NATIONAL & KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE

DEPARTMENT OF INFORMATICS & TELECOMMUNICATIONS

PROGRAM OF POSTGRADUATE STUDIES

PhD DISSERTATION

Study and application of acoustic information for the detection of harmful content, and fusion with visual information

Theodoros D. Giannakopoulos

ATHENS

JULY 2009

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Μελέτη και χρήση ακουστικής πληροφορίας για τον εντοπισμό επιβλαβούς περιεχομένου και ενσωμάτωση σε οπτική πληροφορία

Θεόδωρος Δ. Γιαννακόπουλος

ΑΘΗΝΑ

ΙΟΥΛΙΟΣ 2009

PhD DISSERTATION

Study and application of acoustic information for the detection of harmful content, and fusion with visual information

Theodoros D. Giannakopoulos

ADVISOR: Sergios Theodoridis, Professor NKUA

THREE-MEMBER ADVISING COMMITTEE:
Sergios Theodoridis, Professor NKUA
Stavros Perantonis, Senior Researcher, NCSR “Demokritos"
Emmanuil Sagkriotis, Associate Professor NKUA

SEVEN-MEMBER EXAMINATION COMMITTEE

Sergios Theodoridis
Professor NKUA

Stavros Perantonis
Senior Researcher, NCSR “Demokritos"

Nicolaos Fakotakis
Professor University of Patras

Kostas Berberidis
Professor University of Patras

Manolis Sagriotis
Associate Professor NKUA

Eleftheriadis Alexandros
Associate Professor NKUA

Aggelos Pikrakis
Lecturer University of Piraeus

Examination Date: 17/7/2009

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Μελέτη και χρήση ακουστικής πληροφορίας για τον εντοπισμό επιβλαβούς περιεχομένου και ενσωμάτωση σε οπτική πληροφορία

Θεόδωρος Δ. Γιαννακόπουλος

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Σέργιος Θεοδωρίδης Καθηγητής ΕΚΠΑ

ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ:
Σέργιος Θεοδωρίδης Καθηγητής ΕΚΠΑ
Σταύρος Περαντώνης Ερευνητής Α, ΕΚΕΦΕ "ΔΗΜΟΚΡΙΤΟΣ"
Εμμανουήλ Σαγκριώτης Αναπ. Καθηγητής, ΕΚΠΑ

ΕΠΤΑΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Σέργιος Θεοδωρίδης
Καθηγητής ΕΚΠΑ

Σταύρος Περαντώνης
Ερευνητής Α, ΕΚΕΦΕ "ΔΗΜΟΚΡΙΤΟΣ"

Νικόλαος Φακωτάκης
Καθηγητής Παν/μιου Πατρών

Κών/νος Μπερμπερίδης
Καθηγητής Παν/μίου Πατρών

Εμμανουήλ Σαγκριώτης
Αναπ. Καθηγητής, ΕΚΠΑ

Αλέξανδρος Ελευθεριάδης
Αναπληρωτής Καθηγητής ΕΚΠΑ

Άγγελος Πικράκης
Λέκτορας, Παν/μίου Πειραιώς

Ημερομηνία Εξέτασης: 17/7/2009

Περίληψη

Η παρούσα διατριβή σκοπεύει στην μελέτη και την ανάπτυξη τεχνικών για κατάτμηση και ταξινόμηση ηχητικών σημάτων, με βάση το περιεχόμενο, με έμφαση στην ανάλυση του περιεχομένου ταινιών. Επιπλέον, μέρος της διατριβής σχετίζεται με μεθόδους εντοπισμού ηχητικών κατηγοριών που σχετίζονται με περιεχόμενο *βίας* (π.χ. πυροβολισμοί, κραυγές, κ.α.).

Αρχικά παρουσιάζεται μία εισαγωγική διερεύνηση διαφόρων ηχητικών χαρακτηριστικών, και ο τρόπος διαφοροποίησής τους για ορισμένες ηχητικές κλάσεις. Το κύριο μέρος της παρούσας διατριβής αρχίζει με την παρουσίαση μίας νέας μεθόδου διαχωρισμού μουσικής - ομιλίας σε σήματα από ραδιοφωνικές εκπομπές, η οποία αντιμετωπίζει το πρόβλημα μέσω ενός αλγορίθμου μεγιστοποίησης πιθανοτήτων. Στην συνέχεια, παρουσιάζεται ένας αλγόριθμος *εντοπισμού* μουσικής σε ηχητικά σήματα από ταινίες.

Τα δύο προηγούμενα προβλήματα εμπεριέχουν τις έννοιες της κατάτμησης και της ταξινόμησης ηχητικών σημάτων. Μέρος της συγκεκριμένης διατριβής σχετίζεται με την μελέτη και την υλοποίηση μίας αποδοτικής μεθόδου κατάτμησης, η οποία αντιμετωπίζεται μέσω ενός προβλήματος κατηγοριοποίησης για τον *εντοπισμό αλλαγών στο περιεχόμενο* ενός σήματος. Στην συνέχεια προτείνεται ένας αλγόριθμος ταξινόμησης ηχητικών τμημάτων σε πολλαπλές κατηγορίες, οι οποίες επιλέχθηκαν έτσι ώστε να περιγράφουν περιεχόμενο βίας (π.χ. πυροβολισμοί) και μη βίας (π.χ. μουσική, ομιλία, κ.α.).

Στο τελευταίο τμήμα η διατριβή επικεντρώνεται στην αναγνώριση *συναισθηματικών καταστάσεων* με βάση την ομιλία. Η μέθοδος έχει εκτιμηθεί με χρήση ηχητικών τμημάτων ομιλίας από κινηματογραφικές ταινίες. Επιπλέον, προτείνεται μία μέθοδος χαρακτηρισμού ταινιών με βάση το συναισθηματικό περιεχόμενο της ομιλίας.

Τέλος, σημειώνεται ότι στην παρούσα διατριβή έχει δοθεί ιδιαίτερη έμφαση στην ανάπτυξη βάσεων ηχητικών δεδομένων, οι οποίες χρησιμοποιήθηκαν για εκπαίδευση και για δοκιμή των διαφόρων μεθόδων αναγνώρισης και κατάτμησης. Για τον σκοπό αυτό, έχουν σχηματιστεί δύο διαφορετικές κατηγορίες βάσεων δεδομένων. Η μία από ηχογραφήσεις που προέρχονται από ραδιοφωνικές εκπομπές, η οποία χρησιμοποιήθηκε στην μέθοδο διαχωρισμού μουσικής - ομιλίας. Η δεύτερη βάση δεδομένων περιέχει περισσότερες κατηγορίες (και υποκατηγορίες) ηχητικών σημάτων και δημιουργήθηκε από δεδομένα που προέρχονται από κινηματογραφικές ταινίες, ενώ χρησιμοποιήθηκε από τις υπόλοιπες μεθόδους της διατριβής.

Θεματική περιοχή: *Ανάλυση ηχητικής πληροφορίας*

Keywords: *Ταξινόμηση ήχων, ηχητική κατάτμηση, ανάλυση πολυμεσικής πληροφορίας, εντοπισμός βίας.*

Abstract

This thesis aims at investigating and developing techniques for content-based segmentation and classification of multimedia files, based on audio information. Emphasis has been given to analyzing the content of films based on audio information. In addition, part of the thesis is focused on the detection of audio classes related to *violent* content (e.g., gunshots, screams, etc).

An introductory investigation of several audio features is fist presented, into the context of their respective classification performance. The main part of the current thesis starts with a novel method for *speech-music discrimination* of radio broadcasts, which treats the problem as a maximization task. In this context, Bayesian Networks are used as probability estimators. Then an algorithm for locating the parts of an audio stream that contain music (i.e. *music tracking*) is presented.

Speech-music discrimination and music tracking build upon the concepts of both segmentation and classification. Thus, another major part of this thesis is related to the development of a computationally efficient audio *segmentation* algorithm, which treats the problem as a classification task for *detecting changes in an audio stream's content*. The purpose of this algorithm is to extract audio segments of *homogenous* content, which can then be fed as input to a classification scheme. In the sequel, a *multi-class classification* scheme for audio segments from films, is proposed in this thesis. The audio classes were selected to describe both violent (e.g., gunshots, screams) and non-violent (e.g., music, speech) content, while the method is based on a classifier combination technique.

In the final part, the focus on this thesis shifts on the task of recognizing the *emotional* state of the speaker given a speech segment. Towards this end, a regression approach has been proposed for mapping audio features to a dimensional representation of the affective content. This method has been evaluated on speech segments from movies. In addition, a method for movie characterization is proposed, based on the speech emotions detected in a movie.

Finally, an emphasis in this thesis was to develop annotated databases which were used, both for training and evaluating the several segmentation and classification methods. To this end, two different types of databases have been used. One from radio recordings, which was used in the speech-music discrimination task, and another one from a number of movies.

Subject Area: *Audio Analysis*

Keywords: *Audio classification, audio segmentation, multimedia analysis, violence detection.*

Acknowledgements

First of all, I would like to express my gratitude to my advisor, Professor Sergios Theodoridis (Department of Informatics and Telecommunications, University of Athens), for the support and trust he has shown during the last years. Apart from the scientific advise, knowledge and the several insightful discussions, I am particularly grateful for the possibility he gave me to conduct research with the required level of freedom. Also, I would like to express my gratitude to Dr. Aggelos Pikrakis (Lecturer, Department of Informatics, University of Piraeus), for his continuous guidance. Aggelos has provided me substantial help, through unnumbered discussions regarding both theoretical and practical issues related to my research work.

In addition, I am grateful for the scientific advices that Dr. Stavros Perantonis (Senior Researcher, Institute of Informatics and Telecommunications, National Center for Scientific Research "Demokritos") and Assistant Professor Manolis Sagriotis (Department of Informatics and Telecommunications, University of Athens) have offered me. Furthermore, I would like to express my gratitude to Professor Dionisis Cavouras (Dept. of Medical Instrumentation Technology, Technological Educational Institution of Athens) for his guidance to my graduate thesis in 2002. Professor Theodoridis and Professor Cavouras introduced me to the fields of signal and image analysis and pattern recognition when I was an undergraduate student in the Department of Informatics and Telecommunications. In addition, I would like to thank Professor Nikos Fakotakis (Electrical and Computer Engineering Department, University of Patras, Greece) for his guidance in my Master thesis in 2004.

