visisted during my Ph.D. studies from October 2004 to the end of March 2005. They all provided me with an unforgettable experience. A warm thank to Emilio Parrado-Hernández, Sándor Szedmák, Jason Farquhar, Andriy Kharechko and Hongying Meng for endless lunch-club discussions and many social events.

Finally, I am endebted to my girlfriend and coming wife Sarah, who has been indulgent during the final period of my thesis and helped me with proofreading the thesis. Also a warm thank to my family and friends, which have supported me during my thesis.

Contents

Summary **i**

Resumé **iii**

Preface **v**

Acknowledgements **vii**

1 Introduction **3**

1.1 Organisation of music 4

2 Music genre classification **9**

2.1 Music taxonomies - genre 9

2.2 Music genre classification 13

3 Selected features for music organisation **15**

3.1 Preprocessing 16

3.2 A general introduction to feature extraction 17

3.3 Feature extraction methods 19

3.4 Frame- and hop-size selection 28

3.5 Discussion 32

4 Temporal feature integration 33

4.1 Definition 34

4.2 Stacking 35

4.3 Statistical models 36

4.4 Filterbank coefficients (FC) 45

4.5 Extraction of tempo and beat 46

4.6 Stationarity and frame-size selection 49

4.7 Discussion 51

5 Kernels for aiding temporal feature integration 53

5.1 Kernel methods 54

5.2 Kernels for music 56

5.3 High-level kernels 57

5.4 The effect of normalisation 64

5.5 Discussion 65

6 Experiments in music genre classification 67

6.1 Classifier technologies 69

6.2 Performance measures and generalisation 74

6.3 Datasets 77

6.4 Feature ranking in music genre classification 82

6.5 Temporal feature integration for music genre classification 90

6.6 Kernel methods for music genre classification 101

6.7 Discussion 104

7 Conclusion 107

A Derivation of kernel function for the multivariate AR-model 111

A.1 Memorandum 114

B Simple PCA 115

B.1 Computational complexity 116

C Homepages as of March 2006 119

D Contribution: EUSIPCO 2004 131

E Contribution: ICASSP 2005 137

F Contribution: ISMIR 2005 143

G Contribution: MIREX 2005 151

H Contribution: IEEE 2006 155

Notation and Symbols

$\mathbf{z}_k$	Short-time feature of dimension D extracted from frame k
$\tilde{\mathbf{z}}_{\tilde{k}}$	Feature vector of dimension $\tilde{D}$ extracted from frame $\tilde{k}$ using temporal feature integration over the short-time features in the frame.
f_{s_z}	Frame-size for temporal feature integration over a frame of short-time features.
h_{s_z}	Hop-size used when performing temporal feature integration over a frame of short-time features.
$\mathcal{C}_l$	Identifies class no. l
f_s	Frame-size used when extracting short-time features from the music
h_f	Hop/frame-size ratio h_s/f_s
h_s	Hop-size used when extracting short-time features from the music
$p(\mathbf{z} \mid \boldsymbol{\theta})$	Probability density model of $\mathbf{z}$ given some parameters $\boldsymbol{\theta}$ of the model
$P(\cdot)$	Probability
s_r	Samplerate of audio signal
$x[n]$	Digitalised audio signal at time instant n
BPM	Beats per minute
DCT	Discrete Cosine Transform
HAS	Human Auditory System
i.i.d.	Independent and Identically Distributed
ICA	Independent Component Analysis
IIR	Infinite Impulse Response

LPC	Linear Predictive Coding
MFCC	Mel Frequency Cepstral Coefficient
MIR	Music Information Retrieval
MIREX	Music Information Retrieval evaluation exchange
PCA	Principal Component Analysis
PPK	Product Probability Kernel
SMO	Sequential Minimal Optimisation
STE	Short Time Energy
STFT	Short-Time Fourier Transform
SVC	Support Vector Classifier
SVD	Singular Value Decomposition

Chapter 1

Introduction

Music has the ability to awaken a range of emotions in human listeners regardless of race, religion and social status. Finding music that mimics our immediate state of mind can have an intensifying effect on our soul. This effect is well-known and frequently applied in the movie industry. For example, a sad scene combined with carefully selected music can strengthen the emotions, ultimately causing people to cry. Music can have a soothing or exciting effect on our emotions, which makes it an important ingredient in many people's everyday lives - lives governed by increasing levels of stress. Enjoying music in our spare time after a long hectic day at work can have the soothing effect required[1]. Being in a more explorative mode, new music titles (and styles) can intrigue our mind, develop and move boundaries in the understanding of our own personal music taste. The discovery of new music titles are usually restricted by our personal taste, which makes it hard to find titles outside the ordinary. Listening to radio or e.g. dedicated playlists from Internet sites can to some extent provide users with intriguing new music that is out of the ordinary.

With the increased availability of digital media during the last couple of years music has become a more integrated part of our everyday life. This is mainly due to consumer electronics such as memory and hard discs becoming cheaper, changing our personal computers into dedicated media players. Similarly, the boom of portable digital media players, such as the 'iPod' from Apple Computer, which easily stores more than 1500 music titles, enables us to listen

[1] In a recently published article [122], the authors investigated people's stress levels, indicated by changes in blood pressure, before, during and after the test persons where exposed to two different genres - rock and classical where classical was the preferred genre of the test persons. Classical music actually lowered the test persons stress level.

to a big part of our private music collection wherever we go. With increasing digitisation music distribution is no longer limited to physical media but can be acquired from large online web-portals such as e.g. `www.napster.com` or `www.itunes.com`, where users currently have access to more than a million music titles[2]. Furthermore, a large number of radio and TV-stations allow free streaming from the Internet to ones favourite music player, or to be stored on a personal computer for later use.

The problem of organising and navigating these seemingly endless streams of multimedia information is inherently at odds with the currently available systems for handling non-textual data such as audio and video. During the last decade[3] research in the field of 'Music Information Retrieval' (MIR) has boomed and has attracted attention from large system providers such as Microsoft Research, Sun Microsystems, Philips and HP-Invent. These companies were sponsors for last year's International Symposium on Music Information Retrieval (ISMIR). Also the well known provider of the successful Internet searcher `www.google.com` have provided the "Google desktop" for helping users to navigate the large amounts of textual files on their personal computers. To date the principal approach for indexing and searching digital music is via the metadata stored inside each media file. Metadata currently consists of short text fields containing information about the composer, performer, album, artist, title, and in some cases, more subjective aspects of music such as genre or mood. The addition of metadata, however, is labor-intensive and therefore not always available. Secondly, the lack of consistency in metadata can render the media files difficult or impossible to retrieve.

Automated methods for indexing, organising and navigating digital music is of great interest both for consumers and providers of digital music. Thus, devising fully automated or semi-automated (in terms of user feedback) approaches for music organisation will simplify the navigation and provide the user with a more natural way of browsing their burgeoning music collection.

1.1 Organisation of music

There are many ways of organising a music collection. Figure 1.1 illustrate some ways of organising music collections [65]. The ways of organising the music titles can be classified either as objective or subjective. Typical objective measures are instrumentation, artist, whether the song has vocal or not, etc.

[2] A press release of February 2006, stated that 'iTunes' had their $1,000,000,000$ music download.

[3] Research activities really started accelerating in this field.

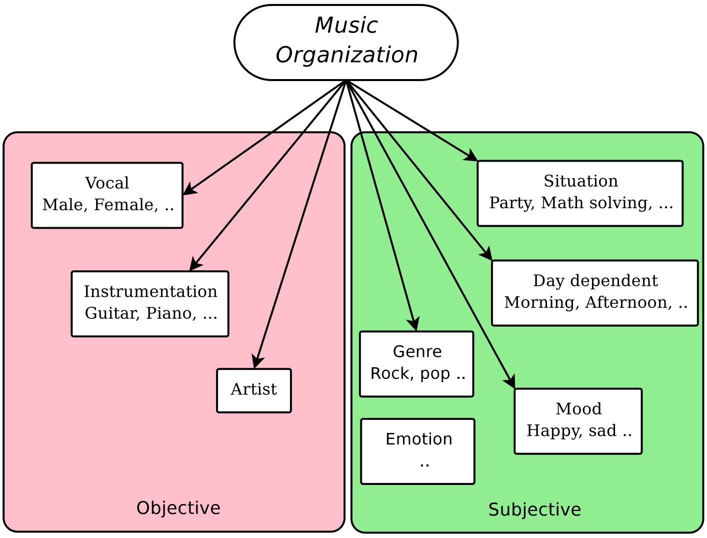


Figure 1.1: Approaches to music organisation. The music organisation methods have been divided into either subjective or objective.

This information can be added by the artist as metadata. Subjective measures such as mood, music genre and theme, can also be added by the artist, but will indeed depend on the artist's understanding of these words. The music might, in the artist's view, seem uplifting and happy, however it could be perceived quite differently by another individual. The degree of subjectivity of a given organisation method can be assessed in terms of the level of consensus achieved in a group of people believed to represent a larger group with a similar cultural background.

A small scale investigation of the Internet portals `www.amazon.com`, `www.mp3.com`, `www.allmusic.com` and `www.garageband.com` showed that music genre has been selected as the primary method for navigating these repertoires of music. This implies that, even though music genre is a subjective measure, there must be a degree of consensus, which makes navigation in these large databases possible. Only at `www.allmusic.com`, it was possible to navigate music by mood, theme, instrumentation or which country the music originates from. Another frequently used navigation method is by artist or album name, which was possible on all the sites investigated.

Having acknowledged that music genre is a descriptor commonly used for navigating Internet portals, music libraries, music shops, etc. this descriptor has been selected as a good starting point for developing methods for automated organisation of music titles. Organisation of digital music into for example music genre indeed requires robust machine learning methods. A typical song is around 4 minutes in length. Most digital music is sampled at a frequency of $s_r = 44100\,\mathrm{Hz}$, which amounts to approximately $21,000,000$ samples per song (for a stereo signal). A review of the literature on automated systems for organisation of music leads to a structure similar to Figure 1.2. In Figure 1.2, solid boxes indicate a common operation performed, whereas, the dotted boxes indicate operations devised by some authors, see e.g. [14] and ([98], appendix E).

This thesis will focus on methods for extraction of high-level metadata such as music genre of digital music from its acoustic contents.
Music information retrieval (MIR) should in this context be understood in terms of a perceptual similarity between songs such as e.g. genre or mood and not as an exact similarity in terms of its acoustic content, which is the task of audio fingerprinting.
The work has concentrated on existing feature extraction methods and primarily on supervised machine learning algorithms to devise improved methods of temporal feature integration in MIR. Temporal feature integration is the process of combining all the feature vectors in a time-frame into a single new feature vector in order to capture the relevant information in the frame.
Music genre classification with flat genre taxonomies has been applied as testbeds for evaluation and comparison of proposed and existing temporal feature integration methods for MIR.

The main contributions of this thesis (and corresponding articles) are listed below:

- Ranking of short-time features at larger time-scales using the proposed method of 'consensus feature analysis'. The research related to this work was published in [4]. Reprint in appendix D.
- A systematic comparative analysis of the inclusion of temporal information of short-time features by using a multivariate (and univariate) AR model to that of existing temporal feature integration methods have been conducted on a music genre classification task, see [98, 99, 3] and appendix E, H and G for reprints.
- Combining existing temporal feature integration methods with kernel methods for improving music organisation tasks. Furthermore, the combination of a product probability kernel with the multivariate AR model was investigated, see [100] and reprint in appendix F.

The content of this thesis has been structured in the following way:

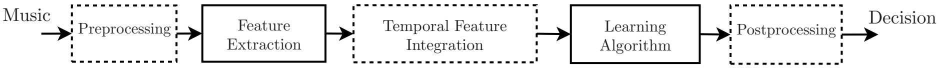


Figure 1.2: The figure shows a flow-chart of a system usually applied for automated music organisation. The solid boxes indicate typical operations performed whereas the dotted boxes are applied by some authors.

Chapter 2 provides an introduction to problems inherent to music genre. Furthermore, the rationale for considering automated systems for music genre classification is given.

Chapter 3 introduces feature extraction and presents the various feature extraction methods, which have been used in the thesis. A method for selection of hop-size from the frame-size is provided.

Chapter 4 present a general formulation of temporal feature integration. Furthermore, different statistical models as well as signal processing approaches are considered.

Chapter 5 introduces two methods for kernel aided temporal feature integration, namely the 'convolution kernel' and the 'product probability kernel'.

Chapter 6 is a compilation of selected experiments on two music genre datasets. The datasets, which have been applied in the various papers are explained in detail. Furthermore, a short introduction to the investigated classifiers is given. This involves ways of assessing the performance of the system and methods for selecting the best performing learning algorithm on the given dataset. Selected sections from articles published as part of this Ph.D. describing the experiments conducted for feature ranking at different time-scales, temporal feature integration and kernel aided temporal feature integration for music genre classification are presented and discussed.

Chapter 7 summaries the work, concludes the thesis, and gives direction for future research.

Appendix A Derivation of the product probability kernel for the multivariate autoregressive model (MAR).

Appendix B A detailed explanation of the simple PCA applied in [4, appendix D].

Appendix C A snapshot of the different URL-addresses reported in the thesis as of March 2006.

Appendix D-H Contains reprints of the papers authored and co-authored during the Ph.D. study.

Chapter 2

Music genre classification

Spawned by the early work of [148] that proposed a system for classification of audio snippets using audio features such as pitch, brightness, loudness, bandwidth and harmonicity, researchers have been intrigued by the task of music genre classification. Some of the earlier papers on music genre classification is by [80, 66]. In [66] the authors investigated a simple 4 class genre problem using neural networks (ETMNN[1]) and Hidden Markov Models (HMM). The work by [80] investigated a simple 3 genre classification setup consisting of only 12 music pieces in total. Recent work has considered even more realistic genre taxonomies [18], and has progressively adopted even more versatile taxonomies [94]. This chapter presents inherent problems of music genre classification that every researcher faces. Furthermore, the existence of music genre classification task is motivated as a valid test-bed for envisaging improved algorithms that increases our understanding of music similarity measures.

2.1 Music taxonomies - genre

Music genre is still the most popular music descriptor for annotating the contents of large music databases. It is used to enable effective organisation, distribution and retrieval of electronic music. The genre descriptor simplifies navigation and organisation of large repositories of music titles. Music genre taxonomies are used by the music industry, librarians and by consumers to organise their expanding collections of music stored on their personal computers. The num-

[1] Explicit Time Modelling Through Neural Networks

ber of music titles in the western world is currently around 10 million. This figure would be close to 20 million if the music titles produced outside the western world were added [108]. The task of assigning relevant metadata to this amount of music titles can be an expensive affair. The music genome project `www.pandora.com` is a music discovery service designed to help users find and enjoy music they like. In a recent interview with the funder of the this project *Tim Westergren*, he reckons that each individual song takes on average around $25 - 30$ minutes to annotate. The music genes contain information about the music such as melody, harmony, rhythm, instrumentation, orchestration, arrangement, lyrics and vocal quality.

Organisation of music titles by genre has been selected by music distributors and naturally been accepted by users. There are problems related with musical genre however. An analysis performed by [109] investigated taxonomies applied in various music environments such as: record company catalogues (like Universal, Sony Music etc.), web-portals, web-radios and specialised books. The first problem encountered was that music genre taxonomies can either be based on music titles, artists or on albums. Record companies still sells collections of music titles in form of CD's, which means that the genre taxonomy is 'album-oriented'. Transforming a detailed album-oriented genre taxonomy into a 'music-title-oriented' taxonomy are bound to create confusion among the different genres, since artists might span different music styles on their albums. However, for very distinct music genres such as rock and classical, the confusion is minimal. An Internet database such as `http://www.freedb.org`, is an example of a free metadata service to music players, which uses an album-oriented taxonomy. Furthermore, record companies might distribute an album under a certain mainstream music genre, such as rock to increase sales, which also leads to greater confusion among genres. Another problem is inconsistencies between different genre taxonomies [109]. Three different web-portals[2]: `allmusic.com` (AMG, with 531 genres), `amazon.com` (with 719 genres) and `mp3.com` (with 430 genres) were used in this study. Their analysis revealed that only 70 genre words were in common between the three taxonomies. A more detailed analysis showed little consensus in the music titles shared among these genres, hence the same rule set is not being applied to the different taxonomies. Another problem is that of redundancies in the genre taxonomy. As an example consider the genre 'import' from `www.amazon.com`, which refers to music from other countries. Moving into the internal node of 'import' the next level of genre nodes is more or less similar to the level of the root node. Hence, the sub-genre rock, classical etc. is repeated, just for all imported music.

From the above inconsistencies, the authors of [109] made an attempt to create

[2] The investigations where conducted in 2000, however, there is no reason to believe the outcome of this analysis has changed.

a taxonomy of music titles as objectively as possible, minimising some of the problems inherent to the investigated taxonomies. As mentioned in [7] they did not solve the task due to several reasons: 1) bottom taxons of the hierarchy were very difficult to describe objectively and 2) the taxonomy was sensitive to music evolution (new appearing genres).

A basic assumption is that music titles belong to one genre only. However, some music is not confined to only one genre. An example of two very different genres combined into something musically provocative was provided by the Danish band 'Sort Sol' producing music in the genre rock/goth. This was in a duet with the famous Norwegian singer 'Sissel Kyrkjebø' (folk/classical/pop) on the track 'Elias Rising' of their album 'Snakecharmer'. Other rock and roll groups have performed with live Symphony orchestras adding an extra dimension to their music. Examples are the 'The Scorpions', 'Metallica' and most recently 'Evanescence' just to name a few. All of these groups use the orchestra to enhance their own sound. These examples illustrate that diverse music styles can be combined without affecting humans decision on the resulting music genre. Acoustically, the mixture of classical and rock will confuse the learning algorithm and would require a different labelling of the music.

The above examples of irregularities found in different music genre taxonomies serve to illustrate that music genre is inherently an ill-defined concept and that care must be taken when developing methods for such systems.

2.1.1 Different types of genre taxonomies

A brief investigation of different taxonomies found at various web-portals[3] reveals that genre taxonomies are either based on a hierarchical or a flat genre structure. In Figure 2.1 an 11 genre taxonomy from `www.soundvenue.com` is illustrated as an example of a flat genre taxonomy. An extract from a hierarchical genre structures from `www.allmusic.com`, is shown in Figure 2.2. After the second level, one discriminates between different music styles and sub-styles. An artist typically produces music in a single sub-genre, but can belong to several styles. One such example is 'Madonna' who's songs belongs to the sub-genres rock with styles dance-pop, adult contemporary, pop-rock and club-dance. Other Internet portals that use a hierarchical structure are e.g. `amazon.com`, `mp3.com`, `mymusic.dk`.

[3] `www.amazon.com`, `www.mp3.com`, `www.garageband.com`, `www.soundvenue.dk`, `www.mymusic.dk`, for snapshots see Appendix C.

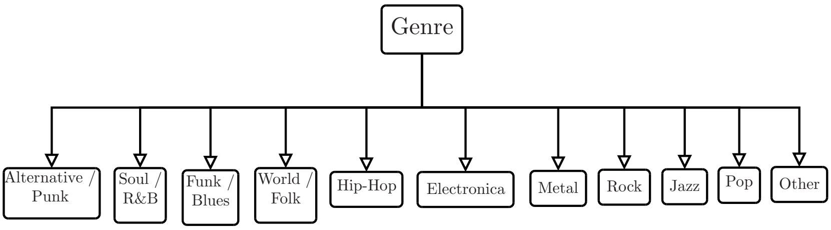


Figure 2.1: Example of flat genre structure from www.soundvenue.com a Danish music portal for music exchange.

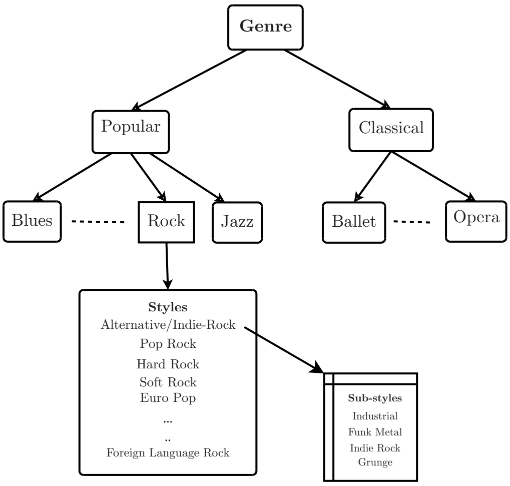


Figure 2.2: An example of a music genre hierarchy found at www.allmusic.com. Only the popular genre contains different styles and sub-styles, whereas the classical genre is at its leaf node.

2.1.2 Other taxonomies in music

As indicated in Chapter 1 music genre is not the only method for organising and navigating music titles. Several other approaches could be applied. From the investigated Internet providers, only `allmusic.com` provided other ways of navigating their repertoires. Here it is possible to navigate by mood, theme, country of origin or by instrumentation. They present 179 mood categories such as 'angry', 'cold' and 'paranoid'. This large taxonomy of moods, will naturally lead to inconsistencies, since how does one discriminate between the moods 'sexy' and 'sexual'? The theme taxonomy consists of 82 different music themes such as 'anniversary', 'in love', 'club', 'background music'. Some of the more objective navigation methods are by country of origin or instrumentation. It is only recently researchers have looked at supervised systems for mood and emotion detection from the music acoustics [137, 114, 94]. Also more elaborate taxonomies are spawned from collaborative filtering of metadata provided by users of the service from `www.moodlogic.com`. Moodlogic delivers a piece of software for generation of playlists and/or music organisation of the users personal music collections. Here, metadata such as mood, tempo and year is collected from user feedback and shared among the users of their system to provide a consistent labelling scheme of music titles.

Another interesting initiative was presented at `www.soundvenue.com`, where only 8 genres were considered. Each artist was rated with a value between $1-8$ to indicate the artists activity in the corresponding genre. This naturally lead to quite a few combinations, and since each artist was represented in multiple genres, this made browsing interesting. Furthermore, most users have difficulty in grasping low level detailed taxons from large hierarchical genre taxonomies, but have an idea if the artist, for example should be more rock-oriented. From the current homepage, `www.soundvenue.com`, this taxonomy is no longer in use.

2.2 Music genre classification

Having acknowledged that music genre is an ill-defined concept, it might seem odd that much of the MIR research has focussed on this specific task, see e.g. [110, 93, 2, 71, 86, 18, 97, 85, 141, 142] and [99, appendix H] just to mention a few. The contributions have primarily focused on small flat genre taxonomies with a limited number of genres. This minimises confusion and makes analysis of the results possible. One can consider these small taxonomies as "playpens" for creating methods, which works on even larger genre taxonomies. Furthermore, machine learning methods that have shown success in music genre classification, see e.g. [93, 13] have also been applied successfully to tasks such as artist identification [14, 93], or active learning [94] of personal taxonomies.

Due to the risk of copyright infringements when sharing music databases, it has been normal to create small databases for test purposes. It is only recently, that larger projects such as [22] make large scale evaluations on common taxonomies possible. It is projects like this, and contents like MIREX [38], which will lead to a better understanding of important factors of music genre classification and related tasks.

There are in principle two approaches to music genre classification, either from the acoustic data (the raw audio) or from cultural metadata, which is based on subjective data such as music reviews, playlists or Internet based searches. The research direction in MIR has primarily focused on building systems to classify acoustic data into different simple music taxonomies. However, there is an interplay between the cultural metadata and the acoustic data, since most of the automatic methods work in a supervised manner, thus requiring some annotated data. Currently, most research has focused primarily on flat genre taxonomies with single genre labels to facilitate the learning algorithm. Only a few researchers have been working with hierarchical genre taxonomies, see e.g. [18, 142].

The second approach to music genre classification is from cultural metadata which can be extracted from the Internet in the form of music reviews, Internet based searches (artists, music titles etc.) or from playlists (personal playlists, streaming radio, mix from DJ's). People have been using co-occurrence analysis or simple text-mining techniques for performing tasks such as hit detection [33], music genre classification [78, 67], and classification of artists [12, 78].

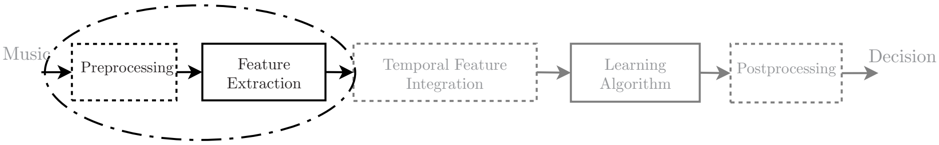
The shortcomings of the cultural metadata approach is that textual information on the music titles are needed, either in the form of review information, or from other relational data. This drawback enforces methods, which are based on purely acoustical data and learns relationships with appropriate cultural metadata.

The current level of performance of music genre classification is close to average human performance, see e.g. [99, appendix H], for reasonably sized genre taxonomies and furthermore, easily handles music collections of 1000 or more music titles with a modern personal computer.

Music genre is to date the single most used descriptor of music. However, as argued, music genre is by nature an ill-defined concept. It was argued that current taxonomies have various shortcomings such as album, artist or title oriented taxonomies, non-consistency between different taxonomies and redundancies in the taxonomomies. Devising systems, which can help users in organising, navigating and retrieving music from their increasing number of music titles is a task that interests many researchers. Simple taxonomies have been investigated using various machine learning approaches. Various music genre classification systems have been investigated by several researchers and it is recognised as an important task, which in combination with subjective assessment makes the quality and predictability of such a learning system possible.

Chapter 3

Selected features for music organisation



This chapter will focus on feature extraction at short-time scales[1], denoted as short-time features. Feature extraction is one of the first stages of music organisation, and it is recognised that good features can decrease the complexity of the learning algorithm while keeping or improving the overall system performance. This is one of the reasons for the massive investigations of short-time features in speech related research such as automatic speech recognition (ASR) and in MIR.

Features or feature extraction methods for audio can be divided into two categories, either into a physical or perceptual category. Perceptually inspired features are adjusted according to the human auditory system (HAS), whereas physical features are not. An example is 'loudness', or intensity of a sound perceived by humans. Sound loudness is a subjective term describing the strength of the ear's perception of a sound. It is related to sound intensity but can by no means be considered identical. The sound intensity must be factored by the ear's sensitivity to the particular frequencies contained in the sound. This information is typically provided in the so-called 'equal loudness curves' for the

[1] Typically in the range of $5 - 100$ms.

human ear, see e.g. [119]. There is a logarithmic relationship between the ear's response of an increasing sound intensity to the perception of intensity. A rule-of-thumb for loudness states that the power must be increased by a factor of ten to sound twice as loud. This is the reason for relating the power of the signal to the loudness simply by applying the logarithm ($\log 10$).

The individual findings by researchers in MIR does seem to move the community in a direction of perceptual inspired features. In [97], the authors investigated several feature sets, including a variety of perceptual inspired features in a general audio classification problem including a music genre classification task. They found a better average classification using the perceptual inspired features. In the early work on general audio snippet classifications and retrieval by [148], the authors considered short-time features such as loudness, pitch, brightness, bandwidth and harmonicity. Since then, quite a few features applied in other areas of audio have been investigated for MIR. Methods for compression, such as wavelet features were investigated in [143] for automatic genre classification. Features developed for ASR such as Mel Frequency Cepstral Coefficients (MFCC) and Linear Predictive Coefficients (LPC) have also been investigated for applications in MIR, see e.g. [98, 48, 7, 19].

The below mentioned music snippets have been applied in various illustrations throughout this thesis:

- S_1: Music snippet of 10 sec from the song 'Masters of revenge' by the hard rock band 'Body Count'.
- S_2: Music snippet from the song 'Fading like a flower' by the pop/rock group 'Roxette'. A music snippet of length 10 sec and 30 sec was generated.

3.1 Preprocessing

In this thesis music compressed in the well known MPEG-1 layer III format (MP3) as well as the traditional PCM format have been used. A typical preprocessing of the music files consists in converting the signal to mono. A real music stereo recording will contain information, which can aid extraction of e.g. the vocal or instruments playing, if these are located in spatially different locations, see e.g. [111], which considers independent component analysis (ICA) for separating instruments. The presented work in this thesis has focussed on information obtained from mono audio. The music signals is down-sampled by a factor of two from 44100 Hz to 22050 Hz, with only a limited loss of perceptual information. The impact of such a down-sampling was briefly analysed in a

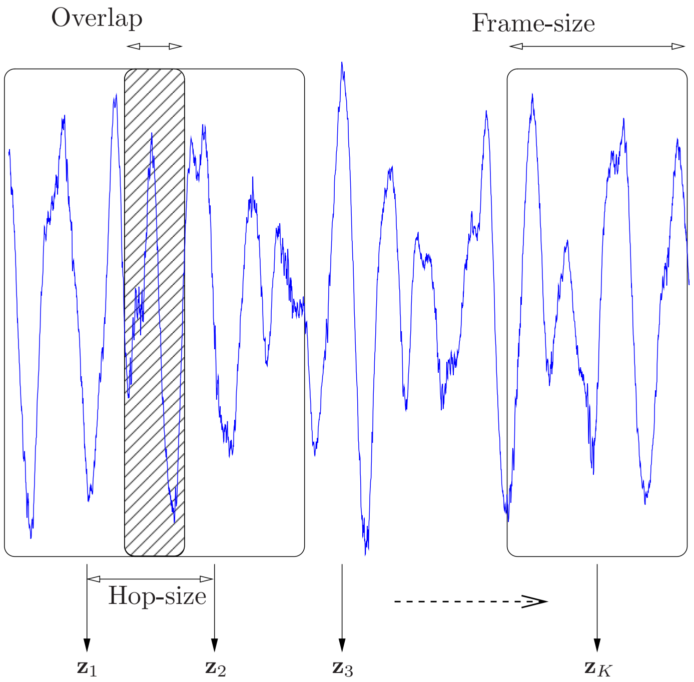


Figure 3.1: A 100ms audio snippet of the music piece S_1 illustrating the idea of frame-, hop-size and overlap. The hatched area indicates the amount of overlap between subsequent frames. The short-time feature vector extracted from the music signal is denoted by $\mathbf{z}$.

music similarity investigation in [9], and was found negligible. The digital audio signal extracted from the file is represented as $x[n] \in \mathbb{R}$ for $n = 0, \ldots, N-1$, where N is the number of samples in the music file. As a final preprocessing stage the digital audio signal is mean adjusted and power normalised.

3.2 A general introduction to feature extraction

Feature extraction can be viewed as a general approach of performing some linear or nonlinear transformation of the original digital audio sequence $x[n]$ into a new sequence $\mathbf{z}_k$ of dimension D for $k = 0, \ldots, K-1$. A more strict formulation of the feature extraction stage can be written as

$$z_{d,k} = g_d\left(x[n]w[h_s k + f_s - n]\right) \quad \text{for} \quad n = 0, 1, \ldots, N-1. \tag{3.1}$$

where $w[m]$ is a window function[2], which can have the function of enhancing the spectral components of the signal. Furthermore, the window is selected such that

$$\begin{array}{ll} w[n] \geq 0 & 0 \leq n \leq f_s - 1 \\ w[n] = 0 & \text{elsewhere.} \end{array} \tag{3.2}$$

The hop- and frame-size are denoted as h_s and f_s, respectively[3], and are both positive integers. The function $g_d(\cdot)$ maps the sequence of real numbers into a scalar value, which can be real or complex. Figure 3.1 illustrates the block based approach to feature extraction showing a frame-size of approximately 30 ms and a hop-size of 20 ms. The hatched area indicate the amount of overlap (10 ms) between subsequent frames.

3.2.1 Issues in feature extraction

Feature extraction is the first real stage of compression and knowledge extraction, which makes it really important for the overall system performance. Issues such as frame/hop-size selection, quality of the features in the global setting as well as the complexity of the methods are relevant. Frame/hop-size selection has an impact on the complexity of the system as well as the quality of the resulting system. If the frame-size is selected too large, detailed time-frequency information of the music instruments, vocal etc. is lost and a performance drop of the complete system is observed. Conversely, using too small a frame-size results in a noisy estimate of especially the lower frequencies.

[2] Typical windows applied is the rectangular, Hann or Hamming type of windows.

[3] If the hop- or frame-size are provided in milliseconds, they can be converted to an integer by multiplying with the samplerate (s_r) and rounding to the nearest integer.

3.3 Feature extraction methods

In the coming sections, the different feature extraction methods investigated in the present work is discussed. Their computational complexity are based on a single frame. The quality of the different features are measured in terms of their impact of the complete system performance, which will be discussed further in Chapter 6. In addition to the perceptual / non-perceptual division of audio features, they can be further grouped as belonging to either temporal or spectral features. In the present work the following audio features have been considered:

- **Temporal features:** Zero Crossing Rate (ZCR) and STE.
- **Spectral features:** Mel Frequency Cepstral Coefficients (MFCC), Linear Predictive Coding (LPC), MPEG-7: Audio Spectrum Envelope (ASE), Audio Spectrum Centroid (ASC), Audio Spectrum Spread (ASS) and Spectral Flatness Measure (SFM).

These features have been selected from previous works on various areas of music information retrieval.

3.3.1 Spectral features

The feature extraction methods presented in this section are all derived from the spectral domain. From a spectral investigation over a frame where the music signal is considered stationary, one is left with a magnitude and phase for each frequency component. The phase information for humans at the short-time scales considered is less important than the magnitude. However, recent studies have shown that phase information can be an important factor for music instrument recognition [40]. In [40], the authors found the phase information in the sustained part of the played instrument important. Also in [146], the phase information was found useful for onset detection in music. The onset detection algorithm was a part of a larger system for music genre classification. The spectral feature methods considered in this thesis, is only using the magnitude spectrum. Thus, we do not include any phase information. The spectral features described in this section all have the discrete short-time Fourier transformation (STFT) in common, see e.g. [116]:

$$z^{STFT}[d,k] = \sum_{n=0}^{N-1} x[n]w[kh_s + f_s - n]e^{-j2\pi dn/f_s} \tag{3.3}$$

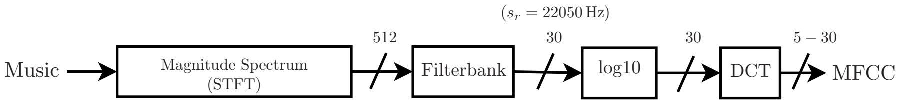


Figure 3.2: The figure illustrates a MFCC extraction scheme. The numbers above each stage expresses the dimension of a typical dimensionality reduction taking place in such a feature extraction stage.

where $d = 0, \ldots, f_s/2$ when f_s is even and $d = 0, \ldots, (f_s - 1)/2$ when f_s is odd.

3.3.1.1 Mel Frequency Cepstral Coefficient (MFCC)

These features were originally developed for automatic speech recognition for decoupling the vocal excitation signal from the vocal tracts shape [29], but have found applications in other fields of auditorial learning, including music information retrieval. Just to mention a few, audio retrieval: [94, 9, 82, 104] and [51], audio fingerprinting: [20, 21], automatic genre classification: [142, 93], [4, appendix D] and [98, appendix E], audio segmentation: [48] and [87]. The MFCCs are in principle a compact representation of the general frequency characteristics important for human hearing. They are ranked in such a way that the lower coefficients contain information about the small variations of the spectral envelope. Hence, adding a coefficient will increase the detail level of the envelope. These features belong to the group of perceptual features and have shown to be good models of 'timbre' spaces, see e.g. [139], where timbre is a catch all term referring to all aspects of sound independent of its pitch and loudness[4]. Timbre is not a frequently applied term in speech related research, however, it is more often applied in the context of modelling music sounds and especially applied in connection with the modelling of music instruments.

There is no single method for extraction of MFCCs, and the chosen approach can differ from author to author. The original procedure for extracting the MFCCs is illustrated in Figure 3.2, where the numbers above the various steps gives an intuitive idea of the dimension. The audio is transformed to frequency domain using a short-time Fourier transformation after which the power (or amplitude) of each frequency component are summed in critical bands of the human auditorial system using the Mel scale. The output of the filterbank is weighted logarithmically, and finally applied a discrete cosine transform (DCT) to decorrelate and sort the outputs of the Mel-filters. The Mel-filters are usually triangular shaped. Other types of windows can be applied such as Hamming,

[4] The timbre definition is a definition of what timbre is not, and not what it actually is, which makes the interpretation of the term timbre weak. There is a common understanding of timbre being multidimensional [72].

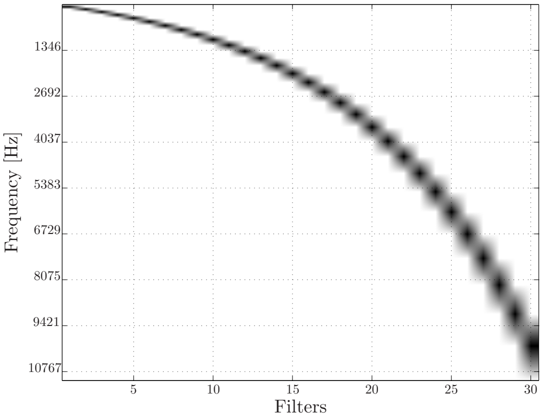


Figure 3.3: The figure shows the Mel-scaled filterbank with 30 filters distributed in the frequency range from $0 - 11025$ Hz.

Hann or rectangular windows. The MFCC extraction can be formulated in a more strict manner as,

$$z^{MFCC}[d,k] = \mathrm{DCT}_d\left(\log_{10}\left[\mathbf{W}_m^T\left|\mathbf{z}_k^{STFT}\right|\right]\right) \tag{3.4}$$

where $\mathbf{W}_m$ is the Mel-scaled filterbank of dimension $\frac{f_s}{2} \times N_f$, assuming that f_s is even. The absolute operator is applied to each scalar of the vector $\mathbf{z}_k^{STFT}$, independently. The DCT_d is a linear operation on the elements in the parenthesis, and expresses the d'th basis function of the DCT. Figure 3.3 shows a triangular filterbank with 30 filterbanks in the frequency range $0 - 11025$ Hz ($s_r = 22050$ Hz). Some authors apply the loudness transformation ($\log_{10}$ operation) after the STFT, hence, they swap the filterbank operation and log-scaling, see e.g. [51]. Furthermore, some authors normalises the filterbanks to unit power [139], where others do not [99, appendix H]. No apparent proof or clarifying experiment of preferring one to the other has been found. The delta MFCCs have been included in initial investigations of the short time features for music genre classification, and simply amounts to calculating

$$z^{DMFCC}[d,k] = z^{MFCC}[d,k] - z^{MFCC}[d,k-1]. \tag{3.5}$$

These features encode information about the local dynamics of the MFCC features. When there is a high correlation between frames, this feature will be zero, or close to zero. The feature is likely to be more discriminative with little or no overlap between subsequent frames, since for a large overlap little temporal change will be observed. The implementation, which have been used in the experiments are from the 'voicebox' by [16]. The number of filters is by default set to $N_f = 3\log(s_r)$, which amounts to 30 filters at a samplerate of $s_r = 22050$ Hz. In principle various authors are presenting the number of MFCCs applied, but

typically does not state the number of Mel-filters applied, which can be an important information.
The complexity of the MFCC calculation is dominated by the complexity of the STFT, which amounts to $\mathcal{O}(f_s \log_2(f_s))$.

3.3.1.2 Linear Predictive Coding (LPC)

Linear predictive coding originally developed for speech coding and modelling, see e.g. [91] represents the spectral envelope of the digital audio signal in a compressed form in terms of the LPC coefficients. It is one of the most successful speech analysis techniques and is useful for encoding good quality speech at low bit-rates. LPC is based on the source filter model, where a pulse train (with a certain pitch) is passing trough a linear filter. The filter models the vocal tract of the speaker. For a music instrument the vocal tract is exchanged with the resonating body of the instrument. In linear prediction we estimate the coefficients of the AR-model [91]:

$$x[n] = \sum_{p=1}^{P} a_p x[n-p] + u[n], \tag{3.6}$$

where the a_p's for $p = 1, \ldots, P$ is the filter coefficients of an all-pole model, which controls the position of poles in the spectrum and $u[n]$ is a noise signal with zero mean and finite variance (finite power).

Several different extensions to the LPC have been proposed. One example is the perceptual LPC (PLP[5]) [64], which extends the normal LPC model using both frequency warping according to the Bark scale [138] and approximate equal loudness curves. The authors of [64] illustrate that a 5th order PLP model is performing just as well as a 14th order LPC model when suppressing speaker dependent information from speech. The traditional LPCs have been applied for singer identification in [75], where both the traditional and a warped LPC [61] were compared. The warped LPC consists of a warping of the frequency axis according to the Bark scale (similar to the PLP-coefficients). The warping results in a better modelling of the spectral components at the lower frequencies as opposed to the traditional LPC method where the spectral components of the whole frequency span are weighted equally. Their investigation, however, did not reveal any apparent gain from the warped LPC in terms of accuracy for singer identification. In [150], the LPC model was applied for fundamental frequency estimation. The model order has to be high enough to ensure a proper modelling of the peaks in the spectra. They used a model order of 40

[5] Perceptual Linear Prediction

and only considered spectras where the peaks were clearly expressed. Using the greatest common divisor between clearly expressed peaks, reveals if there is any harmonicity in the signal. Furthermore, the frequency corresponding to the greatest common divisor was selected as an estimate of the fundamental frequency. In this thesis the traditional LPC coefficients have been investigated for music genre classification. These features are not perceptually inspired, and will to some extent be correlated with the fundamental frequency. The LPC-derived feature becomes

$$\mathbf{z}_k^{LPC} = \left[\begin{array}{ccccc} \hat{a}_1 & \hat{a}_2 & \ldots & \hat{a}_P & \hat{\sigma}^2 \end{array} \right]^T . \tag{3.7}$$

There are several approaches to estimating the parameters of an autoregressive model, see e.g. [115]. The voicebox [16] have been applied for estimating the autoregressive parameters, which implements the autocorrelation approach. The LPC model will be discussed further in Chapter 4.
The complexity of the inversion amounts to $\mathcal{O}(P^3)$, however, the process of building the autocorrelation matrix is $\mathcal{O}(\frac{P(P-1)}{2} f_s)$. Exploiting the symmetry in the autocorrelation matrix, the inversion problem can be solved in $\mathcal{O}(P^2)$ operations.

3.3.1.3 MPEG-7 framework

The MPEG-7 framework (Multimedia Content Description Interface) standardised in 2002 has been developed as a flexible and extensible framework for describing multimedia data. The successful MPEG-1 and MPEG-2 standards mostly focused on efficiently encoding of multimedia data. The perceptual coders use psychoacoustic principles to control the removal of redundancy to minimise the perceptual difference of the original audio signal to that of the coded for a human listener.

MPEG-4 is using structured coding methods, which can exploit structure and redundancy at many different levels of a sound scene. According to [127] this will in many situations improve the compression by several orders of magnitude compared to the original MPEG-1 and MPEG-2 encodings. Researchers finalised their contributions to the MPEG-4 framework in 1998 and it became an international standard in 2000. The encoding scheme has since then been adopted by Apple computer in products such as iTunes and Quicktime.

To discriminate between the different standards one could say that the MPEG-1, 2, 4 standards were designed to represent the information itself, while the MPEG-7 standard [68] is designed to represent information about the information [95].

In this thesis the short-time features from the 'spectral basis' group have been investigated. This group consists of the Audio Spectrum Envelope (ASE), Audio Spectrum Centroid (ASC), Audio Spectrum Spread (ASS) and the Audio Spectral Flatness (ASF). Detailed information about the MPEG-7 audio standard can be found in [68]. In [114] the authors investigated how the MPEG-7 low level descriptors consisting of ASC, ASS, ASF and audio harmonicity performed in a classification of music into categories such as perceived tempo, mood, emotion, complexity and vocal content. The ASF was investigated in [5] for robust matching (audio fingerprinting) applications, investigating robustness to different audio distortions (cropping/encoding formats/dynamic range compressions). In [145] the ASE features were investigated for audio thumbnailing using a self-similarity map similar to [48]. [18] considered the ASC, ASS and ASF features and others for hierarchical music genre classification. The Sound Palette is an application for content based processing and authoring of music. It is compatible with the MPEG-7 standard descriptions of audio [24].

3.3.1.4 Audio Spectrum Envelope (ASE)

The ASE describes the power content of the audio signal in octave spaced frequency bands. The octave spacing is applied to mimic the 12-note scale, thus, the ASE is not a purely physical feature. The filterbank has one filter from 0 Hz to $loEdge$, a sequence of filters octave spaced between $loEdge$ and $hiEdge$ and a single filter from $hiEdge$ to half the sampling rate s_r. The $loEdge$ frequency is selected such that at least one frequency component is present at the lower frequency bands. The resolution in octaves is specified by r.