Finally, I want to express my sincere appreciation to my colleagues and friends Panagiotis Tsakanikas, Iasonas Antonopoulos, Michael Mavroforakis, Harris Georgiou, Margaritis Sdralis, Kostas Rizogiannis, Stelios Tzikopoulos, Kostas Xenoulis, Alexandros Katsiotis, Gerasimos Mileounis, Alexandros Makris and Thanasis Perperis for their continuous cooperation and support.

List of Publications

This thesis has been based on material from the following papers:

Journals (2):

1. A. Pikrakis, T. Giannakopoulos, S. Theodoridis, "A Speech/Music Discriminator of Radio Recordings based on Dynamic Programming and Bayesian Networks", IEEE Transactions on Multimedia, Volume: 10 Issue: 5, Aug. 2008, Page(s): 846-857.

2. T. Giannakopoulos , A. Pikrakis, S. Theodoridis, "Speech emotion recognition in audio streams from movies", IEEE trans on Audio, Speech and Language Processing, (under review, IEEE Transactions on Speech, Audio and language Processing)

Book Chapters (1):

1. A. Pikrakis, T. Giannakopoulos, S. Theodoridis, "An Overview of Speech - Music Discrimination Techniques in the Context of Audio Recordings" in Multimedia Services in Intelligent Environments (Advanced Tools and Methodologies, Studies in Computational Intelligence), Publisher: Springer Berlin / Heidelberg, Volume 120 / 2008, ISBN: 978-3-540-78491-3

Conferences (10):

1. Theodoros Giannakopoulos, Aggelos Pikrakis and Sergios Theodoridis "A dimensional approach to emotion recognition of speech from movies" , 34th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2009).

2. Theodoros Giannakopoulos, Aggelos Pikrakis and Sergios Theodoridis "A Novel Efficient Approach for Audio Segmentation", 19th International Conference on Pattern Recognition, 2008 (ICPR2008).

3. Theodoros Giannakopoulos, Aggelos Pikrakis and Sergios Theodoridis "Music Tracking in Audio Streams from Movies" , 2008 International Workshop on Multimedia Signal Processing, IEEE Signal Processing Society (MMSP2008).

4. A. Pikrakis, T. Giannakopoulos and S. Theodoridis "Gunshot detection in audio streams from movies by means of dynamic programming and Bayesian networks", 33rd International Conference on Acoustics, Speech, and Signal Processing (ICASSP2008)

5. Theodoros Giannakopoulos; Aggelos Pikrakis, Sergios Theodoridis, "A multi-class audio classification method with respect to violent content in movies using bayesian networks" 2007 IEEE International Workshop on Multimedia Signal Processing, Chania, Crete, Greece, October 1-3, 2007 (MMSP2007), [1]

6. A. Pikrakis, T. Giannakopoulos and S. Theodoridis, "A Dynamic Programming Approach to Speech/Music Discrimination of Radio Recordings, 15th European Signal Processing Conference (EUSIPCO2007), Poznan, Poland from Sept 3 - 7, 2007

7. A. Pikrakis, T. Giannakopoulos and S. Theodoridis, "A computationally efficient speech/music discriminator for radio recordings", Proceedings of the 2006 International Conference on Music Information Retrieval and Related Activities (ISMIR2006), 8-12 October 2006, Victoria, BC, Canada.

8. Aggelos Pikrakis, Theodoros Giannakopoulos and Sergios Theodoridis: "Speech/Music Discrimination for radio broadcasts using a hybrid HMM-Bayesian Network architecture", In 14th European Signal Processing Conference (EUSIPCO06), September 4-8, 2006, Florence, Italy.

9. Theodoros Giannakopoulos, Aggelos Pikrakis, Sergios Theodoridis: "A Speech/music Discriminator for Radio Recordings Using Bayesian Networks", 2006 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2006), May 14-19, Toulouse, France.

10. Giannakopoulos Theodoros, Kosmopoulos Dimitrios, Aristidou Andreas, Theodoridis Sergios: "Violence Content Classification Using Audio Features", 4th Hellenic Con-

[1] Student paper award in IEEE 2007 International Workshop on Multimedia Signal Processing

ference on Artificial Intelligence (SETN2006), Heraklion, Crete, Greece, May 18-20, 2006.

Contents

List of tables

List of figures

List of figures

Chapter 1

Introduction

During the last decades, with the advances in the Word Wide Web and in the storage technology, an enormous increase of the available multimedia files has occurred. This explosion in the amount of the multimedia files being stored, transmitted and accessed has led to several research efforts focused on automatically and semantically analyzing the respective information. This is the task of **content-based multimedia analysis**.

1.1 Multimedia analysis

In this paragraph, a general description of the scientific area of content-based multimedia analysis is given. A general categorization of the related tasks is also presented, along with the related bibliography. Obviously, much attention has been paid to methods that deal with *audio* information.

Many scientific areas have contributed in order to build content-based multimedia analysis systems: pattern recognition, signal processing, image and video analysis and artificial intelligence are some of those scientific fields. Many tasks can benefit from content-based multimedia analysis. In the sequel, we describe some general categories of such applications (though, we have focused more on methods applied on audio data):

- **Search-Retrieval.** Large multimedia databases or file collections can contain thousands of multimedia files. Such examples are libraries of movies and videos, digital music collections and image archives. Furthermore, in many cases, multimedia files are not text-annotated, or the annotations are incomplete. It is therefore obvious that

the task of accessing and browsing such data resources is not easy. Towards this end, several content-based indexing and retrieval methods have been proposed during the last years ([1], [2]). It has to be noted that, even if a satisfactory annotation exists in a multimedia database, the use of content-based retrieval can lead to performance boosting compared to the simple text-based search methods. The first research works on multimedia retrieval and indexing focused on image files ([3], [4], [5]). During the last years, much attention has been paid to retrieval applications for video files ([6],[7],[8]). Algorithms for retrieval of audio data have mainly focused on speech or music ([9], [10],[11],[12], [13]). Especially for the case of music information retrieval, several types of applications have appeared. Query by music example (QBME) and query by humming (QBH) are the most basic music-related retrieval applications ([14], [15], [16]). They allow the user to easily search for a music file and therefore, they are both essential for systems with large music databases (especially when the user does not know the artist's or the song's name, which means that a text-based retrieval will not be possible). In a QBH system the input query is a human-hummed melody (monophonic signal), while the database in this case can be symbolic, which means that the music is represented based on musical scores (MIDI representation). On the other hand, in QBME systems the input query is a recorded part of a music file (polyphonic signal).

- **Classification.** Several methods have focused on classifying image, audio or video files to pre-defined categories. The categorical taxonomy differs between the various applications. For the case of audio information, some examples of classification methods are: algorithms for recognizing the *musical genre* of a music file ([17], [18],[19]), methods for recognizing a *musical instrument* from audio data ([20]) and also *speaker recognition* and identification methods ([21], [22]). Furthermore, during the last years, in the field of audio classification, much attention has been paid to recognizing affective content (i.e, *emotions*) in speech ([23], [24]) and music ([25], [26]). Apart from that, emotion recognition methods have also been proposed for visual (or audio-visual) information ([27], [28], [29]).

- **Segmentation.** Segmentation is the procedure of detecting segments (in an audio or video stream) which have a acoustically or visually homogenous content, while the

criterion of homogeneity depends on the particular task. Video segmentation methods ([1]) have mainly focused on *"shot detection"* (also referred to as "shot boundary detection" or "transition detection"). Shots are the most basic components of a video file and therefore their detection is fundamental for video segmentation. In most of the cases, shot detection methods use color and motion criteria from the visual information ([30]), while several methods have been proposed that use other types of data, such as audio and text ([31], [32]). In the case of audio segmentation some segmentation applications are: speaker change detection ([33]), audio event detection and general content-change detection ([34], [35], [36]). Especially for the case of audio data, the segmentation and classification stages are very often combined in a single system. In general, there are two ways to achieve this: a) By *sequentially* applying the two modules: in this case, first the segmentation stage is applied in order to detect homogenous segments, and then, each segment is classified to any of the adopted audio classes. Most speech-music discrimination methods follow this sequential approach ([37], [38]). b) By *jointly* performing the two tasks of segmentation and classification ([39], [40]). Towards this end, dynamic programming or region growing techniques are usually used.

- **Abstraction.** The purpose of abstraction is to represent the multimedia content in a more compact manner. In the particular case of *video* data ([41]), the purpose can be a) to extract static images (called *key frames*) that represent the overall content of the video or b) to extract a shorter and representative video clip (known as *storyboard*). In the first case, the process is called *video summarization*, while in the second case it is referred to as *video skimming* ([42], [43], [44]). Video segmentation and video abstraction are two tasks that usually overlap, since in most video skimming methods, a scene (or shot) detection stage is required. In the case of audio data, several summarization methods have been proposed, especially for music files ([45], [46], [47], [48]). This process of extracting representative audio parts from music tracks is often called *audio thumbnailing*.

1.2 Violence detection in video data

A very important issue related to the increase of the multimedia files (especially for those available through the World Wide Web) is that they are easily accessible by large portions of the population, with limited central control. It is therefore obvious that the need of protection of sensitive social groups (e.g., children) is imperative. Towards this end, several methods for violence detection in video files have been detected ([49], [50], [51], [44]).

In most of these methods, no audio information is used, or such use is limited to simple, energy-based features. However, the **audio** channel of a movie (or any video file) is very informative with respect to the content-based classification, especially when violence is the main target, because most violence-related content classes can be more easily detected through the usage of the audio data. For example, it is difficult (and most of the times impossible) to detect a gunshot in a video file by using only visual cues, but using the audio signal this task is much easier. The same happens for other violent events such as human screams or oral violence.

Therefore, an important part of the present thesis is to detect violent content in videos, using audio classification techniques. This has been achieved through the following two tasks:

- Multi-class audio classification of audio segments. The audio classes for this task have been selected to contain both violent and non-violent content.
- Speech emotion recognition in movies. The violent content of many movies lies in the *oral* part: anger, fear, sadness and disgust are some human feelings that are closely related to *psychological violence*.

1.3 Thesis contribution

As stated before, the purpose of this thesis is to develop methods for audio-based characterization of multimedia data. This includes segmentation, classification, event tracking and speech emotion recognition methods. Furthermore, much priority has been given to the definition of content classes related to violence, e.g. gunshots, fights, screams, oral violence,

etc, and to the corresponding features and characterization methods. More specifically, the present thesis has focused on the following:

- Speech - music discrimination: Before presenting the multi-class case, this thesis focuses on solving the binary classification-segmentation task of speech-music discrimination. Towards this end, a new method has been proposed and evaluated on real radio streams. The heart of this method is an algorithm based on dynamic programming, while Bayesian networks have been used as probability estimators. This method solves the speech-music discrimination task using a joint classification-segmentation approach, i.e., the task of segmentation and of classification of the segments is executed on a single step.

- Music tracking: this is the task of locating the parts of an audio stream that contain music. Focus has been given on using music-oriented features. The main music tracking method combines histogram-based weak learners, each one trained to the binary classification task of "music Vs other audio types". The algorithm has been evaluated on real audio streams from several genres of films. The proposed method can be used in an overall audio-based segmentation-classification scheme as a preprocessing stage which detects the music parts of the audio stream. Having detected the music segments, one can use this information for movie categorization based on the soundtrack of the film.

- General audio segmentation for detecting homogenous segments in uninterrupted audio streams. In particular, a method that faces the segmentation problem as a classification task is proposed. The proposed method is general and can be applied to any other type of signal change detection (e.g., silence detection or speaker change detection), just by changing the training data. Furthermore, the proposed method has a low computational cost, since experiments showed that the average execution time does not exceed 1% of the input audio data length.

- Content-based classification of multimedia files, with respect to violent content, using the audio medium. In particular, a method for **multi-class** audio classification of audio segments from films is proposed, with respect to violence detection. The main

targets of this part of the thesis were:

- To define a representative set of content audio **classes**, focused on the particular type of media (movie files).
- To find a good **feature** representation of the audio segments for the particular classification task.
- To develop an efficient method for multiclass classification of audio segments. Towards this end, a classifier combination scheme has been proposed, based on Bayesian Networks.

Note that Bayesian Networks have been selected, not only due to their ability to extract probabilities for each class, but also because they can be used to combine decisions from other sources, as visual information and subtitles.

- Emotion recognition of speech segments and movie summarization in terms of emotion labels. In particular, a method for speech emotion recognition is proposed based on a regression technique (instead of classification) and using a dimensional representation of the speech emotional states (Emotion Wheel). The first goal of this work is to investigate whether the Emotion Wheel offers a good representation for emotions associated with speech signals. Second, each speech segment is represented by a vector of ten audio features and three regression techniques have been evaluated for "mapping" the feature space to the dimensional representation of emotions. The results indicate that the Emotion Wheel is a good representation of emotional content of speech segments and that the resulting regression method can estimate emotion states of speech segments from movies, with sufficient accuracy. Finally, a scheme to extract affective content from uninterrupted audio streams from movies has been proposed.

1.4 Thesis outline

The present thesis is organized as follows:

- Chapter 2 presents an introduction to some basic features and respective statistics computed over audio segments, which can be used for classification and segmentation.

- Chapter 3 presents the binary segmentation-classification task of speech-music discrimination, along with the proposed approach.
- Chapter 4 presents the problem of music tracking in audio streams from movies. An effective approach to this task is proposed, based on a combination of histogram classifiers.
- Chapter 5 describes the proposed segmentation method. This segmenter can detect signal changes, according to the audio content. It can be used to extract audio segments of homogenous content, and therefore as a pre-processing step to the multi-class classification stage, described in Chapter 6.
- Chapter 6 proposes the multi-class classification method for audio segments from films. As described before, much attention has been paid to the definition of violence-related classes.
- Chapter 7 presents the proposed method for emotion recognition of speech segments from movies. Furthermore, it proposes how this approach can be used to several applications, such as multimedia summarization and violence detection.
- Chapter 8 presents the overall conclusions of the thesis, along with some possible future directions that stem from the current research.

Chapter 2

Audio Features Extraction

Feature extraction, as in any pattern recognition problem, is a very important stage for audio analysis and processing tasks. In this chapter some important audio features, which can be used for classification and segmentation methods, are presented. The choice of the specific features is the result of extensive experimentation and conclusions that stem from the physical meaning of the audio signals. Therefore, among the theoretical description of the audio features, some examples of differentiation of those features for different audio classes are presented.

2.1 Short-term processing for audio feature extraction

Let $x(n), n = 1, \ldots, L$, be the audio signal samples and L the signal length. In order to calculate any audio feature of x, it is needed to adopt a short-term processing technique. Therefore, the audio signal is divided in (overlapping or non-overlapping) short-term windows (frames) and the feature calculation is executed for each frame. The reason that this windowing technique is adopted is that audio signals are non-stationary and therefore their properties vary with time ([52], [53]). So during the time interval of a short frame the audio signal is "quasistationary".

Let $w(n)$ a window sequence of N samples. The simplest window sequence is the rectangular window which is described according to the equation:

$$w(n) = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & \text{elsewhere} \end{cases} \tag{2.1}$$

The windowing process of the original signal is equivalent to the multiplication of the signal with shifted versions of $w(n)$ in the time axis. Therefore, the samples of the $i-$th frame are described using the equation:

$$x_i(n') \equiv x(n)w(n - m_i) \tag{2.2}$$

where m_i is the window shift of the $i-$th frame. The values of m_i obviously depend on the selected window size and step. The window size should be large enough for the feature calculation stage to have enough data. On the other hand, it should be short enough for the (approximate) stationarity to be valid. Common window sizes vary from 10 to 50 msecs, while the window step is associated to the level of overlap. If, for example, 75% of overlap is needed, and the window size is 40 msecs, then the window step is 10 msecs.

In Figure 2.1 an example of windowing process is presented, for a window of 200 samples and a step of 100 samples (50% overlap).

As long as the window size and step is selected the feature value f is calculated *for each frame*. Therefore, an $M-$ element array of feature values $\mathbf{F} = f_j,\ j = 1, \ldots, M$, for the whole audio signal is calculated. Obviously, the length of that array is equal to the number of frames: $M = \lfloor \frac{L-S}{N} \rfloor + 1$, where: N the window length (number of samples), S the window step and L the total number of audio samples of the signal.

2.2 Mid-term processing for audio feature extraction

The process of short-term windowing, described in Section 2.1, leads, for each audio signal, to a sequence $\mathbf{F}$ of feature values. This sequence can be used for processing / analysis of the audio data. Though, a common technique is the processing of the feature in a mid-term basis. According to this technique, the audio signal is first divided into mid-term windows (segments) and then for each segment the short-term process is executed. In the sequel, the sequence $\mathbf{F}$, which has been extracted for each segment, is used for calculating a statistic, e.g., the average value. So finally, each segment is represented by a single value which is the

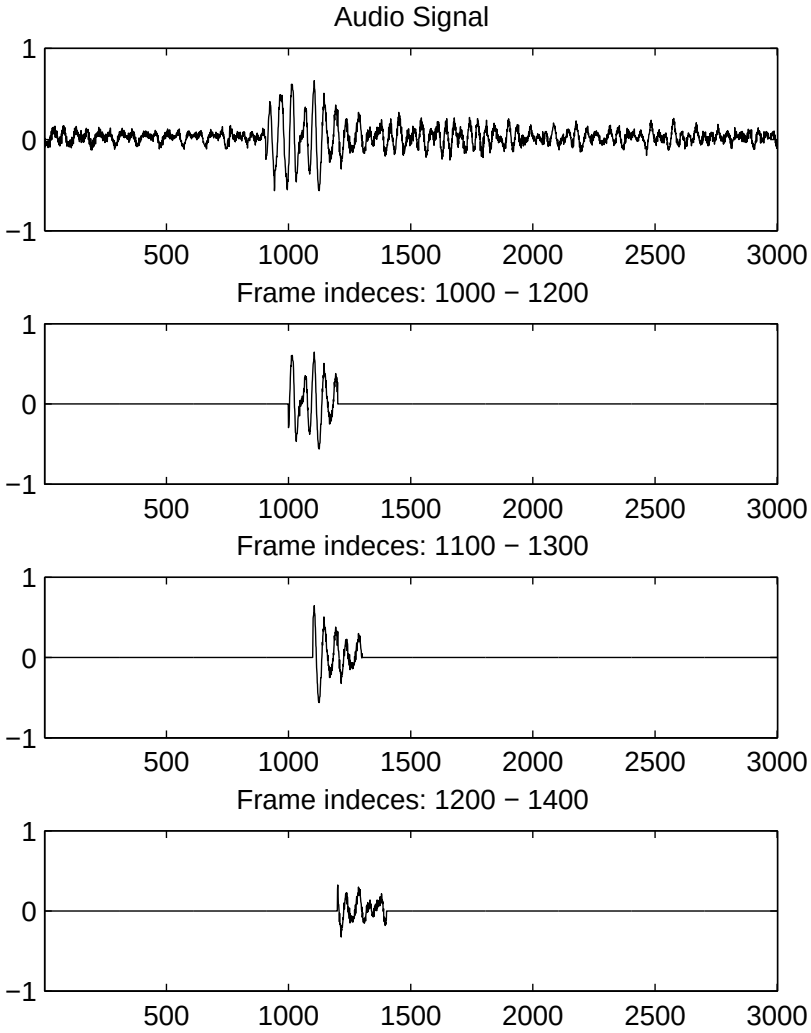


Figure 2.1. Windowing process for an audio signal. Three successive frames are presented. Each frame is 200 long (in samples), while a 50% overlap has been used

statistic of the respective feature sequence. Common durations of the mid-term windows are $1 - 10$ secs. In Figure 2.2, the above process is presented.