Except for $r = 1/8$, the $loEdge$ and $hiEdge$ is related to a 1kHz anchor point by

$$f_m^e = 2^{rm} 1000\,\text{Hz} \tag{3.8}$$

where f_m^e specify edge frequencies for the octave filterbank and m is an integer. With a samplerate of $s_r = 22050\,\text{Hz}$, a frame-size of 1024 samples the low edge frequency is selected to $loEdge = 62.5\,\text{Hz}$ and a high edge frequency of $hiEdge = 9514\,\text{Hz}$ according to the standard. This results in a total of 32 frequency bands. The filterbank consists of rectangular filters, which are designed with a small overlap (proportional to the frequency resolution of the STFT) between subsequent filters. The MPEG-7 filterbank for the above configuration is illustrated in Figure 3.4. As observed from the figure, there are more filters below 1 kHz than the Mel-scaled filterbank. The MPEG-7 ASE can also be written compactly as

$$z^{ASE}[d,k] = c\mathbf{W}_M^T |\mathbf{z}_k^{STFT}|^2, \tag{3.9}$$

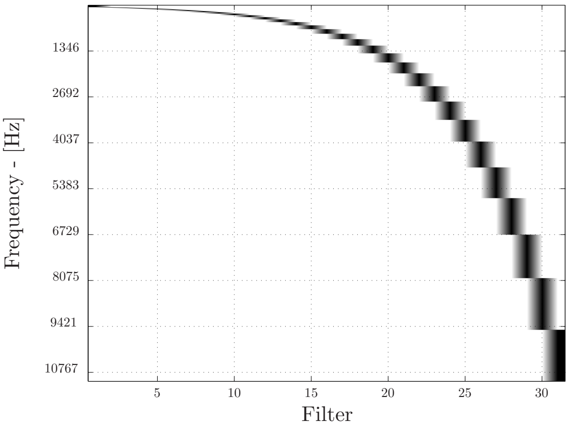


Figure 3.4: The MPEG-7 filterbank for a samplefrequency of $s_r = 22050$ Hz and frame-size of 1024. This results in a *loEdge* frequency of 62.5 Hz and *hiEdge* frequency of 9514 Hz.

where the $|.|^2$ operation applies to each of the elements of the vector $\mathbf{z}_k^{STFT}$ independently, c is a scaling proportional to the length of the frame f_s (or zero-padded[6] length) and $\mathbf{W}_M$ is the MPEG-7 filterbank.

The short-time ASE feature has been applied in applications such as audio thumbnailing [145] and various audio classification tasks [23, 18] and [4, appendix D].
The complexity amounts to $\mathcal{O}(f_s \log_2(fs))$ if f_s is selected such that $\log_2(f_s)$ is an integer (otherwise zero padding is applied).

3.3.1.5 Audio Spectrum Centroid (ASC)

The ASC describes the center of gravity of the octave spaced power spectrum and explains if the spectrum is dominated by low or high frequencies. It is related to the perceptual dimension of timbre denoted as the sharpness of the signal [68]. The centroid, calculated without a scaling of the frequency axis has been applied in classification of different audio samples [148] and in [96] for monophonic instrument recognition. The ASC short time feature was used, among other features, for hierarchical music genre classification in [18]. The

[6] Zero-padding corresponds to "padding" a sequence of zeroes after the signal prior to the DFT. The zero-padding results in a "better display" of the Fourier transformed signal $X(\omega)$, however, does not provide any additional information about the spectrum, see [115].

ASC feature is calculated as

$$z^{ASC}[k] = \frac{\sum_{d=0}^{f_s/2} \log_2(f_d/1000)|z^{STFT}[d,k]|^2}{\sum_{d=0}^{fs/2} |z^{STFT}[d,k]|^2} \tag{3.10}$$

where f_d is the dth frequency component (expressed in Hz) of $z^{STFT}[d,k]$. There is special requirements for the lower edge frequencies, which is further explained in the MPEG-7 audio standard. Having extracted the ASE feature, only $\mathcal{O}(f_s)$ operations is required to extract the ASC.

3.3.1.6 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. A large spread could indicate how noisy the signal is, whereas a small spread could indicate if a signal is dominated by a single tone. Similar to the ASC, the ASS is determined as

$$z^{ASS}[k] = \frac{\sum_{d=0}^{f_2/2} \left(\log_2(f_d/1000) - z^{ASC}[k]\right)^2 |z^{STFT}[d,k]|^2}{\sum_{d=0}^{f_s/2} |z^{STFT}[d,k]|^2}. \tag{3.11}$$

3.3.1.7 Audio Spectral Flatness (ASF)

The audio spectral flatness measure can be used for measuring the correlation structure of an audio signal [39]. Like the audio spectral spread, the ASF can be used for determining how tone, or noise like an audio signal is. The meaning of a tone in this connection, is how resonant the power spectrum is compared to a white noise signal (flat power spectrum). In [39] the spectral flatness measure (SFM) for a continuous spectrum is given as

$$SFM = \frac{\exp\left(\frac{1}{2\pi}\int_{-\pi}^{\pi} \ln\left(S(\omega)\right) d\omega\right)}{\frac{1}{2\pi}\int_{-\pi}^{\pi} S(\omega) d\omega} \tag{3.12}$$

where $S(\omega)$ is the power spectral density function of the continuous aperiodic time signal $x(t)$. In principle, the spectral flatness is the ratio between the geometrical and arithmetical average of the power spectrum. The spectral flatness measure has been applied both for audio fingerprinting [5], and for classification of musical instruments [39].

The ASF is calculated according to the octave spaced frequency axis, hence, the tonality is measured in the given sub-band specified by the MPEG-7 filterbank.

The ASF is defined for a resolution of $r = 1/4$ and the low edge signal is additional required to be 250 Hz. Furthermore, a larger overlap between the filters is applied, such that filters overlap with 10% to their neighbouring filter. The ASF is calculated as

$$z^{ASF}[d,k] = \frac{\left(\prod_{i \in b_d} |z^{STFT}[i,k]|^2\right)^{1/N_d}}{\frac{1}{N_d}\sum_{i \in b_d} |z^{STFT}[i,k]|^2}, \tag{3.13}$$

where b_d are the indices of the nonzero element of the d'th filter and N_d is the number of non-zero elements in the $d'th$ filter (bandwidth estimate of the filter). Furthermore, when no signal is present in the band indexed by b_d, $z^{ASF}[d,k]$ is set to 1. With the above definition $z^{ASF} \in [0,1]$, where 0 and 1 indicate tone like and noise like signals, respectively.

It should be noted that the ASC, ASS and ASF are robust towards scaling of the power-spectrum with some arbitrary constant value c.

3.3.2 Temporal features

3.3.2.1 Short-Time Energy (STE)

The short-time energy is the running estimate across the time signal of the energy and is calculated as

$$z_k^{STE} = \sum_{n=0}^{N-1} \left(x[n]w[kh_s + f_s - n]\right)^2. \tag{3.14}$$

In the investigations a simple rectangular window has been applied, which simply amounts to summing across the samples of $x[n]$, hence,

$$z_k^{STE} = \sum_{d=0}^{f_s-1} x[kh_s + f_s - d]^2. \tag{3.15}$$

It has been applied in a range of music applications, see e.g. [97, 88, 150, 82]. The STE is relevant since it is cheap to calculate, only $\mathcal{O}(f_s)$ operations. Furthermore, its temporal change pattern provide information of the tempo of the music signal.

3.3.2.2 Zero Crossing Rate (ZCR)

Defining the indicator variable $v[n]$ as

$$v[n] = \begin{cases} 1, & x[n] \geq 0 \\ 0, & x[n] < 0 \end{cases}, \tag{3.16}$$

and the squared difference $g[n] = (v[n] - v[n-1])^2$ then the ZCR over a frame is calculated simply as

$$z^{ZCR}[k] = \sum_{d=1}^{f_s-1} g[kh_s + f_s - d]. \tag{3.17}$$

With the above definition it can be shown [74], that there exists the following relation between the autocorrelation of the audio signal $x[n]$ and the number of zero-crossings as

$$\rho_1 = \cos\left(\frac{\pi \mathrm{E}\left[z^{ZCR}[k]\right]}{f_s - 2}\right), \tag{3.18}$$

where $\rho_1 = \frac{\mathrm{E}\{x[n]x[n-1]\}}{\mathrm{E}\{x[n]^2\}}$ since the mean of $x[n]$ is zero. One can consider the system as a binary Markov chain, being in one of its two states. For a random signal with zero mean, the sign change probability of $x[n]$ would be purely chance, hence $p = 0.5$, which amounts to $\mathrm{E}\{z^{ZCR}[k]\} = \frac{f_s-2}{2}$ zero crossings. For a signal with local correlations one would expect either a higher or smaller zero crossing rate[7].

The zero crossing rate has been applied in number of applications such as audio segmentation [125, 150, 43], classification [114, 149], retrieval [82] and speech processing [74, 73, 116]. The ZCR have been considered as a cheap alternative to spectral analysis, see e.g. [74], where the author provides a detailed theoretical investigation of the zero crossing rate.
The complexity of the ZCR amounts to $\mathcal{O}(f_s)$ and must be considered the cheapest of the discussed feature extraction methods.

3.4 Frame- and hop-size selection

The frame-size of an audio signal is normally selected from the local stationarity of the signal. In speech recognition the audio signal is considered stationary for

[7] A high-frequency signal, which changes sign rapidly or a low frequency signal, where sign changes occur less frequently.

intervals of approximately $20 - 40$ ms, see e.g. [30]. For music, the local stationarity varies slightly more and the frame-size is usually selected in the range 10-100 ms. In [8] an investigation of similarity between songs were investigated when varying the frame-size of the short-time features between $10-1000$ ms. The performance dropped when the frame-size increased more than approximately 70 ms[8]. For feature extraction methods that can handle non-stationary data, larger frame-sizes can be used [143], and results in better spectral resolution at lower frequencies.

The hop-size h_s is typically selected such that a 50% overlap between subsequent frames is achieved. From a practical point of view, it would make more sense to have no overlap, since this would decrease complexity considerably [101]. Tasks such as audio fingerprinting are rather sensitive to overlap. Using a small hop-size will compensate for so-called alignment noise, which is defined as the noise resulting from a stored fingerprint is temporally out of phase with the analysis frames [17, 59, 20]. This may happen when a music signal is represented in different encoding formats.

From a signal point of view, it is possible to devise a method for selecting a hop-size that minimises aliasing. This approach to hop-size selection, which have been outlined in [116], will be discussed through a small example.
Many of the proposed feature extraction methods in the literature involves sums of the audio data in the given frame. For the simple temporal short-time energy (STE) feature, applied in the following analysis, a method, which is optimal w.r.t. the Nyquist sampling criteria [115] is explained. The STE for a frame was given as

$$z_k^{STE} = \sum_{n=0}^{N-1} w[kh_s + f_s - n]^2 x[n]^2, \tag{3.19}$$

for $k = 0, 1, \ldots, K-1$. This corresponds to a linear filtering of the squared audio signal $x[n]^2$ using the squared filter $w[kh_s + f_s - n]^2$ [9]. In frequency domain this corresponds to a multiplication of the spectrum of $x[n]^2$ with that of the squared window function $w[n]^2$.

Using a rectangular window function, $w[n] = \frac{1}{f_s}$ for $0 \leq n \leq f_s - 1$ and $w[n] = 0$ elsewhere, the Fourier transformation of the squared filter becomes

$$W(\omega) = \frac{1}{f_s^2} \frac{sin(\omega f_s/2)}{sin(\omega/2)} e^{-j\omega(f_s-1)/2}. \tag{3.20}$$

The low pass filter characteristics of the rectangular window are evaluated by

[8] The MPEG-1 layer III (MP3) (ISO 11172-5) encoding standard, which has been popularised the last couple of years, uses a fixed frame size of 1152 samples. For an audio signal at a sample frequency 44100 Hz this amount to a frame-size of ~ 26.12 ms.

[9] The convolution is given as $y[k] = \sum_{n=0}^{M-1} x[n]h[k-n]$, where $h[n]$ is the filter.

determining the filters zero-crossings, hence, find ω where $W(\omega) = 0$. The zero-crossings are $\omega_g = \frac{2\pi}{f_s} g$ where $g \neq 0$. The approximate bandwidth determined from the first zero-crossing is $B = \frac{2\pi}{f_s}$, thus, the sampling rate of the short-time features (or inverse hop-size) should be selected as

$$\frac{1}{h_s} \geq 2B \rightarrow h_s \leq \frac{1}{2B} = \frac{f_s}{\pi} \tag{3.21}$$

where π represents the half the samplerate $(s_r/2)$. For a samplerate of $s_r = 22.05\,\text{kHz}$ and a frame-size of 46.44 ms $(f_s = 1024)$ the hop-size of the STE feature is determined as

$$h_s \leq \frac{2f_s}{s_r} = 23.2\,\text{ms}, \tag{3.22}$$

which amounts to an overlap of approximately 50% or more required to minimise aliasing in the resulting STE feature. Researchers have been using features with little or no overlap that introduce more aliasing at the higher frequencies of the short-time features. A typical approach for obtaining features at larger time-scales is to calculate the mean and variance across the temporal dimension of the short-time features. This method is an efficient low-pass filter, and therefore would result in little or no detoriation of system performance in tasks such as music genre classification.

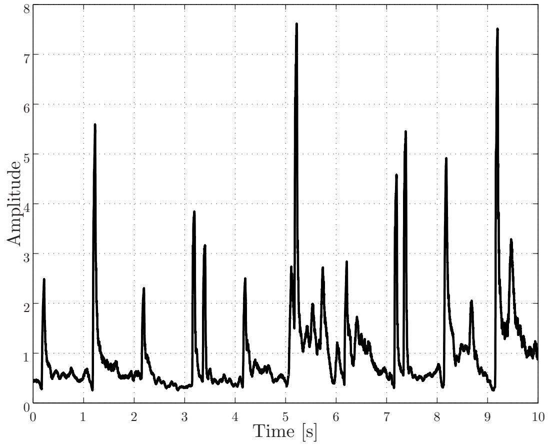
Figure 3.5(a) shows the STE feature extracted from the music signal S_1 as a function of sample index k, and Figure 3.5(b) shows the smoothed absolute difference between the "true" magnitude spectrum of the STE feature across the 10 sec (extracted using a small hop/frame-size ratio $h_f = 1/32$) and that extracted when varying the hop/frame-size ratio $h_f = h_s/f_s$.

The error signal is calculated as

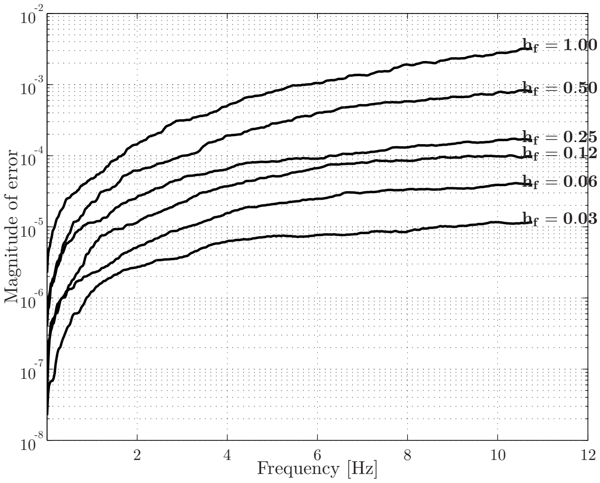
$$E[i] = \left|Z_g^{STE}[i]\right| - \left|Z_{h_f=1/32}^{STE}[i]\right|, \tag{3.23}$$

where $i = 0, 1, \ldots, 511$ and uppercase of z represents the discrete Fourier transform of z_k^{STE}. As observed, aliasing is present at all frequencies, however, the error is increased when selecting a larger h_f ratio. The rectangular window has a larger spectral leakage than e.g. the Hamming or Hann type of windows. The above procedure for selecting hop-size, should only be considered as a rule-of-thumb, since many applications are not relying on the temporal dynamics in the short time features. We will, however, device methods for modelling the temporal dynamics of the short-time features, and therefore select h_f as 1/2 or lower such that 50% overlap or more is applied.

The final impact on the system performance, however, will need to be established through an investigation of the complete system. This have been done in e.g. [99, appendix H].



(a) STE of music piece S_1



(b) Smoothed absolute difference of magnitude spectras

Figure 3.5: Figure (a) shows the STE of music piece S_1 and figure (b) shows the smoothed absolute difference between the “true” magnitude spectrum and that determined from varying the hop-size (overlap ratio).

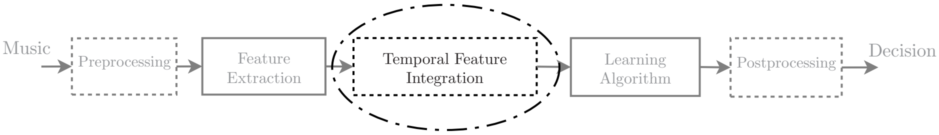
3.5 Discussion

In this section the general idea of feature extraction has been presented. Perceptual and non-perceptual feature extraction techniques were presented. Furthermore, a method for selecting hop-size to minimize aliasing was outlined.
It must be emphasised that many of the short-time features explained in this section are correlated with different perceptual quantities. In principle, when selecting features, they should be selected as independent as possible, to ensure a large span in the space relevant for the specific music organisation task. The short-time feature explained in this section have been investigated in greater detail in Chapter 6 and in published work [4, appendix D], where the short-time features are ranked after discriminative power at different time-scales in a music genre classification task.

The next section will consider methods for creating features at larger time-scales from these short-time features, which have been denoted temporal feature integration.

Chapter 4

Temporal feature integration



Feature extraction is the process of extracting useful information from the audio samples over stationary periods of the audio signal. The features extracted at this time-scale are denoted as short-time features. This section will introduce the notion of temporal feature integration [99, appendix H], which is an important part of this thesis. Essentially, temporal feature integration deals with processing of the short-time features to construct features, which are more discriminative at larger time-scales. Construction of features at a larger time-scale usually results in a sample reduction, and can therefore be considered as an efficient compression of the audio for the specific learning task, whether this is for retrieval or music genre classification.

Methods for summarising information of short-time features have been applied in earlier work in MIR. In the work by [148] simple statistics such as mean and variance as well as a few autocorrelation coefficients were calculated over the temporal dimension of the short-time features[1] of the audio snippets. These integrated short-time features was then applied for general audio classification. In [142, 141], the notion of texture windows is applied for frames of short-time

[1] Pitch, amplitude, brightness, harmonicity and bandwidth.

features. The author finds a texture window of approximately 1.0 sec to be optimal for music genre classification of a 10 genre dataset. The mean and variance of the short-time features (MFCC, spectral centroid, roll off, spectral flux and ZCR[2]) are calculated over the size of the texture window. In [82], the author extracts mean and variance of the short-time features over segments of 600ms. In [97], the authors are investigating different perceptual and non-perceptual short-time features at larger time-scales for general audio classification (music genre included). Periodograms of each short-time feature dimension, independently, are extracted from texture windows of 768 ms after which the power is summarised in 4 predefined frequency bands. With this procedure, temporal fluctuations of the short time features are included. In [44], the authors summarise short-time features over 1 minute windows and apply the temporal integrated features for segmentation and classification of long-duration personal audio.

In the following sections a definition of temporal feature integration is provided. Furthermore, different temporal feature integration models such as the mean-covariance (MeanCov), multivariate AR (MAR) and filterbank coefficient(FC) model are presented. Furthermore, two methods for extracting the tempo of music, the beat spectrum (BS) and beat histogram (BH) are presented.

4.1 Definition

Temporal feature integration is the process of combining all the feature vectors in a time frame into a single feature vector, which captures the temporal information of the frame. The new feature generated does not necessarily capture any explicit perceptual meaning such as tempo or mood of the music, but captures implicit temporal information which is useful for the subsequent learning algorithm.

The temporal feature integration can be expressed more rigorously by observing a sequence of consecutive short-time features $\mathbf{z}_k$ of dimension D where k represents the k'th short-time feature. Using a block-based approach these short-time features are integrated into a new feature $\tilde{\mathbf{z}}_{\tilde{k}}$ of dimension $\widetilde{D}$, hence

$$\tilde{\mathbf{z}}_{\tilde{k}} = \mathbf{f}(\mathbf{z}_{\tilde{k}\cdot h_{s_z}}, \ldots, \mathbf{z}_{\tilde{k}\cdot h_{s_z}+f_{s_z}}), \tag{4.1}$$

where h_{s_z} is the hop-size and f_{s_z} is the frame-size (both defined in a number of samples manner, and $\tilde{k} = 0, 1, \ldots, \widetilde{K} - 1$ is the discrete time index of the larger

[2] The spectral flux is defined as the squared difference between the normalised magnitudes of the current and previous short-time feature window. The roll-off is defined as the frequency f_R below which 85% of the magnitude distribution is concentrated.

time-scale. In the above formulation, multiplication by a rectangular window is indirectly assumed, however, in principle other windows could be applied, see Chapter 3. There exists many functions $\mathbf{f}(\cdot)$ which maps a sequence of short-time features into a new feature vector. The function is not required to be differentiable, however, differentiability allows investigations of the short-time features effect at larger time-scales. In the coming sections, the following matrix denotes a sequence of short-time feature vectors

$$\mathbf{Z}_{\tilde{k}} = \left[\begin{array}{cccc} \mathbf{z}_{\tilde{k}\cdot h_{s_z}} & \mathbf{z}_{\tilde{k}\cdot h_{s_z}+1} & \cdots & \mathbf{z}_{\tilde{k}\cdot h_{s_z}+f_{s_z}} \end{array} \right], \tag{4.2}$$

for some frame $\tilde{k}$.

4.2 Stacking

A simple approach to temporal feature integration is to stack the short-time features of the frame. This operation can be written compactly as

$$\tilde{\mathbf{z}}_{\tilde{k}} = \text{vec}\left(\mathbf{Z}_{\tilde{k}}\right), \tag{4.3}$$

where the vec-notation refers to stacking each column of the matrix $\mathbf{Z}_{\tilde{k}}$ into a single vector[3]. The dimension of $\tilde{\mathbf{z}}_{\tilde{k}}$ becomes $\tilde{D} = D \cdot f_{s_z}$, hence, the data is not compressed in any manner. Without any further preprocessing the learning algorithm can select the most important dimensions which includes time lagged versions of the original feature space. Stacking of short-time features was applied in [4, appendix D] for feature ranking in music genre classification at different time-scales, a process which will be elaborated on in Chapter 6. Stacking of features has also been applied in [126, 66] in the task of music genre classification. In [134, 135] the authors consider stacking of short time features for connecting non-speech sounds with semantic data (words) for audio retrieval and indexing. Hence, a sound of a horse in acoustic space is mapped to the word (or class of words) in a semantic space.

[3] $vec(\mathbf{Z}_{\tilde{k}}) = \begin{bmatrix} \mathbf{z}^T_{\tilde{k}\cdot h_{s_z}} & \cdots & \mathbf{z}^T_{\tilde{k}h_{s_z}+f_{s_z}} \end{bmatrix}^T$

4.3 Statistical models

The short-time features extracted from the music piece can be considered as a multivariate time-series. Figure 4.1 shows the first 6 MFCCs of the music example S_1. Each of the feature dimensions have been normalised to unit variance. From the time series a temporal dependency in the features as well as cross-dependencies among the feature dimensions are observed.

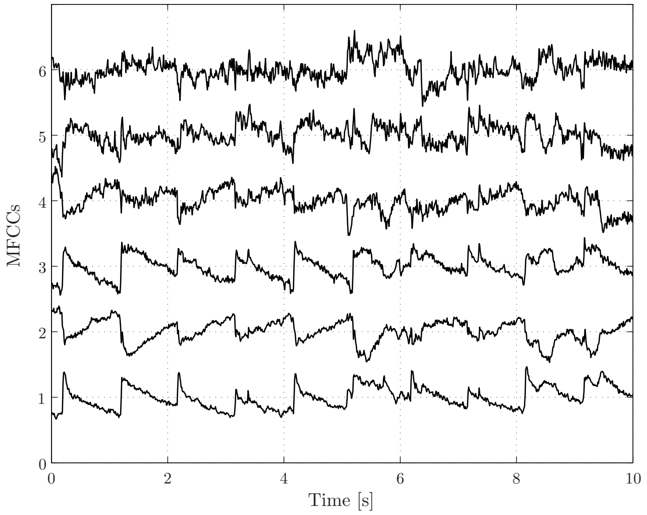


Figure 4.1: The figure shows the first six normalised MFCCs of the music piece S_1. The temporal dependencies as well as cross-dependencies among feature dimensions can be seen.

There exists quite a few statistical models which can be applied for modelling sequences of short-time features. This thesis will focus on the following models for temporal feature integration: the Gaussian model (GM), the Gaussian Mixture Model (GMM) and the multivariate autoregressive model (MAR). Of these three it is only the latter model, which learn information about the dynamics of the short-time features.

4.3.1 Gaussian Model (GM)

The mean and variance of the short-time features have been applied in various papers for capturing the 'dynamics' in a frame. This approach can be formulated more in general terms by the Gaussian model. Assume that the short-time features in a frame are distributed according to a multivariate normal distribution, then

$$\mathbf{z}_k \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{4.4}$$

for the features of that frame. This is a similar assumption as the bag-of-words in text retrieval, which means that short-time features of a frame can be randomly permuted without any change in the parameters of the model ($\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$). The parameters of this model can be used as a new feature at the integrated time-scale. Thus, the feature expressed at the larger time-scale becomes

$$\tilde{\mathbf{z}}_{\tilde{k}} = \begin{bmatrix} \hat{\boldsymbol{\mu}}_{\tilde{k}} \\ \mathrm{vech}\left(\hat{\boldsymbol{\Sigma}}_{\tilde{k}}\right) \end{bmatrix}, \tag{4.5}$$

where vech$(\cdot)$ refers to stacking only the upper triangular part (or lower) of the covariance matrix with the diagonal included. The parameters of the GM can be estimated from the short time features in the frame using e.g. a maximum likelihood approach.

As the model implies, temporal correlations in the short-time features are not modelled, while the covariation between the feature dimensions are. Using only the diagonal of the covariance matrix results in the normal mean-variance approach. In the following, MeanVar refers to the simple statistics and MeanCov refers to the full-covariance model. The spectral properties of the Gaussian model, see e.g. [27] are dominated by low-frequencies, since mean and variance calculation involves weighted sums of the short-time features over finite sized frames. Thus, the parameters of $\tilde{\mathbf{z}}_{\tilde{k}}$ for $\tilde{k} = 0, \ldots, \tilde{K} - 1$ will contain low frequency information of the short-time features. Figure 4.2 illustrates the mean value of $MFCC_0$ using a hop-size of 100 ms and a frame-size of 1000 ms. The figure clearly illustrate that the integrated feature at this timescale has temporal information corresponding to the repetitions in the music. The music piece has a 4/4 time-signature and a tempo of 60 beats per minute. The Gaussian model has been applied with great success in music genre classification together with a support vector classifier, see e.g. [93] and [100, appendix F]. Furthermore, a similar setup as the one applied in [93] won the MIREX [38] contest on music artist identification and finished second on the music genre classification task. The computational complexity for the MeanVar and MeanCov on a frame basis is presented in Table 4.1.

4.3.2 Multivariate Autoregressive Model (MAR)

Music is inherently temporal dependent, otherwise, it wouldn't be music. Like the dynamical change of the vocal tract creates words recognisable by other peo-

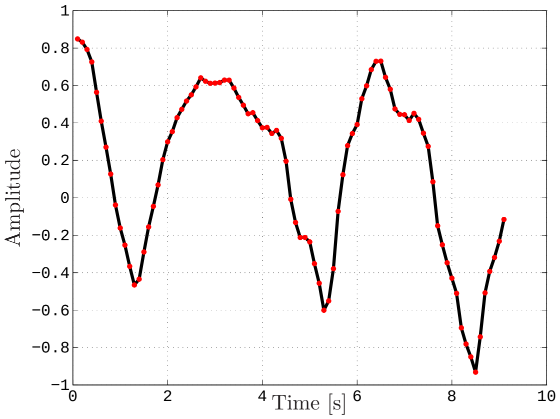


Figure 4.2: The figure shows the mean of the $MFCC_0$ over a frame-size (f_{s_z}) of 1 sec with a hop-size h_{s_z} of 100 ms. As indicated, this features temporal dynamics contain information of the repetitions of the music. The time-signature is known to be 4/4, with a tempo of approx. 60 beats per minute.

Method	Multiplications & Additions
MeanVar	$4Df_{s_z}$
MeanCov	$(D+3)Df_{s_z}$

Table 4.1: Complexity of the MeanVar and MeanCov calculation in temporal feature integration. D is the dimension of the short-time features.

ple, the dynamical changes of the spectral envelope let people recognise music styles, instruments, etc. It seems natural that models of short-time features indeed include information about the temporal structure, whether this is the local dynamics (modulations by instruments, etc.) or the longer temporal structure in music (tempo, chorus etc.). The dynamics of the MFCCs of Figure 4.1 indicated temporal correlations as well as correlations among the different feature dimensions. Authors have recognised that modelling of the temporal dynamics is important in for example, music genre classification [142, 97]. Also when modelling music instruments spectral dynamics have been found to be a relevant descriptor of timbre [72, 56]. In this connection the multivariate autoregressive model has been suggested by the authors for temporal feature integration, see [98, appendix E], [100, appendix F] and [99, appendix H].

Autocorrelation related features have been applied in various other audio mining tasks. High level temporal statistics features such as the autocorrelation function and partial autocorrelation function were applied in [105] for visualisation of music. In [82] correlation like features were extracted from a sequence of MFCC's for general audio retrieval. In [60] the autoregressive model was applied specifically on the MFCCs and was found efficient for stress detection in speech, simply by monitoring single model parameters. Also in the work by [6] the autoregressive model is used in modelling of short-time features over 1 sec frames in terms of ASR. In econometrics autoregressive models have been used with great success[4] [89] and also for Geo-science, where oscillations of a complex systems are sometimes characterised by principal oscillation patterns. This is basically the eigenmodes of a multivariate autoregressive model of first order (MAR-1 model) fitted to the observations [107][5].

In the following, the multivariate autoregressive model is defined. The theory underlying the multivariate autoregressive model is simply too large to be covered in this thesis and would be out of scope. Excellent books exists covering the theory of stochastic processes, see e.g. [89, 27]. For a stationary time series of features $\mathbf{z}_k$ the general multivariate AR model is defined by

$$\mathbf{z}_k = \sum_{p=1}^{P} \mathbf{A}_p \mathbf{z}_{k-I[p]} + \mathbf{u}_k \tag{4.6}$$

where the noise term $\mathbf{u}_k$ is assumed i.i.d. with mean $\mathbf{v}$ and finite covariance $\mathbf{C}$. The above formulation is quite general since I refers to a general set. E.g. for a model order of 3, the set could be selected as $I = \{1, 2, 3\}$ or as $I = \{2, 4, 8\}$ indicating that $\mathbf{z}_k$ is predicted from these previous state vectors. Note that the mean of the noise process $\mathbf{v}$ is related to the mean $\mathbf{m}$ of the time series by

$$\mathbf{m} = \left(\mathbf{I} - \sum_{p=1}^{P} \mathbf{A}_p\right)^{-1} \mathbf{v}. \tag{4.7}$$

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 information in the previous short-time features $\{\mathbf{z}_{k-I[1]}, \ldots, \mathbf{z}_{k-I[P]}\}$ that can be used to model $\mathbf{z}_k$. In this thesis the usual form of the multivariate AR model have been used, hence, $I = \{1, 2, \ldots, P\}$.

4.3.2.1 What can be modelled, and what is modelled?

For simplicity the diagonal multivariate autoregressive model (DAR) is investigated in more detail in the following. The DAR model is simply an AR-model of

[4] Econometrics is a combination of economical mathematic, statistics and economic theory. One of the more important tools of econometrics is time series analysis, where variables across times are monitored. Variables could e.g. be the interest rate. Assuming that the error term is an AR-model, this is also known as GARCH models which are especially useful in Econometrics.

[5] If the model is adequate, the eigenmodes of the AR-model can reveal important dynamic structure of the system.

each feature dimension independently. The DAR model was suggested for temporal feature integration in [98, appendix E] and compared with other temporal feature integration methods for music genre classification. The DAR model can be written as

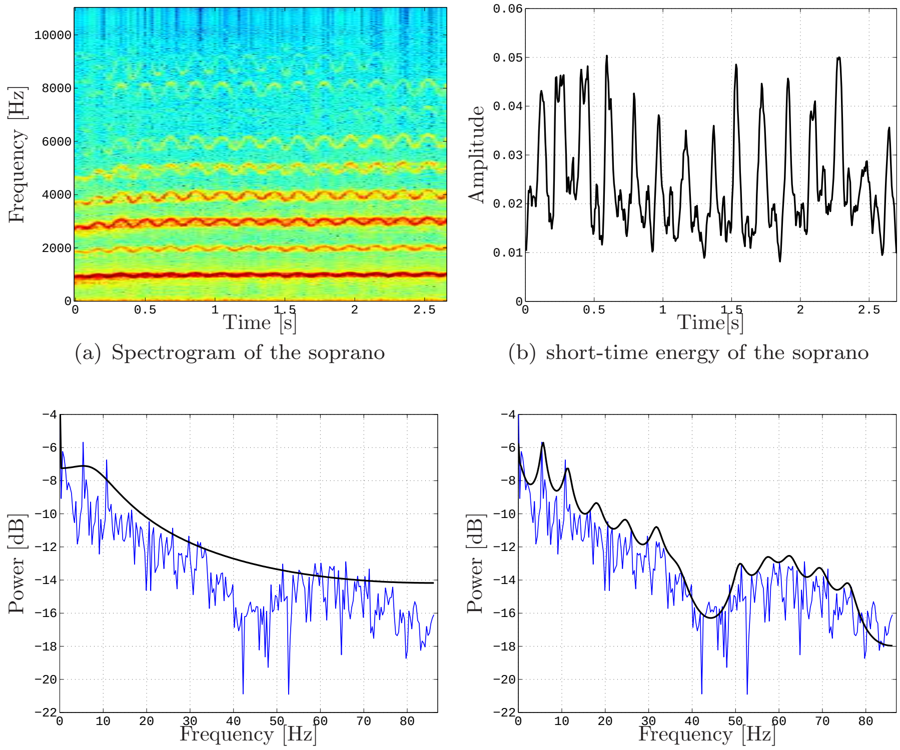
$$z[d,k] = \sum_{p=1}^{P} a_p z[d,k-p] + u[d,k], \tag{4.8}$$

where a_p for $p = 1, \ldots, P$ is the autoregressive coefficients, $u[d,k]$ is the noise term, assumed i.i.d. with finite variance and mean value v. The mean value is related with the mean of the time-series m by $m = \left(1 - \sum_{p=1}^{P} a_p z_{n-p}\right)^{-1} v$. Assuming a white noise process the parameters of the model can efficiently be calculated with a least squares approach (the covariance method [91]). Estimating the model parameters with a least squares approach there are some interesting properties of the modelled spectrum [91]:

- The power spectrum of the autoregressive model, here denoted by $\hat{P}(\omega)$ is a smoothed version of true power spectrum $P(\omega)$ of the process $z[d,k]$.
- The smoothed power spectrum is an unbiased estimator of the true spectrum. Hence, in the limit as the number of samples goes to infinity, the model power spectrum $\hat{P}(\omega) \rightarrow P(\omega)$,
- The 'global property' states that the matching between the model power spectrum and the "true" power spectrum performs uniformly over the whole frequency range, irrespective of the shape of the power spectrum. This means that all frequencies are getting equal importance, irrespective if these frequencies have low or high energy.
- Resonant structures (peaks) of the true power spectrum are better modelled than noisy parts of the signal.

To illustrate the modelling perspectives of the autoregressive model, consider the spectrogram, see Figure 4.3(a), of a music snippet by the famous American-born Greek soprano Maria Callas[6] and the corresponding short-time energy feature (STE) using a hop-size of 5.8 ms and frame-size of 23.2 ms, see Figure 4.3(b). The spectrogram shows presence of a vibrato, with a modulation frequency of approximately 6.4 Hz. Figure 4.3(c) and 4.3(d) show the corresponding periodogram of the STE feature over the 2.7 sec as well as the AR estimate of the power spectrum for a model order of 3 and 24, respectively. Even with a model order of 3, some of the vibrato signal is modelled, although, not very detailed. Increasing the model order, the level of detail of the power spectrum increases as expected.

[6] Collected from `www.findsounds.com`



(a) Spectrogram of the soprano

(b) short-time energy of the soprano

(c) DAR with a model order of 3

(d) DAR with a model order of 24

Figure 4.3: Figure (a) shows the spectrogram of 2.7s of a voiced part of the soprano Maria Callas. The vibrato is clearly observed, and has a modulation frequency around 6.4Hz. Figure (b) shows the corresponding STE of the music snippet. A hop-size of 5.8ms and frame-size of 23.2ms were used. Figure (c) shows the periodogram of the STE (blue) and the power spectrum estimation by the DAR model is shown in black for a model order of 3. Figure (d) shows a model order of 24.

For the MAR model, also cross-correlations between the short-time feature dimensions are modelled. Figure 4.4 and 4.5 shows the autocorrelation and cross-correlations of the first 5 MFCCs of the music snippet S_2 as a function of lag time for the measured and predicted correlation structure (using a MAR model of order 3) from frames of 1.2 sec and 30 sec, respectively. The 1.2 sec frame was selected randomly from the 30 sec music snippet S_2. The figures show that the local correlations in the short-time features are well approximated by the MAR model. Furthermore, the trend of some of the cross-correlations are captured nicely, see e.g. $MFCC_0$ with $MFCC_3$ in Figure 4.4. Increasing the frame-size to 30 sec, see Figure 4.5, the correlations becomes smoother and

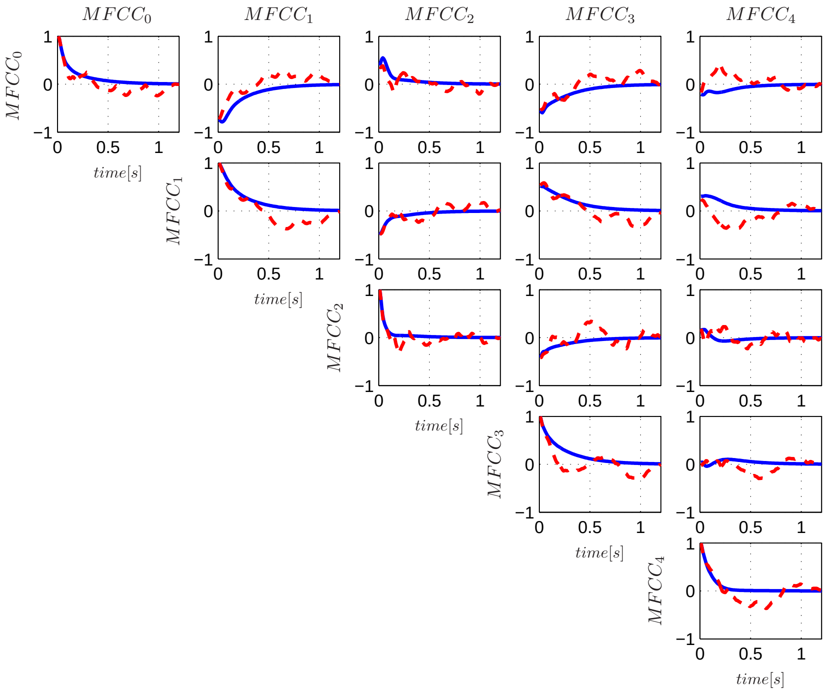


Figure 4.4: The different figures illustrate the estimated and “measured” correlations and cross-correlations between the first 5 MFCCs of the music snippet S_2 by Roxette. The estimated correlations is by a MAR model of order 3. This model order has been found optimal in two datasets for music genre classification, see Chapter 6. The correlations have been calculated for a frame-size ($f_{s_{\tilde{z}}}$) of 1.2 sec. The frame was randomly picked from the music snippet. It seems as if local dynamics of the short-time features are well modelled with a 3 order AR model.

cross-correlations less pronounced[7].

4.3.2.2 Estimation of parameters

There exists quite a few approaches for estimating the parameters of a multivariate autoregressive model. The parameters can be estimated from the time or frequency domain, see e.g. [107, 89, 91]. In [99, appendix H] we used a normal least squares approach for estimating the parameters. In [100, appendix F] we used the ARFIT package [107] that implements a regularised least squares method.

[7] This can be motivated from the fact that the DCT of the MFCCs are in fact decorrelating the feature dimensions.

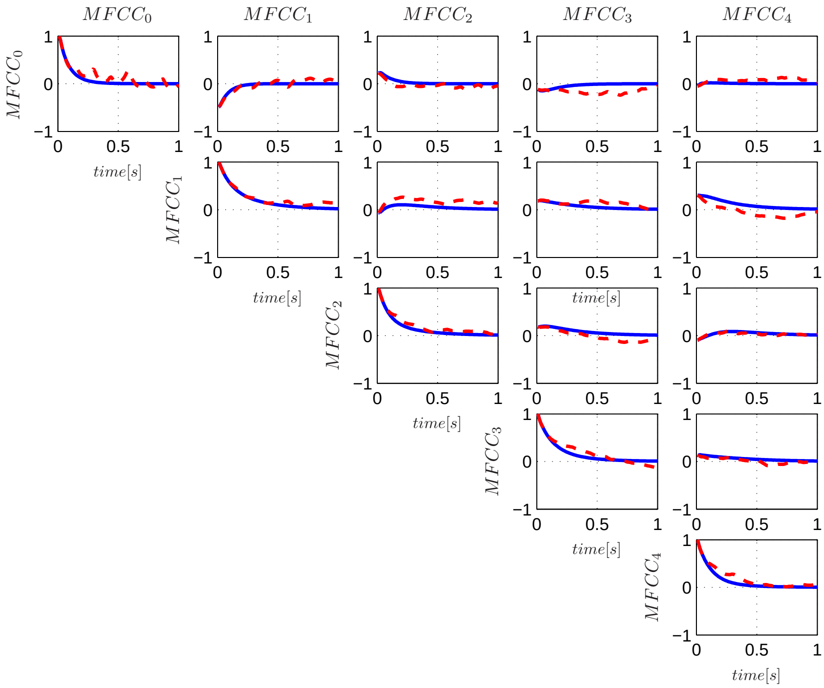


Figure 4.5: The different figures illustrate the estimated and "measured" correlations and cross-correlations between the first 5 MFCCs of the music snippet S_2 by Roxette. The estimated correlations is by a MAR model of order 3. This model order has been found optimal in two datasets for music genre classification, see Chapter 6. The correlations have been calculated from a frame size ($f_{s_{\tilde{z}}}$) of 30 sec, which is corresponding to the length of the song. Only a lag of 1 sec has been illustrated, since the correlations and cross-correlations are decaying to zero.

The parameters, which are estimated from the model are the AR-matrices $\hat{\mathbf{A}}_1, \ldots, \hat{\mathbf{A}}_P$, the intercept term $\hat{\mathbf{v}}$, and the noise covariance $\hat{\mathbf{C}}$. The temporal feature integrated vector of frame $\tilde{k}$ then becomes

$$\tilde{\mathbf{z}}_{\tilde{k}} = \begin{bmatrix} \hat{\mathbf{v}}_{\tilde{k}} \\ \mathrm{vec}\left(\hat{\mathbf{B}}_{\tilde{k}}\right) \\ \mathrm{vech}\left(\hat{\mathbf{C}}_{\tilde{k}}\right) \end{bmatrix}, \tag{4.9}$$

where $\hat{\mathbf{B}}_{\tilde{k}} = \begin{bmatrix} \hat{\mathbf{A}}_{1,\tilde{k}} & \hat{\mathbf{A}}_{2,\tilde{k}} & \ldots & \hat{\mathbf{A}}_{P,\tilde{k}} \end{bmatrix}$ and $\tilde{\mathbf{z}}_{\tilde{k}}$ has a dimension of $(P + 1/2)D^2 + (3/2)D$. For the DAR model, only the diagonals of the $\hat{\mathbf{A}}_p$ and $\hat{\mathbf{C}}$ are used. This amounts to a total of $(P + 2)D$ parameters.

The computational complexity of the DAR and MAR features across a frame f_{s_z}

Method	Multiplications & Additions
DAR	$\frac{D}{3}\left(P+1\right)^3 + \left(\left(P+6\right)\left(P+1\right)+3\right) D f_{s_z}$
MAR	$\frac{1}{3}\left(PD+1\right)^3 + \big(\left(P+4+\frac{2}{D}\right)+\left(D+2\right)\big) D f_{s_z}$

Table 4.2: Complexity of the DAR and MAR approach to temporal feature integration.

in terms of multiplications and additions are shown in Table 4.2. The complexity has been obtained using standard values for matrix inverses.

4.3.3 Other statistical models

The two models considered (GM and MAR) are models with no latent variables. The parameter estimates of these models can be used directly as features at larger time-scales. More general models such as the Gaussian mixture model, hidden Markov model and other linear Gaussian models, see e.g. [120], does not have unique solutions, which renders direct comparison of parameters superfluous. Instead of a direct comparison of parameters one can measure a distance between the model density functions. Researchers in MIR have been evaluating the similarity between songs, simply by evaluating the log-likelihood of a given song with the short-time features of the other songs in the dataset. This method, although not explicitly stated, is in fact related to the estimated KL-divergence between density functions, see e.g. [15]. In [9], the MFCCs of each song are modelled by a GMM. The authors apply the models for accessing the similarity of songs in terms of their timbre for music retrieval. In [47], the authors compare the modelling power of each songs MFCCs using a HMM and a GMM. Using approximately the same number of model parameters the author finds that the HMM is a better model of the MFCCs in terms of log-likelihood, however, there is no direct performance increase observed when comparing the models in a music genre classification task. Detailed information such as confusion matrices are not provided, which makes it difficult to conclude if the models are misclassifying the same examples. In [93], the author considers a symmetric KL-divergence kernel applied with a support vector classifier for music genre classification. Different kernel functions for measuring similarity between probability distributions will be considered in more detail in Chapter 5.