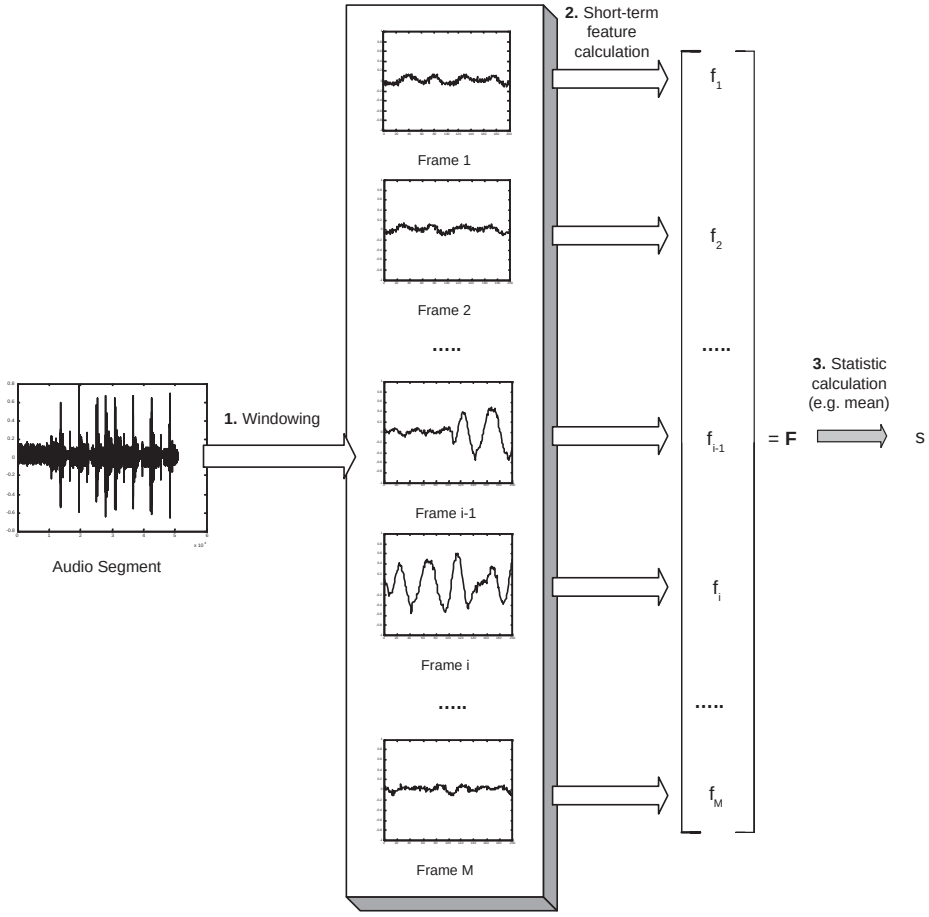


Figure 2.2. Mid-term feature extraction process: each mid-term window (segment) is short-term processed, and then a statistic is calculated on the feature sequence

2.3 Time domain audio features

The audio features that are directly extracted from the time domain, i.e., by the signal samples, are usually simple representations of the signal energy changes. Therefore, they can be used for audio signal discrimination based on energy differentiations. These features offer a simple way of audio analysis, but it is usually necessary to be used in combination with audio features that also contain frequency-related information (Section 2.4).

2.3.1 Energy

Let $x_i(n), n = 1, \ldots, N$ the audio samples of the $i-$th frame, of length N. Then, for each frame i the energy is calculated according to the equation:

$$E(i) = \frac{1}{N} \sum_{n=1}^{N} |x_i(n)|^2 \tag{2.3}$$

This simple feature can be used for detecting silent periods in audio signals, but also for discriminating between audio classes. In Figure 2.3 an example of the energy sequence is presented, for an audio stream that contains a music and a speech part. It is obvious that the variations in the speech part are higher. This is a general observation and it has a physical meaning, since speech signals have many silence intervals between high energy values, i.e., the energy envelope alternates rapidly between high and low energy states. Therefore, a statistic that can be used for the case of discriminating signals with large energy variations (like speech, gunshots etc.) is the standard deviation σ^2 of the energy sequence. In order to achieve energy-independency, the standard deviation by mean ratio ($\frac{\sigma^2}{\mu}$) has also been used ([37]). In Figure 2.3, apart from the energy sequence, those two statistics have also been calculated.

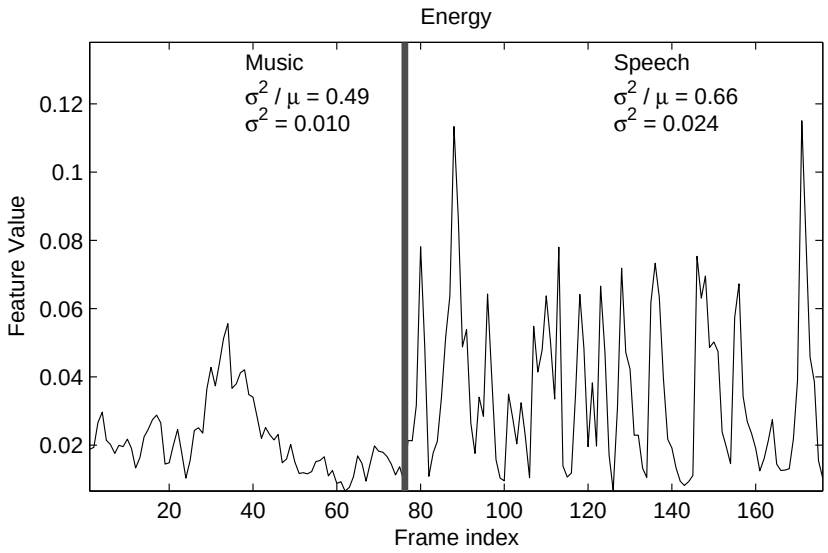


Figure 2.3. Example of energy sequence for an audio signal that contains music and speech

2.3.2 Zero Crossing Rate

Zero Crossing Rate (ZCR) is the rate of sign-changes of a signal, i.e., the number of times the signal changes from positive to negative or back, per time unit. It is defined according to the equation:

$$Z(i) = \frac{1}{2N} \sum_{n=1}^{N} |sgn[x_i(n)] - sgn[x_i(n-1)]| \tag{2.4}$$

where $sgn(\cdot)$ is the sign function:

$$sgn[x_i(n)] = \begin{cases} 1, & x_i(n) \geq 0 \\ -1, & x_i(n) < 0 \end{cases} \tag{2.5}$$

This feature is actually a measure of noisiness of the signal. Therefore, it can be used for discriminating noisy environmental sounds, e.g., rain. Furthermore, in speech signals, the $\frac{\sigma^2}{\mu}$ ratio of the ZCR sequence is high, since speech contains unvoiced (noisy) and voiced parts and therefore the ZCR values have abrupt changes. On the other hand, music, being largely tonal in nature, does not show abrupt changes of the ZCR. In Figure 2.4, an example of a ZCR sequence is presented, for an audio stream that contains three parts: a sound of rain, a music segment and a speech segment. As expected, the average value of the ZCR sequence for the first part (noisy sound) is higher. Furthermore, the $\frac{\sigma^2}{\mu}$ ratio is higher for the speech segment. ZCR has been used for speech-music discrimination ([54], [37]) and for musical genre classification ([17]).

2.3.3 Energy Entropy

This feature is a measure of abrupt changes in the energy level of an audio signal. It is computed by further dividing each frame into K sub-frames of fixed duration. For each sub-frame j, the normalized energy e_j^2 is calculated, i.e., the sub-frame's energy, divided by the whole frame's energy:

$$e_j^2 = \frac{E_{subFrame_j}}{E_{shortFrame_i}} \tag{2.6}$$

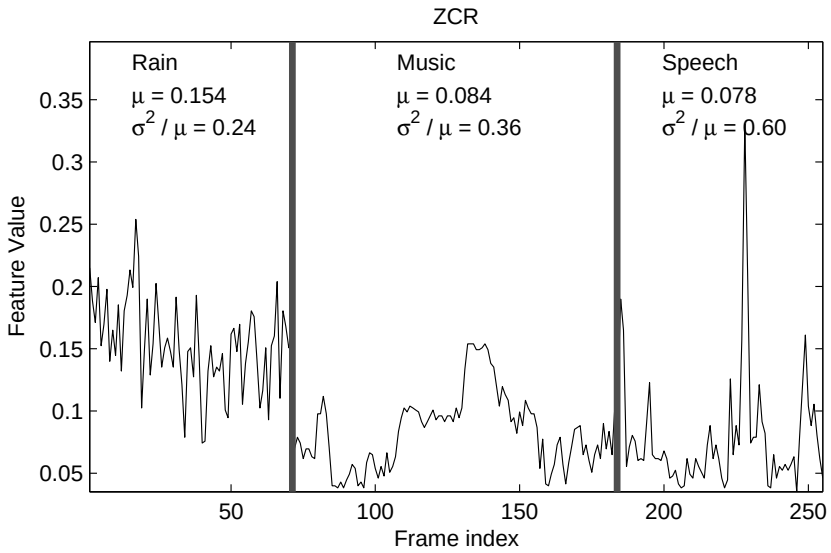


Figure 2.4. Example of ZCR sequence for an audio signal that contains "rain", music and speech

Therefore e_j is a sequence of normalized sub-frame energy values, and it is computed for each frame. Afterwards, the entropy of this sequence is computed using the equation:

$$H(i) = -\sum_{j=1}^{K} e_j^2 \cdot log_2(e_j^2) \tag{2.7}$$

The entropy of energy of an audio frame is lower if there are abrupt changes present in that audio frame. Therefore, it can be used for discrimination of abrupt energy changes, e.g. gunshots, abrupt environmental sounds, etc.. In Figure 2.5 an example of an Energy Entropy sequence is presented for an audio stream that contains: classical music, gunshots, speech and punk-rock music. Also, the selected statistics for this example are the maximum value and the $\frac{\sigma^2}{\mu}$ ratio. It can be seen that the minimum value of the energy entropy sequence is lower for gunshots and speech. Therefore, in order to detect abrupt sounds of violent content (e.g., gunshots, explosions and fights) the feature energy entropy has been used in a number of publications. Though, since this feature only contains energy-related signal information, it has been used in combination with other audio features ([55], [56], [57]), or in combination with visual cues ([44]).

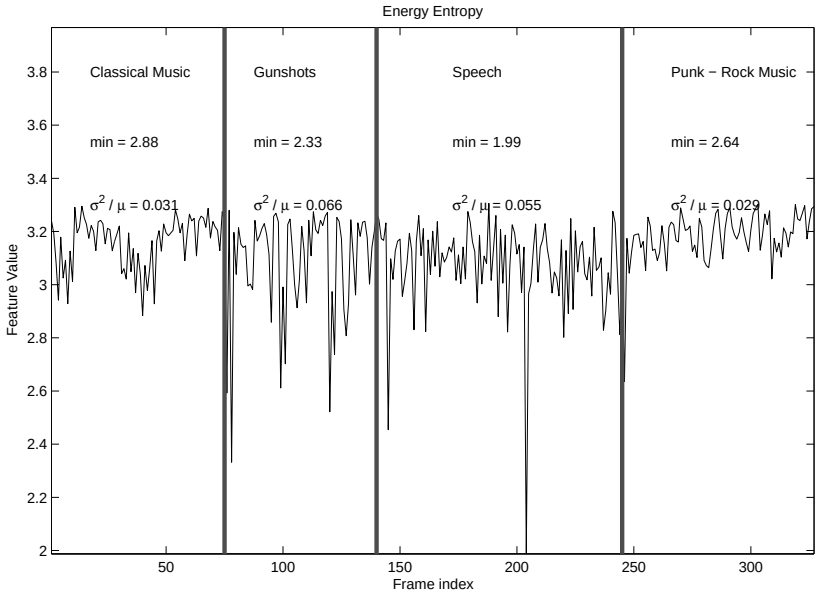


Figure 2.5. Example of Energy Entropy sequence for an audio signal that contains four successive homogenous segments: classical music, gunshots, speech and punk-rock music

2.4 Frequency domain audio features

Frequency domain (spectral) features use as basis the Short-time Fourier Transform (STFT) of the audio signal. Let $X_i(k)$, $k = 1 \ldots, N$, be the Discrete Fourier Transform (DFT) coefficients of the i-th short-term frame, where N is the frame length.

2.4.1 Spectral Centroid

The spectral centroid, C_i, of the i-th frame is defined as the center of "gravity" of its spectrum, i.e.,

$$C_i = \frac{\sum_{k=1}^{N}(k+1)X_i(k)}{\sum_{k=1}^{N} X_i(k)} \tag{2.8}$$

This feature is a measure of the spectral position, with high values corresponding to "brighter" sounds. Experiments have indicated that the sequence of spectral centroid is highly variated for speech segments. In Figure 2.6 an example of a spectral centroid sequence is displayed, for a segment that contains a speech and a scream part. It is obvious that for the scream part, the spectral centroid sequence has very low deviation.

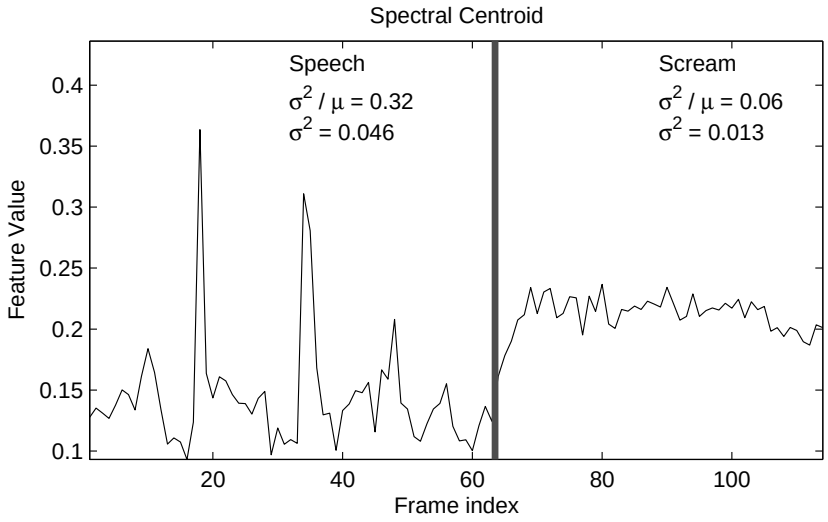


Figure 2.6. Example of Spectral Centroid sequence for an audio stream that contains a speech and a scream segment

2.4.2 Spectral Rolloff

Spectral Rolloff is the frequency below which certain percentage (usually around 90%) of the magnitude distribution of the spectrum is concentrated. This feature is defined as follow: if the m-th DFT coefficient corresponds to the the spectral rolloff of the i-th frame, then the following equation holds:

$$\sum_{k=1}^{m} X_i(k) = C \sum_{k=1}^{N} X_i(k) \tag{2.9}$$

where C is the adopted percentage. It has to be noted that the spectral rolloff frequency is normalized by N, in order to achieve values between 0 and 1. Spectral rolloff is a measure of the spectral shape of an audio signal and it can be used for discriminating between voiced and unvoiced speech ([58], [52]). In Figure 2.7, an example of a spectral rolloff sequence is presented, for an audio stream that contains three parts: music, speech and environmental noise. The mean and the median values of the spectral sequence for each part of the audio streams are also presented. It can be seen that both statistics are lower for the music part, while for the case of the environmental noise they are significantly higher.

In Figure 2.8 we present the histograms of the **median** values of the spectral rolloff sequences for music, speech and gunshots audio segments (C was selected to be equal to

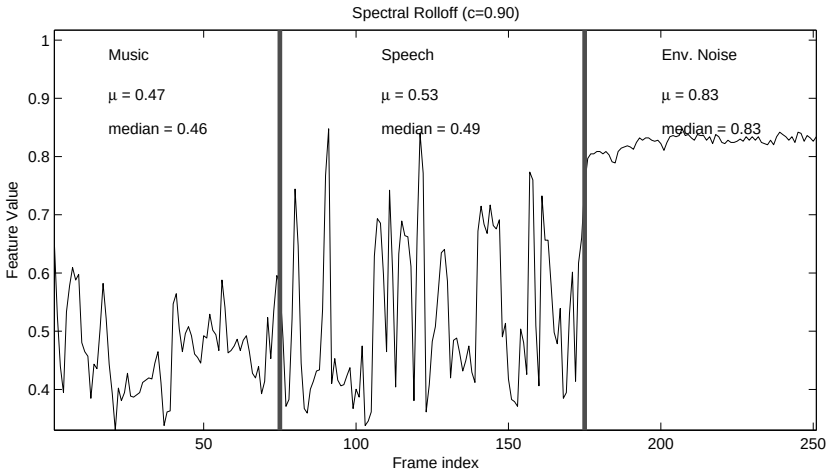


Figure 2.7. Example of a spectral rolloff sequence for an audio signal that contains music and speech and environmental noise.

0.9). One important observation is that for a large majority of the speech segments the statistic's value is around 0.50. Furthermore, for the "gunshots" segments the adopted statistic is significantly higher. This is something expected, since a gunshot is a sound that is characterized by a widely distributed spectrogram (see spectrograms of a gunshot and a music segment in Figure 2.10). Finally, experiments have shown that in 96% of the gunshot segments the median value of the spectral rolloff sequence was higher than 0.5, while the same percentage was 40% for the music and 48% for the speech segments. This discrimination ability of the spectral rolloff feature has lead us to use it for multi-class audio classification ([56]) as described in Chapter 6.

2.4.3 Spectral Flux

This is a measure of the local spectral change between successive frames. It is defined as the squared difference between the normalized magnitudes of the spectra of two successive frames:

$$Fl_{(i,i-1)} = \sum_{k=1}^{N} (EN_i(k) - EN_{i-1}(k))^2 \tag{2.10}$$

where $EN_i(k) = \frac{X_i(k)}{\sum_{l=1}^{N} X_i(l)}$, i.e., $EN_i(k)$ is the k-th normalized DFT coefficient at the i-th

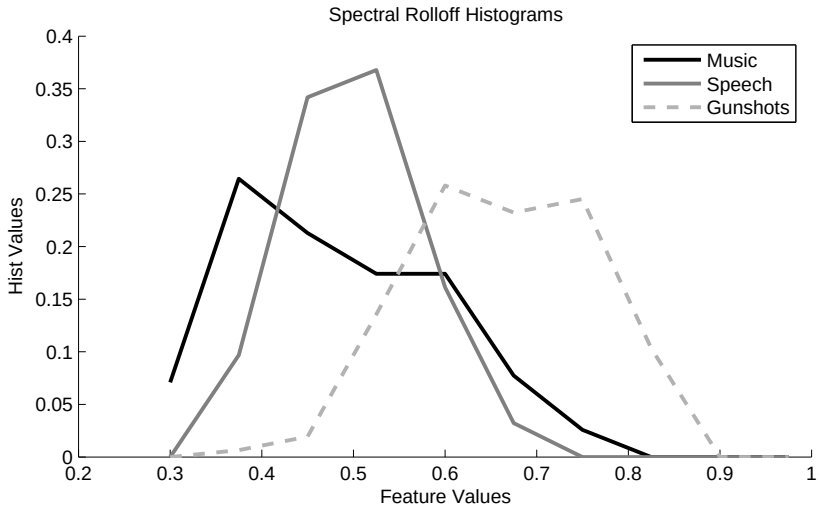


Figure 2.8. Histograms of the median values of the spectral rolloff sequences for three classes of audio segments: music, speech and gunshots.

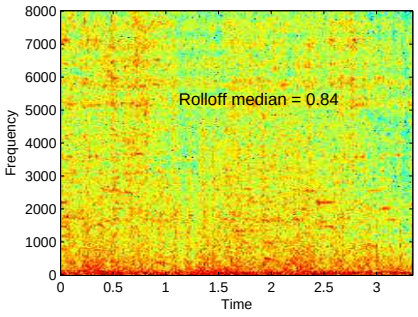


Figure 2.9. Gunshots Spectrogram

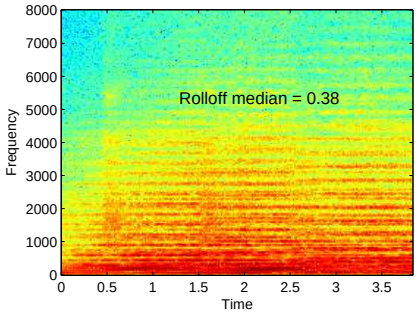


Figure 2.10. Music Spectrogram

frame.