4.3.4 Model order selection

For the MAR model (or the GMM) a model order needs to be selected. The model order which best models the data, and still generalises, can be found using a model order criteria such as the Bayesian Information Criterion (BIC) or Akaike Information Criterion (AIC), see e.g. [107, 136]. For the music organisation tasks considered in Chapter 6, we are interested in finding a single optimal model order across data which has the best generalisation performance. Having different model orders from frame to frame would require specialised classifiers. Classifiers, which rely on divergence based similarity measures are able to compare songs modelled with different model orders. In this thesis, the optimal model order have been selected to maximise the cross-validation test accuracy, which is discussed further in Chapter 6.

4.4 Filterbank coefficients (FC)

The method described in this section was proposed in [97] for including temporal information in the integrated short-time features at larger time-scales for general audio classification (music genre was also investigated). The idea is to extract a summary of the periodogram of each short-time feature dimension, independently. The summary consists of 4 pre-defined frequency bands, in which the periodogram is summarised. The temporal feature integration approach simply amounts to

$$\tilde{\mathbf{z}}_{\tilde{k}} = \mathrm{vec}\left(\mathbf{P}_{\tilde{k}}\mathbf{W}\right), \tag{4.10}$$

where the matrix $\mathbf{P}_{\tilde{k}}$ contains the periodogram of each short-time feature dimension arranged row-wise. The dimension of $\mathbf{P}_{\tilde{k}}$ is $D \times N$, where $N = f_{s_z}/2$ when f_{s_z} is even and $(f_{s_z} - 1)/2$ for odd values. The filter matrix $\mathbf{W}$ of dimension $N \times 4$, summarises the spectral information of selected frequency bands. In [97], the frequency bands investigated were: 1) 0 Hz (Corresponding to the DC-value), 2) $1-2$ Hz, which is on the order of music beat rates, 3) $3-15$ Hz modulation energy (on the order of speech syllabic rates, or vibrato) and $20 - 43$ Hz is a lower range of modulations corresponding to perceptual roughness. The dimension of $\tilde{\mathbf{z}}_{\tilde{k}}$ then becomes $4D$.

The DAR and FC model both involve a modelling of the power spectrum (periodogram for the latter approach). Whereas a smooth estimate of the power spectrum is optimised for the DAR model, a periodogram is summarised in predefined frequency bands for the FC model. The method could be generalised to handle cross-spectras, which would involve Fourier transformations of the cross-correlation spectras.

The selection of four filterbanks in the pre-specified frequency range is rather arbitrary. To obtain best possible performance in some music organisation task, the number of filters as well as their spectral shape could be extracted from the data.

Finding an optimal filter can be approached either supervised or unsupervised.

In principle, using cross-validation and a gradient based approach optimality in sense of generalisation error can be found. However, the method would be expensive and not very practical. Other methods such as canonical correlation analysis (CCA) or partial least squares (PLS) applied in a regression like manner[8] one would obtain the d leading discriminative eigenvectors. These eigenvectors could be interpreted as the filterbanks. Since the methods are applied to the power spectrum, one would need to impose a positivity constraint on each dimension of the eigenvectors. This method has not been considered in greater detail in this thesis.
The spectral approach, is very intuitive since each of the feature dimensions relate to a physical quantity of the short-time features, whether this is a modulation corresponding to beat, vibrato or some higher order modulations. The complexity of the spectral method (assuming a fixed sized filter bank $\mathbf{W}$) is governed by the FFT operation, hence,

$$(4\log_2(f_{s_z})+3)Df_{s_z}. \tag{4.11}$$

The next section introduces two 'perceptual' temporal feature integration methods, the beat histogram and the beat spectrogram.

4.5 Extraction of tempo and beat

Without any detailed level of music experience most people are able to tap according to the beat of the music. Tapping on the beat feels natural since it requires much more attention to tap off the beat. This human phase-locking system illustrate that the phase is important at low frequencies. The problem of finding the phase of the beat has been considered in more detail in [53, 128]. The normal measure of tempo in music is beats per minute (BPM). For most music it is suffices to consider tempi in the range $40-240bpm$, see e.g. [128]. The tempo of music can be important in categorising music after slow or fast music titles. Also when searching for music, a search phrase like: “slow rock”, should retrieve songs from the music genre rock having a slow tempo. Furthermore, with automatic phase locking systems one can devise automatic DJs for mixing playlists, see e.g. [49, 26].

Music tempo extraction has been investigated by several researchers, see e.g. [54, 142, 53, 50, 128]. It is common practice to distinguish between notated and perceptual tempo[9]. In [128], the author discusses the terminology of a strong

[8] Regression on the labels.

[9] This leaves tempo of music a subjective measure. However, the degree of subjectivity depend on the beat strength.

beat. A strong beat refers to large consensus among humans on the perceived tempo, whereas a weak beat results in little or no consensus among humans.

For most music, the instruments are played on and around the beat [130]. This have been exploited in [126] for music genre classification. Temporal feature integration by simple statistics were applied to short-time features in frames centred on the beat, hereby insuring that the short-time features include a mix of the instruments being played. There were, however, no apparent accuracy increase when considering this approach rather than using fixed frames of 1 sec.

Two methods have been investigated in this thesis for music genre classification, one suggested by [142], denoted the beat histogram and another approach suggested by [50, 49] denoted the beat spectrum. Both of these methods, can be considered as being temporal feature integration methods, since short-time features are extracted after which temporal structure is extracted and used as a new feature at a larger time-scale.

4.5.1 Beat Histogram (BH)

The beat histogram was suggested by Tzanetakis in [142] as a rhythmic content feature. The rhythmic feature consists of several processing steps. The complete system for extracting the beat histogram is illustrated in Figure 4.6. The author applies a discrete wavelet transform to the audio signal, which to some extent is similar to decomposing the audio into octave spaced frequency bands. In the implementation, octave space filterbanks have been applied with center frequencies of $62.5, 125, 250, 500, 1000, 2000, 4000, 8000$ Hz. To motivate the octave spaced filterbanks the author of [128] found that his algorithm was less sensitive to different filterbank implementations.

The following listing supports figure 4.6:

- Octave spaced filterbank with center frequencies at 62.5, 125, 250, 500, 1000, 2000, 4000, 8000 Hz
- Full wave rectification, $y_1[n] = |y[n]|$
- Low pass filtering using an IIR[10] filter $y_2[n] = (1 - \alpha)y_1[n] + \alpha y_2[n-1]$. $\alpha = 0.99$ gives a 3dB cutoff normalised frequency of $\sim \omega_c = 0.0032\pi$
- Down-sampling, $z[d, k] = y_2[dn]$, where $d = 16$ is the chosen down-sampling factor[11]

[10] Infinite Impulse Response

[11] Actually a larger down-sampling factor could have been selected.

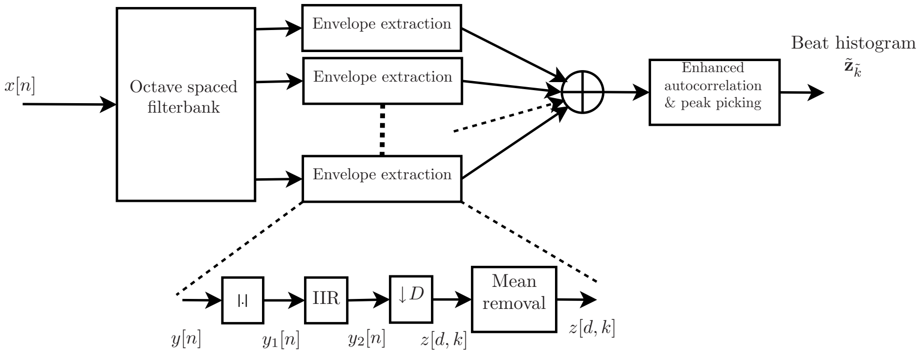


Figure 4.6: The beat histogram similar to the one proposed in [142].

- Mean removal, calculated from the samples in the frame
- Adding the envelopes from each filterbank and applying an enhanced autocorrelation [140] method for detecting periodicities. The three most dominant peaks are added to the beat-histogram.

In [142] the beat histogram was applied to frame-sizes of 3 sec, which allows for a changing tempo through the music piece. The output of the beat histogram is a vector of dimension 6 where the first five dimensions summarise the amount of beats in the ranges $30-55$, $55-80$, $80-105$, $105-160$, $160-250$ bpm and the final feature is correlated with the beat-strength of the corresponding frame.

4.5.2 Beat Spectrum (BS)

The beat spectrum was proposed in [50] for rhythm analysis and tempo extraction. In [49] the authors apply the beat spectrum for arrangement of songs after rhythmic similarity. The beat spectrum is based on acoustic self similarity versus lag-time of spectral short-time features. The beat spectrogram is formed to address rhythmic changes over time. The below listing shows the beat spectrum extraction

- Extract spectral short-time features from the raw audio samples (in our implementation, the MFCCs were used).
- From the short-time features $\mathbf{z}_k$, for $k = 0, \ldots, K-1$, calculate the self

similarity matrix between short-time features $S(\mathbf{z}_i, \mathbf{z}_j)$. Two simple distance measures was suggested

1. Euclidean : $S(\mathbf{z}_i, \mathbf{z}_j) = \mathbf{z}_i^T \mathbf{z}_j$, and
2. Cosine : $S(\mathbf{z}_i, \mathbf{z}_j) = \frac{\mathbf{z}_i^T \mathbf{z}_j}{||\mathbf{z}_i||_2 ||\mathbf{z}_j||_2}$.

The latter ensures that windows with low energy, can still yield a large similarity score.

- From the similarity matrix $\mathbf{S}$, the beat spectrum is calculated as

$$B[l] \approx \frac{1}{|\mathcal{R}|} \sum_{k \in \mathcal{R}} S(k, k+l). \tag{4.12}$$

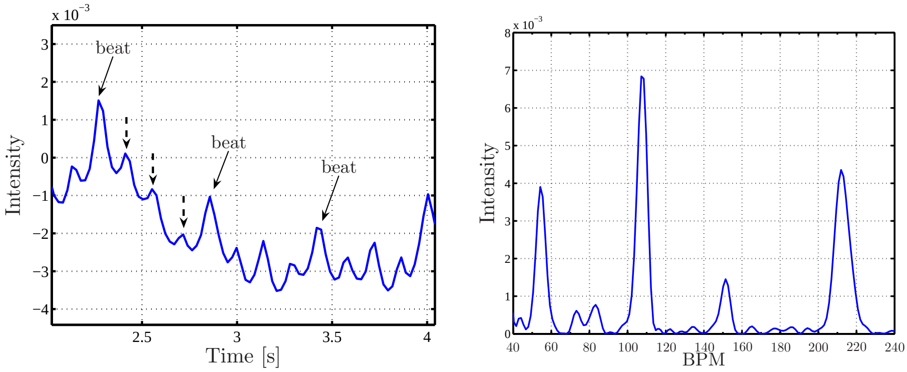
When $l = 0$ the beat spectrum is simply the sum along the main diagonal of $\mathbf{S}$, and when $l = 1$, the sum along the first super-diagonal. $|\mathcal{R}|$ is the cardinality of the number of elements in the l'th diagonal.

The beat spectrum, denoted by $B[l]$ for $l = 0, \ldots, K-1$, expresses the self similarity as a function of lag l. Peaks in the beat spectrum, corresponds to repetitions in the music. The cosine similarity measure was selected due to its robustness. The beat spectrum of the music snippet S_2 has been shown in Figure 4.7(a). The annotated beat have been marked by 'beat'-arrows, while the sub-beats marked in-between with dotted arrows indicate a 4/4 time signature. To extract explicit information about the tempo a DFT is applied to the beat spectrum. Only frequencies in the range $40 - 240$bpm is retained for further processing [100, appendix F]. The DFT of the beat spectrum is illustrated in Figure 4.7(b).

4.6 Stationarity and frame-size selection

Using a block based approach to temporal feature integration, an optimal hop- and frame-size have to be determined. With the presented statistical models, stationarity is indirectly imposed, but not necessarily fulfilled. The frame-size is usually selected to maximise the generalisation performance of the system after which a hop-size can be determined to minimise information loss in terms of aliasing as discussed in Chapter 3.

A simple yet expensive approach for determining an optimal frame-size for a given setup, would be to use cross-validation [15]. Figure 4.8 illustrates the mean classification test accuracy obtained from 10-fold cross-validation on a



(a) Beat spectrum of the music piece S_2

(b) Discrete Fourier transform of the beat spectrum

Figure 4.7: (a) A zoom on the beat spectrum for the music snippet "Fading like a flower" by the music band Roxette (S_2). The 4/4 time signature can be read of the peaks in-between the perceptual beat arrows. (b) illustrates the corresponding DFT of the beat spectrum of (a) in the range $40 - 240$ BPM.

music genre classification task[12] using MFCCs as short-time features and a MAR model of order 3 for temporal feature integration. The hop-size is fixed at 200 ms and the frame-size is varied between 200 ms and 4000 ms. The mean classification test accuracy of two classifiers, the GLM (Generalised Linear Model) and LM (Linear Model) are illustrated. The classification accuracy is calculated at 30 sec from a late fusion technique denoted as the sum-rule, which is related with majority voting.

In [142], the importance of the texture window size (frame-size) was investigated for music genre classification. The author found that a texture window of approximately 1 sec was optimal w.r.t. the classification test accuracy on a 10 genre dataset. In [44], new features were created using simple statistics (mean-variance) over frames of 1 min short-time features. These new features were applied for segmentation and classification of long duration recordings of personal audio[13]. It is believed that the frame-size is selected partly from a computational perspective and partly from the fact that adequate detection of environments can be performed at this time-scale. Lately, a few authors [146, 14] have looked at methods for working with non-fixed frame-sizes, hence selecting frame-sizes from the stationarity of the short-time features. The author of [146] suggests to use an onset detection algorithm for determining optimal segment

[12] An 11 genre dataset, which will be discussed in greater detail in chapter 6.

[13] Audio was recorded from various environments, e.g. library, campus, street, barber, meeting, subway, etc. In total 62 hours of annotated data was investigated. Several spectral short time features was investigated for this analysis.

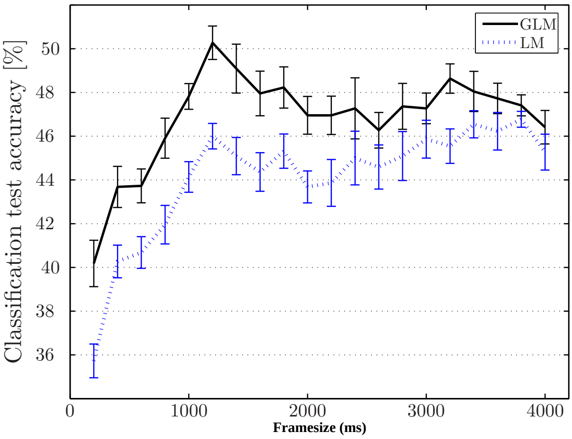


Figure 4.8: The mean classification test accuracy from 10-fold cross-validation on an 11 music genre classification task, plotted for different frame-sizes of the MAR features. The error bars are ± the standard deviation on the mean test accuracy. The results are shown for the LM and GLM classifier. The hop-size was set to 200 ms. The investigated dataset will be discussed in more detail in chapter 6.

boundaries. Thus, by allowing a variable frame-size, important parts of the music can be modelled better, whether this is the attack of the instruments or the sustained part. The authors reported a small increase in classification accuracy in a music genre classification setup when comparing to the block based approach.

4.7 Discussion

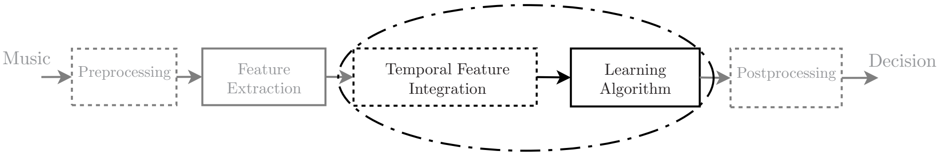
The general idea of temporal feature integration was introduced and different approaches to temporal feature integration were discussed. By considering a sequence of short-time features as a time series, it was illustrated that the temporal characteristics of these short time features can carry important structural information such at beat, vibrato and other temporal structure, which has no perceptual interpretation. The general multivariate autoregressive model (MAR) was suggested for modelling sequences of short-time features. The effect of modelling the temporal information in the short-time features will be elaborated on in Chapter 6, where the different temporal feature integration methods will be compared in a music genre classification task.

The computationally complexity of the different temporal feature integration methods was calculated from typical values of inversion of matrices and fast Fourier transforms and it was noted that the MAR and DAR models are computationally more expensive than the FC, MeanCov and MeanVar approaches.

The next chapter introduces the notion of kernel aided temporal feature integration, where statistical models with a latent structure can be used as input to kernel functions. The generated kernel matrix can then applied in combination with a support vector classifier for e.g. music genre classification.

Chapter 5

Kernels for aiding temporal feature integration



The previous chapter introduced temporal feature integration, which is a method for integrating short-time features into a feature vector at a larger time-scale. Additionally, advanced temporal feature integration models were mentioned, such as Gaussian mixture models and hidden Markov models. These latent generative models cannot be compared directly, since their parameters are not uniquely defined.

In this chapter, kernels which aid temporal feature integration or actually perform temporal feature integration are presented.
The general idea of a kernel method is to embed the input data, which could be the short-time features or the model parameters from the temporal feature integration step, into a vector space denoted as the *feature space*[1] [132]. The learning algorithms, which are devised for learning in the feature space, are implemented in such a way that the coordinates of the embedded points in feature space are not needed, but only their pairwise inner products. *Kernel functions* indirectly calculates the inner product in the feature space. This is frequently referred to as the "kernel trick" and enables efficient calculations

[1] Also denoted as the Reproducing Kernel Hilbert Space.

of high-dimensional embeddings. Algorithms devised in the feature space are normally linear, however, when transformed back to input space they become non-linear. This can lead to versatile solutions, which can detect and effectively use non-linear relations in the input space.

Two of the investigated kernel functions, the product probability kernel (PPK) [70], and the symmetric KL-divergence kernel (KL) [102], take density functions as inputs. Closed form solutions of the Gaussian model, multivariate AR and Gaussian mixture model in a PPK can be obtained analytically. For the KL kernel only the MAR and GM model can be derived analytically.

5.1 Kernel methods

The indirect embedding into a potentially infinite dimensional feature space using kernel functions allow well known dimensionality reduction methods such as principal component analysis (PCA), canonical correlation analysis (CCA), partial least squares (PLS) to have efficient implementations in feature space. In that manner, possible non-linear relationships in input data can be learned, see e.g. [132].

Let the input space be denoted by $\mathcal{X} \subseteq \mathbb{R}^D$, and the feature space by $\mathcal{F} \subseteq \mathbb{R}^L$, where, in principle, L can be infinite dimensional. Then,

> The kernel of $\mathcal{F}$ is a function κ that for all $\mathbf{x}, \mathbf{z} \in \mathcal{X}$ satisfies
>
> $$\kappa(\mathbf{x}, \mathbf{z}) = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{z}) \rangle, \tag{5.1}$$
>
> where $\boldsymbol{\phi}$ is a mapping from $\mathcal{X}$ to feature space $\mathcal{F}$, hence
>
> $$\boldsymbol{\phi} : \mathbf{x} \rightarrow \boldsymbol{\phi}(\mathbf{x}) \in \mathcal{F}, \tag{5.2}$$
>
> and the operation $\langle \cdot, \cdot \rangle$ denotes the inner product.

For a function to be a valid kernel of some feature space, it must be symmetric and fulfil Mercer's theorem. By combining simple kernel functions, even more complex kernels can be constructed. In [132], it is shown that kernel functions fulfil various closure properties. For example a kernel function is closed under addition and multiplication[2].

[2] Thus, both operations shown in Equation 5.3 produce a new valid kernel function,

$$\kappa(\mathbf{z}, \mathbf{x}) = \kappa_1(\mathbf{z}, \mathbf{x}) + \kappa_2(\mathbf{z}, \mathbf{x}) \quad \text{and/or} \quad \kappa(\mathbf{z}, \mathbf{x}) = \kappa_1(\mathbf{z}, \mathbf{x}) \cdot \kappa_2(\mathbf{z}, \mathbf{x}). \tag{5.3}$$

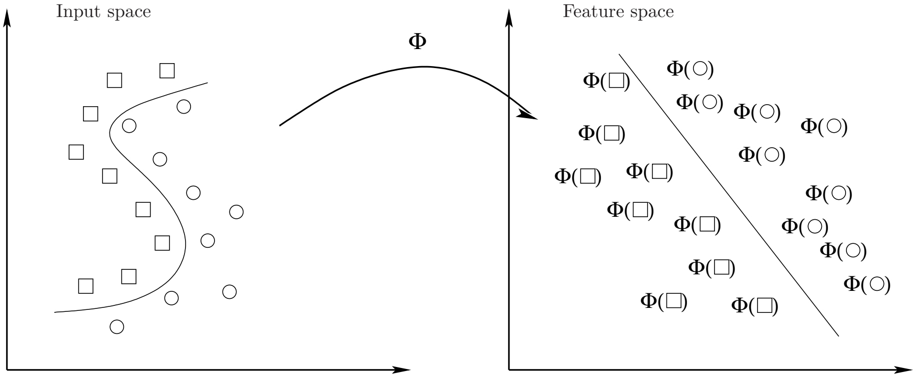


Figure 5.1: The function Φ embeds the input data into feature space. The linear decision boundary learned in feature space transforms to a non-linear decision boundary in input space.

The support vector classifier is a robust discriminative classifier, which typically shows "state of the art" performance in many areas of machine learning. In particular it has shown to be useful for the music genre classification task, see e.g. [93, 14, 126]. The support vector classifier (SVC) [144, 132] exploits the implicit embedding of data into feature space through the kernel function for creating non-linear decision boundaries. The SVC will be discussed shortly in Chapter 6.

Figure 5.1, shows a hypothetical classification example consisting of a two class problem with a non-linear decision boundary. The points in input space are embedded into the feature space using the embedding function $\boldsymbol{\phi}(\mathbf{x})$ where $\mathbf{x}$ denotes a point in the input space and $\boldsymbol{\phi}(\mathbf{x})$ its corresponding projection. The support vector classifier finds the linear discriminating boundary in feature space with maximum margin between the two classes using only the input data, the kernel function and the labels from the dataset. As indicated, the kernel function is a crucial point in any kernel based learning algorithm, hence, special emphasis must be placed on selecting an appropriate kernel which reflects our underlying beliefs of the input space. A kernel function such as the Gaussian kernel embeds the data into a potentially infinite dimensional feature space, which favours the expressive power of the classifier.

5.2 Kernels for music

As noted in the previous section, the kernel function is calculated between vectors of the input space, from which some similarity measure is obtained[3]. For the music organisation task, one is typically faced with sets of short-time features.
The temporal feature integration technique discussed in the previous chapter was applied to create a single new feature at a larger time-scale. In the limit, a single vector could represent an entire song. Two kernel-based approaches can be used for comparing sets of features: either creating a kernel matrix for each feature in the set (similar to the beat spectrum in Chapter 4) or to create a kernel function, which produces a single number representing the similarity between the sets of features. The latter kernel has been denoted the high-level kernel [32]. Examples of high-level kernels will be presented in the following subsections.

Consider the short-time features of two music snippets, denoted here as $\mathbf{Z} = \left[\mathbf{z}_1, \ldots, \mathbf{z}_{f_{s_z}}\right]$ and $\mathbf{Z}' = \left[\mathbf{z}'_1, \ldots, \mathbf{z}'_{f'_{s_z}}\right]$, then a single score value between the music pieces can be calculated through a high-level kernel as $\kappa(\mathbf{Z}', \mathbf{Z})$.

5.2.1 Previous work

In the last couple of years, kernel methods, especially in connection with the support vector classifier, have achieved increasing attention from researchers in MIR. In [104], the authors investigated the usage of a Fisher kernel [69] for web audio classification. The Fisher kernel is constructed by modelling the short-time features of each class with a generative model[4]. After that, the generative model is used to map short-time features of variable length into a linear space of fixed dimension.
In [57] the task of audio classification and retrieval was approached using the Gaussian kernel

$$\kappa(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{||\mathbf{z} - \mathbf{x}||_2}{2\sigma^2}\right), \tag{5.4}$$

where σ controls the width of the kernel. Temporal feature integration is performed on the short-time features[5] using mean and variance to obtain a single feature for each audio snippet. The audio database presented in [148] was investigated, and the authors reported an improved performance on both the retrieval

[3] The cosine measure applied for tempo extraction in Chapter 4 is an example of a valid kernel.

[4] A GMM was applied.

[5] Timbre based features + pitch information

and classification tasks. At the 6th International Symposium on Music Information Retrieval (ISMIR, 2005) a few authors suggested high-level kernels. The symmetric KL-divergence kernel was investigated by [93] for music genre classification in combination with a support vector classifier. Each song in the dataset was first modelled with a generative model[6] and then a kernel was evaluated to measure the distance between the models of different songs. Although a closed form solution exist for the Gaussian model, sampling techniques are required to calculate a distance between e.g. Gaussian mixture models. The combination of using a Gaussian model on each song and measuring the distance using the symmetric KL-divergence kernel won the (MIREX, 2005) [38] 'artist identification' contest and finished second in the 'music genre classification' contest.

A different approach to music genre classification was outlined in [83] which considered an LZ78-based string kernel. A finite size vocabulary was learned using a vector quantiser from sequences of short-time features. From the vocabulary each song was represented by a finite alphabet. Using an LZ78 algorithm, see e.g. [123], a similarity between songs of varying length was obtained. A comparison of the LZ78-based string kernel with the approach of [142] on a 10 genre dataset investigated by the latter author showed a slight improvement when using the LZ78 string kernel, although differences were not statistically significant. The vocabulary was build from the MFCCs using a frame length of 25ms. Better results might have been obtained by generating the codebook from feature vectors of a larger time-scale as proposed in [2].

In the same line, a string kernel was investigated in [124] to identify famous concert pianists from their playing style. The so-called 'performance worm' was used to extract relevant information of the movements. The performance worm is an animation of the tempo and loudness as a function of time [37]. The tempo is smoothened such that local changes in timing do not change the tempo significantly. A performance alphabet of the recorded songs are generated by cutting the trajectories into short segments of fixed length (2 beats) and clustering the segments into groups of similar patterns. A given performance can then be transcribed in terms of this performance alphabet and be compared by using string matching techniques.

5.3 High-level kernels

Three high-level kernels for sets of short-time features are explained in this section. Two of the methods, the convolutive kernel and the product probability kernel have been suggested by the author for music genre classification in [100, appendix F].

[6] Both a Gaussian mixture model (GMM) and Gaussian model (GM) were considered.

5.3.1 Convolution kernel

The convolution kernel [62] handles all kinds of discrete structures such as strings, trees and graphs. In this work, the convolution kernel has been applied to measure the distance between sets of short-time features[7]. The convolution kernel between two songs represented by their short-time features $\mathbf{Z}$ and $\mathbf{Z}'$ of finite size is given by

$$\kappa(\mathbf{Z}, \mathbf{Z}') = \frac{1}{f_{s_z}^2} \sum_{i,j=0}^{f_{s_z}-1} \kappa_I(\mathbf{z}_i, \mathbf{z}'_j), \tag{5.5}$$

where $\kappa_I(\mathbf{z}, \mathbf{z}')$ must be a valid kernel and f_{s_z} represents a frame of short-time features. In the above formulation it is assumed that the songs are of same length, however, this is not a requirement. The theory covering R-convolution kernels is provided in [62]. The convolution kernel performs an averaging in feature space since

$$\begin{aligned} \kappa(\mathbf{Z}, \mathbf{Z}') &= \sum_i \sum_j \kappa_I(\mathbf{z}_i, \mathbf{z}'_j) \\ &= \sum_i \sum_j \langle \boldsymbol{\phi}_I(\mathbf{z}_i), \boldsymbol{\phi}_I(\mathbf{z}'_j) \rangle \\ &= \left\langle \sum_i \boldsymbol{\phi}_I(\mathbf{z}_i), \sum_j \boldsymbol{\phi}_I(\mathbf{z}'_j) \right\rangle \\ &= \langle \boldsymbol{\phi}(\mathbf{Z}), \boldsymbol{\phi}(\mathbf{Z}') \rangle \,. \end{aligned} \tag{5.6}$$

Hence, the convolution kernel amounts to calculate an inner product between the smoothed embedding functions $\boldsymbol{\phi}(\mathbf{Z}) = \sum_i \boldsymbol{\phi}(\mathbf{z}_i)$ and $\boldsymbol{\phi}(\mathbf{Z}') = \sum_j \boldsymbol{\phi}(\mathbf{z}'_j)$. The kernel matrix can be shown to be positive semidefinite, since

$$\begin{aligned} \sum_{j=0,i=0}^{N-1} \alpha_i \alpha_j \kappa(\mathbf{Z}^i, \mathbf{Z}^j) &= \sum_{j=0,i=0}^{N-1} \alpha_i \alpha_j \langle \boldsymbol{\phi}(\mathbf{Z}^i), \boldsymbol{\phi}(\mathbf{Z}^j) \rangle \\ &= \left\langle \sum_{i=0}^{N-1} \alpha_i \boldsymbol{\phi}(\mathbf{Z}^i), \sum_{j=0}^{N-1} \alpha_j \boldsymbol{\phi}(\mathbf{Z}^j) \right\rangle \\ &= \left\| \sum_{i=0}^{N-1} \alpha_i \boldsymbol{\phi}(\mathbf{Z}^i) \right\|_2^2 \geq 0, \end{aligned} \tag{5.7}$$

where $\alpha_i \in \mathbb{R}$ for $i = 0, \ldots, N-1$.

[7] It is also possible to include an intermediate step of temporal feature integration before applying the convolutive kernel.

It is noted, that for kernels where the feature space can be explained explicitly by $\boldsymbol{\phi}(\mathbf{z})$, the computational complexity of the convolution kernel amounts to $\mathcal{O}(f_{s_z})$. For the linear kernel this simply amounts to calculate the mean value across the short-time features. When the inner products is evaluated through their kernel functions an evaluation between two sets $(\mathbf{Z}(\mathbf{Z}'))$ amounts to $\mathcal{O}(f_{s_z}^2)$ kernel evaluations. It should be noted that a normalisation in feature space of the convolution kernel is proportional to the cosine Euclidean distance between the feature space vectors.

5.3.2 Product Probability Kernel

The product probability kernel (PPK) was introduced by [70] as a method for handling different type of sets[8]. The product probability kernel function is one out of the many kernels proposed in the literature for handling distances between sets of data. There are, however, some practical aspects of this kernel, which make it especially useful. In the following definition of the PPK, $\boldsymbol{\theta}$ refers to the parameters of the statistical model applied. Hence, for a Gaussian model, this would amount to the mean and covariance matrix. From [70]

> Let $p(\mathbf{z}|\boldsymbol{\theta})$ and $p(\mathbf{z}|\boldsymbol{\theta}')$ be probability distributions on a space Ω and ρ be a positive constant. Assume that $p(\mathbf{z}|\boldsymbol{\theta})^\rho, p(\mathbf{z}|\boldsymbol{\theta}')^\rho \in L_2(\Omega)$, i.e. that $\int_\Omega p(\mathbf{z}|\boldsymbol{\theta})^{2\rho} d\mathbf{z}$ and $\int_\Omega p(\mathbf{z}|\boldsymbol{\theta}')^{2\rho} d\mathbf{z}$ are well defined (hence, not infinite), then the product probability kernel between the distributions $p(\mathbf{z}|\boldsymbol{\theta})$ and $p(\mathbf{z}|\boldsymbol{\theta}')$ is defined as
>
> $$\kappa_\rho(\boldsymbol{\theta}, \boldsymbol{\theta}') = \int_\Omega p(\mathbf{z}|\boldsymbol{\theta})^\rho p(\mathbf{z}|\boldsymbol{\theta}')^\rho d\mathbf{z} = \langle p(\mathbf{z}|\boldsymbol{\theta})^\rho, p(\mathbf{z}|\boldsymbol{\theta}')^\rho \rangle_{L_2}, \qquad (5.8)$$
>
> where $L_2(\Omega)$ is a Hilbert space.

In the above formulation sets of features are modelled using some statistical model, which is similar to the temporal feature integration stage, presented in the previous section. The parameters of the statistical model are then passed to a kernel, which returns a measure of similarity between the chosen statistical model.

Figure 5.2 illustrates the product probability kernel, with $\rho = 1/2$ between two univariate Gaussian distributions. The upper figure shows the individual distributions and the lower figure illustrates the product between the two. The area under the product between the two distributions is calculated efficiently by the evaluation of $\kappa(\boldsymbol{\theta}, \boldsymbol{\theta}')$.

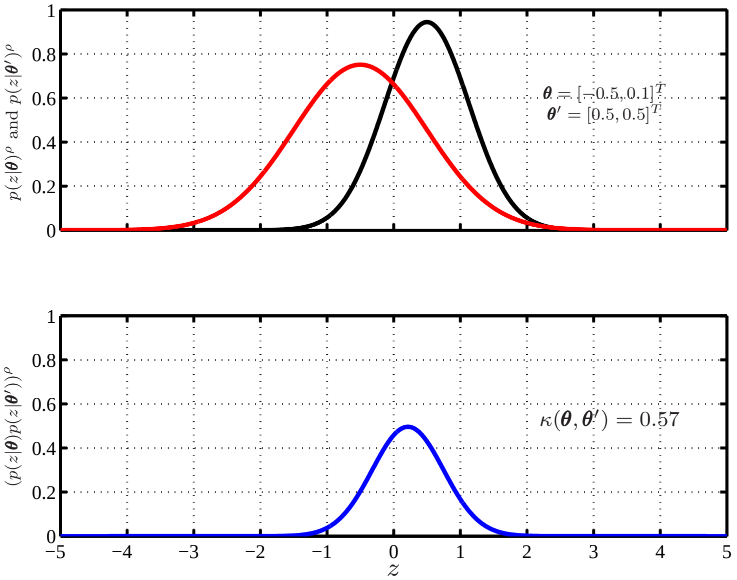


Figure 5.2: The upper figure shows to univariate Gaussian distributions with mean value of $\mu = -0.5$ and $\mu' = 0.5$, and variance of $\sigma^2 = 0.1$ and ${\sigma'}^2 = 0.5$, respectively. The product of the two distributions is illustrated in the lower figure with $\rho = 1/2$. Evaluating the area under this product gives the output of the kernel function, namely $\kappa(\boldsymbol{\theta}, \boldsymbol{\theta}') = 0.57$.

Next, we introduce three different models that can be used as inputs to the product probability kernel, the GM, MAR and GMM.

5.3.2.1 Gaussian model

Assuming that the short-time features over a frame are independent and identically Gaussian distributed, hence $\mathbf{z} \sim \mathcal{N}(\mathbf{m}, \mathbf{C})$, then the PPK can be evaluated in closed form in terms of the model parameters [70]:

$$\kappa_\rho(\boldsymbol{\theta}, \boldsymbol{\theta}') = \int_\Omega \mathcal{N}(\mathbf{m}, \mathbf{C})^\rho \mathcal{N}(\mathbf{m}', \mathbf{C}')^\rho d\mathbf{z} = (2\pi)^{(1-2\rho)D/2} \rho^{-D/2} \cdot$$
$$|\mathbf{C}^\dagger|^{1/2} |\mathbf{C}|^{-\rho/2} |\mathbf{C}'|^{-\rho/2} \mathrm{e}^{\left(-\frac{\rho}{2}\left(\mathbf{m}^T\mathbf{C}^{-1}\mathbf{m} + \mathbf{m}'^T\mathbf{C}'^{-1}\mathbf{m}' - {\mathbf{m}^\dagger}^T\mathbf{C}^\dagger\mathbf{m}^\dagger\right)\right)}, \quad (5.9)$$

where $\mathbf{C}^\dagger = \left(\mathbf{C}^{-1} + \mathbf{C}'^{-1}\right)^{-1}$ and $\mathbf{m}^\dagger = \mathbf{C}^{-1}\mathbf{m} + \mathbf{C}'^{-1}\mathbf{m}'$.

[8] Both structured, such as timeseries, and unstructured data sets.

Assuming independence among the short-time feature dimensions (univariate) the PPK matrix can be generated by addition or multiplication of each dimensions individual PPK matrix.
A rough estimate of the complexity of a single kernel operation scales as $\mathcal{O}\left(D^3\right)$ given that the matrix inverses and determinants each scale with $\mathcal{O}\left(D^3\right)$, where D is the input dimension of $\mathbf{z}$. For the univariate case, the complexity is only $\mathcal{O}(D)$[9]

5.3.2.2 Multivariate autoregressive model

The MAR model was suggested in Section 4.3.2 for temporal feature integration to include temporal information of the short-time features. This was the primary motivation for extracting the closed form solution of this model. The derivation of the PPK for a MAR model have been shown in appendix A.1. In order to obtain a closed form solution, we indirectly assume that the noise process is Gaussian distributed.
The closed form solution of the product probability kernel of the MAR model parameters can be expressed as

$$\kappa(\boldsymbol{\theta}, \boldsymbol{\theta}') = (2\pi)^{(1-2\rho)(P+1)D/2} \rho^{-(P+1)D/2} \\ |\mathbf{M} + \mathbf{M}'|^{-1/2} |\mathbf{C}|^{-\rho(P+1)/2} |\mathbf{C}'|^{-\rho(P+1)/2}, \quad (5.10)$$

where P is the model order of the MAR, and $\mathbf{M}$ and $\mathbf{M}'$ are square symmetric matrices of size $D(P+1)$ created from the parameters of the MAR model: $\mathbf{A}_i(\mathbf{A}'_i)$ for $i = 1, \ldots, P$ and $\mathbf{C}(\mathbf{C}')$, respectively.
The complexity of a single kernel evaluation amounts to $\mathcal{O}\left((D(P+1))^3\right)$. The univariate case is computationally cheaper than the multivariate case, although it still scales with the model order $\mathcal{O}\left(D(P+1)^3\right)$ for a single kernel evaluation.

5.3.2.3 Other statistical models

In [70], a range of different statistical models is suggested from latent models such as the GMM and HMM to other linear dynamical systems. The GMM has been used for timbre modelling of short-time features for music retrieval in [9]. A similarity function was obtained by measuring a symmetric likelihood established on a sample basis from the individual songs. This distance measure, however, requires sampling, and unavoidable becomes an expensive affair. This

[9] It should be noted that the complexity does not compare directly with those of the convolutive kernel, since the parameters are given as inputs to the PPK.

technique has further been applied for measuring artist similarity and for music genre classification in [93].

The GMM has been investigated by the author in a product probability kernel since it has a closed form solution that makes the method computationally attractive. An exact closed form solution can be obtained between song GMMs using a PPK with $\rho = 1$. The Gaussian mixture model

$$\mathbf{z} \sim \sum_{p=0}^{P-1} \pi_p \mathcal{N}\left(\mathbf{m}_p, \mathbf{C}_p\right), \tag{5.11}$$

where $\sum_p \pi_p = 1$ and P is the number of clusters is inserted into the PPK function 5.8 and evaluates to

$$\kappa(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sum_{i=0}^{P-1} \sum_{j=0}^{P-1} \pi_i \pi'_j \int_\Omega \mathcal{N}(\mathbf{m}_i, \mathbf{C}_i) \mathcal{N}(\mathbf{m}'_j, \mathbf{C}'_j) d\mathbf{z}. \tag{5.12}$$

For $\rho \neq 1$ the following approximation is applied

$$\kappa(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sum_{i=0}^{P-1} \sum_{j=0}^{P-1} \left(\pi_i \pi'_j\right)^\rho \int_\Omega \mathcal{N}(\mathbf{m}_i, \mathbf{C}_i)^\rho \mathcal{N}(\mathbf{m}'_j, \mathbf{C}'_j)^\rho d\mathbf{z}. \tag{5.13}$$

In [100, appendix F], we investigated the PPK kernel for a GMM with $\rho = 1/2$ for music genre classification. The parameters of the GMM were found using the NETLAB software package [106].
The complexity of the PPK with a GMM amounts to $\mathcal{O}\left(P^2 D^3\right)$. Modelling each dimension independently (univariate), the complexity reduces to $\mathcal{O}\left(P^2 D\right)$.

A more versatile (and complex) model have been considered by [102, 42] applying the Bhattacharyya kernel (PPK with $\rho = 1/2$). This approach embeds the input space, which could be the short-time features, into feature space using some kernel $\kappa_1(\mathbf{z}, \mathbf{z}')$. In this feature space, the data is assumed Gaussian, which is a crude assumption. The Bhattacharyya kernel is then applied between the sets of embedded feature vectors. In other words, a statistical model is learned from each set of feature space vectors $\Phi_1(\mathbf{Z})$, where $\Phi_1(\mathbf{Z}) = [\boldsymbol{\phi}_1(\mathbf{z}_1), \ldots, \boldsymbol{\phi}_1(\mathbf{z}_\ell)]$. Principal component analysis is applied in feature space to reduce the dimensions of the feature space vectors prior to modelling, after which one can evaluate the Bhattacharrya kernel. In [45] instrument recognition in polyphonic music was tackled using the above procedure.

5.3.3 The symmetric KL-divergence kernel

The use of the symmetric KL-divergence kernel for multimedia applications was suggested in [103]. The kernel has further been applied for music genre classification in [93]. Moreover, it won the artist identification task of MIREX (2005) [38].
The KL-divergence between two distributions is defined as

$$KL(p(\mathbf{z}|\boldsymbol{\theta})||p(\mathbf{z}|\boldsymbol{\theta}')) = \int_{\Omega} p(\mathbf{z}|\boldsymbol{\theta}) \log \frac{p(\mathbf{z}|\boldsymbol{\theta})}{p(\mathbf{z}|\boldsymbol{\theta}')} d\mathbf{z}. \tag{5.14}$$

Symmetry and positive definiteness is obtained using the symmetric version of KL-divergence and applying the exponential function, hence,

$$\kappa(\boldsymbol{\theta}, \boldsymbol{\theta}') = e^{-\frac{\gamma}{2}\left(KL(p(\mathbf{z}|\boldsymbol{\theta})||p(\mathbf{z}|\boldsymbol{\theta}')) + KL(p(\mathbf{z}|\boldsymbol{\theta}')||p(\mathbf{z}|\boldsymbol{\theta}))\right)}, \tag{5.15}$$

where it is observed that the kernel is normalised since the KL-divergence is zero for similar density models.
As with the product probability kernel, different kernels are obtained when using different models for $p(\mathbf{z}|\boldsymbol{\theta})$. Next, we give the closed form solution for the Gaussian model, although the MAR model also results in a closed form expression.

5.3.3.1 Gaussian Model

When considering the multivariate Gaussian model, $\mathbf{z} \sim \mathcal{N}(\mathbf{m}, \mathbf{C})$ the kernel function can be evaluated in closed form:

$$\kappa(\boldsymbol{\theta}, \boldsymbol{\theta}') = \mathrm{e}^{-\gamma\left(\mathrm{tr}\left(\mathbf{C}^{-1}\mathbf{C}' + \mathbf{C}'^{-1}\mathbf{C}\right) + (\mathbf{m}-\mathbf{m}')^T(\mathbf{C}^{-1}+\mathbf{C}^{-1})(\mathbf{m}-\mathbf{m}') - 2D\right)}, \tag{5.16}$$

where tr denotes the trace operation[10]. This kernel have certain similarities with the PPK using a Gaussian model. In Equation 5.16, γ is an additional parameter that controls the decay of eigenvalues of the kernel matrix. It can be determined from cross-validation on the given dataset.
The kernel is computationally similar with the product probability kernel and scales with D as $\mathcal{O}(D^3)$.

[10] Sum of the main diagonal of a matrix.

5.3.4 Other relevant kernels

One of the drawback of the probability based kernels are the problems inherent to density estimation in high dimensional spaces. Naturally, one can consider a dimensionality reduction step to increase the power of the density based methods or, as suggested in [32], to use level sets of probability density functions. Where by the level set of a density function they refer to the part of the space that contains a given fraction of the probability mass.

5.4 The effect of normalisation

Normalisation is a common preprocessing stage for most machine learning tasks. Typically, normalisation improves the generalisation error, or at least does not make it worse. The effect of normalisation of kernels can be considered in both the input space as well as the feature space. Normalisation of the input space [55] according to

$$\tilde{\mathbf{x}} = \frac{\mathbf{x}}{||\mathbf{x}||_2} \tag{5.17}$$

amounts to placing every data sample in a hypersphere of $\mathbb{R}^D$. For the linear kernel (given as $\kappa(\mathbf{x}, \mathbf{z}) = \mathbf{x}^T\mathbf{z}$) this amounts to measuring the angle between vectors (see also subsection 4.5.2), since

$$\kappa(\tilde{\mathbf{x}}, \tilde{\mathbf{z}}) = \frac{\mathbf{x}^T\mathbf{z}}{||\mathbf{x}||_2||\mathbf{z}||_2} = cos(\theta). \tag{5.18}$$

Normalisation in feature space can be performed efficiently through the kernel function since

$$\tilde{\kappa}(\mathbf{x}, \mathbf{z}) = \left\langle \frac{\phi(\mathbf{x})}{||\phi(\mathbf{x})||_2}, \frac{\phi(\mathbf{z})}{||\phi(\mathbf{z})||_2} \right\rangle = \frac{\kappa(\mathbf{x}, \mathbf{z})}{\sqrt{\kappa(\mathbf{x}, \mathbf{x})\kappa(\mathbf{z}, \mathbf{z})}}. \tag{5.19}$$

The normalisation in feature space results in every point is placed on an unit hypersphere. It should be noted, however, that the normalisation is no longer confined to a preprocessing stage since the kernel matrix will be needed in order to perform the normalisation. In [63], the author argues that normalisation improves generalisation error. Both theoretical justification and numerical simulation of two benchmark datasets illustrates the importance of normalisation. For the above mentioned reasons, normalisation has been applied in all experiments involving kernels.