2.4.4 Spectral Entropy

Spectral entropy ([59]) is computed by dividing the spectrum of the short-term frame into L sub-bands (bins). The energy E_f of the f-th sub-band, $f = 0, \ldots, L-1$, is then normalized by the total spectral energy, yielding $n_f = \frac{E_f}{\sum_{f=0}^{L-1} E_f}$, $f = 0, \ldots, L-1$. The entropy of the normalized spectral energy n is then computed by the equation:

$$H = -\sum_{f=0}^{L-1} n_f \cdot log_2(n_f) \tag{2.11}$$

In Figure 2.11 an example of the spectral entropy sequence is presented, for an audio stream that contains a speech and a music part. It is obvious that the variations in the music part are significantly lower. A variant of the spectral entropy called "chromatic entropy" has been used in [60] and [40] in order to discriminate in an efficient way speech from music. More details are given in Chapter 3.3.

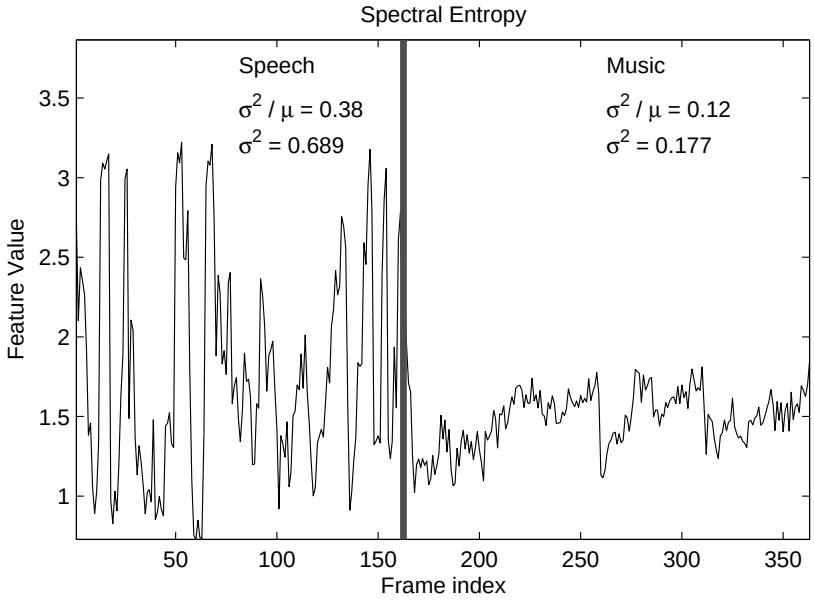


Figure 2.11. Example of Spectral Entropy sequence for an audio stream that contains a speech and a music segment

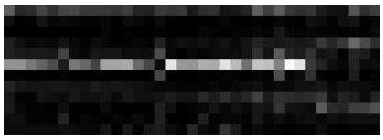


Figure 2.12. Music Chroma

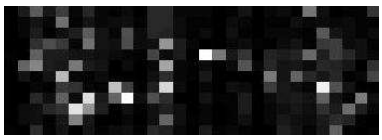


Figure 2.13. Speech Chroma

2.4.5 Fundamental Frequency

Harmonic signals, as is the case with the signals produced from musical instruments or voiced speech segments, possess the distinct characteristic of fundamental frequency, which may vary a lot for music signals. Fundamental frequency tracking of audio signals (in the general case) is not an easy task and a large number of techniques have been proposed in the published literature mainly in the context of speech and music signals (e.g., [61], [62]).

2.4.6 Chroma based Features

Based on early studies on the human perception of pitch [63], Wakefield proposed in [64] a 12-element representation of the spectral energy of a music signal, known as the **"Chroma Vector"**. Each element of the vector corresponds to one of the twelve traditional pitch classes (i.e., twelve notes) of the equal-tempered scale of the Western music. The chroma vector encodes and represents harmonic relationships within a particular music signal and can be easily computed for each short-term window using the DFT coefficients.

In particular, the chroma vector is computed by the logarithmic magnitude of the DFT:

$$v_k = \sum_{n \in S_k} \frac{X_i(n)}{N_k}, k \in 0..11 \tag{2.12}$$

where S_k is a subset of the frequency space and N_k is the number of elements in S_k. Each of the bins S_k expresses one of the 12 pitch classes existing in western music, and therefore each of the chroma bands is separated by one semitone. The chroma vector v_k is computed for each frame i of the audio segment, resulting in a matrix V with elements $V_{k,i}$. The resulting sequence of chroma vectors is known as the **chromagram** (as an analogy to the spectrogram). In Figures 2.12 and 2.13 the chromagrams of a music and a speech signal are presented.

In this work, two features, based on the chromagram, are proposed:

- **Chroma Feature 1:** The first chroma-based feature is based on the observation that, for music segments, there are usually two or three dominant chroma coefficients, while all other coefficients have values close to zero. In order to calculate this first chroma-based feature, the deviation between chroma coefficients $k \in 0..11$ in each frame i is calculated. For this feature, non-overlapping windows of 100 msecs have been adopted. Finally, the *mean value* of that feature sequence is used as the final statistic value.

- **Chroma Feature 2:** The second feature based on the chroma vector is a measure of deviation between successive frames for each chroma element. This stems from the observation that in music segments there is at least one chroma element with low deviation for a short period of time (e.g., 200msecs), i.e., there is at least one "stable" chroma coefficient. In order to compute this feature, a short-term window of 20 msecs is adopted for the computation of the chromagram. Then, the deviation of each chroma coefficient is computed for every 10 frames (i.e., a mid-term window of 200 msecs is used), and the **minimum deviation** is kept for each 200 msecs block. Finally, the *median value* of those minimum deviations is computed. This feature is a measure of the minimum (per 200 msecs block) chroma coefficient variation and, as explained above, it is lower for music signals.

The above chroma-based features encode the way the chroma coefficients are distributed, especially for music signals. They have therefore been used for music tracking ([65]) and speech/music discrimination ([40]). Though, experimental results have indicated that those features have a high discrimination ability even for *multi-class* audio classification tasks ([56]). In Figure 2.14, the histograms of the second chroma-based feature (i.e. the median value of the second chroma feature vector) is presented for three classes: Music, Speech and Shots.

2.4.7 Mel-frequency cepstral coefficients (MFCCs)

The MFCCs have been very popular in the field of speech processing [52]. MFCC is actually a type of cepstral representation of the signal, though, the frequency bands are computed using the mel-scale, instead of the linearly-spaced approach. In order to extract the MFCCs from a frame, the DFT is computed and the resulting spectrum is given as input to a mel-

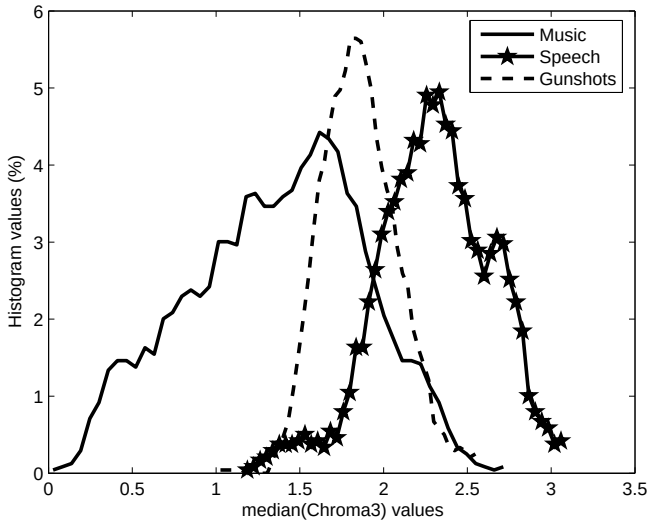


Figure 2.14. Histograms of the 2nd chroma-based feature for "Music", "Speech" and "Shots" audio segments.

scale filter bank that consists of L *overlapping triangular* filters. Over the years a number of frequency warping functions have been proposed, e.g. ([66]),

$$f_w = 1127.01048 * \log(f/700 + 1)$$

The above conversion equation is presented in Figure 2.15.

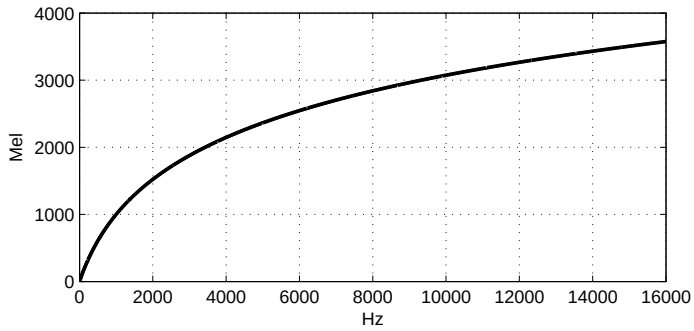


Figure 2.15. Frequency warping function for the computation of the MFCCs

If $\widetilde{O_k}, \quad k = 1, \ldots, L$, is the output power of the k-th filter, then the resulting MFCCs are given by the equation

$$c_m = \sum_{k=1}^{L} (\log \widetilde{O_k}) \cos[m(k - \frac{1}{2})\frac{\pi}{L}], \quad m = 1, \ldots, L \tag{2.13}$$

Chapter 3

Speech - Music Discrimination

This chapter focuses on the binary problem of speech / music discrimination and proposes a multi-stage robust method for this task. Speech/Music discrimination refers to the problem of segmenting an audio stream and labelling (i.e., classifying) each segment as either speech or music.

In this chapter, besides covering the major bibliography in the field, a method for speech/music discrimination, which is based on a three-step procedure is also proposed. The first step is a computationally efficient scheme consisting of a region growing technique. This is used as a preprocessing stage and yields segments with high music and speech precision at the expense of leaving certain parts of the audio recording unclassified. The unclassified parts of the audio stream are then fed as input to a more computationally demanding scheme, which treats speech/music discrimination of radio recordings as a probabilistic segmentation task, where the solution is obtained by means of *dynamic programming*. At a final stage, an algorithm that performs boundary correction is applied to remove possible errors at the boundaries of the segments (speech or music) that have been previously generated. The proposed system has been tested on radio recordings from various sources. The overall system accuracy is approximately 96%. Performance results are also reported on a musical genre basis and a comparison with existing methods is given.