5.5 Discussion

This chapter introduced the notion of kernel aided temporal feature integration and introduced kernels such as the convolution kernel and product probability kernel, which can be evaluated between sets of short-time features. It was illustrated that the convolution kernel implicitly performs temporal feature integration, since it corresponds to performing an averaging of the embedded short-time features in feature space. The product probability kernel efficiently encodes distances between density functions modelling songs and can be regarded as a measure of acoustical similarity between songs. Closed form solutions have been derived for the Gaussian, multivariate AR and Gaussian mixture density models. Not explicitly discussed in this chapter is combinations of kernels. Especially, when initially applying a product probability kernel after which the convolution kernel is applied. This corresponds to an averaging in feature space, hence, an averaging across the density models from each frame of the song. Another interesting density model, which has not been discussed is the simple factor analyser model. Combined with stacking of the short-time features, important temporal information can be included in the density model in a highly compressed manner.

The increased interest in kernels for MIR comes from the versatile modelling of possible non-linear relationships between songs together with the good generalisation obtained from the discriminatively trained support vector classifiers.

Chapter 6

Experiments in music genre classification

This chapter will present results of published [4, 98, 100, 3], see Appendix D, E, F, G, and yet unpublished work [99, appendix H]. The returning topic of this chapter will be that of temporal feature integration. Special emphasis is put on ranking of short-time features at different time-scales, on the multivariate autoregressive model (MAR) for temporal feature integration, as well as on kernel aided temporal feature integration. Furthermore, a detailed description of the datasets and problems of estimating the 'ground-truth' of these is given. A small description of the different classifiers: Linear Model (LM), Generalised Linear Model (GLM), Support Vector Classifier (SVC), Gaussian Mixture Model (GMM) and the Gaussian Classifier (GC), which have been used in the various experiments, is provided. Late information fusion (postprocessing) schemes and methods for accessing the performance of algorithms for the music genre classification task are discussed. The different experiments presented have an impact on the whole system chain illustrated in Figure 6.1, and therefore care must be taken when comparing different methods.

The selected experiments will be presented in three different sections:

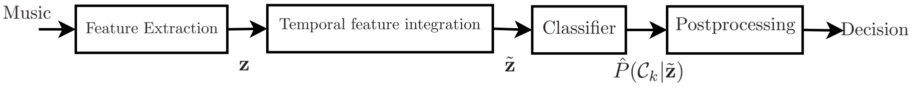


Figure 6.1: The figure illustrates the flowchart of the whole system chain. Dimensionality reduction can be enforced at several places in the system chain and has therefore not been illustrated.

Section 6.4: Discriminative features; this section describes and utilises a method for finding discriminative short-time features at different time-scales for music genre classification using the consensus feature ranking method. This method is not limited to music genre but generalises to other supervised music organisation tasks. The results of this section have been published in [4, appendix D].

Section 6.5: Temporal feature integration; the different temporal feature integration models discussed in Chapter 4 are compared on two different datasets. This investigation deals with temporal feature integration for music genre classification. Two datasets are investigated by combining temporal feature integration and late information fusion. Existing temporal feature integration models such as the FC, MeanVar and MeanCov are compared with the MAR and DAR feature models. This section primarily holds results from [98, appendix E] and [99, appendix H].

Section 6.6: Kernel methods; selected kernel methods from Chapter 5 are compared in a music genre classification task. Selected models are put in a context comparable with the results of the previous section. Special emphasis will be put on the Gaussian model, Gaussian mixture model and the multivariate AR model in a product probability kernel for music genre classification. The different kernels investigated were presented in [100, appendix F].

6.1 Classifier technologies

Quite a few tasks in MIR are formulated in a supervised manner, in which classifiers are an inevitable part. Classifiers can usually be divided according to the models they build. A generative model is a model for randomly generating observed data, and usually involves latent variables. The generative model can be used for modelling the data of a given class after which the conditional distribution of class membership can be found using Bayes rule. A discriminative model does not model the data, but is formulated to achieve the best possible discriminative behaviour between classes. Lately, discriminative methods have achieved a lot of attention in MIR due to their good generalisation performance, see for example [93, 94, 14]. Several workarounds have been suggested in the literature to obtain good discrimination with generative models. One such example is given in [117], where a HMM is trained discriminately by minimising a distance between the true and estimated labels.

The following classifiers have been considered in the present work,

- **Discriminative models** : Linear model classifier (LM), generalised linear model classifier (GLM) and Support Vector Classifier (SVC)
- **Generative models** : Gaussian classifier (GC), Gaussian mixture model classifier (GMM)

and will be briefly discussed in the following subsections.

6.1.1 Discriminative Classifiers

6.1.1.1 Linear Model (LM)

One can formulate the binary classification task as being a single neuron with linear activation function. The model then takes a feature vector $\mathbf{x}$ as input, and outputs a single value y. To obtain a multi-class architecture (c classes) a single neuron can be added for each class [90, 15] and each input is connected to all the output neurons[1]. The weights of the model are found by minimising the sum-of-squares error between the true labels and outputs of the neurons across all the training examples. The method is fast, non-iterative and inherently discriminately trained due to the selected cost-function. The training simply

[1] one-v-all

amounts to finding the pseudo-inverse of the data matrix. Since the classifier is trained using hard assigned classes, the target values are selected as $t^i_{\tilde{k}} = \delta_{i,l}$ for $l = 1, \ldots, c$. Hence the target vector is only different from zero at the corresponding index of the class. A more thorough introduction to the linear model classifier is presented in [15]. To estimate the posterior probability of a class $\mathcal{C}_l$ given some data $\tilde{\mathbf{z}}_{\tilde{k}}$, the *softmax* function has been applied to the outputs of the linear model classifier, hence

$$\hat{P}(\mathcal{C}_i|\tilde{\mathbf{z}}_{\tilde{k}}) = \frac{\exp(y^i_{\tilde{k}})}{\sum_{l=1}^{c} \exp(y^l_{\tilde{k}})} \tag{6.1}$$

where $y^i_{\tilde{k}} = \mathbf{w}_i^T \tilde{\mathbf{z}}_{\tilde{k}} + b_i$ and b_i is the offset. The number of parameters to estimate is $c(\tilde{D}+1)$ when the input data $\tilde{\mathbf{z}}$ has dimension $\tilde{D}$.

The LM for multi-class problems has been applied in ([4, 98, 100], appendix D,E,F), for music genre classification, and have shown good and robust performance.

6.1.1.2 Generalised Linear Model (GLM)

In the linear model it is indirectly assumed that the true class can be approximated by a smooth function with additive Gaussian noise. The sum-of-squares error cost-function is then found to be optimal when minimising the log-likelihood. This assumption is reasonable for most regression problems. However, for classification purposes, the labels are typically binary, hence, each example belongs to a single class exclusively. Restricting the output $y^i_{\tilde{k}}$ of the linear model between 0 and 1 using e.g. the softmax function, a more reasonable error function is the cross entropy for multiple classes [15]. Assuming independence among the c outputs of the linear model with a softmax function, the conditional distribution of the true class and the input data to the classifier is given as

$$p(\mathbf{t}_{\tilde{k}}|\tilde{\mathbf{z}}_{\tilde{k}}) = \prod_{l=1}^{c} \left(y^l_{\tilde{k}}\right)^{t^l_{\tilde{k}}}, \tag{6.2}$$

where $y^i_{\tilde{k}}$ for $i = 1, \ldots, c$ represents the bounded output. The cost-function is created by forming the negative log likelihood, hence

$$\mathcal{E}_m = -\sum_{\tilde{k}=0}^{N_{\text{train}}-1} \sum_{l=1}^{c} t^l_{\tilde{k}} \ln y^l_{\tilde{k}}, \tag{6.3}$$

where N_{train} represents the number of training examples. The NETLAB package [106] was used in the experiments involving the GLM classifier. The package, includes a weight decay regulariser, which for the multi-class problem means that the following term is added to the cost function:

$$\mathcal{E}_w = \frac{\alpha}{2} \sum_{l=1}^{c} ||\mathbf{w}_l||_2. \tag{6.4}$$

The regularisation constant α is known as the weight decay rate, and helps the classifier not to overfit the training data, thereby improving the generalisation error. Cross-validation has been applied for selecting an optimal value for α in the experiments involving the GLM. As with the LM $c(\tilde{D}+1)$ parameters have to be estimated. The GLM classifier has been applied in [99, appendix H] for music genre classification.

6.1.1.3 Support Vector Classifier (SVC)

The support vector machine [144, 28] and related methods have lately received a lot of attention in different fields of MIR. The SVM has the ability to work in very high dimensional feature spaces, which ensures a high flexibility but apparently seems to be at odds with the curse of dimensionality. Normally, a good fit on the training data leads to a poor generalisation error, but support vector machines manage to avoid this problem by optimising a bound on the generalisation error in terms of quantities that do not depend on the dimension of the feature space [131], hence enabling good performance unaffected by the curse of dimensionality.

The SVC optimisation criterion involves the 2-norm of the weight vector realizing a linear function that separates the positive and negative examples with maximum margin together with a sum over any violations of this margin constraint.

The margin optimisation problem with constraints is usually formulated in a primal version and transformed to a dual version[2] involving only inner products between feature vectors. The inner products can then be calculated using the kernel-trick. For further details regarding generalisation bounds and issues on the cost-functions the author encourages the reader to look for more information in [28], which provides a good introduction to the support vector machines and the mathematics that support them.

There exists many different SVC implementations, which are available from the Internet. The LIBSVM software package[3] [25] has been applied in experiments involving the SVC. The constrained quadratic problem, inherent to most support vector classifiers is solved in this package using a *Sequential Minimal*

[2] Using the Karush-Kuhn-Tucker(KKT) conditions at the solution.

[3] Although this implementation is written in C-code, there exist several wrappers for using the package in Matlab, R and other high-language programs.

Optimisation (SMO) algorithm. SMO is an efficient (and memory friendly) algorithm for solving quadratic problems when the solution involves sparseness. The uniqueness of the convex quadratic problem ensures a single solution for the given dataset. This make comparisons between different implementations easy.

The support vector machine is inherently a two-class discriminator. However, there exists several methods for handling multi-class problems. The LIBSVM package implements a one-v-one approach, hence, having c classes the number of classifiers, which need to be trained is $\binom{c}{2}$. Each of the training rounds, however, involves fewer datapoints than when using the the one-v-all scheme. In the one-v-all scheme a single decision boundary is determined between the class under investigation and the remaining classes. This results in training c different classifiers. Also, error correcting codes have been suggested to maximise the difference between outputs of the individual one-v-one classifiers, see e.g. [36]. A thorough investigation of the different approaches for the multi-class problem was performed by [118], where the author compared different approaches on several benchmark toy-sets. No significant differences among the various ways of combining the outputs were found. However, there was a weak evidence in favour of the one-v-one combination scheme on small datasets consisting of many classes and an overall high test error.
An early paper on audio organisation applied the support vector machine [104]. Here, the authors investigated a simple audio classification problem combining the modelling capabilities of a GMM with the good discriminative performance of a support vector machine using a Fisher kernel [69]. In [57], the support vector machine was applied with success in a content-based audio classification and retrieval task comparing their approach with that of [148]. Recently, [93, 83] and [100, appendix F] investigated more complicated kernel functions for the task of music genre classification.

6.1.2 Generative Classifiers

6.1.2.1 Gaussian Mixture Model (GMM)

Instead of focussing on good discriminative performance, which in principle says little on the generating process, the task of a generative model is to learn the underlying structure of the data. The probability density function of a GMM is defined as

$$p(\tilde{\mathbf{z}}) = \sum_{l=0}^{L-1} \pi_l p(\tilde{\mathbf{z}}|\boldsymbol{\theta}_l), \tag{6.5}$$

where L is the number of clusters, $\{\pi_l\}$ are a set of mixing coefficients satisfying $\sum_{l=0}^{L-1} \pi_l = 1$, and $p(\tilde{\mathbf{z}}|\boldsymbol{\theta}_l) \sim \mathcal{N}(\mathbf{m}_l, \boldsymbol{\Sigma}_l)$. When applied in a supervised manner for music genre classification, this amounts to fitting a GMM to each music genre. Hence, $p(\tilde{\mathbf{z}}|\mathcal{C}_l) \sim$ GMM.
The NETLAB package [106] has been applied in experiments involving the GMM. The number of cluster components L can be estimated from the data using model order selection methods such as e.g. Bayesian Information Criterion (BIC) [136], or by methods such as cross-validation. In our case, the number of clusters were found using cross-validation. The GMM has $cL\frac{\tilde{D}(\tilde{D}+1)}{2}+cL(\tilde{D}+1)$ parameters, which need to be estimated when using a full covariance structure. With a diagonal covariance structure the number of parameters reduces to $cL(2\tilde{D}+1)$.

To some extent, the GMM compares with the K-means algorithm [90]. Whereas the normal K-means algorithm is utilising hard assignment of data points to clusters, the GMM is using soft assignments. Hence, a single data point can belong to several clusters. The traditional K-means algorithm has been used as an initial starting guess to fit the GMM parameters, after which a traditional EM-algorithm [31] was applied.
The GMM has been applied in MIR both for modelling individual songs [93, 9] and for the task of structural learning in, for example, music genre classification [99, 10] and singer identification [75].

6.1.2.2 Gaussian Classifier (GC)

The Gaussian classifier [41] can be seen as a GMM with a single cluster. The probability density function for a class $\mathcal{C}_l$ is assumed to follow a Gaussian distribution, thus

$$p(\tilde{\mathbf{z}}|\mathcal{C}_l) \sim \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l). \tag{6.6}$$

Using the negative log-likelihood of the observed data as the error function makes the training fast and non-iterative. When $\boldsymbol{\Sigma}_l$ has a full covariance structure for all l, the decision surface becomes non-linear, whereas for an isotropic, diagonal covariance and when $\boldsymbol{\Sigma}_l = \mathbf{C}\ \forall l$, where $\mathbf{C}$ is a full covariance matrix, the decision surface is linear. The number of parameters to be estimated for this model is similar to the GMM using $L = 1$.

6.1.3 Postprocessing and late information fusion

The task of combining classifier outputs has been considered by several researchers in machine learning, see e.g. [76, 34]. One can, for example, combine several classifier outputs and reach a consensus across the different classifiers. Adaboost type algorithms where several weak-learners are combined to obtain a decision of improved quality, have proved to be one of the most successful combination strategies, see e.g. [121]. In [14] an adaboost algorithm is applied with great success in music genre classification.

In the current work, late information fusion or temporal fusion, as noted in [34], have been applied to obtain decisions at larger timescales. Late information fusion is the task of combining a sequence of outputs from a classifier into a single consensus decision.
Late information fusion has successfully been applied in combination with temporal feature integration for music genre classification in [98, appendix E] where different voting methods such as majority-voting, sum-rule, and median-rule were investigated. The sum-rule over a frame of length $\tilde{K}$ of temporal integrated features amounts to

$$\text{argmax}_l \sum_k \hat{P}(\mathcal{C}_l|\tilde{\mathbf{z}}_{\tilde{k}}) \quad \text{for} \quad \tilde{k} = 0, \ldots, \tilde{K} - 1, \tag{6.7}$$

where for the median-rule, the median is calculated instead of the sum. In majority voting one counts the hard decisions from the classifier and select the class that received the most votes.

6.2 Performance measures and generalisation

For music genre classification (and related tasks of MIR) a common applied error measure is the classification accuracy. Researchers usually apply either the normalised or unnormalised accuracy. The normalised accuracy is calculated by taking prior information of the different classes into account. The two datasets described in Section 6.3 (A and B) have the same number of examples in the different classes, which corresponds to an uniform prior on the classes. The classification accuracy is computed as the number of correctly classified examples out of a test dataset of size N_{test}. Another frequently applied measure for accessing the quality of a classifier is the 'confusion matrix'. For the normalised confusion matrix, all numbers in one row, which are estimations for the same true class should sum to 1. Each column contains the probability of estimating a given class for all the true classes. Table 6.1 illustrates a simple two class problem with the true classes being A and B and the estimated classes given by a and b.

Comparing only the classification accuracy of two algorithms does not provide enough knowledge of the algorithms individual behaviour. Two classifiers C-I

$\hat{P}(a\|A)$	$\hat{P}(b\|A)$
$\hat{P}(a\|B)$	$\hat{P}(b\|B)$

Table 6.1: Simple confusion matrix illustrated with estimated probabilities. All numbers in one row, which are estimations for the same true class should sum up to 1. Each column contains the probability of estimating a given class for all true classes.

and C-II may have the same classification accuracy without failing on the same examples. E.g. having a similar classification accuracy, classifier C-I might be better at detecting jazz, whereas classifier C-II could be especially good at detecting rock. Confusion matrices provide a more complete characterisation of a classifier performance and can be especially useful when combining several classifiers for obtaining a better overall consensus accuracy. Confusion matrices have been applied constructively in hierarchical music genre classification by aggregating classes with large confusion, see e.g. [10, 84].

The above error measures are relatively simple when considering hard decisions. However, more complex error measures can be thought of when songs, for example, are allowed to belong to more than one class. Also from a user perspective, it might be more relevant to penalise some errors harder than others. If a user is very keen on jazz music he/she might consider errors made in this specific genre more important than other types of errors. This type of information can, in principle, be included in the cost-function of the classifier. In line with a more specialised cost-function, the notion of 'graceful' errors was mentioned in [7]. Graceful errors refer to some errors being weighted less than others. E.g. confusion between rock and alternative should not be penalised as hard as errors between rock and classical.

6.2.1 Generalisation error and finding the best learning algorithm

When devising new learning algorithms for machine learning the optimal goal is to prove that the suggested method indeed performs better than other methods irrespective of the dataset applied. The generalisation error expresses the overall error of a model irrespective of the dataset applied, and it simply becomes a question of achieving better generalisation than the competing algorithms. There is, however, a problem, since the generalisation error is not easily obtained. One of the major topics of support vector machines is to obtain, and improve bounds on the generalisation error. The bounds, however, are provided with a certain confidence, see e.g. [131, 63].
A frequently applied suboptimal approach for accessing the generalisation error is to evaluate the algorithms on several benchmark datasets. Hence, if the sug-

gested learning algorithm performs better on most of the different benchmark datasets, it must be preferred.

In [35], the author provides a good overview of statistical tests which can be applied when comparing supervised learning algorithms. The author concludes that the McNemar test and the k-fold cross-validated paired t-test have reasonable type-I errors[4].
When using cross-validation, see e.g. [15], the dataset is divided at random into S non overlapping segments. The learning algorithm is then trained using data from $S - 1$ segments and evaluated on the remaining set. This procedure is repeated for each of the possible choices of segments. The leave-one-out procedure is the limit of performing N-fold cross validation (where N is the number of samples in the dataset).

In the experiments learning algorithms are compared using the McNemar test when applied on fixed size training/test set, whereas the cross-validation paired t-test has been applied when the test accuracy is obtained by cross-validation. Furthermore, it should be noted that nuisance parameters of e.g. the SVC have been determined from cross-validation. It is noted that using too few folds when determining nuisance parameters results in larger discrepancy between the training and training/test splits which can lead to sub-optimal performance of the classifier.

Ensuring that the classifier is not overfitting the training data, which leads to a poor generalisation error is an important aspect when comparing different systems for music organisation. To ensure the best possible generalisation error of the different systems on the given dataset, learning-curves can be produced [77]. A learning-curve is produced by plotting the test accuracy as a function of an increased number of training examples. Increasing the number of training samples the test accuracy at some point levels out. If not, dimensionality reduction might be the only possibility to ensure a fair comparison between the different systems. Cross-validation (or a separate validation set) has been used to obtain the learning curves in the different experiments where learning-curves have been used.

[4] A type-I error occurs when one rejects the null hypothesis even though it was true.

6.3 Datasets

The field of MIR have had, and still has, problems with legal obstacles when sharing music between sites. This can lead to suboptimal solutions where researchers apply in-house datasets with unwanted side effects to evaluate their methods. One such example is the album-effect pointed out by Berenzweig [11][5]. In related areas such as text retrieval, and ASR, benchmark datasets have been available for a long time [81, 46]. This have led to a common ground for comparing algorithms. Furthermore, the generalisation performance of an algorithm has been accessed by measuring the performance over several such benchmark datasets.

It has shown to be possible to share datasets across different labs, see [12], where the short-time features of the USPOP2002 dataset[6], which consists of approximately 8700 different songs distributed across 400 different US music pop artists, were shared for joint work between *LabROSA* (Columbia), *MIT* and *HP Labs*(Cambridge). Lately, a benchmark dataset for music classification (and clustering) was proposed in [65], avoiding legal obstacles by only releasing 10 sec music snippets sampled randomly from the full length songs. Furthermore, initial investigations of this dataset have been given in [65, 105]. A common repository of audio, metadata, ontologies and algorithms is proposed in [22], as a framework where researchers across the globe can test and evaluate their algorithms on various audio mining tasks.

One of the main problems when creating a dataset for MIR is how to obtain "ground truth". The work by [12], which investigated similarity among artists, considered various sources of ground truth, such as 'a survey', 'expert opinion', 'playlist co-occurrence', 'user collections' and 'web text'. The different approaches for obtaining ground truth can be divided into three main categories [108]:

- 'Editorial metadata', which is literally inserted by the editor. Typically this information is added by experts
- 'Cultural metadata', can be a result of analysing emerging patterns and categories. Methods such as collaborative filtering and machine learning approaches can be used to learn similarities of data (e.g. co-occurrence in playlists)

[5] Consider two music pieces $\mathcal{X}$,$\mathcal{Y}$ from the same album. Let $\mathcal{Y}_2$ represent the same recording of the song $\mathcal{Y}$, just sampled from another album. Let $d(\mathcal{X},\mathcal{Y})$ be a measure of acoustic similarity between two songs. The consequence of the album effect is that $d(\mathcal{X},\mathcal{Y}) > d(\mathcal{X},\mathcal{Y}_2)$, even though $\mathcal{Y}$ and $\mathcal{Y}_2$ are exactly the same songs. This effect stems from producer artifacts and might blur the conclusions of the experiments.

[6] http://labrosa.ee.columbia.edu/projects/musicsim/uspop2002.html

Dataset A		
Genre	Artists	Total
Classical	9 5	9
Rock	8 5	8
Jazz	15 4	18
Pop	12 5	13
Techno	13 5	18

Table 6.2: The five genres, number of artists in cases where the dataset was divided into a training and test set, and the total number of artists in the corresponding genre. The scheme should be read as follows : for rock a total of 8 artists is present. The training set contains 8 artists, which corresponds to a complete overlap of artists in the test and training set. Jazz and techno have distinct artists in training and test samples.

- 'Acoustic metadata' is the purely objective information in the music data. Vocal presence, silence/non-silence could be objective measures.

Cultural metadata can be accessed in various ways. A common approach is the use of co-occcurence methods. For instance, finding co-occurring words with a genre or an artist. Co-occurrence can be based on closeness on a web-page, or in playlists, where playlists can be from the radio, from users[7] or from album compilations, see e.g. [113].

6.3.1 "Dataset A" for music genre classification

This dataset was constructed by the author and a fellow researchers own collection (in-house) and was created for music genre classification task. Ground truth was provided by the authors. This dataset has been applied in various experiments, see e.g. appendix D, E and H. It consists of 100 music titles from 66 different artists. The music pieces are extracted in mono PCM format[8] at a samplerate of 22050 Hz. The music distributes evenly among the 5 music genres: classical, hard rock, jazz, pop and techno. Table 6.2 shows a detailed overview of the number of artists in the dataset. The middle column shows the partition of artists when the data is divided into a fixed training and test set. A splitting ratio of 75/25 was used when the dataset was applied in a fixed training/test scenario.

[7] www.audioscrobbler.com

[8] Microsoft wave-format.

6.3.1.1 Human evaluation - (cultural metadata)

Although the selected genres were chosen as distinct as possible (some overlap with pop), the genre labels are still subjective. To access the ground truth of the dataset 22 persons[9] were asked to evaluate the test set. This test set is believed to be representative of the complete dataset. From the middle of each music piece 30 seconds were extracted. Each music snippet was divided into 5 overlapping 10 second snippets. This resulted in a total of 125, 10 seconds music snippets. Each of the evaluators were kindly instructed to classify each music piece into one of the 5 genres on a forced choice basis. No prior information except for genre names was given. Considering all classifications performed across the different persons as that of a single "average" human[10], a 95% binomial confidence interval is $[97\%, 98\%, 99\%]$ ([lower bound, mean, upper bound]).

The human evaluation provides a measure of complexity of the applied taxonomy and how well the used metadata show consensus with the average human perception of the music titles.

6.3.2 "Dataset B" for music genre classification

This dataset consists of 1317 music pieces distributed evenly among the eleven music genres: alternative, country, easy listening, electronica, jazz, latin, pop & dance, rap & hip-hop, r&b and soul, reggae and rock, except for latin, which had 117 music pieces. The dataset consists of 720 different artists, which amounts to approximately 2 music pieces per artist, on the average. This dataset is considered complex w.r.t. the number of artists per music title compared to previously suggested datasets in MIR. For instance, the USpop dataset [12] consists of 8700 music pieces distributed among 6 genres with a total of 400 artists. Another recently applied dataset is that collected from Magnatune[11], see e.g. [110]. This dataset consists of 10 music genres with a total of 3248 music tracks distributed among 147 different artists which amounts to 22 music pieces per artist.

The music pieces of dataset B are all encoded in MPEG1-layer 3 format (stereo) at a bitrate of 128 kbps. In the experiments, the music pieces were downsampled to 22050 Hz and a mono signal extracted. A more detailed overview of the database can be seen from Table 6.3. In [100, appendix F], the dataset was applied using a fixed training/test partition of 1097/220. Accessing the gen-

[9] specialists and non-specialists

[10] Assuming independence among samples.

[11] Music pieces collected from `www.magnatune.com`.

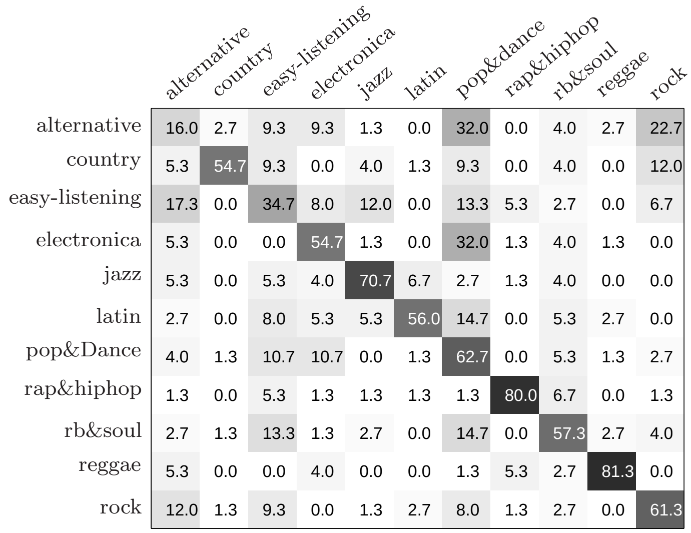
Dataset B		
Genre	Artists	Total
Alternative	55 20	73
Country	15 19	31
Easy Listening	44 20	62
Electronica	64 19	81
Jazz	48 18	60
Latin	24 19	38
Pop & Dance	73 20	89
Rap & Hiphop	49 20	62
R&B & Soul	63 20	76
Reggae	47 20	65
Rock	67 20	83

Table 6.3: The eleven genres, number of artists in cases where the dataset was divided into a training and test set, and the total number of artists in the corresponding genre. As indicated there is little or no overlap of artists when the dataset is divided into a single training/test set.

eralisation performance using cross-validation results in little overlap between training / test splits. It is believed that this dataset is a good representation of music genre. The labels have been provided by an external reference.

6.3.2.1 Human evaluation

The genre labels obtained by an external reference were investigated by 25 persons. The test set, consisting of 220 music snippets of each 30 seconds sampled from the middle of each music piece, was considered as being representative of the complete dataset. Each of the subjects was asked to classify 33 music snippets into one of the 11 genre categories on a forced choice basis. No prior information, except for the genre names was given. Considering all the classifications performed across the different test individuals as independent a 95% binomial confidence interval becomes [0.54 0.58 0.61]. The consensus accuracy obtained by voting across 3 or more persons vote results in a 95% binomial confidence interval of [0.61 0.68 0.75]. Only 172 music snippets out of the 220 did have three or more votes. Another interesting measure is the confusion among



	alternative	country	easy-listening	electronica	jazz	latin	pop&dance	rap&hiphop	rb&soul	reggae	rock
alternative	16.0	2.7	9.3	9.3	1.3	0.0	32.0	0.0	4.0	2.7	22.7
country	5.3	54.7	9.3	0.0	4.0	1.3	9.3	0.0	4.0	0.0	12.0
easy-listening	17.3	0.0	34.7	8.0	12.0	0.0	13.3	5.3	2.7	0.0	6.7
electronica	5.3	0.0	0.0	54.7	1.3	0.0	32.0	1.3	4.0	1.3	0.0
jazz	5.3	0.0	5.3	4.0	70.7	6.7	2.7	1.3	4.0	0.0	0.0
latin	2.7	0.0	8.0	5.3	5.3	56.0	14.7	0.0	5.3	2.7	0.0
pop&Dance	4.0	1.3	10.7	10.7	0.0	1.3	62.7	0.0	5.3	1.3	2.7
rap&hiphop	1.3	0.0	5.3	1.3	1.3	1.3	1.3	80.0	6.7	0.0	1.3
rb&soul	2.7	1.3	13.3	1.3	2.7	0.0	14.7	0.0	57.3	2.7	4.0
reggae	5.3	0.0	0.0	4.0	0.0	0.0	1.3	5.3	2.7	81.3	0.0
rock	12.0	1.3	9.3	0.0	1.3	2.7	8.0	1.3	2.7	0.0	61.3

Figure 6.2: The figure shows the confusion matrix from evaluations of the 25 people. The "true" genres are shown on along the different rows, hence, each row sums to 100%. The diagonal of the confusion matrix illustrates the accuracy of each genre separately.

the music genres. The "average" single human confusion matrix has been illustrated in Figure 6.2. As illustrated in the confusion matrix, music genres such as reggae, rap&hiphop and jazz are recognised with accuracies above 70%, whereas music genres such as alternative and easy-listening are not well understood by the test subjects.

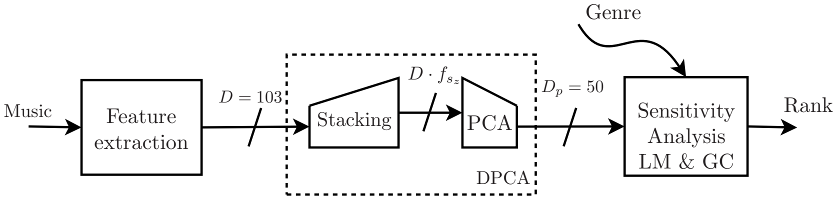


Figure 6.3: Feature ranking for music genre classification using Dynamic PCA and sensitivity analysis. The short-time features of dimension 103 were investigated at lag values of $f_{s_z} = 0, 50, 100$ after which PCA was applied. It was found optimal to project the high dimensional spaces into the leading 50 eigenvectors.

6.4 Feature ranking in music genre classification

Finding an optimal set of features for the task of automated music genre classification is not easy, since music genre is not restricted to a single time-scale. Instrumentation, rhythm, vocal, etc. are all characteristics of music that makes discrimination between different genres possible for a human. The results presented in this section have been published in [4, appendix D].

The main questions considered were: 1) does the ranking of short-time features change with a different decision time horizons, 2) which features are generally ranking the better at these time-scales.
To work with the short-time features at larger time-scales stacking was applied (see Section 4.2). A sensitivity analysis was performed using the proposed technique of consensus feature ranking. The flowchart of the complete ranking system have been illustrated in Figure 6.3.
The following sub-sections will discuss the approach and results in more detail. Dataset A has been applied for this investigation.

6.4.1 Short-time features

The investigated short-time features are listed in Table 6.4. The short-time features are extracted using a frame-size of 30 ms, and a hop-size of 10 ms, which minimises aliasing along the temporal dimension.

S.T.F.	Dim.	Comments
MFCC	13	0th order MFCC included
DMFCC	13	
LPC	13	12th order model+gain
DLPC	12	No delta gain
ASE	27	$r = 1/4$ $loEdge = 125\,\text{Hz}, hiEdge = 9514\,\text{Hz}$
ASC,ASS	2	
ASF	21	$r = 1/4$ $loEdge = 250\,\text{Hz}, hiEdge = 9514\,\text{Hz}$
STE, ZCR	2	
Total	103	

Table 6.4: The different investigated short-time features. Each of the features are explained in detail in Section 3. S.T.F. is an acronym for short-time feature.

6.4.2 Feature stacking and dimensionality reduction

As discussed in Chapter 4, stacking is one approach to temporal feature integration. The stacked features were denoted as $\tilde{\mathbf{z}}$, where the dimension becomes $\tilde{D} = f_{s_z} D$. The largest possible frame-size (lag time) to work with at that time was $f_{s_z} = 100$, which corresponds to an effective time-scale of 1 sec. Maximum overlap between frames was applied, hence, $h_{s_z} = 1$. With a short-time feature dimension of $D = 103$, the dimensionality of the stacked vector becomes $\tilde{D} = 10300$ for $f_{s_z} = 100$, which is high, even for a linear classifier and would lead to overfitting on dataset A. Dimensionality reduction in form of principal component analysis (PCA) was applied. Applying PCA to time-stacked features is known as *dynamic PCA*, see e.g. [79]. Dynamic PCA extends regular PCA by both decorrelating feature dimensions and temporal correlations in the short-time features. In principle, one can view the eigenvectors in which the high-dimensional data is projected onto as filters, where relevant temporal information in the short-time features are retained. The leading principal directions will contain information of the lower frequencies across the temporal dimension of the short-time features. Figure 6.4 shows the leading 25 principal directions extracted from the 0'th order MFCC stacked with a lag value of $f_{s_z} = 100$ and a hop-size of $h_{s_z} = 10$, from dataset A. As indicated, several of the projections carry information on specific frequencies, and show resemblance with cosine basis functions. The eigenvalues and eigenvectors can be determined by forming the covariance matrix of the temporal feature integrated vectors, after which an eigenvalue decomposition is performed. Collecting the stacked short-time features ($\tilde{\mathbf{z}}$) of each song in the training data in the matrix $\tilde{\mathbf{Z}}$

$$\tilde{\mathbf{Z}} = \left[\tilde{\mathbf{z}}_1^1, \ldots, \tilde{\mathbf{z}}_{\tilde{K}}^1, \ldots, \tilde{\mathbf{z}}_1^N, \ldots, \tilde{\mathbf{z}}_{\tilde{K}}^N\right], \qquad (6.8)$$

of dimension $\tilde{D} \times N\tilde{K}$, where $\tilde{K}$ indicate that music pieces are of same length, then the biased covariance matrix can be formed as

$$\mathbf{C} = \tilde{\mathbf{Z}}\tilde{\mathbf{Z}}^T, \qquad (6.9)$$

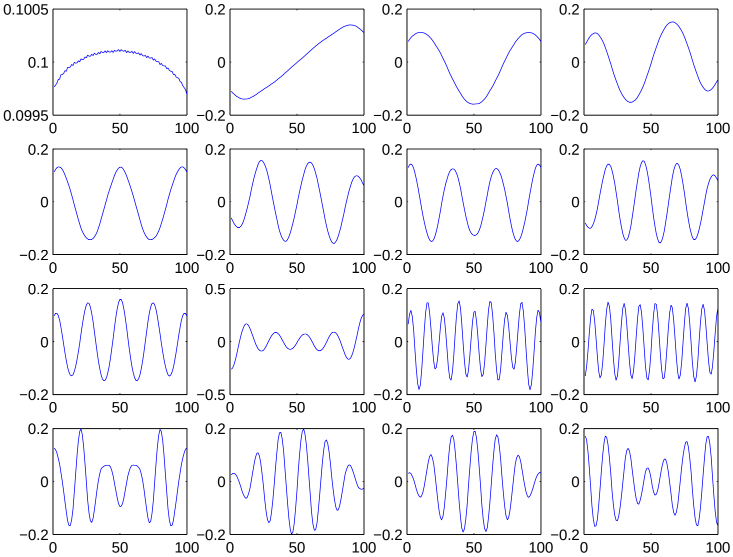


Figure 6.4: Basis functions extracted from the leading eigenvalue/vector pairs of the covariance matrix **C**. As indicated many of the basis functions show similar behaviour as the DCT basis functions. The upper left figure shows the eigenvector corresponding largest eigenvalue.

which has a dimension of $\tilde{D} \times \tilde{D}$. For large frame-sizes it becomes computationally expensive to build, and calculate eigenvector/value pairs of the covariance matrix. Furthermore, the memory requirements increases drastically. To overcome some of the memory and time problems a method that we denoted the 'simple' PCA was applied, see Appendix B.1. The method estimates the leading principal components by sampling the training data quite heavily after which the PCA problem is solved in the smaller of the two spaces [132]. A small experiment was conducted to verify the robustness of the method. With a lag of $f_{s_z} = 50$, the classification accuracy was measured on a separate validation set when the number of randomly selected training points were varied between $200 - 1500$. The variation in the mean accuracy using only the leading 50 eigenvectors was less than one percent, indicating robust behaviour. This is ascribed to the large redundancy in the data due to the small hop-size (oversampling). Experimenting with the dimensionality of the projection, and to avoid overfitting problems, learning curves revealed that a projection into the leading 50 principal components for both the GC and LM classifier was optimal.

6.4.3 Ranking features

Feature ranking concerns the problem of ranking features after importance for the given learning problem. It is closely related to the task of feature selection where a subset of discriminant features is found. A good introduction to feature selection can be found in [58], which also describes inherent problems of different ranking methods. Faced with a rather large amount of features, a fairly simple approach was taken. We are interested in finding the short-time features, which have the largest impact on the classification accuracy of music genre. Thus, being able to rank the short-time features after relevance.

A simple measurement of how much the classifier reacts on each input is the *sensitivity map*, see e.g [133, 151]. The perturbation of the classifier cost w.r.t. each of the inputs is found by calculating the gradient of each output with respect to all the inputs. For the investigated classifiers, this amounts to evaluate the derivative of the estimated posterior $\hat{P}(\mathcal{C}_l|\tilde{\mathbf{z}})$ of each class with respect to the individual feature dimension. $\mathcal{C}_l$ is the lth genre. One way of calculating the sensitivity map for a given system is the *absolute value average sensitivity* [133],

$$\mathbf{s} = \frac{1}{\ell_{test} c} \sum_{l=1}^{c} \sum_{n=0}^{\ell_{test}} \left| \frac{\partial \hat{P}(\mathcal{C}_l|\tilde{\mathbf{z}}_n^p)}{\partial \tilde{\mathbf{z}}_n} \right|, \tag{6.10}$$

where $\tilde{\mathbf{z}}_n$ is the n'th stacked short-time feature of the test set and $\tilde{\mathbf{z}}_n^p$ is feature projected into the $p = 50$ leading eigenvectors. ℓ_{test} is the total number of test frames and $c = 5$ is the number of genres. The absolute operator works on each element of the vector. Taking the absolute value of each test example's perturbation is necessary to avoid possible cancellation of positive and negative values because of multiple decision boundaries. Averaging has been performed across samples and genres, since we are interested in features, which have the largest discriminative power on the complete system. It should be noted that the sensitivity map expresses the importance of each feature individually, hence, if large correlations exists between feature dimensions, similar features might be ranked as being equally important.

Figure 6.5 illustrates the absolute value average sensitivity in a simple classification setup. Two classes in 2 dimensions, shown by red and black dots, were learned by a Gaussian classifier. The sensitivity was evaluated by sampling 1000 points from each distribution and evaluating Equation (6.10). The $\mathbf{s}$ values have been indicated on the axis's of the 4 figures as s_x and s_y. A larger numeric value illustrate that the feature is more relevant. The contours illustrated on the figures, show the absolute value of the derivative of the posterior distribution, which as expected, are only non-zero around the classification boundary. The values of $\mathbf{s}$ shown on the x-y axis, have been normalised, hence $\tilde{\mathbf{s}} = \frac{\mathbf{s}}{||\mathbf{s}||_{max}}$.

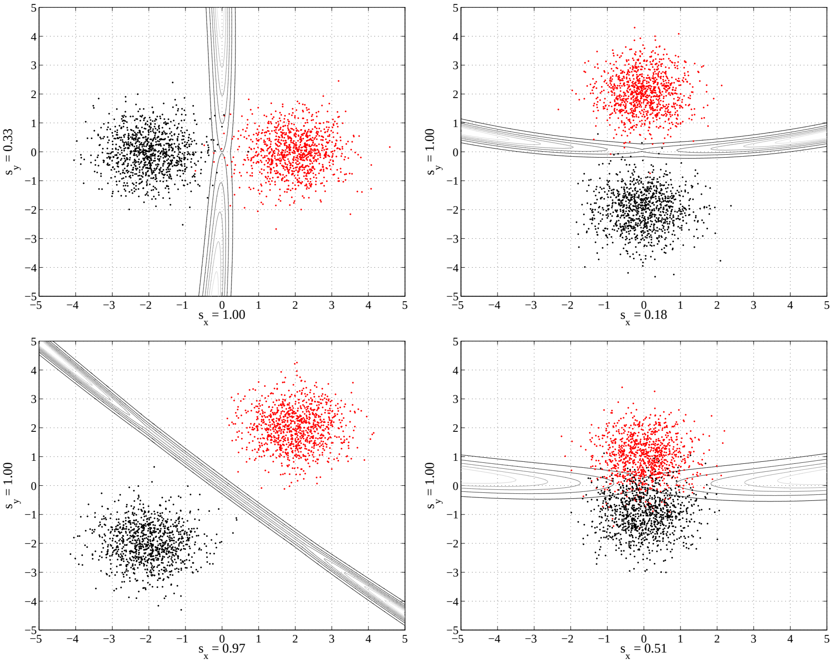


Figure 6.5: Feature sensitivities in different configurations. Upper left: discriminate power in x_1, upper right : discriminative power in x_2, lower left : discriminative power in both x_1 and x_2 and lower right: discriminative power in x_2 for overlapping data.

Working with a fixed training/test set, this ranking technique will provide one single ranking. Random sampling of the training set will provide new estimates of the posterior distribution. Furthermore, at lag values larger than 0, we need some way of combining the sensitivity ranks to convey information about the short-time features at larger lags. A simple method that we denoted 'consensus feature ranking' was proposed to handle these iterations for finding the most discriminate short-time features at different time-scales for the music genre classification task. The consensus feature ranking method is listed in Table 6.5. The matrix $\mathbf{R}$ is denoted the sensitivity rank matrix. Each row contains a histogram of the ranks achieved by the specific feature. A very narrow histogram (little spread), indicate robustness of that feature, hence, achieve a similar ranking in each permutation of the training set, whereas wider histograms reveals that the feature is very sensitive to the training splits. Figure 6.6 shows the sensitivity rank matrix at lag $f_{s_z} = 100$ using the LM classifier with `noperms=50`, where the posterior distribution is estimated using the softmax function. The dots shown in each row illustrate the median of the histogram. Furthermore, the average classification accuracy of the LM classifier on the 5 genre music setup at this lag size was $\sim 71\%$, where random guessing would give 20%. Table (6.6)

Table 6.5: Consensus feature ranking procedure as applied in [4, appendix D].

```
Initialise $\mathbf{R} = \mathbf{0}$ of dimension $D \times D$
Repeat noperms times
    1. Train classifier

    2. Calculate sensitivity rank using Equation (6.10)

    3. For $f_{s_z} > 0$ apply the mean value of the sensitivity
       across time of each feature dimension to produce
       $\mathbf{s}_{avg}$ of dimension $D$

    4. Update $\mathbf{R}$ as

               $\mathbf{v} = \mathrm{sort}(\mathbf{s}_{avg})$
               for $j = 1$ to $D$
                   $R_{v_j,j} = R_{v_j,j} + 1$
               end for

       where $\mathbf{v}$ represents a vector of indices to the sorted
       rank values. Hence, $v_1$ contains the index to the
       feature with the highest rank.
Output the sensitivity rank matrix $\mathbf{R}$
```

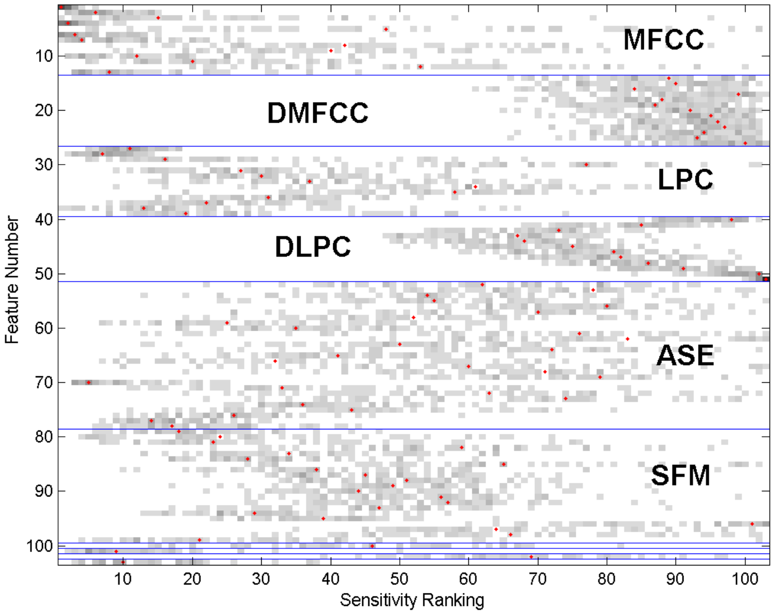


Figure 6.6: The figure shows the sensitivity rank matrix **R** obtained from the consensus feature ranking procedure outlined in Table 6.5 at lag $f_{s_z} = 100$, which corresponds to approximately 1 sec. The matrix was obtained using a LM classifier and 50 permutations of the training data. The last 4 features are the ASC, ASS, STE and ZCR, respectively. The ordinate show each of the 103 features investigated, whereas the abscissa shows the obtained rank in each of the 50 permutations of the training set. The dots in each row illustrates the median of the "histogram" from each row.

shows the 10 best ranked features of the LM classifier using the consensus feature ranking for the lags $f_{s_z} = 0, 50, 100$. The best ranked features have been selected from the median of the sensitivity rank matrix. Hence, we do not take the robustness into account, but rather look at the best ranked values across the different permutation runs.

6.4.4 Discussion

The consensus feature ranking analysis showed that selected feature dimensions from the MFCCs and LPCs were the most salient at three different time-scales, namely at 30 ms, 500 ms and 1000 ms for discrimination between the five music genres: classical, hard rock, jazz, pop and techno. From the experiments in

	$f_{s_z} = 0$	$f_{s_z} = 50$	$f_{s_z} = 100$
1	LPC-2	MFCC-0	MFCC-0
2	LPC-1	MFCC-3	MFCC-3
3	MFCC-1	MFCC-5	MFCC-5
4	LPC-3	MFCC-1	MFCC-6
5	MFCC-3	LPC-2	ASE-19
6	LPC-4	MFCC-6	MFCC-1
7	LPC-5	ASE19	LPC-2
8	GAIN	LPC-1	MFCC-12
9	MFCC-0	ASS	ASS
10	MFCC-12	MFCC-9	ZCR

Table 6.6: The 10 best ranked features at the three time-scales: 30 ms, 500 ms and 1000 ms obtained from the LM classifier with a softmax activation function, have been illustrated. They have been obtained from the median value of each row in the sensitivity rank matrix $\mathbf{R}$.

[4, appendix D] , it was further concluded that the DMFCC and DLPC performed poorly at the different time-scales. The MPEG-7 features did not show robust behaviour, and generally ranked lower than the LPCs and MFCCs. The structure of the basis functions displayed cosine behaviour across the temporal dimension. Projection of the test-data into these basis functions will correspond to a filtering of the test-data, picking out dynamics with most energy.

One could argue that the dataset is not very representative given the few genres represented, however, for the present analysis, the good consensus achieved among humans indicate little subjectivity on the labels provided by the author. This makes the dataset relevant for small scale analysis as the one presented above.

6.5 Temporal feature integration for music genre classification

This section will present and discuss results from published work [98, appendix E] , [3, appendix G] and yet unpublished work [99, appendix H] . The DAR and MAR models are suggested for temporal feature integration for music genre classification. Section 6.5.1 provide an overview of the experiments and main conclusions of the work presented in [98, appendix E] . Section 6.5.2 takes the previous work a step further introducing the MAR model for temporal feature integration, extending the investigation with more classifiers, and a more realistic dataset.

6.5.1 Initial time-scale investigations

Inspired from the previous work on ranking short-time features, where temporal feature integration in terms of feature stacking did have a positive effect on the classification accuracy [4, appendix D], it seemed natural to investigate models that include information about the dynamics of the short-time features. Furthermore, it was illustrated how the basis functions found from PCA indeed show cosine behaviour at the time-scales investigated. This enforces methods with bases similar to the cosine transform. In this section we compare previous suggested temporal feature integration methods such as the Filterbank coefficients (FC), mean-variance (MV) against a diagonal AR model (DAR) at preselected time-scales.

The following three time-scales where considered in these experiments:

1. A "short time-scale" with a frame and hop-size of 30 ms and 10 ms, respectively. Instant frequency characteristics.
2. A "medium time-scale" with a frame and hop-size of 740 ms and 370 ms, respectively. Information such as modulations of instruments and the voice can be extracted at this time-scale.
3. And a "long time-scale" with a frame and hop-size of 9.62 sec and 4.82 sec, respectively. Information about beat and long structural correlations in the data.

The MFCCs were selected as short-time features, and the first 6 MFCCs were found adequate for a decent classification accuracy. Furthermore, it must be emphasised that our main interest stems in investigating methods for performing

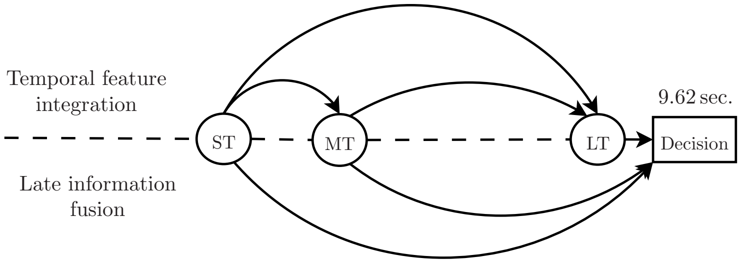


Figure 6.7: Investigated combinations of temporal feature integration and late information fusion techniques. ST, MT and LT denotes short, medium and long time-scale, respectively. Arrows above the dotted line means temporal feature integration, while lines below denotes late information fusion (postprocessing).

temporal feature integration and not as much on the actual performance obtained. The largest decision time horizon considered was approximately 10 sec, corresponding to the long time-scale. To reach a decision at this time-scale, combinations of temporal feature integration and late information fusion can be used. The different combinations investigated are illustrated in Figure 6.7. For temporal feature integration the DAR model, the static mean-variance (MeanVar) and the Filterbank coefficients (FC) were investigated. Figure 6.8 illustrates the features that were extracted at the three time-scales. The parenthesis indicates the dimensionality of the features. The feature named MV_{12a} indicates temporal feature integration of the DAR medium time features to the long time-scale using a MeanVar model. The model order of the DAR was accessed for each of the combinations considered and was determined from resampled cross-validation on the training set[12]. In addition to the features created from temporal feature integration, a few features, explicitly derived for the medium and long time-scales were investigated. These were the LSHZ at the medium time-scale, beat spectrum (BS) and beat histogram (BH) at the long time-scale. The LSHZ is a concatenation of LSTER and HZCRR. These have not been described previously, but integrate information from the STE and ZCR short-time features, respectively[13]. The different combinations were investigated on two datasets (dataset A) and an additional 6 genre dataset consisting of 354 music pieces (described in more detail in appendix E). Both datasets where divided into a fixed training/test set as described in Section 6.3.1. Two classifiers, the GC and LM classifier were applied in the experiments. The classification accuracy of the different combinations has been illustrated in Figure 6.9. The higher dimensional features (DAR_{23a}, DAR_{23m} and MV_{23a} and the combined features at the long time-scale denoted as *All* were projected onto the 20 lead-

[12] We later found that this method is less robust to type-I errors [35].

[13] **HZCRR**(High Zero-Crossing Rate Ratio) : Number of frames whose ZCR are above 1.5 the average. **LSTER**(Low short-time energy ratio) : ratio of number of frames whose STE is less than 0.5 times the average.

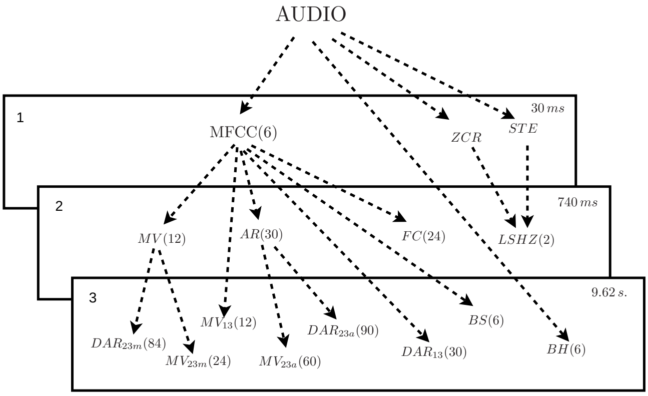


Figure 6.8: Short(1), medium(2) and long(3) time features and their relationships. The arrow from e.g. MV to the long-time feature AR_{23m} indicate temporal feature integration from medium to long time-scale. Thus, for each of the 12 time-series of MV coefficients, a 5th order DAR model is fitted, resulting in a $7 \cdot 12 = 84$ dimensional vector (mean+gain). The model orders of the DAR models have been selected from a validation set. The LSHZ consists of HZCRR and LSTER.

ing principal components of a PCA, to avoid overfitting the data. The number of leading principal components were determined from learning curves using resampled cross-validation.

6.5.1.1 Discussion

The best results obtained were from a three-step procedure[14]: 1) Extract MFCCs, 2) Temporal feature integration using the DAR model, 3) Late information fusion using the sum-rule to reach a final decision at the long time-scale. A McNemar test was applied and it was found that the three-step procedure using the DAR model differed from the MV and FC features on a 1% significance level. The beat spectrum (BS) and beat histogram (BH) features provided individual classification accuracies better than random at the long time-scale, however, no clear indication of a performance increase was observed when combining these with features created from temporal feature integration. This could imply that the temporal information present in the integrated features implicitly holds information about the tempo of the music. One of the larger problems when performing several layers of temporal feature integration is the drastic increase of feature dimensions. PCA was applied for projecting data into a lower dimensional subspace, however, there is no guarantee that the subspace is optimal in sense of music genre. Other projection techniques, such as the canonical corre-

[14] Observed on both datasets.

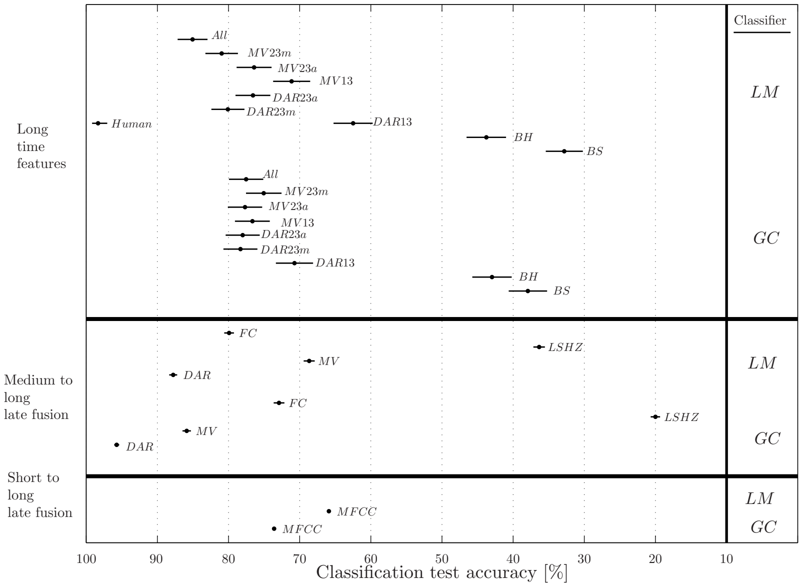


Figure 6.9: The figure illustrates the classification test accuracy for dataset A. The accuracy obtained from the test set is shown for the long decision time horizon of approximately 10 sec. Thus the block “Medium to long late fusion” includes all the medium-time features, such as DAR and the FC features, where the sum-rule has been used to fuse information from the medium to long time-scale. The single “average” human accuracy has been indicated with a 95% binomial confidence interval.

lation analysis (CCA) or partial least squares (PLS) could have been considered as extensions, ensuring a better subspace for discrimination. Also more robust classifiers such as support vector classifier could have been considered.

6.5.2 Extending the model

The performance increase observed when applying the three-step procedure of short-time feature extraction (MFCC), temporal feature integration and late information fusion using the sum-rule were investigated in more detail in [99, appendix H] . Since, the MFCCs are not decorrelated at the time-scales investigated, see Figure 4.4, the multivariate autoregressive model that also has the capability of modelling cross-correlations was suggested. Our hypothesis is that the MAR model is performing better than existing temporal feature integration methods, usually applied in music genre classification.

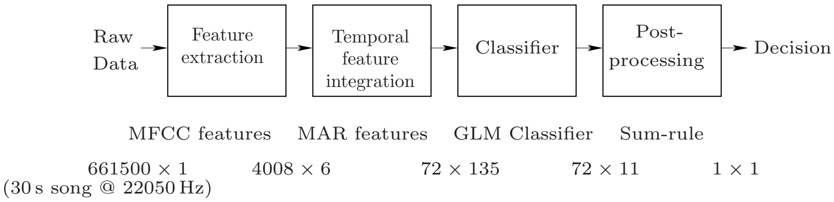


Figure 6.10: The full music genre classification system. The names below the flow-chart are the specific choices which gave the best performing system. The numbers in the bottom part of the figure illustrate the dimensionality reduction which takes place in such a system. Here the number of genres are 11.

6.5.2.1 Experiment description

The system presented as a flow-chart in Figure 6.10 was considered in this investigation. From each music piece the first six MFCCs were extracted. The MeanVar, MeanCov, Filterbank coefficients (FC), DAR and MAR models were considered for temporal feature integration. The LM, GLM, GC and GMM classifier were investigated in this setup. Preliminary investigations of dimensionality reduction of the high-dimensional feature vectors MAR and DAR did not prove beneficial for the GLM and LM classifier, however, it was necessary for the GC and GMM to avoid overfitting. Furthermore, the effect of whitening the data[15] did not show any significant effect w.r.t. classification test accuracy.

The optimisation of the system follows the data stream, which means that the MFCCs were optimised first (frame- hop-size, whether to use normalisation etc.). Afterwards, the temporal feature integration stage was optimised, and so forth. In order to ensure a fair comparison between the different temporal feature integration methods, their optimal hop- and frame-sizes were examined individually, since especially the frame-size is important for the classification accuracy (see Section 4.6). For the MFCCs we found an optimal hop- and frame-size of 7.5 ms and 15 ms, respectively. The optimal hop-size was 400 ms for the DAR, MAR, MeanVar and MeanCov features and 500 ms. for the FC features. The frame-sizes were 1.2 sec for the MAR features, 2.2 sec for the DAR, 1.4 sec for the MeanVar, 2 sec for the MeanCov and 2.4 sec for the FC features. Furthermore, the optimal parameter of the DAR and MAR models were determined to 5 and 3, respectively. The resulting feature dimensions were: MAR-135, DAR-42, FC-42, MeanCov-27 and MeanVar-12.

The above results were obtained using 10-fold cross-validation using the GLM and LM classifiers, since these classifiers are robust towards overfitting.

[15] Zero mean and unit variance of each feature dimension independently.

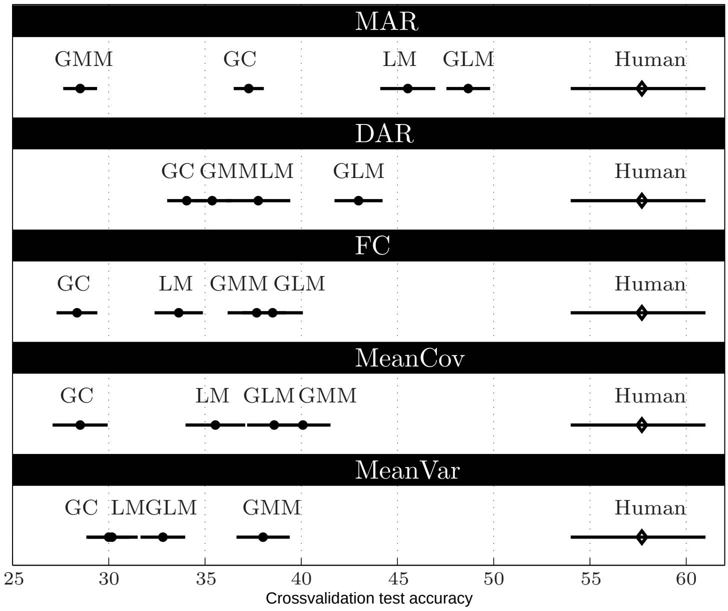


Figure 6.11: The figure show the music genre classification test accuracies for the GC, GMM, LM and GLM classifiers on the five different temporal integrated features. The mean accuracy of 10-fold cross-validation is shown along with error bars which are one ± standard deviation of the mean to each side. 95% binomial confidence intervals have been shown for the human accuracy. Using random guessing, an accuracy of 9.1% would be obtained.

Figure 6.11 shows the achieved classification test accuracies of the different combinations investigated. The mean accuracy of a 10-fold cross-validation with errorbars of ± the standard deviation of the mean value is plotted. Furthermore, the average human classification accuracy obtained from this dataset has been illustrated with its 95% binomial confidence interval. The lower bound on the classification test accuracy is $\approx 9.1\%$ on this dataset. Comparing the MAR features against the other four temporal feature integration schemes using a cross-validation paired t-test gave t-statistics estimates all above 3.90 which is well above the 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 DAR features gave t-statistics of 2.67 and 2.83 for the FC and MeanVar features, but only 1.56 for the MeanCov, hence, the null hypothesis cannot be rejected for the MeanCov features. Table 6.7 illustrates the complexity measured on a framebasis of the investigated temporal feature integration methods. As seen, in exchange for an increased performance, the computational complexity increases.

The confusion matrices obtained by the average 'single' human performance, as

Model	Estimated multiplications & additions
MeanCov	3.21
FC	15.6
DAR	27.2
MAR	32.25

Table 6.7: Calculated complexities from Chapter 4 using the frame-sizes optimal for the individual temporal feature integration methods. The complexities are scaled by the MeanVar complexity. Thus, the complexity of the MeanVar model is 1.

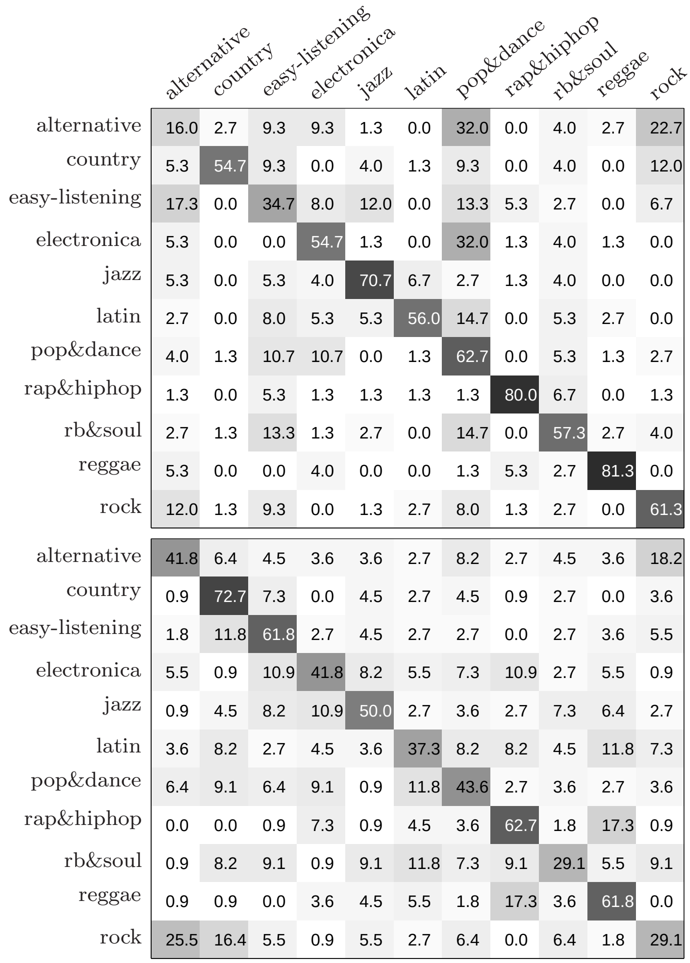
well as best performing combination is illustrated in Figure 6.12. All the classes is well above random performance.

We further investigated the robustness of the best performing method for music genre classification to different encoding qualities. For that purpose we selected the MP3 encoding standard. A small scale experiment was carried out on dataset A (which have not been through a perceptual coder). The best combination scheme for dataset A (MAR with LM classifier) was used. The mono PCM samples were encoded using the `LAME version 3.96.1`[16] into $16, 32, 64, 128$ kbps, respectively. The classification test accuracy was accessed by 10-fold cross validation. In each fold, the test classification accuracy was calculated for each of the 4 encodings considered, as well as PCM. The above experiment was repeated with the different encodings for training. Figure 6.13 shows the different classification accuracies obtained from this analysis. Training with PCM, 128 kbps, 64 kbps or 32 kbps, did not provide a significant performance decrease, whereas a performance decrease was observed when encoding the music in 16 kbps. To make the results comparable, all the music signals were downsampled to 16kHz. This ensures that the Mel-filters are similar for the different sampling rates.

6.5.2.2 MIREX contest on music genre classification

The best combination from the above analysis on dataset B was a MAR model of order 3 extracted from the initial 6 MFCCs over a frame-size corresponding to 1.2 sec. Classification was done using the GLM classifier and a sum-rule was applied to achieve a classification at a decision time of 30 sec. This combination was submitted to MIREX [38] music genre classification task, see [3, appendix G] . Two independent datasets, the 'USpop' and 'Magnatune' dataset were considered. The USpop dataset consists of a training/test set of 940/474 songs distributed unevenly between the 6 genres: country, electronica&dance, newage, rap&hiphop, reggae and rock. The Magnatune dataset consists of a training/test

[16] http://lame.sourceforge.net/



	alternative	country	easy-listening	electronica	jazz	latin	pop&dance	rap&hiphop	rb&soul	reggae	rock
alternative	16.0	2.7	9.3	9.3	1.3	0.0	32.0	0.0	4.0	2.7	22.7
country	5.3	54.7	9.3	0.0	4.0	1.3	9.3	0.0	4.0	0.0	12.0
easy-listening	17.3	0.0	34.7	8.0	12.0	0.0	13.3	5.3	2.7	0.0	6.7
electronica	5.3	0.0	0.0	54.7	1.3	0.0	32.0	1.3	4.0	1.3	0.0
jazz	5.3	0.0	5.3	4.0	70.7	6.7	2.7	1.3	4.0	0.0	0.0
latin	2.7	0.0	8.0	5.3	5.3	56.0	14.7	0.0	5.3	2.7	0.0
pop&dance	4.0	1.3	10.7	10.7	0.0	1.3	62.7	0.0	5.3	1.3	2.7
rap&hiphop	1.3	0.0	5.3	1.3	1.3	1.3	1.3	80.0	6.7	0.0	1.3
rb&soul	2.7	1.3	13.3	1.3	2.7	0.0	14.7	0.0	57.3	2.7	4.0
reggae	5.3	0.0	0.0	4.0	0.0	0.0	1.3	5.3	2.7	81.3	0.0
rock	12.0	1.3	9.3	0.0	1.3	2.7	8.0	1.3	2.7	0.0	61.3

alternative	41.8	6.4	4.5	3.6	3.6	2.7	8.2	2.7	4.5	3.6	18.2
country	0.9	72.7	7.3	0.0	4.5	2.7	4.5	0.9	2.7	0.0	3.6
easy-listening	1.8	11.8	61.8	2.7	4.5	2.7	2.7	0.0	2.7	3.6	5.5
electronica	5.5	0.9	10.9	41.8	8.2	5.5	7.3	10.9	2.7	5.5	0.9
jazz	0.9	4.5	8.2	10.9	50.0	2.7	3.6	2.7	7.3	6.4	2.7
latin	3.6	8.2	2.7	4.5	3.6	37.3	8.2	8.2	4.5	11.8	7.3
pop&dance	6.4	9.1	6.4	9.1	0.9	11.8	43.6	2.7	3.6	2.7	3.6
rap&hiphop	0.0	0.0	0.9	7.3	0.9	4.5	3.6	62.7	1.8	17.3	0.9
rb&soul	0.9	8.2	9.1	0.9	9.1	11.8	7.3	9.1	29.1	5.5	9.1
reggae	0.9	0.9	0.0	3.6	4.5	5.5	1.8	17.3	3.6	61.8	0.0
rock	25.5	16.4	5.5	0.9	5.5	2.7	6.4	0.0	6.4	1.8	29.1

Figure 6.12: 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.

set of 1005/510 distributed unevenly among the 10 music genres: ambient, blues, classical, electronic, ethnic, folk, jazz, newage, punk and rock. It should be noted that no prior datasets were released, and that the evaluations were performed exclusively by the MIREX team in a dedicated software environment. See [38] for technical details.

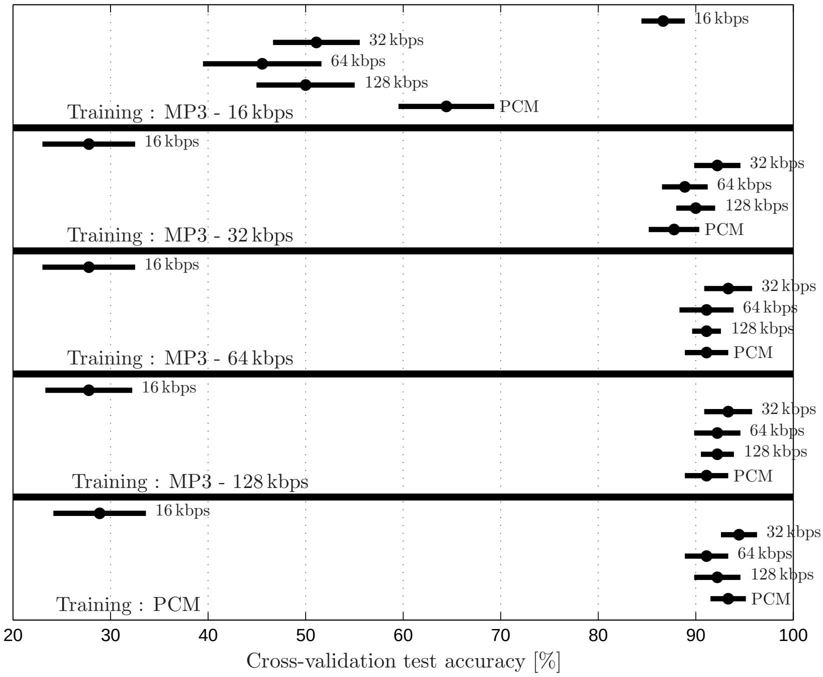


Figure 6.13: The figure illustrate the test classification accuracy of the best performing setup(MAR and LM) on dataset A when encoding the music snippets in the following audio formats : PCM (Microsoft wave), MP3-128 kbps, MP3-64 kbps, MP3-32 kbps and MP3-16 kbps. The top illustration, shows a training with MP3-16 kbps. The mean cross-validation accuracy and standard deviation of the mean value have been calculated from the 10 folds.

The unnormalised classification test accuracy of the two datasets as well as the combined accuracy on the two datasets has been illustrated in Figure 6.14 for the contributions of the different researchers. A 95% binomial confidence interval has been included together with the test accuracy to indicate the uncertainty on the specific dataset. The best performing method was suggested by [13] which considered a total of 402 short-time features consisting mostly of spectral features and included among other features the MFCCs and ZCR. Temporal feature integration using the MeanVar model were calculated over frames of short-time features of 13.9 sec. Using an Adaboost classifier an overall classification accuracy of $\sim$ 82% was obtained. The second best method was suggested in [92], which used the initial 20 MFCCs as short-time features. Temporal feature integration over the short-time features was performed a Gaussian model (MeanCov) for the entire song. A support vector classifier was applied using the symmetric KL-divergence kernel that was discussed in Chapter 5, resulting in an overall performance on the two datasets of $\sim$ 79%.

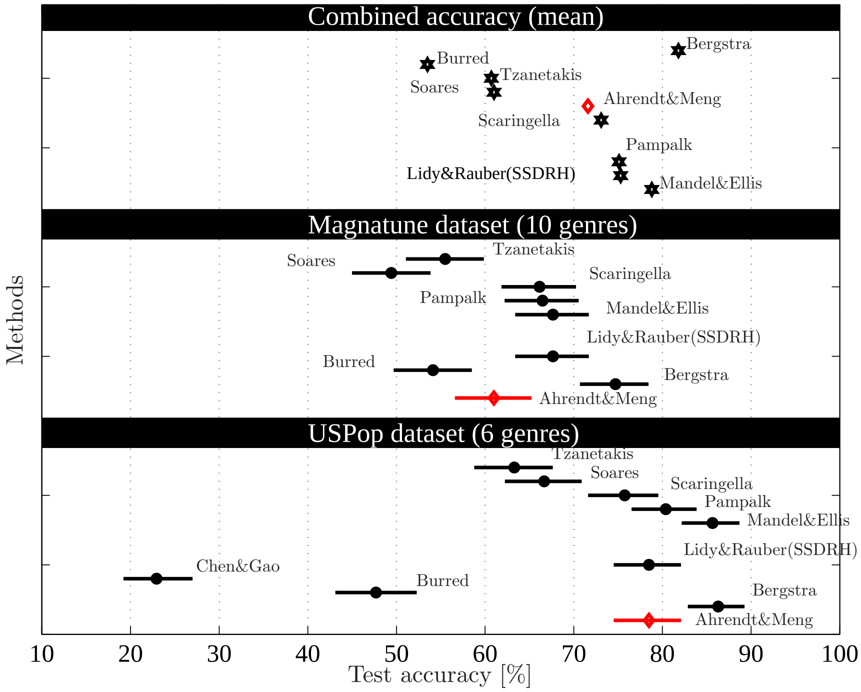


Figure 6.14: The classification accuracies obtained on the two benchmark datasets, the Magnatune and USpop dataset as well as the combined mean accuracy illustrated under the bar 'Combined accuracy (mean)'. A 95% binomial confidence interval has been added to the mean accuracies obtained on the individual datasets. The method prepared by the author is denoted 'Ahrendt&Meng'.

Without optimisation of the model order of the MAR-model, frame-size optimisation and optimisation of regularisation parameter of the classifier an accuracy of 72 % was achieved. Comparing with the other approaches it must be emphasised that only the initial 6 MFCCs were used. Comparing the confusion matrices on the Magnatune dataset of the algorithms of Bergstra, Mandel and Ahrendt&Meng (our method) several differences were found. For example, an accuracy of 11%[17] was achieved on the genre ambient by Mandel whereas Bergstra and Ahrendt&Meng obtained accuracies of 76% and 59%, respectively. Another example is the genre punk that had an accuracy of ∼ 85% with the Bergstra and Mandel algorithms, where Ahrendt&Meng had an accuracy of 97%. These examples illustrates that combining the different learning methods most likely will boost the overall performance even further.

[17] Which is pure chance

The method proposed by Mandel won the artist identification task, and will be compared with the product probability kernel using both the Gaussian model, GMM and the MAR model in a music genre classification setup in section 6.6

6.5.2.3 Discussion

This section presented results of work presented in [98, appendix E], [99, appendix H] and [3, appendix G]. A comparison was performed on existing temporal feature integration models, the MeanVar, MeanCov and FC with the proposed MAR and DAR models using different classifiers. The different parameters of the system were optimised to provide a basis for comparison between the different methods. The MAR feature has an overall larger computational complexity, however, also better accuracy on the investigated datasets. With the best combination: a MAR model of order 3, classification using a GLM classifier and finally late information fusion using the sum-rule to reach a decision at 30 sec, a mean test accuracy on dataset B of $\sim 48\%$ was obtained. The average human performance on this dataset was $\sim 57\%$. The dataset is believed to generalise well due to its high average song-per-artist complexity (~ 2 songs per artist). It was further noticed that the DAR features only performed 5% worser than the MAR features on this dataset, which indicate that some of the cross-correlations provide little or no information.
Comparing the confusion matrices of the average human performance and the confusion matrix obtained from the MAR combination, it was observed that the genres that humans often classify correctly, i.e. country, rap& hiphop and reggae are the same genres that our system typically classifies correctly.
The results from the MIREX contest illustrated that the MAR model in the prescribed combination achieved comparable or better results than some of the other contributions. Again, it should be emphasised that the model was not optimised to the specific datasets, which indicates that the setup is rather robust to different datasets and is expected to generalise well.

Using the MAR model in the three-step procedure results in a 135 dimensional feature vector for each 1.2 sec of music. This is a rather hard compression considering the original size of a song.
A further compression is possible without cutting the mean accuracy, which will be the topic of Section 6.6. Selected kernel aided temporal feature integration methods, which have been discussed in Chapter 5, are investigated on dataset B for music genre classification.

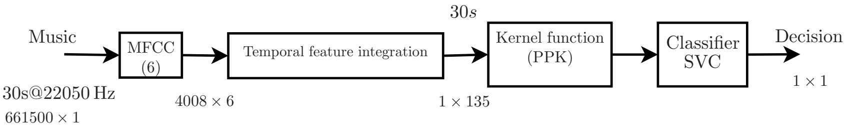


Figure 6.15: Flow-chart of kernel aided temporal feature integration. The numbers below the flow-chart indicate the dimensionality reduction of a single music piece which takes place. The numbers above the flow-chart indicate the time-scale of the feature.

6.6 Kernel methods for music genre classification

Selected methods from the work in [100, appendix F] are put in a context comparable to the results of sub-section 6.5.2. The reader is encouraged to read [100, appendix F] before continuing this sub-section.

The experiments in this section are reported for dataset B using cross-validation to access the classification accuracy. The investigations in [100, appendix F] considered different approaches to obtain a decision for a music snippet at 30 sec. This section have focused on selected temporal feature integration methods for creating feature sets at a song time-scale. Three different models, the GM, MAR and GMM model for temporal feature integration have been considered in combination with PPK and RBF kernel (only GM and MAR). Furthermore, the KL-divergence kernel with a Gaussian model have been included due to its good performance at this years MIREX [38]. Figure 6.15 shows the flow-chart of the setup. As indicated, the only classifier applied in this investigation is the support vector classifier.

The numbers below the flow-chart indicate the dimensionality reduction taking place with one of the best performing methods, a MAR model of order 3 in a PPK.
As in the previous sub-section, the MFCCs have been extracted using a frame-size of 15 ms and a hop-size of 7.5 ms. Again, the first 6 MFCCs have been applied in these experiments. The optimal model order for the MAR model was determined to 3 using cross-validation. Also, the single best model order was determined to 3 using full covariance matrices for the Gaussian mixture model. Different values of $\rho = 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}$ were investigated. There was a small drop in accuracy for $\rho = 1$, otherwise, the classification test accuracy was constant for $\rho = 1/2, 1/4, 1/8$ and $\rho = 1/16$. A value of $\rho = 1/2$ was selected due to the automatic normalisation of the kernel. A similar investigation was conducted for the KL-divergence kernel where an optimal value of $\gamma = 1/4$ was found. The nuisance parameter C of the support vector classifier is selected

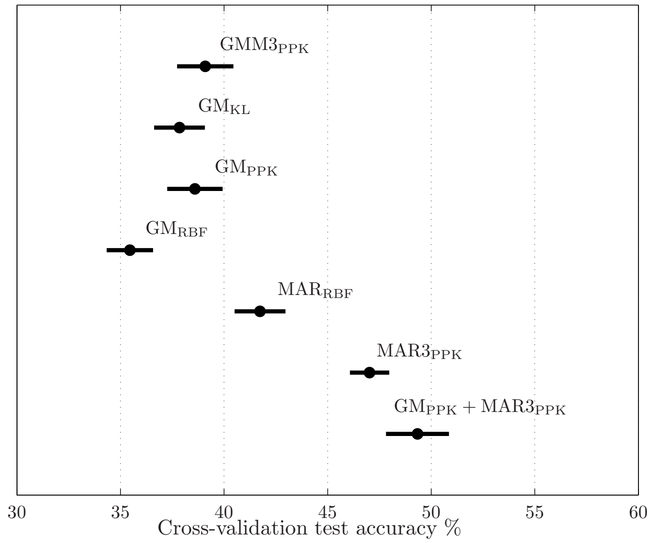
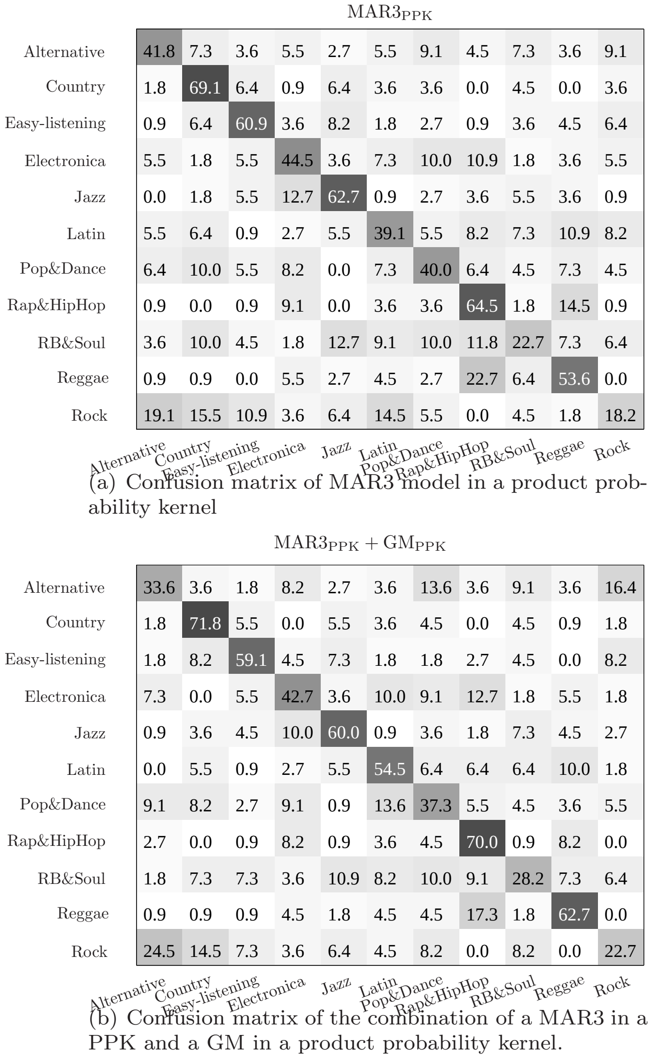


Figure 6.16: The figure shows the mean cross-validated classification accuracy on dataset B from selected kernel approaches from [100, appendix F]. The mean accuracy of 10-fold cross-validation is shown along with error bars, which are one ± standard deviation of the mean to each side. The short-time features (6-MFCCs) of each music piece 30 sec is modelled either by the Gaussian model, MAR or GMM with 3 mixtures, after which different kernels have been applied and investigated with a SVC. The lowercase indicate kernel type. Two kernels, the GM$_{\text{PPK}}$ and MAR3$_{\text{PPK}}$ where added to produce a new kernel GM$_{\text{PPK}}$ + MAR3$_{\text{PPK}}$, which provided a small increase in performance. For all experiments involving the product probability kernel, $\rho = 1/2$.

from cross-validation on the training data from each of the 10 folds.

Figure 6.16 illustrate the mean classification test accuracy obtained from 10-fold cross validation. The mean value ± the standard deviation of the mean has been plotted. It is observed that the temporal information in the lower 6 MFCCs has a big influence on the music genre classification task. An increase of approximately 8% is achieved by using the MAR model in a PPK compared to the mean accuracy obtained using a Gaussian model in a PPK. Furthermore, a performance increase is clearly observed when applying the product probability kernel instead of the RBF kernel on the MAR or GM model parameters. For the MAR model a performance increase of approximately 5% is achieved. It should be noted that there is no statistical difference in performance whether one is using the symmetric KL-divergence kernel or a PPK with a Gaussian model on this dataset. Adding the different kernels can provide a boost in performance, which has been indicated by the adding of the GM$_{\text{PPK}}$ and MAR3$_{\text{PPK}}$ without any weighting (hence they have not been combined in an optimal manner). This combination provided a small performance boost, but it is believed that combining kernels in a proper manner, which individually expresses good performance in different music genres, will boost the performance even further. This issue, however, have not been considered further in this work.

The confusion matrices of the two best performing kernels have been illustrated in Figure 6.17. Comparing the confusion matrices of the MAR3 model in a



$\mathrm{MAR3}_{\mathrm{PPK}}$

	Alternative	Country	Easy-listening	Electronica	Jazz	Latin	Pop&Dance	Rap&HipHop	RB&Soul	Reggae	Rock
Alternative	41.8	7.3	3.6	5.5	2.7	5.5	9.1	4.5	7.3	3.6	9.1
Country	1.8	69.1	6.4	0.9	6.4	3.6	3.6	0.0	4.5	0.0	3.6
Easy-listening	0.9	6.4	60.9	3.6	8.2	1.8	2.7	0.9	3.6	4.5	6.4
Electronica	5.5	1.8	5.5	44.5	3.6	7.3	10.0	10.9	1.8	3.6	5.5
Jazz	0.0	1.8	5.5	12.7	62.7	0.9	2.7	3.6	5.5	3.6	0.9
Latin	5.5	6.4	0.9	2.7	5.5	39.1	5.5	8.2	7.3	10.9	8.2
Pop&Dance	6.4	10.0	5.5	8.2	0.0	7.3	40.0	6.4	4.5	7.3	4.5
Rap&HipHop	0.9	0.0	0.9	9.1	0.0	3.6	3.6	64.5	1.8	14.5	0.9
RB&Soul	3.6	10.0	4.5	1.8	12.7	9.1	10.0	11.8	22.7	7.3	6.4
Reggae	0.9	0.9	0.0	5.5	2.7	4.5	2.7	22.7	6.4	53.6	0.0
Rock	19.1	15.5	10.9	3.6	6.4	14.5	5.5	0.0	4.5	1.8	18.2

(a) Confusion matrix of MAR3 model in a product probability kernel

$\mathrm{MAR3}_{\mathrm{PPK}} + \mathrm{GM}_{\mathrm{PPK}}$

	Alternative	Country	Easy-listening	Electronica	Jazz	Latin	Pop&Dance	Rap&HipHop	RB&Soul	Reggae	Rock
Alternative	33.6	3.6	1.8	8.2	2.7	3.6	13.6	3.6	9.1	3.6	16.4
Country	1.8	71.8	5.5	0.0	5.5	3.6	4.5	0.0	4.5	0.9	1.8
Easy-listening	1.8	8.2	59.1	4.5	7.3	1.8	1.8	2.7	4.5	0.0	8.2
Electronica	7.3	0.0	5.5	42.7	3.6	10.0	9.1	12.7	1.8	5.5	1.8
Jazz	0.9	3.6	4.5	10.0	60.0	0.9	3.6	1.8	7.3	4.5	2.7
Latin	0.0	5.5	0.9	2.7	5.5	54.5	6.4	6.4	6.4	10.0	1.8
Pop&Dance	9.1	8.2	2.7	9.1	0.9	13.6	37.3	5.5	4.5	3.6	5.5
Rap&HipHop	2.7	0.0	0.9	8.2	0.9	3.6	4.5	70.0	0.9	8.2	0.0
RB&Soul	1.8	7.3	7.3	3.6	10.9	8.2	10.0	9.1	28.2	7.3	6.4
Reggae	0.9	0.9	0.9	4.5	1.8	4.5	4.5	17.3	1.8	62.7	0.0
Rock	24.5	14.5	7.3	3.6	6.4	4.5	8.2	0.0	8.2	0.0	22.7

(b) Confusion matrix of the combination of a MAR3 in a PPK and a GM in a product probability kernel.

Figure 6.17: The figure shows the average of the confusion matrices over the 10 cross-validation runs of the two best performing kernel methods $\mathrm{MAR3}_{\mathrm{PPK}}$ and the combination of the $\mathrm{GM}_{\mathrm{PPK}}$ and $\mathrm{MAR3}_{\mathrm{PPK}}$. The latter kernel have only been included to illustrate that if the kernels produce diverse results their combinations can provide increased performance.

PPK with that of figure 6.12, there is good agreement. Comparing the MAR in a PPK and the combined kernel shows an improvement in the two genres latin and reggae.

6.6.1 Discussion

Selected experiments from [100, appendix F] were put in a context which made them comparable to those of the previous section. Special emphasis was placed on illustrating the performance obtained using the product probability kernel with the three temporal feature integration models the GM, GMM and MAR model. Furthermore, the Gaussian model in a symmetric KL-divergence kernel, which won the MIREX artist identification task were compared with the PPK using the same density model. Using a cross-validation paired t-test the null hypothesis that the two models are equal could not be rejected. The obtained mean test accuracy obtained with the MAR model of order 3 in a product kernel was comparable with the mean accuracy of a MAR model in combination with a GLM classifier and late information using the sum-rule.