The chapter is organized as follows: Section 3.3 describes the CES segmentation scheme, Section 3.4 presents the maximization technique and 3.5 describes the post processing stage. Results, experiments and comparison with other methods are presented in Section 3.6. Finally, conclusions are drawn in Section 3.7.

3.1 Previous works

Since the first attempts in the mid 90's, a number of speech / music discrimination systems have been implemented in various application fields. In [67], a real-time technique for speech/music discrimination was proposed, focusing on the automatic monitoring of radio stations, using features related to the short-term energy and zero-crossing rate (ZCR). In [54], thirteen audio features were used in order to train different types of multidimensional classifiers, such as a Gaussian MAP estimator and a nearest neighbor classifier. A scheme based on models for speech recognition was used in [68]. The work in [69] employs Gaussian Mixture Models to classify homogeneous (pre-segmented) audio samples as speech or music. In [70], a set of "One vs all" classifiers was used for the classification of pre-segmented data. In [71], a combination of line spectral frequencies (LSFs) and zero-crossing-based features was used for frame-level speech/music discrimination. In [38], energy, ZCR and fundamental frequency were used as features in order to achieve analysis of audiovisual data. Segmentation/classification was achieved by means of a procedure based on heuristic rules. A framework based on a combination of standard Hidden Markov Models and Multilayer Perceptrons (MLP) was used in [39] for speech/music discrimination of broadcast news. An Adaboost - based algorithm, applied on the spectrogram of the audio samples, was used in [72] for frame-level discrimination of speech and music. In [37], energy and ZCR were employed as features and classification was achieved by means of a set of heuristic criteria in an attempt to exploit the nature of speech and music signals.

The majority of the previously described methods deal with the problem of speech/music discrimination in two separate steps: first, the audio signal is split into segments by detecting abrupt changes in the signal statistics and at a second step the extracted segments are classified as speech or music by using standard classification schemes. The work in [39] differs in the sense that the two tasks are performed jointly by means of a standard HMM, where, for each state, a MLP is used as an estimator of the continuous observation densities required by the HMM.

3.2 Proposed method - General

The proposed system proposed is based on a three-stage philosophy (see Figure 3.1):

(a) A computationally efficient scheme is first employed as a preprocessing stage. It is based on a region growing technique that bears its origins in the field of image segmentation and operates on a single feature, which is a variant of the spectral entropy. A useful property of this very simple algorithm is that it can easily be *tuned to maximize speech or music precision* at the expense of leaving certain parts of the audio recording unclassified. To exploit this property, the algorithm is applied twice on the original recording: in the first pass, it is tuned to detect music segments with a high precision rate and during the second pass to yield speech segments with a high precision rate. After the application of this scheme, an amount of data is left unclassified. However, the *precision rate* of those which have been classified is *over* 98%. In the sequel, we will refer to this first-stage segmentation scheme as the Chromatic Entropy Segmenter (*CES*).

(b) At a second stage, a more sophisticated and computationally demanding algorithm is applied on the regions left unclassified. Each one of these regions is first split into a number of short-term frames by means of a short-term processing window and five features are extracted per frame. Speech/music discrimination is then treated as a maximization task. In other words, the method processes the feature sequence in order to *group features together and form the sequence of segments* and the *respective class labels* (i.e., speech/music) that *maximizes* the product of posterior probabilities, given the data that contribute to each one of the segments. In order to estimate the required posterior probabilities, a Bayesian Network (BN) Combiner is trained and used. Since an exhaustive approach to this solution is unrealistic, we resort to dynamic programming to solve this maximization task. The use of a BN as a conditional probability estimator is the most natural choice, since BNs are tailored for such a job by their definition. Moreover, the use of the BN offers a computationally simple way of overcoming the assumption of statistical independence among the data residing within a segment. In the sequel, we will refer to this second-stage segmentation/classification scheme as the Dynamic Programming Based Segmenter (*DPBS*).

(c) In the final stage, a boundary correction algorithm is applied on the previously obtained discrimination results, in order to improve the overall system's accuracy.

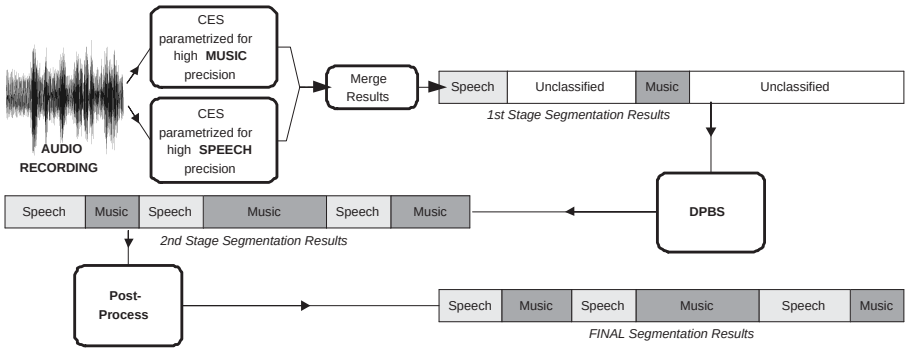


Figure 3.1. Overall Architecture: CES detects music and speech segments with a precision rate higher than 98%. The unclassified audio regions are subsequently fed as input to the DPBS. At a final step, a boundary correction algorithm is applied.

3.3 The CES Stage

This first scheme bears its origins in the field of image segmentation. The main idea is that if speech/music discrimination is treated as a segmentation problem (where each segment is labelled as either speech or music), then each of the segments can be the result of a segment (region) growing technique, where one starts from small regions and keeps expanding them as long as certain criteria are fulfilled. This approach has been used in the past in the context of image segmentation, where a number of pixels are usually selected as candidates (seeds) for region growing. In image segmentation, regions grow by attaching neighboring pixels, provided that certain criteria are fulfilled. These criteria usually examine the relationship between statistics drawn from the region and the pixel values to be attached.

To this end, a feature sequence is first extracted from the audio recording by means of a short-term processing technique. Once the feature sequence is generated, a number of frames are selected as candidates for region expansion. Starting from these *seeds*, segments grow and keep expanding as long as the standard deviation of the feature values in each region remains below a predefined threshold. In the end, adjacent segments are merged and short (isolated) segments are ignored. All segments that have survived are labelled as music. As

it will become apparent later on, this is due to the choice of certain algorithmic parameters, that are tuned towards the music part of the signal.

3.3.1 Feature extraction

At a first step, the audio recording is split into a sequence of non-overlapping short-term frames (50ms long). Computational efficiency is the only reason that non-overlapping frames were used. From each frame, a variant of the spectral entropy [59] (see Chapter 2.4.4) is extracted by taking into account the frequency range up to approximately 2KHz (by its definition, entropy is a measure of the uncertainty or disorder in a given distribution [73]):

- All computations are carried on a mel-scale, i.e., the frequency axis is warped according to the equation

$$f = 1127.01048 * \log(f_l/700 + 1)$$

where f_l is the frequency value on a linear scale.

- The mel-scaled spectrum of the short-term frame is divided into L sub-bands (bins). The center frequencies of the sub-bands are chosen to coincide with the frequencies of semitones of the chromatic scale, i.e.,

$$f_k = 1127.01 * \log(\frac{f_0 * 2^{\frac{k}{12}}}{700} + 1), k = 0, \ldots, L-1$$

where f_0 is the center frequency of the lowest sub-band of interest on a linear scale. In our study, $f_0 = 13.75$ Hz and $L = 86$, i.e., the last bin center is located at 1975.5 Hz on a linear scale. We have found that, dealing with music signals, such a choice is more natural and has a beneficial effect on the performance

- The energy X_i of the i-th sub-band, $i = 0, \ldots, L-1$, is then normalized by the total energy of all the sub-bands, yielding

$$n_i = \frac{X_i}{\sum_{i=0}^{L-1} X_i}, i = 0, \ldots, L-1$$

The entropy of the normalized spectral energy is then computed by the equation:

$$H = -\sum_{i=0}^{L-1} n_i \cdot log_2(n_i) \tag{3.1}$$

In the sequel, we will refer to this feature by the term "chromatic entropy". Thus, at the end of the feature extraction stage, the audio recording is represented by the feature sequence $\mathbf{F}$, i.e., $\mathbf{F} = \{H_1, H_2, \ldots, H_T\}$, where T is the number of short-term frames. Figure 3.2 presents the feature sequence that has been extracted from a BBC radio recording, the first half of which corresponds to speech and the second half corresponds to music. It can be observed that the standard deviation of the chromatic entropy is significantly lower for the case of music.

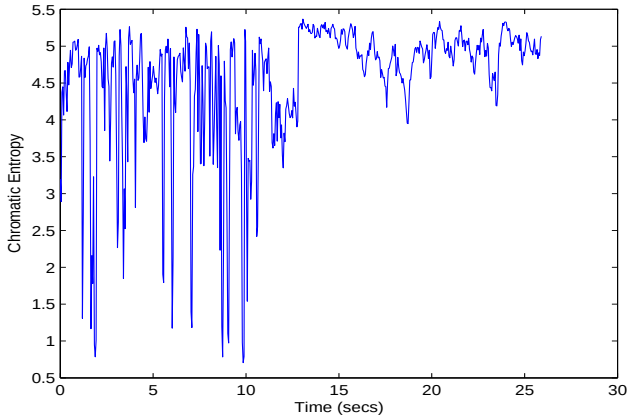


Figure 3.2. Chromatic entropy over time for 26 seconds of a BBC radio recording.

3.3.2 Region Growing

Region growing consists of three stages. An initialization stage, a region growing stage via an iterative procedure and, finally, of a termination stage. More specifically:

Initialization step - Seed generation: If T is the length of the feature sequence, a "seed" is chosen every M frames, M being a pre-defined constant. If K is the total number of seeds and i_k is the frame index of the k-th seed, then the frame indices of the seeds form the set $\{i_1, i_2, \ldots, i_K\}$. The k-th seed is considered to form a region, R_k, consisting of a single frame, i.e., $R_k = \{H_{i_k}\}$ where H_{i_k} is the feature value of the respective frame.

Iteration: In this step, every region, R_k, is expanded by examining the feature values of the two frames that are adjacent to the boundaries of R_k. To this end, let l_k and r_k be the indices (at the current iteration step) that correspond to the leftmost and rightmost frames that are part of R_k, respectively. Clearly, if R_k consists of a single frame, then $l_k = r_k = i_k$.