Regarding memory requirements, the effect of retaining only a single vector per song compresses the large database. For the larger dataset, which consist of 1210 songs we only need to store 135×1210 datapoints using a MAR model of order 3. This corresponds to a compression factor of 156800 when compared to the original music pieces. Training can be done using a support vector classifier, or e.g. a K-nearest neighbour classifier, since the kernel matrix can be converted to a distance measure.

6.7 Discussion

This chapter introduced the different classifiers, which were applied in the experiments, presented the investigated datasets and discussed methods for comparing learning algorithms on the given datasets. Furthermore, it was argued that especially dataset B was a good representation of music genre since the large amount of different artists led to as few as ~ 2 songs per artist on the average. The ground truth on the datasets provided by the authors and an external reference, were accessed by a human evaluation on selected parts of the datasets.

Three experiments were described: 1) feature ranking of short-time features at different time-scales, 2) temporal feature integration for music genre classification where the multivariate AR model was investigated in detail, and finally 3) kernel aided temporal feature integration for music genre classification investigating different high-level kernels.

The consensus sensitivity ranking approach showed that the MFCC short-time features generally ranked better at three time-scales: 30 ms, 500 ms and 1000 ms. The eigenvectors of the stacked short-time features indicated cosine like basis-functions, which motivates spectral methods for modelling the dynamical structure of short-time features. An experiments, involving features extracted at a medium and long time-scale illustrated that the best overall classification accuracy in a music genre eksperiment was obtained from a three-step procedure consisting of:

1. Extraction of short-time features (MFCCs)
2. Temporal feature integration to an intermediate time-scale (Medium time-scale)
3. Late information fusion using e.g. the sum-rule

Using the three-step procedure with the multivariate AR model for temporal feature integration of the first 6 MFCCs across a frame of 1.2sec on dataset B, resulted in a mean cross-validation accuracy of 48% ± 2% using the GLM classifier and sum-rule. This result was statistical significant in comparison to the other temporal feature integration methods.
The final results of the kernel aided temporal feature integration illustrated that a similar performance could be obtained using the temporal information of the initial 6 MFCCs. A MAR model of order 3 was created for each 30 sec music snippet, and a kernel matrix was created using the product probability kernel. A cross-validated accuracy of 47% ± 1% was obtained using this kernel matrix with a support vector classifier on dataset B for music genre classification. A 95% binomial confidence interval for the 'average' single human on dataset B was estimated to [54% 57% 61%].

Chapter 7

Conclusion

The field of music information retrieval is expanding, which is an inevitable process due to the large digitalisation of all kinds of information, and especially music. The increasing interest from researchers will spawn new ideas in the coming future, where this thesis is no exception. In this thesis, we have provided several methods that have shown to be useful for unveiling music organisation, and therefore can be of great importance for music retrieval systems. This includes a method for ranking short-time features at larger time-scales, a framework, which we denoted as temporal feature integration, where existing methods were compared with the proposed MAR model. Moreover, kernel functions, which directly or indirectly aid temporal feature integration were suggested for music organisation.

Ranking of short-time features for music organisation
A method that we denoted 'consensus sensitivity ranking' was suggested for ranking of short-time features at larger time-scales. Temporal feature integration by stacking was applied to include temporal information of the short-time features. Two classifiers, the Gaussian and the linear model classifier, were applied in an investigation on a 5 genre music genre classification setup. Short-time features such as the MFCC, LPC, STE, MPEG-7: ASE, ASC, ASS, ASF and ZCR were carefully analysed. The features that had the best overall consensus ranking at the three time-scales: 30 ms, 500 ms and 1000 ms, mainly consisted of the lower order MFCCs.

Temporal feature integration
A general introduction to temporal feature integration was provided, and various existing methods such as the mean-variance, mean-covariance, and filterbank

coefficient approach were described. The general multivariate autoregressive model (MAR) was proposed for temporal feature integration. It was illustrated that this model captures the local dynamics of the short-time features.

A three-step procedure was found optimal for music genre classification, which consists of:

1. Extract short-time features (MFCC)
2. Temporal feature integration to an intermediate time-scale (1.2 sec was optimal for the MAR model on dataset B).
3. Perform late information fusion using e.g. the sum-rule.

A thorough evaluation of the MAR (and DAR) model on dataset B in a music genre classification setup was provided, comparing it with existing temporal feature integration methods using several classifiers carefully optimised to provide a standard of reference. Using a cross-validated paired t-test, it was shown that the MAR model in the three step procedure, significantly outperformed competing temporal integration schemes. To measure the subjectivity of the applied datasets a human evaluation was conducted. A 95% binomial confidence interval for an average "single" human on dataset B was [54% 57% 61%], where the best mean cross-validated accuracy obtained from the three-step procedure was 48% ± 2%. Moreover, the MIREX contest showed robust behaviour of the three-step procedure without any optimisation of the system.

Kernel aided temporal feature integration
An overview of different existing kernel functions, which handles sequences of short-time features denoted as high-level kernels, was provided. The product probability kernel and convolutive kernel were investigated for music genre classification. A Gaussian model, Gaussian mixture model, and a MAR model were investigated in a product probability kernel. A closed form solution of the MAR model in a product probability kernel was derived. Experiments on dataset B illustrated that a significant increase in accuracy was obtained when modelling a single song with a MAR model in a product probability kernel. A mean cross-validation test accuracy of 47% ± 1% was obtained, close to that of the three-step procedure. Using this approach, we are only required to store a 135 dimensional feature vector per song.

Suggestions for further work
Currently a fixed frame-size is used when performing temporal feature integration. However, from the varying stationarity of a music signal, it seems more appropriate to impose variable frame length, such that local stationarities can

be modelled more efficiently. A method considered (not implemented) is to apply several parallel adaptive filters using different forgetting factors (what is equivalent to using different frame-sizes) and select the filter with the best modelling capabilities using an intelligent selection algorithm. With this approach, interesting audio cues would achieve more attention.
Another issue which would be interesting to investigate would be the modelling of the different short-time feature dimensions using different density models. E.g. cross-correlations were found to be more pronounced for the lower MFCCs than the higher coefficients. Thus, by using appropriate density models one could cut the dimensions of the final feature vector and likely improve generalisation.

It was hypothesized that the product probability kernel could handle different model orders, which were partly investigated in [100, appendix F]. This idea is interesting since the level of detail required will differ from song to song.

As a final remark, it should be noted that the methods presented in this work are of great benefit for both consumers and companies for navigation and retrieval of music in large databases and constitutes a range of business opportunities in the near future.

Appendix A

Derivation of kernel function for the multivariate AR-model

In this appendix a closed form derivation of the product probability kernel for a multivariate autoregressive model is provided. The parameters of the AR model is considered known a priory. The normal multivariate AR-model is repeated here for completeness,

$$\mathbf{z}_k = \sum_{p=1}^{P} \mathbf{A}_p \mathbf{z}_{k-p} + \mathbf{v} + \mathbf{u}, \tag{A.1}$$

where $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$ and $\mathbf{v}$ is related to the mean value of the time-series by $\mathbf{m} = (\mathbf{I} - \sum_{p=1}^{P} \mathbf{A}_p)^{-1}\mathbf{v}$ and P is the model order. For simplicity, we will write the model in its mean adjusted form ($\mathbf{z}_k^0 = \mathbf{z}_k - \mathbf{m}$), thus

$$\begin{aligned}
\mathbf{z}_k &= \sum_{p=1}^{P} \mathbf{A}_p \mathbf{z}_{k-p} + \mathbf{v} + \mathbf{u} \\
\mathbf{z}_k &= \sum_{p=1}^{P} \mathbf{A}_p \mathbf{z}_{k-p} + \left(\mathbf{I} - \sum_{p=1}^{P} \mathbf{A}_p \right) \mathbf{m} + \mathbf{u} \\
\mathbf{z}_k - \mathbf{m} &= \sum_{p=1}^{P} \mathbf{A}_p (\mathbf{z}_{k-p} - \mathbf{m}) + \mathbf{u} \\
\mathbf{z}_k^0 &= \sum_{p=1}^{P} \mathbf{A}_p \mathbf{z}_{k-p}^0 + \mathbf{u}. \qquad (A.2)
\end{aligned}$$

We are interested in calculating the joint distribution between the correlated variables, denoted as $p(\mathbf{z}_0^0, \mathbf{z}_1^0, \ldots, \mathbf{z}_{f_{s_z}}^0 | \boldsymbol{\theta})$, however, since we are using a finite order MAR model it is adequate to consider only the initial $P+1$ variables, since the density model repeats. Thus, we only consider the following joint distribution

$$p(\mathbf{z}_0^0, \mathbf{z}_1^0, \ldots, \mathbf{z}_P^0) = p(\mathbf{z}_0^0) p(\mathbf{z}_1^0 | \mathbf{z}_0^0) \ldots p(\mathbf{z}_p^0 | \mathbf{z}_{P-1}^0, \ldots, \mathbf{z}_0^0). \qquad (A.3)$$

Each of the above densities is Gaussian, due to noise assumption on $\mathbf{u}$. Thus,

$$\begin{aligned}
p(\mathbf{z}_0^0) &= \mathcal{N}(\mathbf{0}, \mathbf{C}) \\
p(\mathbf{z}_1^0 | \mathbf{z}_0^0) &= \mathcal{N}(\mathbf{A}_1 \mathbf{z}_0, \mathbf{C}) \\
&\vdots \\
p(\mathbf{z}_P^0 | \mathbf{z}_{P-1}, \ldots, \mathbf{z}_0^0) &= \mathcal{N}\left(\sum_{p=1}^{P} \mathbf{A}_p \mathbf{z}_{P-p}^0, \mathbf{C} \right). \qquad (A.4)
\end{aligned}$$

Stacking the variables of $\mathbf{z}^0$ into a single vector, simplifies further calculation:

$$\hat{\mathbf{z}} = \begin{bmatrix} \mathbf{z}_0^0 \\ \mathbf{z}_1^0 \\ \vdots \\ \mathbf{z}_P^0 \end{bmatrix}, \qquad (A.5)$$

since the operation allows the densities to be written compactly as

$$\begin{aligned}
p(\mathbf{z}_0^0) &= c_0 \cdot \exp\left\{-\frac{1}{2}\left(\hat{\mathbf{A}}_0\hat{\mathbf{z}}\right)^T \mathbf{C}^{-1}\left(\hat{\mathbf{A}}_0\hat{\mathbf{z}}\right)\right\} \\
p(\mathbf{z}_1^0|\mathbf{z}_0^0) &= c_0 \cdot \exp\left\{-\frac{1}{2}\left(\hat{\mathbf{A}}_1\hat{\mathbf{z}}\right)^T \mathbf{C}^{-1}\left(\hat{\mathbf{A}}_1\hat{\mathbf{z}}\right)\right\} \\
&\vdots \\
p(\mathbf{z}_P^0|\mathbf{z}_{P-1}^0,\dots,\mathbf{z}_0^0) &= c_0 \cdot \exp\left\{-\frac{1}{2}\left(\hat{\mathbf{A}}_P\hat{\mathbf{z}}\right)^T \mathbf{C}^{-1}\left(\hat{\mathbf{A}}_P\hat{\mathbf{z}}\right)\right\}. \qquad \text{(A.6)}
\end{aligned}$$

where $c_0 = \frac{|\mathbf{C}|^{-1/2}}{(2\pi)^{D/2}}$ and

$$\begin{array}{lcllllll}
\hat{\mathbf{A}}_0 & = & [& \mathbf{I} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \quad] \\
\hat{\mathbf{A}}_1 & = & [& -\mathbf{A}_1 & \mathbf{I} & \mathbf{0} & \dots & \mathbf{0} \quad] \\
\vdots & = & & & & \vdots & & \\
\hat{\mathbf{A}}_P & = & [& -\mathbf{A}_1 & -\mathbf{A}_2 & \dots & -\mathbf{A}_P & \mathbf{I} \quad].
\end{array} \qquad \text{(A.7)}$$

Multiplying the distributions of Equation A.6 together as in Equation A.3, the following result is obtained:

$$p(\hat{\mathbf{z}}|\boldsymbol{\theta}) = c_0^{P+1} \exp\left\{-\frac{1}{2}\hat{\mathbf{z}}^T\left(\sum_{i=0}^{P}\hat{\mathbf{A}}_i^T\mathbf{C}^{-1}\hat{\mathbf{A}}_i\right)\hat{\mathbf{z}}\right\}. \qquad \text{(A.8)}$$

Setting $\mathbf{M} = \sum_{i=0}^{P}\hat{\mathbf{A}}_i^T\mathbf{C}^{-1}\hat{\mathbf{A}}_i$, the following density model is obtained:

$$p(\hat{\mathbf{z}}|\boldsymbol{\theta}) \sim |\mathbf{C}|^{-(P+1)}|\mathbf{M}|^{-1/2}\mathcal{N}(\mathbf{0},\mathbf{M}^{-1}). \qquad \text{(A.9)}$$

Using rule 8.82 in [112] on block-matrices, see Equation A.10

$$\left|\left[\begin{array}{cc} \mathbf{A}_{m\times m} & \mathbf{B}_{m\times n} \\ \mathbf{C}_{n\times m} & \mathbf{D}_{n\times n} \end{array}\right]\right| = |\mathbf{A}_{m\times m}| \cdot |\mathbf{C2}_{n\times n}|, \qquad \text{(A.10)}$$

where $\mathbf{C2} = \mathbf{D}_{n\times n} - \mathbf{C}_{n\times m}\mathbf{A}_{m\times m}^{-1}\mathbf{B}_{m\times n}$,

it is possible to show that

$$|\mathbf{M}|^{-1/2} \cdot |\mathbf{C}|^{-(P+1)} = 1, \qquad \text{(A.11)}$$

hence

$$p(\hat{\mathbf{z}}|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0},\mathbf{M}^{-1}). \qquad \text{(A.12)}$$

The product probability kernel between two multivariate Gaussian models, can then be calculated simply as (using Equation 5.9 and Equation A.12)

$$\kappa(\boldsymbol{\theta}, \boldsymbol{\theta}') = (2\pi)^{(1-2\rho)(P+1)D/2} \rho^{-(P+1)D/2} |\mathbf{M}^{\dagger}|^{1/2} |\mathbf{M}|^{\rho/2} |\mathbf{M}'|^{\rho/2}, \qquad \text{(A.13)}$$

where $\mathbf{M}^{\dagger} = \mathbf{M} + \mathbf{M}'$. Applying Equation A.11 reduces the kernel function to

$$\kappa(\boldsymbol{\theta}, \boldsymbol{\theta}') = (2\pi)^{(1-2\rho)(P+1)D/2} \rho^{-(P+1)D/2} \\ |\mathbf{M} + \mathbf{M}'|^{-1/2} |\mathbf{C}|^{-\rho(P+1)/2} |\mathbf{C}'|^{-\rho(P+1)/2}. \qquad \text{(A.14)}$$

The complexity of this evaluation lies in the determinant of the matrix $\mathbf{M} + \mathbf{M}'$, which is a square symmetric matrix of size $D(P+1)$, hence the complexity becomes $\mathcal{O}\left((D(P+1))^3\right)$. It is also straightforward to implement the MAR model in a symmetric KL-divergence kernel, using the expression in Equation A.12.

A.1 Memorandum

A similar approach was considered in [147]. The author investigated how correlated variables can be considered independent by difference observations. The same results of the inverse covariance matrix $\mathbf{M}$ was reported. However, since cyclic boundary conditions is assumed the inverse covariance matrix $\mathbf{M}$ becomes singular.

Appendix B

Simple PCA

This appendix will introduce the 'simple' PCA, which has been applied in [4, appendix D] . The method to be presented draw inspiration from the ideas presented in [129]. The general idea is to sample the data retaining only a small percentage of the original data, which is believed to be representative of the original data space. If there is a lot of redundancy in the data, decimation, using random sampling techniques is adequate. Using a more intelligent sampling method, which selects the most informative data vectors would be an extension to the applied method, see e.g. [52]. Consider a sequence of stacked features[1] using a frame-size (lag) of f_{s_z}. Then each music snippet "i" can be represented as a matrix

$$\mathbf{Z}^{(i)} = \begin{bmatrix} \mathbf{z}_1^{(i)} & \mathbf{z}_2^{(i)} & \dots & \mathbf{z}_{\tilde{K}^{(i)}}^{(i)} \end{bmatrix}_{Df_{s_z} \times \tilde{K}^{(i)}}, \tag{B.1}$$

for $i = 0, \dots, N-1$, where N represents number of music snippets in the dataset and $\tilde{K}^{(i)}$ represent the number of stacked features in each music snippet i.
In the following derivation the music snippets are assumed to have the same length, however, this is not a requirement.

Randomly select r feature vectors from each $\mathbf{Z}^{(i)}$, for $i = 0, \dots, N-1$ and construct the matrix

$$\mathbf{Z}_r = \left[\mathbf{Z}_r^{(0)}, \dots, \mathbf{Z}_r^{(N-1)}\right], \tag{B.2}$$

which is of dimension $Df_{s_z} \times rN$. In the following we assume that $Df_{s_z} >> rN$. Form the smaller of the two biased covariance matrices $\mathbf{Z}_r^T \mathbf{Z}_r$ of dimension

[1] The method works for general feature sequences, hence, it is not required that the features are stacked.

$rN \times rN$, and perform an eigenvalue decomposition, thus

$$\mathbf{Z}_r^T \mathbf{Z}_r = \mathbf{U}_r \Lambda_r \mathbf{U}_r^T, \tag{B.3}$$

where Λ_r is a square symmetric matrix with ordered eigenvalues, of size rN. $\mathbf{U}_r$ contains the corresponding eigenvectors.
Next, calculate the outer product eigenvectors of $\mathbf{Z}_r \mathbf{Z}_r^T$ using the relationship between inner and outer products from the singular value decomposition (SVD, economy description). Furthermore, we only retain the $p \leq rN$ eigenvectors with largest eigenvalues, hence

$$\mathbf{Z}_r = \mathbf{V}_p \mathbf{S}_p \mathbf{U}_p^T \quad \rightarrow \quad \mathbf{V}_\mathrm{p} = \mathbf{Z}_\mathrm{r} \mathbf{U}_\mathrm{p} \mathbf{S}_\mathrm{p}^{-1} \tag{B.4}$$

where $\mathbf{V}_p$ is of dimension $Df_{s_z} \times p$, $\mathbf{S}_p$ is the singular values of dimension $p \times p$, and $\mathbf{U}_p$ is of dimension $rN \times p$. For stability reasons, $\mathbf{V}_p$ is obtained by calculating $\mathbf{Z}_r \mathbf{U}_p$ and normalising the columns of $\mathbf{V}_p$ such that $||\mathbf{v}_i||_2 = 1$, for $i = 1, \ldots, p$.

Retaining only the p largest eigenvalue/eigenvector pairs, one finds that the eigenvectors is an approximation to the corresponding p largest eigenvector/eigenvalue pair of the complete matrix

$$\mathbf{Z}\mathbf{Z}^T = \mathbf{V}\Lambda\mathbf{V}^T \tag{B.5}$$

which is of dimension $Df_{s_z} \times Df_{s_z}$, hence $\mathbf{V}_p \Lambda_r \mathbf{V}_p^T \approx \mathbf{Z}\mathbf{Z}^T$.

Calculating the eigenvalue decomposition of the $rN \times rN$ matrix instead of the $p \times p$ matrix results in better approximations of the eigenvalue/vector pairs when $rN > p$, see [129].

New data $\mathbf{Z}_\mathrm{test}(Df_{s_z} \times N_\mathrm{test})$ can be projected into the p leading eigenvectors as

$$\mathbf{Z}'_\mathrm{test} = \mathbf{V}_p^T \mathbf{Z}_\mathrm{test}, \tag{B.6}$$

where $\mathbf{Z}'_\mathrm{test}$ is of dimension $p \times N_\mathrm{test}$.

B.1 Computational complexity

The computational complexity can be divided into three parts,

1. Creation of $\mathbf{Z}_r^T \mathbf{Z}_r$. This amounts to approximately $\left(\frac{(rN)^2 - rN}{2}\right) Df_{s_z}$ operations.

2. Eigenvalue decomposition of $\mathbf{Z}_r^T \mathbf{Z}_r$ amounts to $\mathcal{O}\left((rN)^3\right)$ operations.
3. Calculation of the outer product eigenvector $\mathbf{V}_p$ amounts to $\approx (Df_{s_z}) \cdot rN \cdot p$ operations.

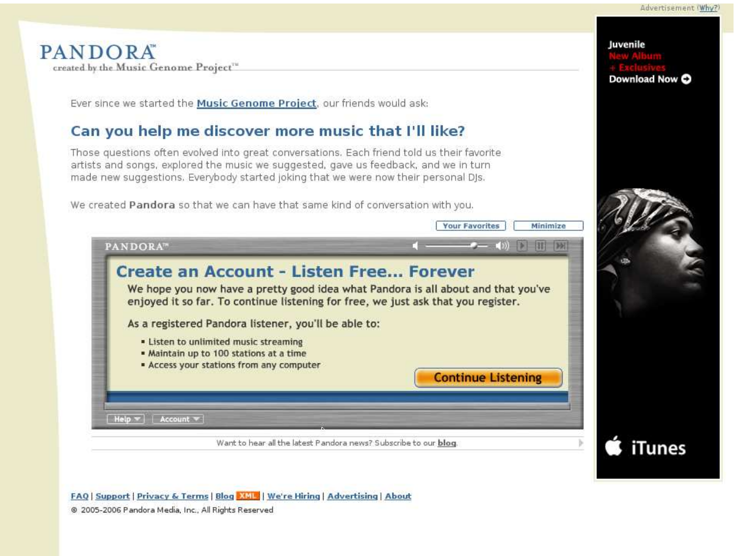
For the investigations conducted in [4, appendix D] , the values which largest computational complexity was $f_{s_z} = 100$, $D = 103$. Only the largest $p = 50$ eigenvalue/vector pairs were used. The amount of operations required to perform the eigenvalue decomposition of the original matrix of size $Df_{s_z} \times Df_{s_z}$, would amount to approximately 10^{12} operations. This is without considering the complexity of creating the matrix. The above suggested scheme with $r{\cdot}N = 1500$ (as applied in the article), results in a complexity less than $1.2 \cdot 10^{10}$ operations, thus, an overall decrease in complexity of approximately 10^2.

Appendix C

Homepages as of March 2006

Links to screenshots of relevant homepages

- `www.pandora.com`: see page 120
- `www.freedb.org`: see page 120
- `www.itunes.com`: see page 121
- `www.napster.com`: see page 122
- `www.allmusic.com`: see page 123
- `www.amazon.com`: see page 123
- `www.garageband.com`: see page 124
- `www.soundvenue.dk`: see page 125
- `www.mymusic.dk`: see page 126
- `www.mp3.com`: see page 127
- `www.findsounds.com`: see page 128
- `www.audioscrobbler.com`: see page 128
- `www.google.com`: see page 129
- `www.magnatune.com`: see page 130



PANDORA™
created by the Music Genome Project™

Ever since we started the Music Genome Project, our friends would ask:

Can you help me discover more music that I'll like?

Those questions often evolved into great conversations. Each friend told us their favorite artists and songs, explored the music we suggested, gave us feedback, and we in turn made new suggestions. Everybody started joking that we were now their personal DJs.

We created **Pandora** so that we can have that same kind of conversation with you.

Your Favorites | Minimize

PANDORA™

Create an Account - Listen Free... Forever

We hope you now have a pretty good idea what Pandora is all about and that you've enjoyed it so far. To continue listening for free, we just ask that you register.

As a registered Pandora listener, you'll be able to:

- Listen to unlimited music streaming
- Maintain up to 100 stations at a time
- Access your stations from any computer

Continue Listening

Help | Account

Want to hear all the latest Pandora news? Subscribe to our blog.

FAQ | Support | Privacy & Terms | Blog XML | We're Hiring | Advertising | About

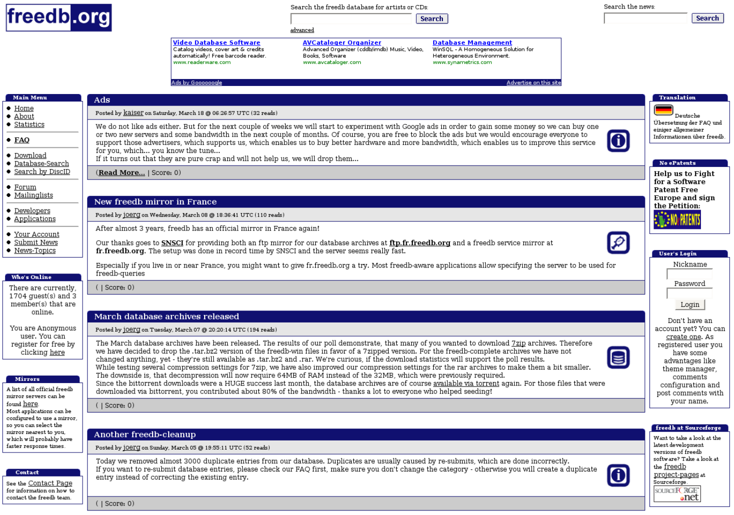
© 2005-2006 Pandora Media, Inc., All Rights Reserved

Advertisement (Why?)

Juvenile
New Album
+ Exclusives
Download Now

iTunes

Figure C.1: The figure shows a screen dump of the Internet portal 'www.pandora.com'.



freedb.org

Search the freedb database for artists or CDs: Search
advanced

Search the news: Search

Video Database Software
Catalog videos, cover art & credits automatically! Free barcode reader.
www.readerware.com

AVCataloger Organizer
Advanced Organizer (cddb/imdb) Music, Video, Books, Software
www.avcataloger.com

Database Management
WinSQL - A Homogeneous Solution for Heterogeneous Environment.
www.synametrics.com

Ads by Goooooogle | Advertise on this site

Main Menu

- Home
- About
- Statistics

- **FAQ**

- Download
- Database-Search
- Search by DiscID

- Forum
- Mailinglists

- Developers
- Applications

- Your Account
- Submit News
- News-Topics

Who's Online

There are currently, 1704 guest(s) and 3 member(s) that are online.

You are Anonymous user. You can register for free by clicking here

Mirrors

A list of all official freedb mirror servers can be found here.
Most applications can be configured to use a mirror, so you can select the mirror nearest to you, which will probably have faster response times.

Contact

See the Contact Page for information on how to contact the freedb team.

Ads

Posted by kaiser on Saturday, March 18 @ 06:26:57 UTC (32 reads)

We do not like ads either. But for the next couple of weeks we will start to experiment with Google ads in order to gain some money so we can buy one or two new servers and some bandwidth in the next couple of months. Of course, you are free to block the ads but we would encourage everyone to support those advertisers, which supports us, which enables us to buy better hardware and more bandwidth, which enables us to improve this service for you, which... you know the tune...
If it turns out that they are pure crap and will not help us, we will drop them...

(**Read More...** | Score: 0)

New freedb mirror in France

Posted by joerg on Wednesday, March 08 @ 18:36:41 UTC (110 reads)

After almost 3 years, freedb has an official mirror in France again!

Our thanks goes to **SNSCI** for providing both an ftp mirror for our database archives at **ftp.fr.freedb.org** and a freedb service mirror at **fr.freedb.org.** The setup was done in record time by SNSCI and the server seems really fast.

Especially if you live in or near France, you might want to give fr.freedb.org a try. Most freedb-aware applications allow specifying the server to be used for freedb-queries

(| Score: 0)

March database archives released

Posted by joerg on Tuesday, March 07 @ 20:20:14 UTC (194 reads)

The March database archives have been released. The results of our poll demonstrate, that many of you wanted to download 7zip archives. Therefore we have decided to drop the .tar.bz2 version of the freedb-win files in favor of a 7zipped version. For the freedb-complete archives we have not changed anything, yet - they're still available as .tar.bz2 and .rar. We're curious, if the download statistics will support the poll results.
While testing several compression settings for 7zip, we have also improved our compression settings for the rar archives to make them a bit smaller.
The downside is, that decompression will now require 64MB of RAM instead of the 32MB, which were previously required.
Since the bittorrent downloads were a HUGE success last month, the database archives are of course available via torrent again. For those files that were downloaded via bittorrent, you contributed about 80% of the bandwidth - thanks a lot to everyone who helped seeding!

(| Score: 0)

Another freedb-cleanup

Posted by joerg on Sunday, March 05 @ 19:55:11 UTC (52 reads)

Today we removed almost 3000 duplicate entries from our database. Duplicates are usually caused by re-submits, which are done incorrectly.
If you want to re-submit database entries, please check our FAQ first, make sure you don't change the category - otherwise you will create a duplicate entry instead of correcting the existing entry.

(| Score: 0)

Translation

Deutsche Übersetzung der FAQ und einiger allgemeiner Informationen über freedb.

No ePatents

Help us to Fight for a Software Patent Free Europe and sign the Petition:

NO ePATENTS

User's Login

Nickname

Password

Login

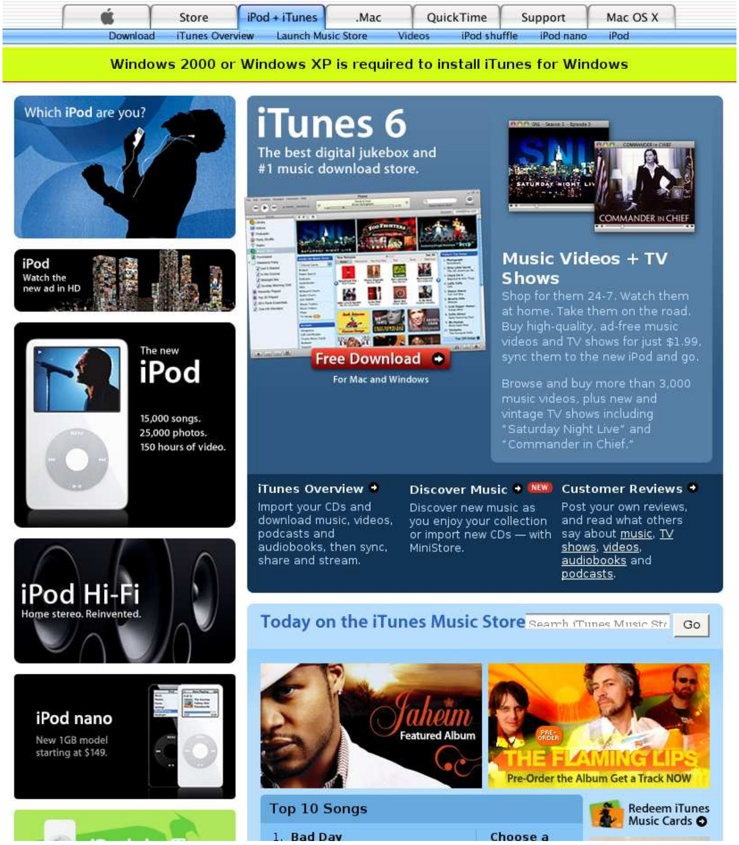
Don't have an account yet? You can create one. As registered user you have some advantages like theme manager, comments configuration and post comments with your name.

freedb at Sourceforge

Want to take a look at the latest development versions of freedb software? Take a look at the freedb project-pages at Sourceforge.

SOURCEFORGE.net

Figure C.2: The figure shows a screen dump of the Internet portal 'www.freedb.org'.



Windows 2000 or Windows XP is required to install iTunes for Windows

Which iPod are you?

iPod Watch the new ad in HD

The new iPod

15,000 songs. 25,000 photos. 150 hours of video.

iPod Hi-Fi Home stereo. Reinvented.

iPod nano New 1GB model starting at $149.

iTunes 6

The best digital jukebox and #1 music download store.

For Mac and Windows

Music Videos + TV Shows

Shop for them 24-7. Watch them at home. Take them on the road. Buy high-quality, ad-free music videos and TV shows for just $1.99, sync them to the new iPod and go.

Browse and buy more than 3,000 music videos, plus new and vintage TV shows including "Saturday Night Live" and "Commander in Chief."

iTunes Overview

Import your CDs and download music, videos, podcasts and audiobooks, then sync, share and stream.

Discover Music NEW

Discover new music as you enjoy your collection or import new CDs — with MiniStore.

Customer Reviews

Post your own reviews, and read what others say about music, TV shows, videos, audiobooks and podcasts.

Today on the iTunes Music Store

Search iTunes Music Sto

Jaheim Featured Album

PRE-ORDER THE FLAMING LIPS Pre-Order the Album Get a Track NOW

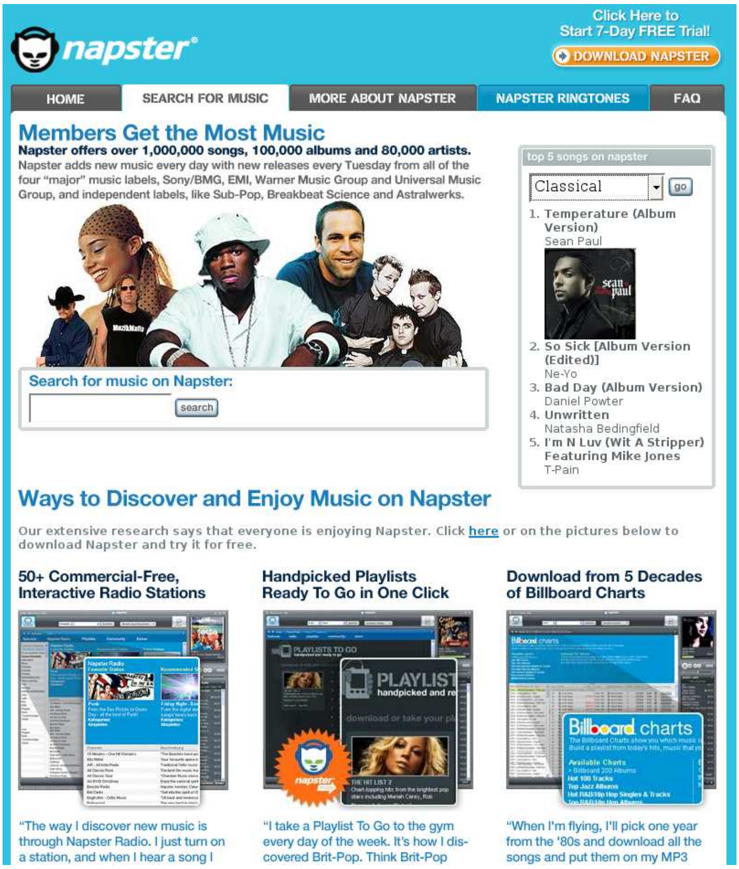
Top 10 Songs

1. Bad Day

Choose a

Redeem iTunes Music Cards

Figure C.3: The figure shows a screen dump of 'http://www.apple.com/itunes/'.



napster®

Click Here to
Start 7-Day FREE Trial!

DOWNLOAD NAPSTER

HOME | SEARCH FOR MUSIC | MORE ABOUT NAPSTER | NAPSTER RINGTONES | FAQ

Members Get the Most Music

Napster offers over 1,000,000 songs, 100,000 albums and 80,000 artists.
Napster adds new music every day with new releases every Tuesday from all of the four "major" music labels, Sony/BMG, EMI, Warner Music Group and Universal Music Group, and independent labels, like Sub-Pop, Breakbeat Science and Astralwerks.

Search for music on Napster:

search

top 5 songs on napster

Classical ▾ go

1. Temperature (Album Version)
   Sean Paul
2. So Sick [Album Version (Edited)]
   Ne-Yo
3. Bad Day (Album Version)
   Daniel Powter
4. Unwritten
   Natasha Bedingfield
5. I'm N Luv (Wit A Stripper) Featuring Mike Jones
   T-Pain

Ways to Discover and Enjoy Music on Napster

Our extensive research says that everyone is enjoying Napster. Click here or on the pictures below to download Napster and try it for free.

50+ Commercial-Free, Interactive Radio Stations

"The way I discover new music is through Napster Radio. I just turn on a station, and when I hear a song I

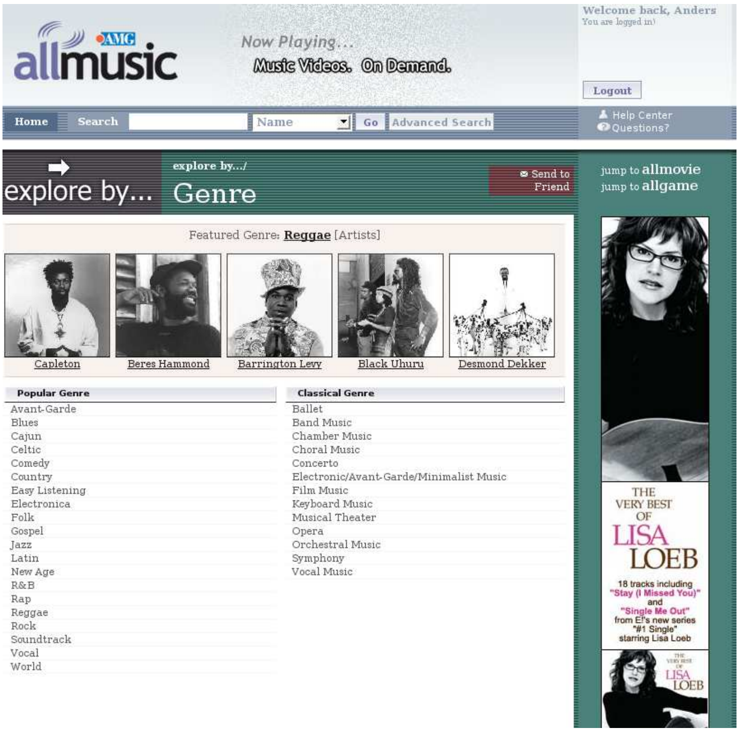
Handpicked Playlists Ready To Go in One Click

"I take a Playlist To Go to the gym every day of the week. It's how I discovered Brit-Pop. Think Brit-Pop

Download from 5 Decades of Billboard Charts

"When I'm flying, I'll pick one year from the '80s and download all the songs and put them on my MP3

Figure C.4: The figure shows a screen dump of 'http://www.napster.com'.



Featured Genre: Reggae [Artists]

Capleton · Beres Hammond · Barrington Levy · Black Uhuru · Desmond Dekker

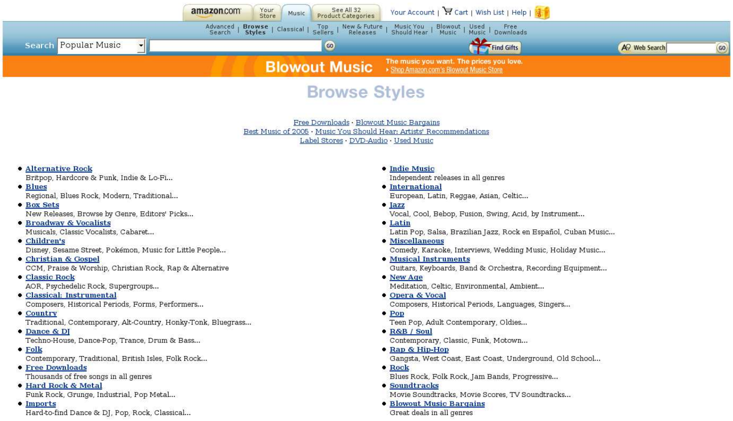
Popular Genre

Avant-Garde
Blues
Cajun
Celtic
Comedy
Country
Easy Listening
Electronica
Folk
Gospel
Jazz
Latin
New Age
R&B
Rap
Reggae
Rock
Soundtrack
Vocal
World

Classical Genre

Ballet
Band Music
Chamber Music
Choral Music
Concerto
Electronic/Avant-Garde/Minimalist Music
Film Music
Keyboard Music
Musical Theater
Opera
Orchestral Music
Symphony
Vocal Music

Figure C.5: The figure shows a screen dump of the Internet portal 'www.allmusic.com'.

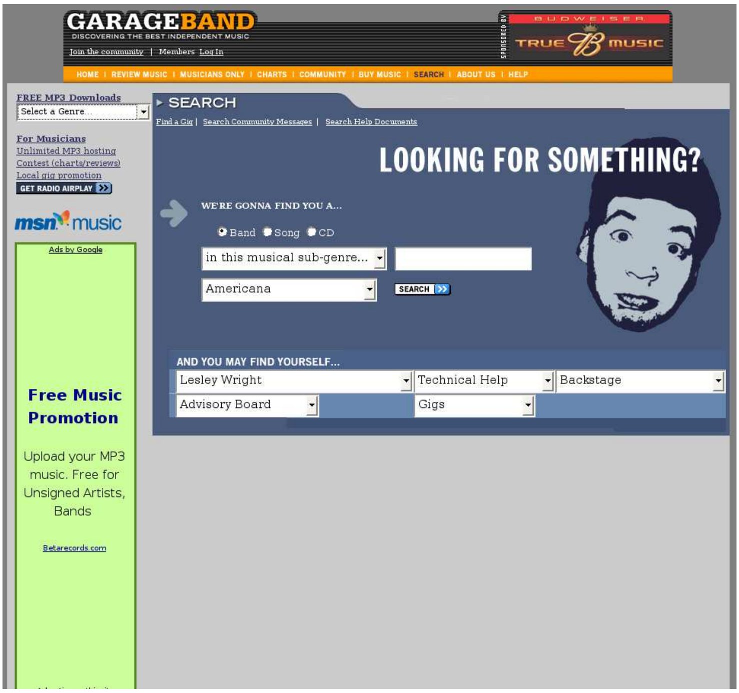

amazon.com
Your Store
Music
See All 32 Product Categories
Your Account | Cart | Wish List | Help |
Advanced Search | Browse Styles | Classical | Top Sellers | New & Future Releases | Music You Should Hear | Blowout Music | Used Music | Free Downloads
Search
Popular Music
GO
Find Gifts
Web Search
GO
Blowout Music
The music you want. The prices you love.
Shop Amazon.com's Blowout Music Store

Browse Styles

Free Downloads · Blowout Music Bargains
Best Music of 2005 · Music You Should Hear: Artists' Recommendations
Label Stores · DVD-Audio · Used Music

- **Alternative Rock**
  Britpop, Hardcore & Punk, Indie & Lo-Fi...
- **Blues**
  Regional, Blues Rock, Modern, Traditional...
- **Box Sets**
  New Releases, Browse by Genre, Editors' Picks...
- **Broadway & Vocalists**
  Musicals, Classic Vocalists, Cabaret...
- **Children's**
  Disney, Sesame Street, Pokémon, Music for Little People...
- **Christian & Gospel**
  CCM, Praise & Worship, Christian Rock, Rap & Alternative
- **Classic Rock**
  AOR, Psychedelic Rock, Supergroups...
- **Classical: Instrumental**
  Composers, Historical Periods, Forms, Performers...
- **Country**
  Traditional, Contemporary, Alt-Country, Honky-Tonk, Bluegrass...
- **Dance & DJ**
  Techno-House, Dance-Pop, Trance, Drum & Bass...
- **Folk**
  Contemporary, Traditional, British Isles, Folk Rock...
- **Free Downloads**
  Thousands of free songs in all genres
- **Hard Rock & Metal**
  Funk Rock, Grunge, Industrial, Pop Metal...
- **Imports**
  Hard-to-find Dance & DJ, Pop, Rock, Classical...
- **Indie Music**
  Independent releases in all genres
- **International**
  European, Latin, Reggae, Asian, Celtic...
- **Jazz**
  Vocal, Cool, Bebop, Fusion, Swing, Acid, by Instrument...
- **Latin**
  Latin Pop, Salsa, Brazilian Jazz, Rock en Español, Cuban Music...
- **Miscellaneous**
  Comedy, Karaoke, Interviews, Wedding Music, Holiday Music...
- **Musical Instruments**
  Guitars, Keyboards, Band & Orchestra, Recording Equipment...
- **New Age**
  Meditation, Celtic, Environmental, Ambient...
- **Opera & Vocal**
  Composers, Historical Periods, Languages, Singers...
- **Pop**
  Teen Pop, Adult Contemporary, Oldies...
- **R&B / Soul**
  Contemporary, Classic, Funk, Motown...
- **Rap & Hip-Hop**
  Gangsta, West Coast, East Coast, Underground, Old School...
- **Rock**
  Blues Rock, Folk Rock, Jam Bands, Progressive...
- **Soundtracks**
  Movie Soundtracks, Movie Scores, TV Soundtracks...
- **Blowout Music Bargains**
  Great deals in all genres

Figure C.6: The figure shows a screen dump of the Internet portal 'www.amazon.com'.

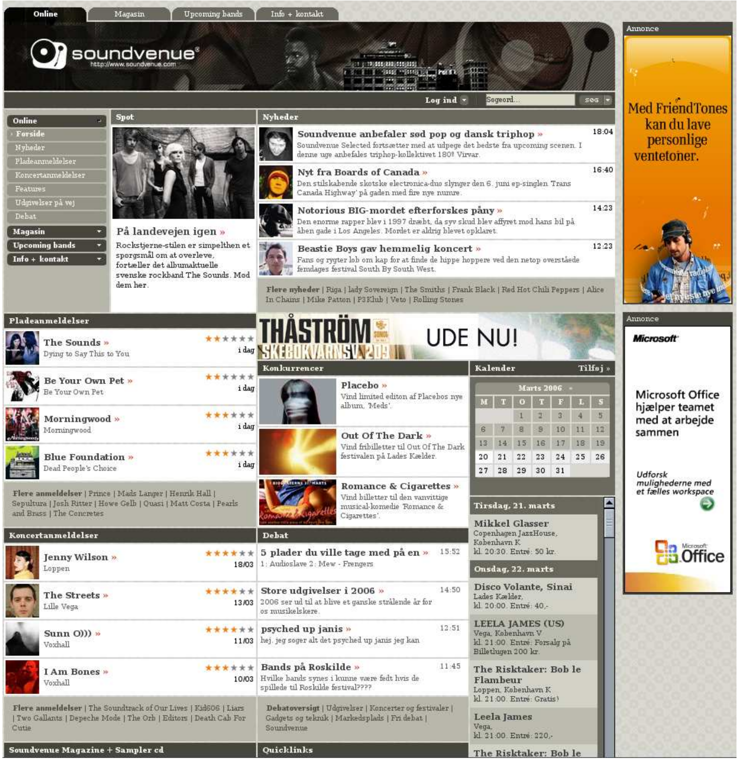

GARAGEBAND
DISCOVERING THE BEST INDEPENDENT MUSIC
Join the community | Members Log In
SPONSORED BY
BUDWEISER
TRUE MUSIC
HOME | REVIEW MUSIC | MUSICIANS ONLY | CHARTS | COMMUNITY | BUY MUSIC | SEARCH | ABOUT US | HELP
FREE MP3 Downloads
Select a Genre...
For Musicians
Unlimited MP3 hosting
Contest (charts/reviews)
Local gig promotion
GET RADIO AIRPLAY
msn music
Ads by Google
SEARCH
Find a Gig | Search Community Messages | Search Help Documents
LOOKING FOR SOMETHING?
WE'RE GONNA FIND YOU A...
● Band ○ Song ○ CD
in this musical sub-genre...
Americana
SEARCH
AND YOU MAY FIND YOURSELF...
Lesley Wright
Technical Help
Backstage
Advisory Board
Gigs

Free Music Promotion

Upload your MP3 music. Free for Unsigned Artists, Bands

Betarecords.com

Figure C.7: The figure shows a screen dump of the Internet portal 'www.garageband.com'.



Spot

På landevejen igen »

Rockstjerne-stilen er simpelthen et spørgsmål om at overleve, fortæller det albumaktuelle svenske rockband The Sounds. Mød dem her.

Nyheder

Soundvenue anbefaler sød pop og dansk triphop » 18:04
Soundvenue Selected fortsætter med at udpege det bedste fra upcoming scenen. I denne uge anbefales triphop-kollektivet 180º Virvar.

Nyt fra Boards of Canada » 16:40
Den stilskabende skotske electronica-duo slynger den 6. juni ep-singlen 'Trans Canada Highway' på gaden med fire nye numre.

Notorious BIG-mordet efterforskes pany » 14:23
Den enorme rapper blev i 1997 dræbt, da syv skud blev affyret mod hans bil på åben gade i Los Angeles. Mordet er aldrig blevet opklaret.

Beastie Boys gav hemmelig koncert » 12:23
Fans og rygter løb om kap for at finde de hippe hoppere ved den netop overstående femdages festival South By South West.

Flere nyheder | Riga | lady Sovereign | The Smiths | Frank Black | Red Hot Chili Peppers | Alice In Chains | Mike Patton | P3Klub | Veto | Rolling Stones

Pladeanmeldelser

The Sounds » ★★☆☆☆☆ i dag
Dying to Say This to You

Be Your Own Pet » ★★☆☆☆☆ i dag
Be Your Own Pet

Morningwood » ★★★☆☆☆ i dag
Morningwood

Blue Foundation » ★★★☆☆☆ i dag
Dead People's Choice

Flere anmeldelser | Prince | Mads Langer | Henrik Hall | Sepultura | Josh Ritter | Howe Gelb | Quasi | Matt Costa | Pearls and Brass | The Concretes

Konkurrencer

Placebo »
Vind limited editon af Placebos nye album, 'Meds'.

Out Of The Dark »
Vind fribilletter til Out Of The Dark festivalen på Lades Kælder.

Romance & Cigarettes »
Vind billetter til den vanvittige musical-komedie 'Romance & Cigarettes'.

Kalender Tilføj »

Marts 2006

M	T	O	T	F	L	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Tirsdag, 21. marts

Mikkel Glasser
Copenhagen JazzHouse,
København K
kl. 20:30. Entré: 50 kr.

Onsdag, 22. marts

Disco Volante, Sinai
Lades Kælder,
kl. 20:00. Entré: 40,-

LEELA JAMES (US)
Vega, København V
kl. 21:00. Entré: Forsalg på Billetlugen 200 kr.

The Risktaker: Bob le Flambeur
Loppen, København K
kl. 21:00. Entré: Gratis!

Leela James
Vega,
kl. 21:00. Entré: 220,-

The Risktaker: Bob le

Koncertanmeldelser

Jenny Wilson » ★★★★☆☆ 18/03
Loppen

The Streets » ★★★★☆☆ 13/03
Lille Vega

Sunn O))) » ★★★★☆☆ 11/03
Voxhall

I Am Bones » ★★★☆☆☆ 10/03
Voxhall

Flere anmeldelser | The Soundtrack of Our Lives | Kid606 | Liars | Two Gallants | Depeche Mode | The Orb | Editors | Death Cab For Cutie

Debat

5 plader du ville tage med på en » 15:52
1: Audioslave 2: Mew - Frengers

Store udgivelser i 2006 » 14:50
2006 ser ud til at blive et ganske strålende år for os musikelskere.

psyched up janis » 12:51
hej. jeg søger alt det psyched up janis jeg kan

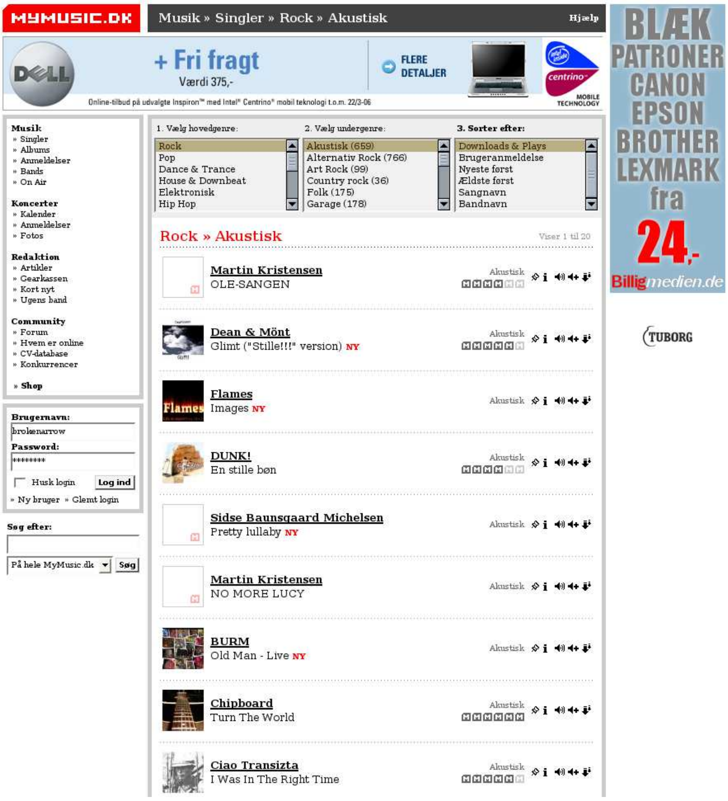
Bands på Roskilde » 11:45
Hvilke bands synes i kunne være fedt hvis de spillede til Roskilde festival????

Debatoversigt | Udgivelser | Koncerter og festivaler | Gadgets og teknik | Markedsplads | Fri debat | Soundvenue

Soundvenue Magazine + Sampler cd

Quicklinks

Figure C.8: The figure shows a screen dump of the Internet portal 'www.soundvenue.dk'.



Brugernavn: brokenarrow

Password: ********

☐ Husk login

Søg efter:

På hele MyMusic.dk

Rock » Akustisk

Viser 1 til 20

Martin Kristensen
OLE-SANGEN
Akustisk

Dean & Mönt
Glimt ("Stille!!!" version) NY
Akustisk

Flames
Images NY
Akustisk

DUNK!
En stille bøn
Akustisk

Sidse Baunsgaard Michelsen
Pretty lullaby NY
Akustisk

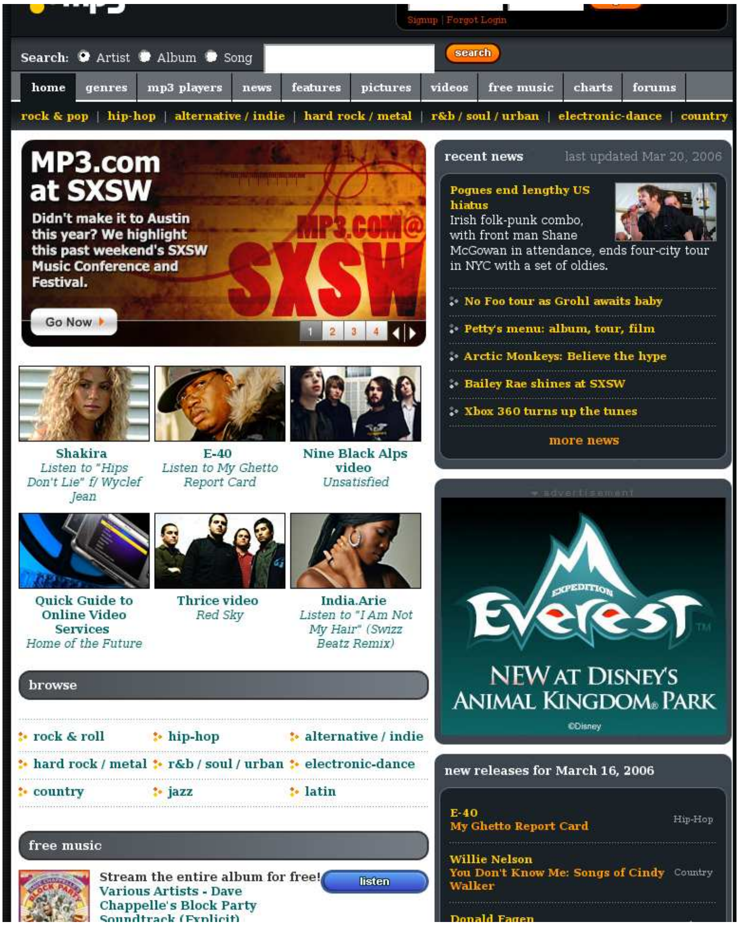
Martin Kristensen
NO MORE LUCY
Akustisk

BURM
Old Man - Live NY
Akustisk

Chipboard
Turn The World
Akustisk

Ciao Transizta
I Was In The Right Time
Akustisk

Figure C.9: The figure shows a screen dump of the Internet portal 'www.mymusic.dk'.



MP3.com at SXSW

Didn't make it to Austin this year? We highlight this past weekend's SXSW Music Conference and Festival.

recent news last updated Mar 20, 2006

Pogues end lengthy US hiatus
Irish folk-punk combo, with front man Shane McGowan in attendance, ends four-city tour in NYC with a set of oldies.

- No Foo tour as Grohl awaits baby
- Petty's menu: album, tour, film
- Arctic Monkeys: Believe the hype
- Bailey Rae shines at SXSW
- Xbox 360 turns up the tunes

more news

Shakira
Listen to "Hips Don't Lie" f/ Wyclef Jean

E-40
Listen to My Ghetto Report Card

Nine Black Alps video
Unsatisfied

Quick Guide to Online Video Services
Home of the Future

Thrice video
Red Sky

India.Arie
Listen to "I Am Not My Hair" (Swizz Beatz Remix)

NEW AT DISNEY'S ANIMAL KINGDOM® PARK
©Disney

browse

- rock & roll
- hip-hop
- alternative / indie
- hard rock / metal
- r&b / soul / urban
- electronic-dance
- country
- jazz
- latin

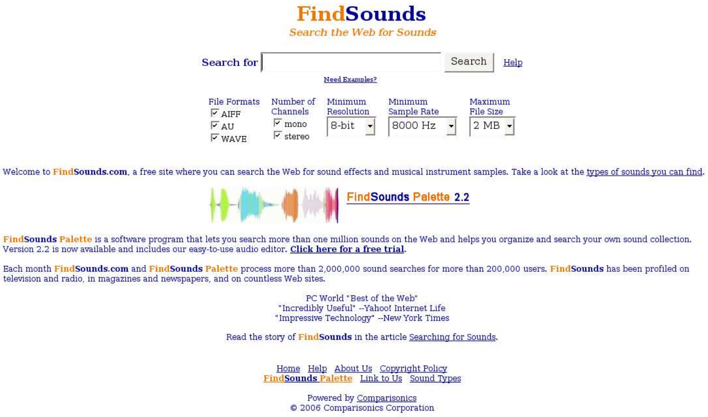
new releases for March 16, 2006

E-40 My Ghetto Report Card	Hip-Hop
Willie Nelson You Don't Know Me: Songs of Cindy Walker	Country
Donald Fagen	

free music

Stream the entire album for free!
Various Artists - Dave Chappelle's Block Party Soundtrack (Explicit)

Figure C.10: The figure shows a screen dump of the Internet portal 'www.mp3.com'



Welcome to FindSounds.com, a free site where you can search the Web for sound effects and musical instrument samples. Take a look at the types of sounds you can find.

FindSounds Palette is a software program that lets you search more than one million sounds on the Web and helps you organize and search your own sound collection. Version 2.2 is now available and includes our easy-to-use audio editor. **Click here for a free trial**.

Each month FindSounds.com and FindSounds Palette process more than 2,000,000 sound searches for more than 200,000 users. FindSounds has been profiled on television and radio, in magazines and newspapers, and on countless Web sites.

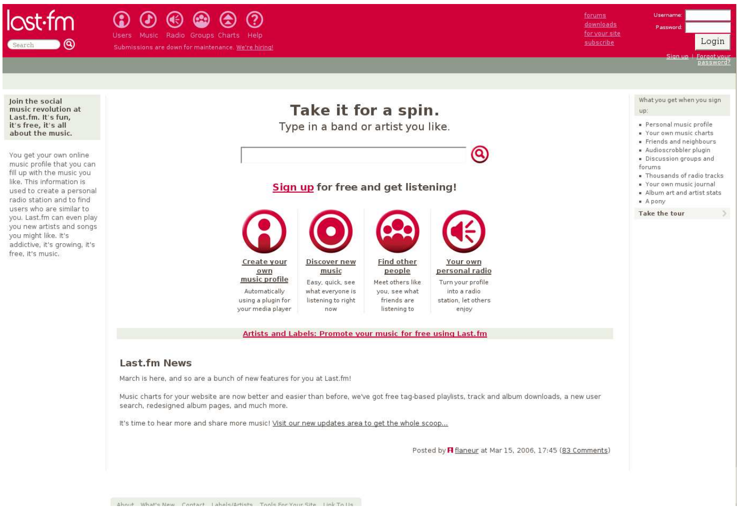
PC World "Best of the Web"
"Incredibly Useful" --Yahoo! Internet Life
"Impressive Technology" --New York Times

Read the story of FindSounds in the article Searching for Sounds.

Home Help About Us Copyright Policy
FindSounds Palette Link to Us Sound Types

Powered by Comparisonics
© 2006 Comparisonics Corporation

Figure C.11: The figure shows a screen dump of the Internet site 'www.findsounds.com'



Submissions are down for maintenance. We're hiring!

Join the social music revolution at Last.fm. It's fun, it's free, it's all about the music.

You get your own online music profile that you can fill up with the music you like. This information is used to create a personal radio station and to find users who are similar to you. Last.fm can even play you new artists and songs you might like. It's addictive, it's growing, it's free, it's music.

Take it for a spin.
Type in a band or artist you like.

Sign up for free and get listening!

Create your own music profile Automatically using a plugin for your media player

Discover new music Easy, quick, see what everyone is listening to right now

Find other people Meet others like you, see what friends are listening to

Your own personal radio Turn your profile into a radio station, let others enjoy

Artists and Labels: Promote your music for free using Last.fm

Last.fm News

March is here, and so are a bunch of new features for you at Last.fm!

Music charts for your website are now better and easier than before, we've got free tag-based playlists, track and album downloads, a new user search, redesigned album pages, and much more.

It's time to hear more and share more music! Visit our new updates area to get the whole scoop...

Posted by flaneur at Mar 15, 2006, 17:45 (83 Comments)

What you get when you sign up:

- Personal music profile
- Your own music charts
- Friends and neighbours
- Audioscrobbler plugin
- Discussion groups and forums
- Thousands of radio tracks
- Your own music journal
- Album art and artist stats
- A pony

Take the tour

Figure C.12: The figure shows a screen dump of the Internet site 'www.audioscrobbler.com', their pages have been moved to 'http://www.last.fm'

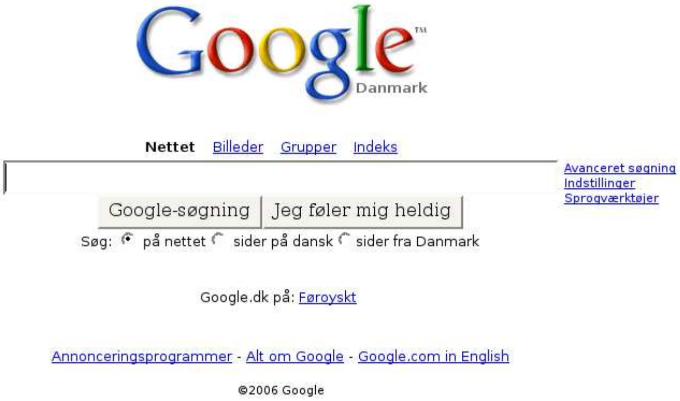
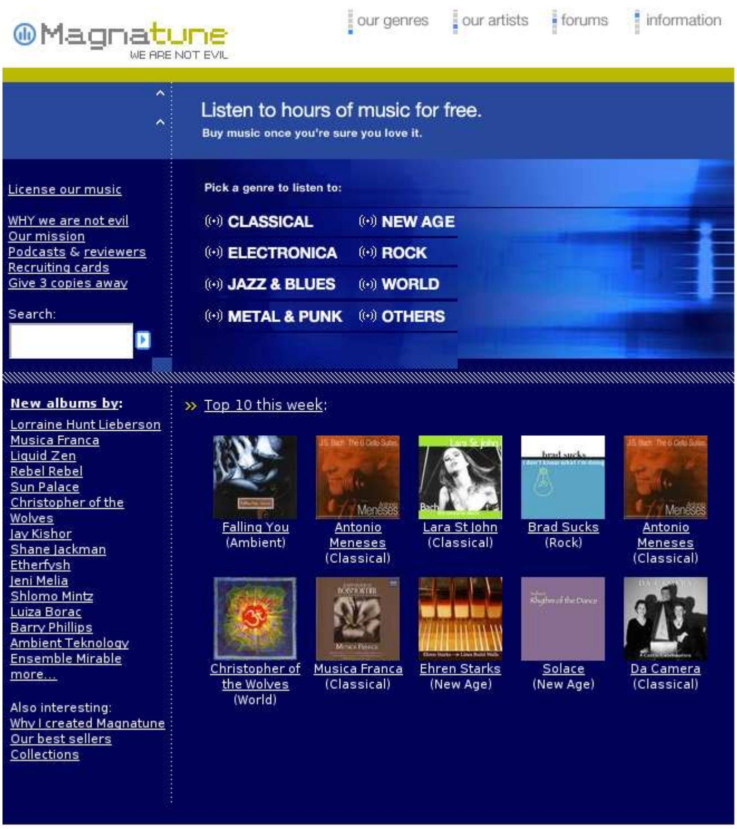


Figure C.13: The figure shows a screen dump of the Internet site 'www.google.com'



Listen to hours of music for free.

Buy music once you're sure you love it.

License our music

WHY we are not evil
Our mission
Podcasts & reviewers
Recruiting cards
Give 3 copies away

Search:

Pick a genre to listen to:

CLASSICAL	NEW AGE
ELECTRONICA	ROCK
JAZZ & BLUES	WORLD
METAL & PUNK	OTHERS

New albums by:
Lorraine Hunt Lieberson
Musica Franca
Liquid Zen
Rebel Rebel
Sun Palace
Christopher of the Wolves
Jay Kishor
Shane Jackman
Etherfysh
Jeni Melia
Shlomo Mintz
Luiza Borac
Barry Phillips
Ambient Teknology
Ensemble Mirable
more...

Also interesting:
Why I created Magnatune
Our best sellers
Collections

» Top 10 this week:

Falling You (Ambient)
Antonio Meneses (Classical)
Lara St John (Classical)
Brad Sucks (Rock)
Antonio Meneses (Classical)
Christopher of the Wolves (World)
Musica Franca (Classical)
Ehren Starks (New Age)
Solace (New Age)
Da Camera (Classical)

Figure C.14: The figure shows a screen dump of the Internet music portal 'www.magnatune.com'

Appendix D

Contribution: EUSIPCO 2004

This appendix contain the article *Decision time horizon for music genre classification using short time features.* In Proceedings of European Signal Processing Conference (EUSIPCO), pages: 1293-1296. Author list: P. Ahrendt, A. Meng and J. Larsen. Own contribution estimated to approximately 40%.

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

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

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.

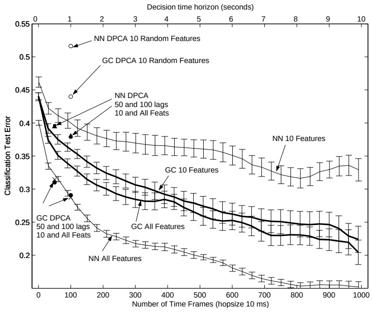


Figure 1: Classification error as a function of the lag of the GC and NN using DPCA and majority voting, respectively.

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 sensitivies 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.

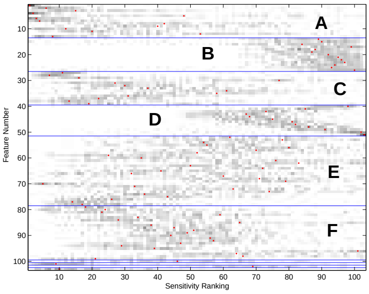


Figure 2: Consensus feature ranking of individual feature at $L = 100$. See text for interpretation. The features are MFCC(**A**), DMFCC(**B**), LPC(**C**), DLPC(**D**), ASE(**E**), SFM(**F**) and the single features ASC, ASS, STE and ZCR. The ten best features in decreasing order are: $\{1, 4, 6, 7,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

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 E

Contribution: ICASSP 2005

This appendix contain the article *Improving Music Genre Classification by Short Time Feature Integration*, in Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages: 497-500. Author list: A. Meng, P. Ahrendt and J. Larsen. Own contribution estimated to approximately 40%.

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.

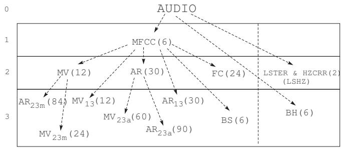


Fig. 1. Short(1), medium(2) and long(3) time features and their relationships. The arrow from e.g. medium time *MV* to the long time feature AR_{23m} indicate feature integration. Thus, for each of the 12 time-series of *MV* coefficients, 7 *AR* features have been found, resulting in a $7 \cdot 12 = 84$ 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.

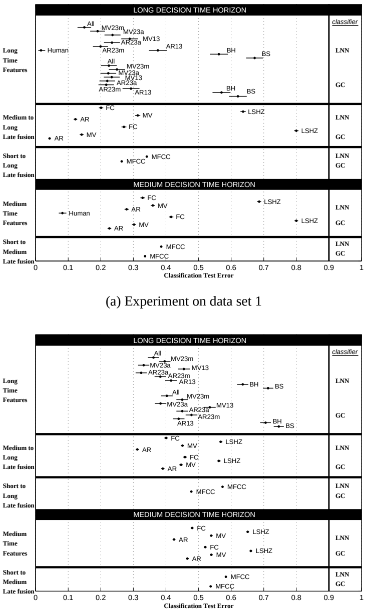
The classification test errors are shown in figure 2 for both of the data sets and both the medium time and long time classification problems.

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



(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.

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 F

Contribution: ISMIR 2005

This appendix contain the article *An investigation of feature models for music genre classification using the support vector classifier*. In Proceedings of International Symposium on Music Information Retrieval (ISMIR), pages: 504-509. Author list: A. Meng and J. Shawe-Taylor. Own contribution estimated to approximately 80%.

AN INVESTIGATION OF FEATURE MODELS FOR MUSIC GENRE CLASSIFICATION USING THE SUPPORT VECTOR CLASSIFIER

Anders Meng
Informatics and Mathematical Modelling - B321
Technical University of Denmark
am@imm.dtu.dk

John Shawe-Taylor
University of Southampton
jst@ecs.soton.ac.uk

ABSTRACT

In music genre classification the decision time is typically of the order of several seconds, however, most automatic music genre classification systems focus on short time features derived from $10-50$ms. This work investigates two models, the *multivariate Gaussian model* and the *multivariate autoregressive model* for modelling short time features. Furthermore, it was investigated how these models can be integrated over a segment of short time features into a kernel such that a support vector machine can be applied. Two kernels with this property were considered, the *convolution kernel* and *product probability kernel*. In order to examine the different methods an 11 genre music setup was utilized. In this setup the *Mel Frequency Cepstral Coefficients* were used as short time features. The accuracy of the best performing model on this data set was $\sim 44\%$ compared to a human performance of $\sim 52\%$ on the same data set.

Keywords: Feature Integration, Product Probability Kernel, Convolution Kernel, Support Vector Machine, Music Genre

1 INTRODUCTION

The field of audio mining covering areas such as audio classification, retrieval, fingerprinting etc. has received quite a lot of attention lately both from academic and commercial groups. Some of this interest stems from an increased availability of large online music stores and growing access to live radio-programs, music stations, news on the internet etc. The big task for the academic world is to find methods for effectively searching and navigating these large amounts of data.

The genre is probably the most important descriptor of music in everyday life, however, it is not an intrinsic property of music such as e.g. tempo, which makes it more difficult to grasp with computational methods. Still, for a limited amount of data and for coherent music databases there seem to be a link between computational methods and human assessment, see e.g. [1, 2].

It is a well established fact that the success of a pattern recognition system is closely related to the task of finding descriptive features. There exist a large amount of descriptive audio features, each designed for a specific audio mining task. The various features can be grouped as perceptual features such as pitch, loudness, beat or as non-perceptual features as the Mel Frequency Cepstral Coefficients (MFCC). The MFCCs have been applied in a range of audio mining tasks, and have shown good performance compared to other features at a similar time scale.

In music genre classification the typical time horizon for a human to classify a piece of music as belonging to a specific genre is of the order of a quarter of a second up to several seconds, see [3]. Typically for automatic music genre classification systems whole pieces of music are available, so the decision time is generally longer than just a few seconds.

Short time features such as the MFCCs are typically derived at time horizons around $10-50ms$ depending on the stationarity of the audio signal. A few authors [4, 5, 1] have looked at methods for integrating (modelling) the short time features to classify at longer time horizons. Integration of short time features (*feature integration*) is also known as early information fusion. Late information fusion is another way of classifying at larger time horizons. The idea of late information fusion is to combine the sequence of outputs from a classifier, like e.g. majority voting. Some techniques of information fusion (both early and late) have been considered in more detail in [4, 2].

The focus of this work was to extend the model of [2] for modelling the temporal structure of short time features and secondly to investigate different methods for handling audio data using kernel methods such as the *Support Vector Machine (SVM)*. The support vector machine is known for its good generalization performance in high-dimensional spaces, furthermore, its ability to work implicitly in a possible high-dimensional feature space makes it possible to investigate non-linear relations in the data.

The paper is structured as follows. An overview of the investigated features as well as a description of the two feature integration models the *multivariate Gaussian* *model (GM)* and the *multivariate autoregressive model (MAR)* are given in section 2. Section 3 briefly explains the classifiers applied to a music genre setup and furthermore explains the idea of information fusion. Section 4 presents the results of an 11 genre music genre setup. Last, but not least a conclusion in section 5.

2 FEATURES

The work presented in this paper will focus on constructing descriptive features at larger time scales by modelling short time features. Earlier work by [2, 1, 5] suggested to work with an intermediate time scale around 1 second. Here three time scales have been considered, a *short time scale* of 30ms where short time features are extracted, a *medium time scale* at 2 seconds (selected from the data set, see section 4) and a *long time scale* of 30 seconds, limited by the length of the music snippets. The long time scale contains information such as the ”mood” of the song as well as long-structural correlations.

2.1 Short Time Features (30ms)

The short time feature extraction stage is really important in all audio processing applications, since it is the first level of feature integration performed[1]. Earlier results [4, 5] indicate good performance in music genre classification using the MFCCs and therefore these will be the preferred choice in this investigation. These features were originally developed for classification of speech, however, they have been applied in various audio mining tasks, see e.g. [6] where they were used in a timbre similarity experiment. The low order MFCCs contain information of the slowly changing spectral envelope while the higher order MFCCs explains the fast variations of the envelope. Several authors report success using only the first $6-10$ MFCCs. In the music genre classification setup, see section 4, we found that the first seven MFCCs were adequate. Furthermore, a hop- and frame-size of 10ms and 30ms, respectively, were used. The larger overlap results in more smooth transitions between consecutive feature vectors.

2.2 Feature Integration ($>$ 30ms)

Feature integration is a method for capturing the temporal information in the features. With a good model the most salient structural information remains and the noisy part is suppressed. The idea of using feature integration in audio classification is not new, but has been investigated in earlier work by e.g. [1, 5, 2] where a performance increase was observed. The idea of feature integration can be stated more strict by observing a sequence of consecutive features

$$\mathbf{x}_{n+1},\ldots,\mathbf{x}_{n+L} \rightarrow \mathbf{f}(\mathbf{x}_{n+1},\ldots,\mathbf{x}_{n+L}) = \mathbf{z}, \quad (1)$$

where the sequence $\{\mathbf{x}_{n+1},\ldots,\mathbf{x}_{n+L}\} \in \mathcal{R}^{D\times L}$ are integrated into a new feature vector denoted as $\mathbf{z} \in \mathcal{R}^{M}$ where typically $M << D\cdot L$ and L indicates the number of short time features used in the integration step. A commonly used feature integration technique is the *mean-variance* of features, which provides a performance increase, but generally does not capture the temporal structure of the short time features. An improvement to this is the *filter-bank* approach considered in [5] to capture the frequency contents of the temporal structure in the short time features. This improvement indicated a performance increase compared to the mean-variance model, see [2]. Recently an autoregressive model [2] was suggested for feature integration and provided a performance increase compared to the mean-variance and filter-bank approach.

Figure 1 shows the first seven normalized MFCCs of a 10 second excerpt of the music piece *Master of Revenge* by the heavy metal group *Body Count*. As observed from the coefficients there is both temporal correlations as well as correlations among features dimensions.

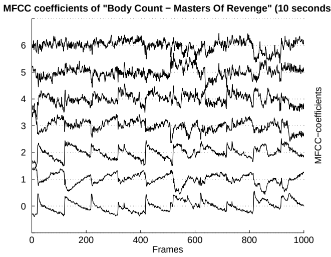


Figure 1: The first seven normalized MFCCs of a 10 second snippet of ”Body Count - Masters of Revenge”. The temporal correlation and correlations among feature dimensions are very clear from this piece of music.

2.2.1 *Multivariate autoregressive model (*MAR*)*

The *multivariate autoregressive* model handles both temporal and correlations among feature dimensions, which makes it a good candidate for feature integration. In [2] a simple autoregressive model was suggested where simple refers to considering each feature dimension independently. The MAR model is popular in time-series modelling and prediction being both simple and well understood, see e.g. [7]. For a stationary time series of state vectors $\mathbf{x}_n \in \mathcal{R}^D$ the MAR model is defined by

$$\mathbf{x}_n = \sum_{p=1}^{K} \mathbf{A}_p \mathbf{x}_{n-I(p)} + \boldsymbol{\mu} + \mathbf{u}_n, \quad (2)$$

where the noise term $\mathbf{u}_n$ (error-term) is assumed to be zero mean Gaussian distributed, hence $\mathbf{u}_n \sim \mathcal{N}(\mathbf{u}_n; \mathbf{0}, \mathbf{C})$.

The D-dimensional parameter vector $\boldsymbol{\mu}$ is a vector of intercept terms that is included to allow for a non-zero mean of the time-series, see [8]. The matrices $\mathbf{A}_p \in \mathcal{R}^{D\times D}$ for $p = 1\ldots K$ are the coefficient matrices of the K'th order multivariate autoregressive model. They

[1] Basically this first step is denoted as feature extraction and not feature integration.

encode how much of the previous information given in $\mathbf{x}_{n-I(1)}, \mathbf{x}_{n-I(2)}, .., \mathbf{x}_{n-I(K)}$ is present in $\mathbf{x}_n$. The above formulation is quite general as I refers to a general set. For a model order of $K = 4$, the set could be selected as $I = \{1, 2, 3, 4\}$ or $I = \{1, 2, 4, 8\}$ indicating that $\mathbf{x}_n$ is predicted from these previous state vectors. In this paper we focus on the standard multivariate autoregressive model where $I = \{1, 2, 3, \ldots, K\}$. When estimating the parameters of the model there is several methods available, see e.g. [7]. The authors have used the *ARFIT* package, a regularized ordinary least squares approach, described in [8]. This package ensures the uniqueness of the estimated parameters of the model.

2.2.2 *Multivariate Gaussian model (*GM*)*

Neglecting the temporal correlations in the data, hence setting the $\mathbf{A}_p$ matrices for $p = 1, \ldots, K$ in equation (2) to zero leads to the much simpler model

$$\mathbf{x}_n = \boldsymbol{\mu} + \mathbf{u}_n, \tag{3}$$

where $\boldsymbol{\mu}$ encode the mean value of the time series and $\mathbf{u}_n \sim \mathcal{N}(\mathbf{u_n}; \mathbf{0}, \mathbf{C})$ is denoted the multivariate Gaussian model. The previous mentioned *mean-variance* model is the mean value $\boldsymbol{\mu}$ and the variance components given from the diagonal of the covariance matrix $\mathbf{v} = \mathrm{diag}\{\mathbf{C}\}$. If the full covariance matrix is used, only the upper (or lower) triangular coefficients are needed due to the symmetry. The multivariate Gaussian model will be considered as the ”base-line” against the MAR model in the experimental section since it performs better than the typical *mean-variance* model.

The two feature integration techniques described above can be used to derive features at the *medium time scale* or used directly to derive features at the *long time scale*. The model order for the MAR model can be selected from e.g. Schwarz's Bayesian Criterion (SBC) [8], which is implemented in the *ARFIT* package or as in our experimental setup, where a separate validation set was used to determine the optimal model order across data examples (music snippets).

2.3 Unique Solutions

Performing feature integration the model parameters are typically used as new feature vectors at the new time scale. If the model does not have a unique solution, two similar audio pieces could risk being classified as dissimilar. Consider using a *mixture of Gaussian (MoG)*, given as

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} p(k) p(\mathbf{x}|k, \boldsymbol{\theta}),$$

where $p(k)$ (and $\sum_{k=1}^{K} p(k) = 1$) are the mixing proportions and $p(\mathbf{x}|k, \boldsymbol{\theta}) \sim \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \mathbf{C}_k)$, as a feature integration model. Optimizing the model parameters from the likelihood function using e.g. the *EM-algorithm* does not necessarily provide a global maximum since the likelihood function has many local maximums. So using these model parameters (mixing proportions, means and covariances) directly in a classifier[2] would make no sense. Recent studies in kernels indicate that it is possible to integrate this type of complicated models in a kernel, see e.g. [9, 10]. The mixture of Gaussian model was considered as modelling music snippets in [6] and will be investigated as a feature integration model in section 4.

3 CLASSIFIERS

Earlier work in the field of music information retrieval (*MIR*) considered simple yet efficient classifiers such as K-nearest neighbors, however, lately more computationally demanding algorithms have been investigated. Only a few researchers within the field of *MIR* have considered support vector machines (*SVM*), see e.g. [11, 12]. In the following subsections the support vector classifier (SVC) and the linear neural network classifier (LNN) will be briefly discussed.

3.1 Support Vector Classifier

The challenge of machine learning is to provide the learner with as broad a range of functions as possible while still ensuring that accurate learning can be achieved. Using high-dimensional feature spaces satisfies the first constraint of ensuring high flexibility, but appears to be at odds with the second since it is undermined by the curse of dimensionality. As a result we would expect that a good fit on the training data could still leave the generalization very poor. Support vector machines [13] manage to avoid this difficulty by optimizing a bound on the generalization error in terms of quantities that do not depend on the dimension of the feature space [14], hence enabling good performance unaffected by the curse of dimensionality. In the present work, the C-library *LIBSVM* [15] was used. This library implements the one-against-one voting terminology to handle more than two classes.

3.1.1 *Kernels*

A typical applied kernel for the support vector classifier is the *linear kernel*, which is defined as
$\kappa(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}'$, hence an inner product between the input vectors. Another well known kernel is the Gaussian kernel (or *RBF-kernel*) with width parameter σ defined as $\kappa(\mathbf{x}, \mathbf{x}') = \exp(- \parallel \mathbf{x} - \mathbf{x}' \parallel^2 / 2\sigma^2)$. Using this kernel the support vector classifier is basically finding discriminating dimensions in an infinite feature space.

The linear and RBF kernel can be used in comparing vector data, however, when handling audio we are typically forced to calculate the distance between two audio snippets of varying lengths, which for two pieces of audio is presented by the sequence of short time features: $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L] \in \mathcal{R}^{D\times L}$ and $\mathbf{X}' = [\mathbf{x}'_1, \mathbf{x}'_2, \ldots, \mathbf{x}'_{L'}] \in \mathcal{R}^{D\times L'}$. The two audio files are not required to be of same length, though in the present investigation they are ($L = L'$). Two different kernels have been investigated, which calculate a similarity between sequences of data, the *convolution kernel* [16] and the *product probability kernel* [9]. These kernels naturally incorporate feature integration.

Convolution Kernel - CK

The convolution kernel [16] handles all kinds of discrete

[2] Stacked in a vector.

structures such as strings, trees and graphs. In this work the convolution kernel measures the distance (correlation) between two audio pieces (between their feature vectors). The kernel is defined as

$$\kappa(\mathbf{X}, \mathbf{X}') = \frac{1}{L^2} \sum_{v=1}^{L} \sum_{v'=1}^{L} \kappa_I \left(\mathbf{x}_v, \mathbf{x'}_{v'}\right), \qquad (4)$$

where $\kappa_I(\mathbf{x}, \mathbf{z})$ must be a valid kernel. It is interesting to note that if a linear kernel is used a fast calculation can be obtained.

Product Probability Kernel - PPK
The *product probability kernel* introduced in [9] measures the distance between probability models of the feature vectors. Other divergence based kernels have been suggested, see e.g. [10], for measuring a similar distance. In [6] the Kullback-Leibler similarity measure was applied to measure the distance between timbre models of music snippets modelled by a mixture of Gaussian, however, no closed form solution could be found using this divergence measure. With the *product probability kernel*, a closed form solution can be determined for e.g. a mixture of Gaussian, furthermore, the *PPK* fulfills the requirement for a kernel to be positive semi-definite. From [9] the *PPK* is given as

$$\kappa(\boldsymbol{\theta}, \boldsymbol{\theta}') = \int p(\mathbf{x}|\boldsymbol{\theta})^{\rho} p(\mathbf{x}|\boldsymbol{\theta}')^{\rho} d\mathbf{x}, \qquad (5)$$

where $\boldsymbol{\theta}(\boldsymbol{\theta}')$ are the parameters from modelling $\mathbf{X}(\mathbf{X}')$, $\rho > 0$ and $p(\mathbf{x}|\boldsymbol{\theta})$ is the probabilistic model of the short time features of a music piece. ρ controls the weighting of low or high density areas of the probability distribution. Selecting $\rho = 1/2$ the *Bhattacharyya* affinity between distributions is found. A nice bi-product of selecting $\rho = 1/2$ is a normalized kernel structure, since $\kappa(\boldsymbol{\theta}, \boldsymbol{\theta}) = \int p(\mathbf{x}|\boldsymbol{\theta}) d\mathbf{x} = 1$. This kernel can directly compute the distance between the models suggested in section 2.2, and thus incorporates feature integration. As mentioned in section 2.3 the problem of uniqueness is alleviated for this kernel, since probabilistic models are compared instead of model parameters.

Closed form solutions of the kernel for the multivariate Gaussian and mixture of Gaussian can be found in [9]. Additionally, we have calculated a closed form solution of the MAR model, but the details have been omitted through lack of space [3].

3.2 Linear Neural Network classifier (LNN)

The linear Neural Network has c outputs and is trained using a squared loss function [17]. This classifier has previously been applied with success in music genre classification, see e.g. [2, 4].

3.3 Fusion Techniques

The early information fusion (feature integration) was discussed in section 2.2. Late information fusion is the problem of combining the results from the classifier. There exist several ways of performing late information fusion, see [18]. In the present work, the majority voting rule was applied due to the SVM classifier. 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.

4 EXPERIMENTS

To evaluate the different feature integration techniques an 11 genre music setup was investigated. As discussed in the introduction, decisions can be made at different time scales. In the present work, the best achievable performance at 30 seconds will be pursued, using the above feature integration techniques, voting technique and combinations of the two.

4.1 Data set

The data set consists of 11 music genres distributed evenly among the following categories: *Alternative, Country, Easy Listening, Electronica, Jazz, Latin, Pop&Dance, Rap&Hiphop, R&B and Soul, Reggae and Rock*. The data set consists of a training set of 1098 music snippets, 100 from each genre except for latin, of each 30 seconds and a separate test set of 220 music snippets each of 30 seconds in length. The music snippets were *MP3* encoded music with a bit-rate $\geq 128kB$ down-sampled with a factor two to 22050Hz.

4.1.1 Human evaluation

To test the integrity of the data set a human evaluation was performed on the music snippets (at a 30 second time scale) of the test set. Each test person out of 9 was asked to classify each music snippet into one of the 11 genres on a forced choice basis. Each person evaluated 33 music snippets out of the 220 music pieces. No information except for the genre of the music pieces was given prior to the test. The average accuracy of the human evaluation across people and across genre was 51.8% as opposed to random guessing, which is $\sim 9.1\%$. The lower/upper 95% confidence limits were $46.0\%/57.7\%$ (results shown in figure 2, upper figure). The human evaluation shows that the common genre definition is less consistent for this data set, however, it is still interesting to observe how an automatic genre system works in this setup.

4.1.2 Results & Discussion

In each genre 90 out of the 100 music snippets from the training set were randomly selected 10 times to assess the variations in the data. In each of these runs the remaining music pieces (10 in each genre, except *latin*) was used as a validation set for tuning parameters such as C in the support vector classifier and σ in the RBF kernel. Optimal model order selection for the MAR models were determined across music samples and evaluated on the validation set. A model order of $K = 3$ at both 2 and 30 seconds was found optimal.

The medium time scale was selected by evaluating the performance at 30 seconds using both the *GMMV* and the

[3] Regarding computational complexity the methods ranked after numerical complexity are (top: least computational intensive): GM, MAR, MoG. The GM and MAR are closer related in complexity than the MAR and MoG.

Table 1: Description of the different combinations investigated. All investigations with the product probability kernel, $\rho = 1/2$ was used.

Scheme	Description
MOG,PPK	Mixture of Gaussian applied to each 30 second music snippet. A PPK kernel was generated (dimension 990×990).
GM,PPK	A multivariate Gaussian is fitted for each 30 second music snippet. A PPK kernel was generated.
GM,PPK,MV	A multivariate Gaussian is fitted for each 2 seconds of music data. A PPK kernel is generated (sampling applied using only 3 samples from each music piece resulting in a kernel of 2970×2970). After classification with SVM, majority voting is applied.
GM,CONV	A multivariate Gaussian is fitted for each 2 seconds of music data and a linear convolution kernel is applied (taking mean of the parameters).
GM,MV	A multivariate Gaussian is fitted for each 2 seconds of music data and majority voting is applied to the outputs of the classifiers. For the SVM a RBF-kernel was applied.
GM	A multivariate Gaussian is fitted for each 30 second music snippet. For the SVM a RBF-kernel was applied.
MAR,PPK	Same as above (see GMPPK), just with a multivariate AR process.
MAR,PPK,MV	Same as above, just with a multivariate AR process.
MAR,PPK,CONV	Same as above, just with a multivariate AR process.
MAR,CONV	Same as above, just with a multivariate AR process.
MAR,MV	Same as above, just with a multivariate AR process.
MAR	Same as above, just with a multivariate AR process.

MARMV method explained in table 1 varying the frame-/hop size of the medium time scale[4]. No big performance fluctuation was observed in this investigation, however, a small favor of a frame-/hop size of $2/1$ second was observed. The various combinations investigated have been described in more detail in table 1. For the mixture of Gaussian model incorporated in a product probability kernel (MOG,PPK) the optimal model order for each music snippet of 30 seconds were selected by varying the model order between $2 - 6$ mixtures, and selecting the optimal order from the Bayesian Information Criterion (BIC).

The average accuracy over the ten runs of the various combinations illustrated in table 1 have been plotted in figure 2 (upper figure) with a 95% binomial confidence applied to the average values. From the accuracy plot there is a clear indication that the MAR model is performing better than the GM for both the *SVM* and *LNN* classifier. Performing a McNemar test, see e.g. [19], on the mixture of Gaussian model (MOGPPK) and the Gaussian model in a product probability kernel (GMPPK) the probability that the two models are equal is 76%, hence the hypothesis that the models are equal cannot be rejected on a 5% significance level. This observation, together with the good performance of the MAR model illustrate the importance of the temporal information in the short time features. Even with the various techniques applied in this setup we are still around $\sim 8\%$ from the average human accuracy of $\sim 52\%$ on this data set, but it is interesting to notice that reasonable performance is achieved with fairly simple feature integration models and fusion techniques using only the first seven MFCCs. The two best performing models are the MAR model in a product probability kernel (MARPPK) and the MAR model modelled at 2 seconds, after which majority voting is applied on the LNN outputs (MARMV), see figure 2 (upper). The McNemar test on these two models showed a 43% significance level thus it can not be rejected that the two models are similar.

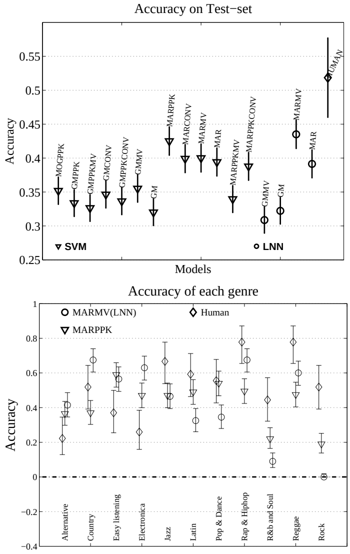


Figure 2: **Upper:** Average accuracy at 30 seconds shown with a 95% binomial confidence interval for all investigated combinations. The larger confidence interval for humans is due to only nine persons evaluating a part of the test-data. **Lower:** Average accuracy with 95% confidential interval of each genre at a time scale of 30 seconds using the two best performing combinations, *MARMV* and *MARPPK*. The average human accuracy in each genre is also shown with a 75% confidence interval.

The advantage of the MARPPK model is that we only need to store the model parameters at 30 seconds, while

[4] The investigated frame-/hop sizes were: {1s/0.5s, 1.5s/0.75s, 2s/1s, 2.5s/1.25s, 3s/1.5s, 3.5s/1.75s}.

for the MARMV model a sequence of model parameters need to be saved for each music snippet. The computational workload though, is a little larger for the MARPPK model when compared to the MARMV model.

Figure 2 (lower) shows the accuracy on each of the 11 genres of the two models MARMV and MARPPK. The MARPPK seem to be more robust in classifying all genres, whereas the MARMV is much better at specific genres such as *Rap & Hiphop* and *Reggae*. However, the MARMV does not capture any of the *Rock* pieces, but generally confuses them with *Alternative* (not shown here). Also illustrated in this figure is the human performance in the different classes. A confidence interval of 75% has been shown on the human performance, due to the few test persons involved in the test. The humans are much better at genres such as *Rap & Hiphop* and *Reggae* than, e.g. *Alternative*, which also corresponds to some of the behavior observed with the MARMV method.

5 CONCLUSION

The purpose of this work has partly been to illustrate the importance of modelling the temporal structure in the short time features, and secondly how models of short time features can be integrated into kernels, such that the support vector machine can be applied. In the music genre setup the best performance was achieved with the MAR model in a product probability kernel (MARPPK) used in combination with an SVM and with the MAR model used in combination with majority voting (MARMV) in a linear neural network. The average accuracy of these two methods were $\sim 43\%$ compared to a human average accuracy of $\sim 52\%$.

Even though the results presented in this article were a music genre setup, the general idea of feature integration and generating a kernel function, which efficiently evaluates the difference between audio-models can be generalized and used in other fields of *MIR*.

ACKNOWLEDGEMENTS

The work was 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] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5), July 2002.

[2] A. Meng, P. Ahrendt, and J. Larsen. Improving music genre classification by short-time feature integration. In *Proc. of ICASSP*, pages 1293–1296, 2005.

[3] D. Perrot and R. Gjerdigen. Scanning the dial: An exploration of factors in identification of musical style. In *Proc. of Soc. Music Perception Cognition*, 1999.

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

[5] M. F. McKinney and J. Breebaart. Features for audio and music classification. In *Proc. of ISMIR*, pages 151–158, 2003.

[6] J-J. Aucouturier and F. Pachet. Music similarity measures: What's the use? In *Proc. of ISMIR*, 2002.

[7] Helmut Lütkepohl. *Introduction to Multiple Time Series Analysis*. Springer-Verlag, 1993.

[8] A. Neumaier and T. Schneider. Estimation of parameters and eigenmodes of multivariate autoregressive models. *ACM Transactions on Mathematical Software*, 27(1):27–57, March 2001.

[9] T. Jebara, R. Kondor, and A. Howard. Probability product kernels. *Journal of Machine Learning Research*, pages 819–844, July 2004.

[10] P. J. Moreno, P. P. Ho, and N. Vasconcelos. A kullback-leibler divergence based kernel for svm classification in multimedia applications. In *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004. MIT Press.

[11] L. Lu, S. Z. Li, and Zhang H.-J. Content-based audio segmentation using support vector machines. In *ACM Multimedia Systems Journal*, volume 8, pages 482–492, March 2003.

[12] S.-Z. Li and G. Guo. Content-based audio classification and retrieval using svm learning. In *First IEEE Pacific-Rim Conference on Multimedia, Invited Talk, Australia*, 2000.

[13] N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK, 2000.

[14] J. Shawe-Taylor and N. Cristianini. On the generalisation of soft margin algorithms. *IEEE Transactions on Information Theory*, 48(10):2721–2735, 2002.

[15] C-C. Chang and C-J. Lin. Libsvm : A library for support vector machines, 2001. (http://www.csie.ntu.edu.tw/ cjlin/libsvm).

[16] David Haussler. Convolution kernels on discrete structures. Technical report, University of California at Santa Cruz, July 1999.

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

[18] 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.

[19] T.G Diettereich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, (10):1895–1924, 1998.

Appendix G

Contribution: MIREX 2005

Music genre classification using the multivariate AR feature integration model, ISMIR content : Music Information Retrieval Evaluation eXchange. Author list: P. Ahrendt and A. Meng. Own contribution estimated to approximately 50%.

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

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

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

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

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.

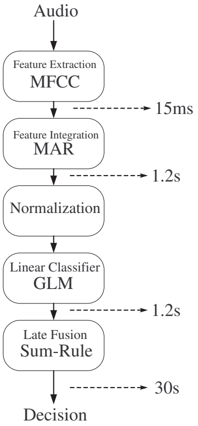


Figure 1: Overview of system from audio to a genre decision at 30s. The time scale at each step is indicated to the right.

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 algorithm, 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

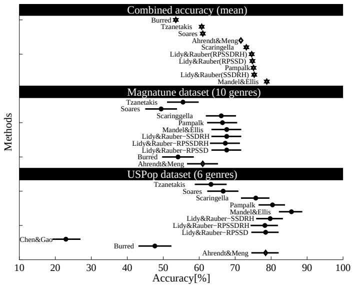
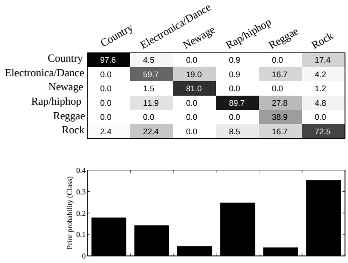


Figure 2: Mean accuracy on both USPop and Magnatune data sets illustrated with a 95% binomial confidence interval. The ”Combined accuracy” is the mean accuracy on the two data sets.

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.

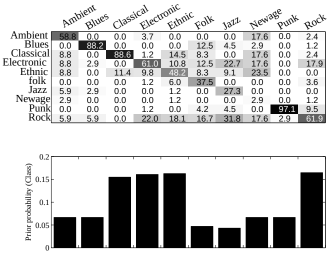
In order to make a proper statistical comparison of the different methods the raw classifications should have been known.



	Country	Electronica/Dance	Newage	Rap/hiphop	Reggae	Rock
Country	97.6	4.5	0.0	0.9	0.0	17.4
Electronica/Dance	0.0	59.7	19.0	0.9	16.7	4.2
Newage	0.0	1.5	81.0	0.0	0.0	1.2
Rap/hiphop	0.0	11.9	0.0	89.7	27.8	4.8
Reggae	0.0	0.0	0.0	0.0	38.9	0.0
Rock	2.4	22.4	0.0	8.5	16.7	72.5

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%.



	Ambient	Blues	Classical	Electronic	Ethnic	Folk	Jazz	Newage	Punk	Rock
Ambient	58.8	0.0	0.0	3.7	0.0	0.0	0.0	17.6	0.0	2.4
Blues	0.0	88.2	0.0	0.0	0.0	12.5	4.5	2.9	0.0	1.2
Classical	8.8	0.0	88.6	1.2	14.5	8.3	0.0	17.6	0.0	2.4
Electronic	8.8	2.9	0.0	61.0	10.8	12.5	22.7	17.6	0.0	17.9
Ethnic	8.8	0.0	11.4	9.8	48.2	8.3	9.1	23.5	0.0	0.0
folk	0.0	0.0	0.0	1.2	6.0	37.5	0.0	0.0	0.0	3.6
Jazz	5.9	2.9	0.0	0.0	1.2	0.0	27.3	0.0	0.0	0.0
Newage	2.9	0.0	0.0	0.0	1.2	0.0	0.0	2.9	0.0	1.2
Punk	0.0	0.0	0.0	1.2	0.0	4.2	4.5	0.0	97.1	9.5
Rock	5.9	5.9	0.0	22.0	18.1	16.7	31.8	17.6	2.9	61.9

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 H

Contribution: IEEE 2006

Temporal feature integration for music genre classification. Submitted to IEEE Trans. on Signal Processing. Author list: A. Meng, P. Ahrendt, J. Larsen and L. K. Hansen. Own contribution estimated to 40%.

Temporal 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: Temporal 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.

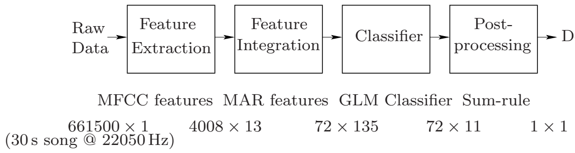


Figure 1. The full music genre classification system. The flow-chart illustrates the different parts of the system, whereas the names just below the chart are the specific choices that gives the best performing system. The numbers in the bottom part of the figure illustrates the (large) dimensionality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

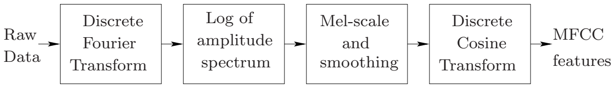

Raw Data
Discrete Fourier Transform
Log of amplitude spectrum
Mel-scale and smoothing
Discrete Cosine Transform
MFCC features

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

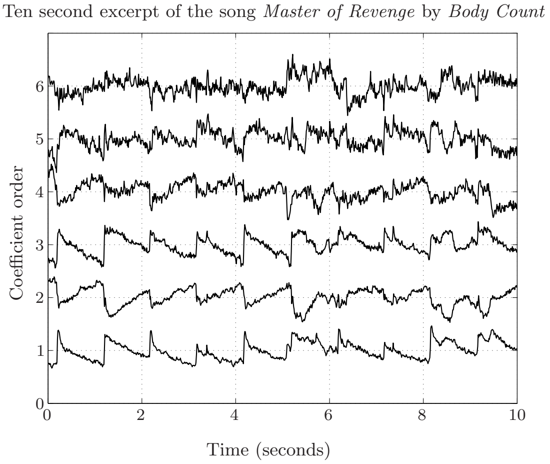


Figure 3. The first six normalized MFCCs of a ten second snippet of "Body Count - Masters of Revenge". The temporal correlations is very clear from this piece of music as well as the cross-correlations among the feature dimensions. This suggests that relevant information is present and could be extracted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

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

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 $\frac{D}{2}(3+D)$.

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

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

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

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) $1 - 2\,\mathrm{Hz}$ modulation energy, 3) $3 - 15\,\mathrm{Hz}$ modulation energy and 4) $20 - 43\,\mathrm{Hz}$ 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.

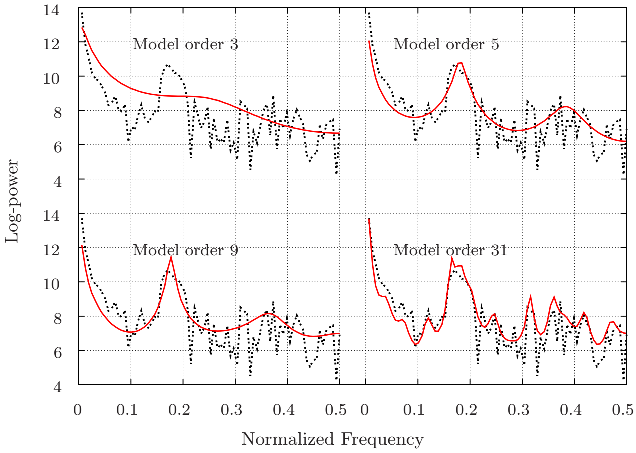


Figure 4. Power density of a first order MFCC of a piano note $A5$ played for a duration of 1.2 s. The four figures show the periodogram as well as the AR-model power spectrum estimates of orders $3, 5, 9$ and 31, respectively.

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	$\frac{1}{3}(PD+1)^3+$ $\big((P+4+\frac{2}{D})(PD+1)+(D+2)\big)\,DF_s$

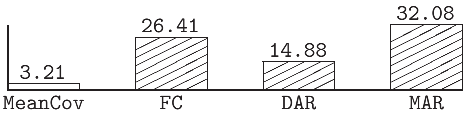


Figure 5. Computational complexity of the music genre setup using the optimized values from the experimental section, hence $P = 3$, $D = 6$ and $F_s = 188, 268, 322, 188, 162$ for the MeanVar, MeanCov, FC, DAR and MAR, respectively. Note that the complexity values are scaled such that the MeanVar has 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.

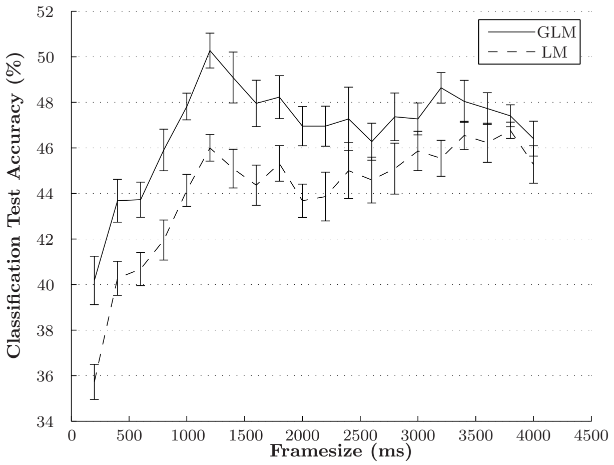


Figure 6. Classification test accuracy is plotted against framesize for the MAR features using the LM and GLM classifiers. The hopsize was 200 ms in these experiments and data set B, section 4.6, was used. The importance of the framesize is clearly seen. The baseline classification accuracy by random 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

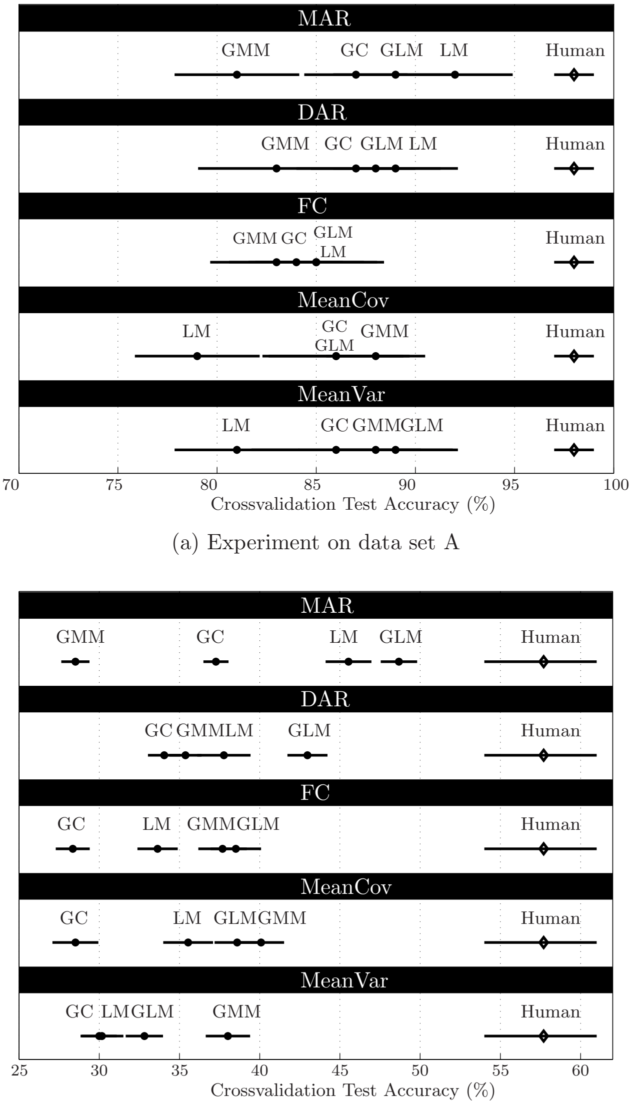
The main classification results are illustrated in figure 7 for both the small and the large data set. The figure compares the classification test accuracies of the FC and MeanCov features and the baseline MeanVar with the newly proposed DAR and MAR features. It is difficult to see much difference in 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



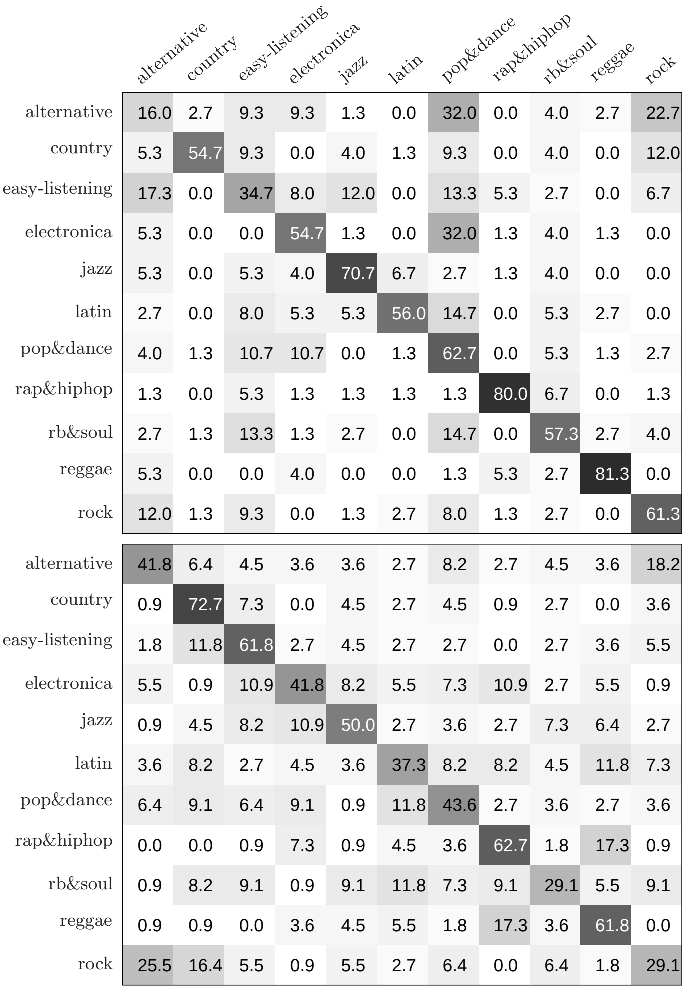
(a) Experiment on data set A

(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.



	alternative	country	easy-listening	electronica	jazz	latin	pop&dance	rap&hiphop	rb&soul	reggae	rock
alternative	16.0	2.7	9.3	9.3	1.3	0.0	32.0	0.0	4.0	2.7	22.7
country	5.3	54.7	9.3	0.0	4.0	1.3	9.3	0.0	4.0	0.0	12.0
easy-listening	17.3	0.0	34.7	8.0	12.0	0.0	13.3	5.3	2.7	0.0	6.7
electronica	5.3	0.0	0.0	54.7	1.3	0.0	32.0	1.3	4.0	1.3	0.0
jazz	5.3	0.0	5.3	4.0	70.7	6.7	2.7	1.3	4.0	0.0	0.0
latin	2.7	0.0	8.0	5.3	5.3	56.0	14.7	0.0	5.3	2.7	0.0
pop&dance	4.0	1.3	10.7	10.7	0.0	1.3	62.7	0.0	5.3	1.3	2.7
rap&hiphop	1.3	0.0	5.3	1.3	1.3	1.3	1.3	80.0	6.7	0.0	1.3
rb&soul	2.7	1.3	13.3	1.3	2.7	0.0	14.7	0.0	57.3	2.7	4.0
reggae	5.3	0.0	0.0	4.0	0.0	0.0	1.3	5.3	2.7	81.3	0.0
rock	12.0	1.3	9.3	0.0	1.3	2.7	8.0	1.3	2.7	0.0	61.3

alternative	41.8	6.4	4.5	3.6	3.6	2.7	8.2	2.7	4.5	3.6	18.2
country	0.9	72.7	7.3	0.0	4.5	2.7	4.5	0.9	2.7	0.0	3.6
easy-listening	1.8	11.8	61.8	2.7	4.5	2.7	2.7	0.0	2.7	3.6	5.5
electronica	5.5	0.9	10.9	41.8	8.2	5.5	7.3	10.9	2.7	5.5	0.9
jazz	0.9	4.5	8.2	10.9	50.0	2.7	3.6	2.7	7.3	6.4	2.7
latin	3.6	8.2	2.7	4.5	3.6	37.3	8.2	8.2	4.5	11.8	7.3
pop&dance	6.4	9.1	6.4	9.1	0.9	11.8	43.6	2.7	3.6	2.7	3.6
rap&hiphop	0.0	0.0	0.9	7.3	0.9	4.5	3.6	62.7	1.8	17.3	0.9
rb&soul	0.9	8.2	9.1	0.9	9.1	11.8	7.3	9.1	29.1	5.5	9.1
reggae	0.9	0.9	0.0	3.6	4.5	5.5	1.8	17.3	3.6	61.8	0.0
rock	25.5	16.4	5.5	0.9	5.5	2.7	6.4	0.0	6.4	1.8	29.1

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] P. Ahrendt. *Music Genre Classification Systems - A Computational Approach.* PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2006. Thesis not yet published.

[2] P. Ahrendt, C. Goutte, and J. Larsen. Co-occurrence models in music genre classification. In *IEEE International workshop on Machine Learning for Signal Processing*, Mystic, Connecticut, USA, Sept. 2005.

[3] P. Ahrendt and A. Meng. Music genre classication using the multivariate AR feature integration model. Music Information Retrieval Evaluation Exchange (MIREX), Sept. 2005.

[4] P. Ahrendt, A. Meng, and J. Larsen. Decision time horizon for music genre classification using short time features. In *EUSIPCO*, pages 1293–1296, Vienna, Austria, sept. 2004.

[5] E. Allamanche and J. Herre. Content-based identification of audio material using MPEG-7 low level description. In *ISMIR*, 2003.

[6] M. Athineos, H. Hermansky, and D. P. W. Ellis. LP-TRAP: Linear predictive temporal patterns. In *Proc. of International Conference on Spoken Language Processing (ICSLP)*, Oct. 2004.

[7] J.-J. Aucouturier and F. Pachet. Representing musical genre: A state of the art. *Journal of New Music Research*, 32(1):83–93, 2003.

[8] J. J. Aucouturier and F. Pachet. Improving timbre similarity: How high is the sky? *Journal of Negative Results in Speech and Audio Sciences*, 1(1), 2004.

[9] J.-J. Aucouturier, F. Pachet, and M. Sandler. The way it sounds : Timbre models for analysis and retrieval of polyphonic music signals. *IEEE Trans. on Multimedia*, 7(6):8, December 2005. (in press).

[10] Ulas Bagcl and Engin Erzin. Boosting classifiers for music genre classification. In *Lecture Notes in Computer Science*, volume 3733, 2005.

[11] A. Berenzweig, D. P. W. Ellis, and S. Lawrence. Using voice segments to improve artist classification of music. In *AES 22nd International Conference on Virtual, Synthetic and Entertainment Audio*, 2002.

[12] A. Berenzweig, B. Logan, D. P. W. Ellis, and B. Whitman. A large-scale evaluation of acoustic and subjective music similarity measures. *Computer Music Journal*, 28:63–76, 2004.

[13] J. Bergstra, N. Casagrande, and D. Eck. MIREX contests on music genre classification, 2005.

[14] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kégl. Meta-features and adaboost for music classification. *Machine Learning Journal : Special Issue on Machine Learning in Music*, 2006.

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

[16] M. Brookes. VOICEBOX (a MATLAB toolbox for speech processing), 1997.

[17] C.J.C. Burges, J.C. Platt, and S. Jana. Distortion discriminant analysis for audio fingerprinting. *IEEE Trans. on Speech and Audio Processing*, 11(3):165–174, May 2003.

[18] J.J. Burred and A. Lerch. A hierarchical approach to automatic musical genre classification. In *Proceedings of the 6th International Conference on Digital Audio Effects*, London, Sept. 2003.

[19] P. Cano, E. Batlle, T. Kalker, and J. Haitsma. A review of algorithms for audio fingerprinting. In *International Workshop on Multimedia Signal Processing*, 2002.

[20] P. Cano, E. Batlle, T. Kalker, and J. Haitsma. A review of audio fingerprinting. *Journal of VLSI Signal Processing*, 41:271–284, 2005.

[21] P. Cano, E. Batlle, H. Mayer, and H. Neuschmied. Robust sound modeling for song detection in broadcast audio. In *Proceedings of 112th AES Convention*, 2002.

[22] P. Cano, M. Koppenberger, S. Ferradans, A. Martinez, F. Gouyon, V. Sandvold, V. Tarasov, and N. Wack. MTG-DB: a repository for music audio processing. In *Proceedings of 4th International Conference on Web Delivering of Music*, Barcelona, Spain, 2004.

[23] M. Casey. MPEG-7 sound recognition tools. *IEEE Trans. on Curcuits and Systems for Video Technology*, 11(6):737–747, June 2001.

[24] O. Celma, E. Gómez, J. Janer, F. Gouyon, P. Herrera, and D. Garcia. Tools for content-based retrieval and transformation of audio using MPEG-7: The SPOffline and the MDTools. In *Proceedings of 25th International AES Conference*, London, UK, 2004.

[25] C-C. Chang and C-J. Lin. LIBSVM: a library for support vector machines, 2001. (http://www.csie.ntu.edu.tw/ cjlin/libsvm).

[26] D. Cliff. Hang the DJ: Automatic sequencing and seamless mixing of dance-music tracks. Technical report, HPL-2000-104, 20000809, 2000.

[27] D. R. Cox and H. D. Miller. *The Theory of Stochastic Processes.* Chapman & Hall, 1995.

[28] N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines.* Cambridge University Press, Cambridge, UK, 2000.

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

[30] John R. Deller, John H. Hansen, and John G. Proakis. *Discrete Time Processing of Speech Signals.* IEEE Press Marketing, 2000.

[31] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 34:1–38, 1977.

[32] F. Desobry, M. Davy, and W.J. Fitzgerald. A class of kernels for sets of vectors. In *ESANN*, April 2005.

[33] R. Dhanaraj and B. Logan. Automatic prediction of hit songs. In *International Symposium on Music Information Retrieval*, pages 488–491, 2005.

[34] C. Dietrich, F. Schwenker, and G. Palm. Classification of time series utilizing temporal and decision fusion. In *Proceedings of Multiple Classifier Systems (MCS), Springer - LNCS 2096*, pages 378–387, 2001.

[35] T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, Oct. 1998.

[36] T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

[37] S. Dixon, W. Goebl, and G. Widmer. The performance worm: Real time visualisation of expression based on langner's tempo-loudness animation. In *Proceedings of the International Computer Music Conference (ICMC), Göteborg, Sweden*, 2002.

[38] J. S. Downie, K. West, A. Ehmann, and E. Vincent. The 2005 music information retrieval evaluation exchange (MIREX 2005): Preliminary overview. In *International Symposium of Music Information Retrieval*, 2005.

[39] S. Dubnov. Generalization of spectral flatness measure for non - gaussian linear processes. *IEEE Signal Processing Letters*, 11:698–701, 2004.

[40] S. Dubnov and X. Rodet. Investigation of phase coupling phenomena in sustained portion of musical instruments sound. *Journal of Acoustic Society of America*, 112(6):348–359, Sept. 2003.

[41] R. O. Duda and P. E. Hart. *Pattern Classsification and Scene Analysis.* New York : John Wiley, 1973.

[42] J. Eichhorn and O. Chapelle. Object categorization with SVM: Kernels for local features. Technical report, Max-Planck-Institut für biologische Kybernetic, 2004.

[43] K. El-Maley, M. Klein, G. Petrucci, and P. Kabal. Speech/music discrimination for multimedia applications. In *Proceedings of ICASSP*, 2000.

[44] D. Ellis and K.S. Lee. 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*, pages 1–6, Jeju, Korea, Oct. 2004.

[45] S. Essid, G. Richard, and B. David. Instrument recognition in polyphonic music based on automatic taxonomies. *IEEE Trans. on Speech and Audio Processing*, 14(1):68–80, January 2006.

[46] W. M. Fisher, G. R. Doddington, and K. M. Goudie-Marshall. The DARPA speech recognition research database : specifications and status. In *proceedings of the DARPA workshop on speech recognition*, pages 93–99, 1986.

[47] A. Flexer, E. Pampalk, and G. Widmer. Hidden markov models for spectral similarity of songs. In *International conference on digital audio effects (DAFx'05)*, 2005.

[48] J. Foote and M. Cooper. Media segmentation using self-similarity decomposition. In *SPIE Storage and Retrieval for Multimedia Databases*, volume 5021, pages 167–175, Jan. 2003.

[49] J. Foote, M. Cooper, and U. Nam. Audio retrieval by rhythmic similarity. In *International Symposium on Music Information Retrieval*, Oct. 2002.

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

[51] J. T. Foote. Content-based retrieval of music and audio. In *Multimedia Storage and Archiving Systems II, Proc. of SPIE*, volume 3229, pages 138–147, 1997.

[52] A. Frieze, R. Kannan, and S. Vempala. Fast monte-carlo algorithms for finding low-rank approximations. In *Proceedings on the Foundations of Computer Science*, pages 370–378, 1998.

[53] M. Goto. An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research*, 30:159–171, 2001.

[54] F. Gouyon and S. Dixon. A review of automatic rhythm description systems. *Computer Music Journal*, 29(1):34–54, 2005.

[55] A. B. A. Graf and S. Borer. Normalization in support vector machines. In *DAGM-Symposium*, pages 277–282, 2001.

[56] J. M. Grey. Multidimensional perceptual scaling of musical timbres. *The Journal of the Acoustical Society of America*, 61(4):1270–1277, May 1977.

[57] G. Guodong and S. Z. Li. Content-based audio classification and retrieval by support vector machines. *IEEE Trans. on Neural Networks*, 14:209–215, 2000.

[58] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[59] J. Haitsma and T. Kalker. A highly robust audio fingerprinting system. In *International Symposium on Music Information Retrieval*, 2002.

[60] J. H. L. Hansen and B. D. Womack. Feature analysis and neural network-based classification of speech under stress. *IEEE Trans. on Speech Audio Processing*, 4:307–313, 1996.

[61] A. Härma, M. Karjalainen, L. Savioja, V. Välimäki, U. K. Laine, and J. Huopaniemi. Frequency-warped signal processing for audio applications. *Journal of the Audio Engineering Society*, 48:1011–1031, 2000.

[62] David Haussler. Convolution kernels on discrete structures. Technical report, University of California at Santa Cruz, July 1999.

[63] R. Herbrich and T. Graepel. A PAC-Bayesian margin bound for linear classifiers: Why SVMs work. In *Advances in Neural Information Systems 13*, 2001.

[64] H. Hermansky. Perceptual linear predictive (PLP) analysis of speech. *J. Acoust. Soc. Am*, 87:1738–1752, 1990.

[65] H. Homburg, I. Mierswa, B. Möller, K. Morik, and M. Wurst. A benchmark dataset for audio classification and clustering. In *International Symposium on Music Information Retrieval*, pages 528–531, 2005.

[66] H.Soltau, T. Schultz, M. Westphal, and A. Waibel. Recognition of music types. In *ICASSP*, 1998.

[67] X. Hu, J. S. Downie, K. West, and A. Ehmann. Mining music reviews: Promising preliminary results. In *Proceedings of ISMIR*, pages 536–539, 2005.

[68] ISO/IEC. FDIS 15938-4 Multimedia Content Description Interface - Audio Part. ISO/IEC, 2002.

[69] T. S. Jaakola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems II*, pages 487–493, July 1999.

[70] T. Jebara, R. Kondor, and A. Howard. Probability product kernels. *Journal of Machine Learning Research 5*, pages 819–844, July 2004. Southampton, Kernels.

[71] H. D. Jennings, P. C. Ivanov, A. M. De Martins, P. C. da Silva, and G. M. Viswanathan. Variance fluctuations in nonstationary time series: a comparative study of music genres. *Physica A Statistical Mechanics and its Applications*, 336:585–594, May 2004.

[72] K. Jensen. *Timbre Models of Musical Sounds*. PhD thesis, DIKU Report 99/7, 1999.

[73] S. Kay. A zero-crossing based spectrum analyzer. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 34:96–104, 1986.

[74] B. Kedem. Spectral analysis and discrimination by zero crossing. *Proc. of IEEE*, 74(11):1477–1493, 1986.

[75] Y. E. Kim and B. Whitman. Singer identification in popular music recordings using voice coding features. In *International Symposium on Music Information Retrieval*, 2002.

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

[77] U. Kjems, L. K. Hansen, and J. Anderson. The quantitative evaluation of functional neuroimaging experiments: Mutual information learning curves. *NeuroImage*, 15(4):772–786, 2002.

[78] P. Knees, E. Pampalk, and G. Widmer. Automatic classification of musical artists based on web-data. *ÖGAI Journal*, 24(1):16–25, 2005.

[79] W. Ku, R. H. Storer, and C. Georgakis. Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 30(1):179–196, November 1995.

[80] T. Lambrou, P. Kudumakis, R. Speller, M. Sandler, and A. Linney. Classification of audio signals using statistical features on time and wavelet transform domains. In *ICASSP*, pages 3621–3624, May 1998.

[81] D. Lewis. Reuters-21578 text categorization test collection. Distribution 1.0, AT&T Labs-Research, http://www.research/.att.com, 1997.

[82] D. Li, I.K. Sethi, N. Dimitrova, and T. McGee. Classification of general audio data for content-based retrieval. *Pattern Recognition Letters*, 22:533–544, 2001.

[83] M. Li and R. Sleep. Genre classification via an LZ78-based string kernel. In *International Symposium on Music Information Retrieval*, 2005.

[84] T. Li and M. Ogihara. Music genre classification with taxonomy. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 197–200, March 2005.

[85] T. Li, M. Ogihara, and Q. Li. A comparative study on content-based music genre classification. In *ACM SIGIR*, pages 282–289, 2003.

[86] S. Lippens, J.P Martens, T. D. Mulder, and G. Tzanetakis. A comparison of human and automatic musical genre classification. In *Proc. of ICASSP*, pages 233–236, May 2004.

[87] L. Lu and H.-J. Zhang adn S. Z. Li. Content-based audio classification and segmentation by using support vector machines. *Multimedia Systems*, 8:482–492, 2003.

[88] L. Lu, H-J. Zhang, and H. Jiang. Content analysis for audio classification and segmentation. *IEEE Trans. on Speech and Audio Processing*, 10(7):504–516, October 2002.

[89] H. Lütkepohl. *Introduction to Multiple Time Series Analysis.* Springer, 2nd edition, 1993.

[90] D. J. C. Mackay. *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, 2003.

[91] J. Makhoul. Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4):561–580, 1975.

[92] M. Mandel and D. Ellis. MIREX contest: Music genre classification, 2005.

[93] M. Mandel and D. Ellis. Song-level features and support vector machines for music classification. In *International Symposium on Musical Information Retrieval*, 2005.

[94] M. Mandel, G. Poliner, and D. Ellis. Support vector machine active learning for music retrieval. *Accepted for publication in ACM Multimedia Systems Journal*, 2006.

[95] B.S. Manjunath, P. Salembier, and T. Sikora. *Introduction to MPEG-7 - Multimedia Content Description Interface.* John Wiley & Sons Ltd., 2002.

[96] K. Martin. Towards automatic sound source recognition: Identifying musical instruments. In *NATO Computational Hearing Advanced Study Institute*, 1998.

[97] M. F. McKinney and J. Breebart. Features for audio and music classification. In *International Symposium on Music Information Retrieval*, pages 151–158, 2003.

[98] A. Meng, P. Ahrendt, and J. Larsen. Improving music genre classification by short-time feature integration. In *proc. of ICASSP*, pages 1293–1296, 2005.

[99] A. Meng, P. Ahrendt, J. Larsen, and L. K. Hansen. Temporal feature integration for music genre classification. Submitted to IEEE Trans. on Speech and Signal Processing, 2006.

[100] A. Meng and J. Shawe-Taylor. An investigation of feature models for music genre classification using the support vector classifier. In *International Conference on Music Information Retrieval*, pages 604–609, 2005.

[101] I. Mierswa and K. Morik. Automatic feature extraction for classifying audio data. *Machine Learning Journal*, 58:127–149, 2005.

[102] P. J. Moreno, P. P. Ho, and N. Vasconcelos. A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications. In *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2003.

[103] P. J. Moreno, P. P. Ho, and N. Vasconcelos. A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications. In *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004.

[104] P. J. Moreno and R. Rifkin. Using the Fisher kernel method for web audio classification. In *Proc. of ICASSP*, 2000.

[105] F. Mörchen, A. Ultsch, M. Thies, I. Löhken, M. Nöcker, C. Stamm, N. Efthymiou, and M. Kümmerer. Musicminer: Visualizing timbre distances of music as topographical maps. Technical report, University of Marburg, 2005.

[106] I. Nabney and C. Bishop. NETLAB package. http://www.ncrg.aston.ac.uk/netlab/index.php, 1995.

[107] A. Neumaier and T. Schneider. Estimation of parameters and eigenmodes of multivariate autoregressive models. *ACM Trans. on Mathematical Software*, 27(1):27–57, March 2001.

[108] F. Pachet. Knowledge management and musical metadata. Encyclopedia of Knowledge Management, 2005.

[109] F. Pachet and D. Cazaly. A taxonomy of musical genres. In *Content-Based Multimedia Information Access Conference (RIAO)*, 2000.

[110] E. Pampalk, A. Flexer, and G. Widmer. Improvements of audio-based music similarity and genre classification. In *International Symposium on Music Information Retrieval*, 2005.

[111] M. S. Pedersen, T. Lehn-Schiøler, and J. Larsen. BLUES from music: Blind underdetermined extraction of sources from music. In *ICA2006*, volume 3889 of *Lecture Notes in Computer Science*, pages 392–399. Springer Berlin / Heidelberg, 2006.

[112] K. B. Petersen and M. S. Pedersen. The Matrix Cookbook. http://www2.imm.dtu.dk/pubdb/p.php?3274, 2005. Version: 20051003.

[113] J. C. Platt, C. J. C. Burges, S. Swenson, C. Weare, and A. Zheng. Learning a Gaussian process prior for automatically generating music playlists. In *Advances in Neural Information Processing Systems 14*, pages 1425–1432, 2002.

[114] T. Pohle, E. Pampalk, and G. Widmer. Evaluation of frequently used audio features for classification of music into perceptual categories. In *Workshop on Content-Based Multimedia Indexing*, 2005. Riga, Latvia.

[115] J. G. Proakis and D. G. Manolakis. *Digitial Signal Processing, Principles, Algorithms, and Applications.* Prentice-Hall International, Inc., 3 edition, 1996.

[116] L. R. Rabiner and R. W. Schafer. *Digital Processing of Speech Signals.* Prentice-Hall, 1978.

[117] M. J. Reyes-Gomez and D. Ellis. Selection, parameter estimation, and discriminative training of Hidden Markov Models for general audio modeling. In *Proc. of ICME*, pages 73–76, 2003.

[118] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

[119] D. W. Robinson and R. S. Dadson. A re-determination of the equal-loudness relations for pure tones. *Br. J. Appl. Phys. 7*, pages 166–181, 1956.

[120] S. Roweis and Z. Ghahramani. A unifying review of linear Gaussian models. *Neural Computation*, 11:305–345, 1999.

[121] Y. Bartlett S. R. Freund. Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics*, 26(5):1651–1686, 1998.

[122] E. Salamon, S. R Bernstein, S.-A. Kim, M. Kim, and G. B. Stefano. The effects of auditory perception and musical preference on anxiety in naive human subjects. *Med. Sci. Monit.*, 9(9):CR396–CR399, Sept. 2003.

[123] D. Salomon. *Data Compression: The Complete Reference.* Springer, second edition, 2000.

[124] C. Saunders, D. R. Hardoon, J. Shawe-Taylor, and G. Widmer. Using string kernels to identify famous performers from their playing style. In *Proceedings of The 15th European Conference on Machine Learning (ECML)*, pages 384–395, 2004.

[125] J. Saunders. Real-time discrimination of broadcast speech/music. In *International Conference on Acoustics, Speech and Signal Processing*, pages 993–996, 1996.

[126] N. Scaringella and G. Zoia. On the modeling of time information for automatic genre recognition systems in audio signals. In *International Symposium on Music Information Retrieval*, 2005.

[127] E. D. Scheirer. Structured audio and effects processing in the MPEG-4 multimedia standard. *Multimedia Systems special issue on audio and multimedia*, 7:11 – 22, 1999.

[128] Eric D. Scheirer. Tempo and beat analysis of acoustic musical signals. *Acoustical Society of America.*, 103:588–601, January 1998.

[129] H. Schweitzer. A distributed algorithm for content based indexing of images by projections on Ritz primary images. *Data Mining and Knowledge Discovery*, 1:375–390, 1997.

[130] W. A. Sethares, R. D. Morris, and J. C. Sethares. Beat tracking of musical performances using low-level audio features. *IEEE Trans. on Speech and Audio Processing*, 13(2), March 2005.

[131] J. Shawe-Taylor and N. Cristianini. On the generalisation of soft margin algorithms. *IEEE Trans. on Information Theory*, 48(10):2721–2735, 2002.

[132] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

[133] S. Sigurdsson, P. A. Philipsen, L. K. Hansen, J. Larsen, M. Gniadecka, and H. C. Wulf. Detection of skin cancer by classification of raman spectra. *IEEE Trans. of Biomedical Engineering*, 51:1784–1793, 2004.

[134] M. Slaney. Mixtures of probability experts for audio retrieval and indexing. In *IEEE International Conference on Multimedia and Expo*, August 2002.

[135] M. Slaney. Semantic- audio retrieval. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2002.

[136] P. Stoica and Y. Selén. Model-order selection: a review of information criterion rules. *IEEE Signal Processing Magasine*, pages 36–47, 2004.

[137] S. Streich and P. Herrera. Detrended fluctuation analysis of music signals: Danceability estimation and further semantic characterization. In *Proceedings of 118th Audio Engineering Society Convention*, Barcelona, 2005.

[138] H. W. Strube. Linear prediction on a warped frequency scale. *The Journal of the Acoustical Society of America*, 68:1071–1076, 1980.

[139] H. Terasawa, M. Slaney, and J. Berger. Perceptual distance in timbre space. In *International Conference on Auditory Display*, pages 61–68, 2005.

[140] T. Tolonen and M. Karjalainen. A computationally efficient multipitch analysis model. *IEEE Trans. on Speech and Audio Processing*, 8:708–716, 2000.

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

[142] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Trans. on speech and audio processing*, 10(5):293–302, July 2002.

[143] G. Tzanetakis, G. Essl, and P. Cook. Audio analysis using the discrete wavelet transform. In *WSES International Conference on Acoustics and Music: Theory and Applications*, 2001.

[144] V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1999.

[145] J. Wellhausen and M. Höynck. Audio thumbnailing using MPEG-7 low level audio descriptors. In *Internet Multimedia Management Systems IV*, pages 65–73, 2003.

[146] K. West and S. Cox. Finding an optimal segmentation for audio genre classification. In *International Symposium on Music Information Retrieval*, pages 680–685, 2005.

[147] C. K. I. Williams. How to pretend that correlated variables are independent by using difference observations. *Neural Computation*, 17(1):1–6, 2005.

[148] E. Wold, T. Blum, D. Keislar, and J. Wheaton. Content-based classification, search, and retrieval of audio. *IEEE Multimedia*, 3(3):27–36, 1996.

[149] C. Xu, N. C. Maddage, X. Shao, F. Cao, and Q. Tian. Musical genre classification using support vector machines. In *Proc. of ICASSP*, pages 429–432, 2003.

[150] T. Zhang and C.-C. Jay Kuo. Audio content analysis for online audiovisual data segmentation and classification. *IEEE Trans. on Speech and Audio Processing*, 9(4):441–457, 05 2001.

[151] J. M. Zurada, A. Malinowski, and I. Cloete. Sensitivity analysis for minimization of input data dimension for feedforward neural network. In *IEEE Symposium on Circuits and Systems*, volume 6, pages 447–450, 1994.