Following this notation, $l_k - 1$ and $r_k + 1$ are the indices of the two frames which are adjacent to the left and right boundary of R_k, respectively. Our algorithm decides to expand R_k to include H_{l_k-1}, if **a)** H_{l_k-1} is not already part of any other region *and* **b)** if the standard deviation of the feature values of the R_k region, *after the expansion*, is below a pre-defined threshold T_h, common to all regions. In other words, if the standard deviation of the feature values for $H_{l_k-1} \cup R_k$ is less than T_h, then, at the end of this step R_k is grown to include one frame to the left. Similarly, if H_{r_k+1} is not already part of any other region and if the standard deviation of the feature values in $R_k \cup H_{r_k+1}$ is less than T_h, then R_k will also grow by one frame to its right. At the end of this step, each R_k is grown by at most two frames. It must be pointed out that, at some iteration step, certain regions may not grow at all (although the region growing criteria are fulfilled). This is because both frames that are adjacent to their boundaries, already belong to other regions. At the end of this step, it is examined whether at least one region has grown by at least one frame. If this is the case, this step is repeated until no more region growing takes place.

Termination: After region growing has been completed, some of the formed regions may be adjacent. Such adjacent regions are merged to form larger segments. Finally, after the merging process is complete, short regions are eliminated by comparing their length with a pre-defined threshold, say T_{min}. The survived segments are labelled as music. This is because the proposed scheme relies on the assumption that music segments exhibit low standard deviation in terms of the adopted feature (see Figure 3.2). Furthermore, T_{min} is an extra "guarantee" for these segments to be music, since segments of small duration, say 0.5 s, cannot be considered as music.

The above suggests that the CES is dependent on three parameters, namely T_h, the threshold for the standard deviation (which controls the region growing procedure), T_{min}, the minimum segment length (used in the final stage of the algorithm) and T_{seed}, the distance (measured in seconds) between successive seeds. Given that we choose one seed per M non-overlapping frames and that the frame length is 50 ms, $T_{seed} = M * 0.05$ s.

The above iterative scheme is applied twice. In a first pass, the parameters are set to maximize music precision and in a second pass they are tuned to maximize speech precision. In practice, if T_h is set to a low value and T_{min} to large value, all segments that are returned as music are, with very high probability, true music segments (high music precision).

The remaining audio stream is treated as unclassified audio, because if music precision is maximized then music recall is expected to decrease. As a result, unclassified segments are expected to consist of speech as well as music.

During the second pass, T_h is set to a high value and T_{min} to a small value. This time and after running the CES, we are more confident that those frames *that have not been* merged in any one of the survived segments *will contribute* to large values of standard deviation. Thus, with high probability, they can be treated as speech. Of course, now, we consider as unclassified all segments that are *not* labelled as speech. The two sets of values for maximizing music and speech precision are presented in Section 3.6.

As a concluding remark, it has to be noted that, since both passes operate on the original feature sequence (hence the two parallel blocks in Figure 3.1), conflicts are resolved by trusting the segmenter which maximizes music precision. In other words, if a frame is given both labels, it is considered to be a frame of music. This is because our experiments have indicated that music precision is slightly higher (see Section 3.6). This is indicated in Figure 3.1 by the block titled "Merge Results".

3.3.3 Computational complexity of the CES

The worst case complexity of the CES is linear with respect to the length of the feature sequence. In order to compute the complexity, we first focus on the worst case scenario for a single seed. Due to the fact that successive seeds are located M frames apart, the region around a seed may grow to include at most $2(M-1)+1$ frames. The cost of each region expansion by one frame is equal to the computation of the standard deviation of chromatic entropy over the region plus the comparison of the resulting value against the adopted threshold. As a result, the first expansion requires the computation of the standard deviation over two frames, the second expansion over three frames and following this line of thinking the l-th expansion over $l+1$ frames. It is well known from statistical analysis that in such a scenario, the mean value and standard deviation can be computed recursively. In other words, if μ_k and σ_k are the mean value and standard deviation of a region consisting of k frames, $\mu_1 = \textit{seed value}$ and $\sigma_1 = 0$, then

$$\mu_k = \frac{(k-1)\mu_{k-1} + x}{k}$$

and

$$\sigma_k^2 = \frac{(k-2)\sigma_{k-1}^2 + \frac{k(x-\mu_k)^2}{(k-1)}}{k-1}$$

where, for simplicity of notation, x is the value of the frame to be included in the region. The above suggest that the cost of an expansion in terms of both additions and multiplications is constant, i.e., $O(1)$. For a single seed, in the worst case at most $2M-1$ expansions are expected to take place, so the expected computational cost is $(2M-1)O(1)$. If $\frac{T}{M}$ is the number of seeds (T is the length of the feature sequence), the total complexity in the worst case is $\frac{T}{M}(2M-1)O(1) = 2TO(1) - \frac{T}{M}O(1)$. We distinguish two cases: (a) If $M \to T$ then $\frac{T}{M} \to 1$ and the second term can be ignored, i.e., complexity is $O(T)$ and (b) if $M \to 1$, then $\frac{T}{M} \to T$ and complexity is again $O(T)$.

3.4 Speech/Music discrimination treated as a maximization task

Once the two CES passes have been completed, three types of segments have been formed. Those classified as music or speech and the rest, which remain unclassified. In a way, the CES decides on the "easy" cases. The decision on the rest is left to a more computationally demanding procedure. The latter treats speech/music discrimination as a maximization task, where the solution is obtained by means of a dynamic programming technique. The proposed scheme seeks the sequence of segments and the respective class labels (i.e., speech/music) that maximize the product of posterior class probabilities, given the data within each one of the segments. To this end, a Bayesian Network combiner is embedded as a posterior probability estimator.

3.4.1 Feature Extraction

At a first step, each unclassified audio segment is split into a sequence of non-overlapping short-term frames (50ms long) and five audio features are extracted per frame. At the end of this feature extraction stage, each audio segment is represented by a sequence $\mathbf{F}$ of five-dimensional feature vectors, i.e.,

$$\mathbf{F} = \{O_1, O_2, \ldots, O_T\}$$

where T is the number of short-term frames. The specific choice of features was the result of extensive experimentation. It must be emphasized that this is not an optimal feature set in any sense and other choices may also be applicable. The adopted features are ([52]):

1. **Short-term Energy:** (Described in Chapter 2.3.1).
2. **Chroma-Vector based features:** We have used the two chroma-based features described in Chapter 2.4.6). It has to be noted, that those features require a different short-term processing (e.g., for the 2nd Chroma-based feature 20 msecs are used). This is not a restriction, as it will be made clear in section 3.4.3, where the feature sequences are fed as input to a Bayesian Network that serves as a posterior probability estimator.
3. **The first two Mel Frequency Cepstral Coefficients (MFCCs)**. In particular, the first two MFCCs have been adopter (see Chapter 2.4.7).

3.4.2 Speech/Music discrimination treated as a maximization task

In this stage, speech/music discrimination is treated as a maximization task, where the solution is obtained by means of dynamic programming. Two assumptions are adopted concerning the length of the segments to be formed: a) a segment has to be at least T_{dmin} frames long and b) its duration cannot exceed T_{dmax} frames. The minimum segment duration is dictated by the nature of the signals under study, i.e., we assume that a segment must be of sufficient duration (we use 0.5s) in order to be interpreted either as speech or music. The need for T_{dmax} (3s in our work) is imposed by computational issues related to the searching of the optimal path. As a result, any segment longer than T_{dmax}, will be partitioned in segments of smaller than T_{dmax} length.

To proceed further, some definitions must be given. Let L be the length of a feature sequence $\mathbf{O}$ that has been extracted from an audio stream. Our goal is twofold: a) Segment the sequence into K segments and b) classify each one of the segments as speech or music. Let $\{d_1, d_2, \ldots, d_{K-1}, d_K\}$ be the frame indices that mark the end of each segment. Clearly, $T_{dmin} \leq d_1 < d_2 \ldots < d_K = L$ and $T_{dmax} \geq d_k - d_{k-1} \geq T_{dmin}$, $k = 2, \ldots, K$. Therefore, the k-th segment starts at frame index $d_{k-1} + 1$ and ends at frame index d_k, with the exception of the first segment, that starts at the first frame and ends at frame index d_1 (initialization

step). Thus, the feature sequence, $\mathbf{F}$, yields the following sequence of pairs

$$\{(1, d_1), (d_1 + 1, d_2), \ldots, (d_{K-1} + 1, L)\},$$

where each pair holds the frame indices of the beginning and the end of the corresponding segment. In addition, let c_k be the class label of the k-th segment, where c_k can indicate either speech or music. To this end, let $p(c_k \mid \{O_{d_{k-1}+1}, \ldots, O_{d_k}\})$, be the posterior probability of class label c_k given the sequence of observations (feature sequence) within the k-th segment.

Following the above notation, for any given sequence of K segments and corresponding class labels, we form the cost function

$$\begin{aligned} J(\{d_1, \ldots, d_K\}, \{c_1, \ldots, c_K\}, K) \equiv \\ p(c_1 \mid \{O_1, \ldots, O_{d_1}\}) \cdot \\ \prod_{k=2}^{K} p(c_k \mid \{O_{d_{k-1}+1}, \ldots, O_{d_k}\}) \end{aligned} \tag{3.2}$$

where independence between successive segments has been assumed. It is now possible to treat speech/music discrimination as a maximization problem. In other words, we seek the *optimal sequence of segments* (i.e., the start and the end point of each segment) *and the corresponding class labels that maximize* J. Equivalently, J needs to be maximized over all possible values of $\{d_1, d_2, \ldots, d_{K-1}, d_K\}$, $\{c_1, c_2, \ldots, c_{K-1}, c_K\}$ and K, under the two assumptions made in the beginning of this section. In other words, the number of segments, K, is an outcome of the optimization process. Obviously, an exhaustive search would amount to an prohibitive computational load. Thus, we resort to dynamic programming to obtain a solution to the problem in an efficient way. Note that this is the first time that, to our knowledge, the segmentation/classification task is cast in such a framework.

To this end, as it is common with dynamic programming techniques, we first construct a grid by placing the feature sequence on the x-axis and the two states (speech/music) on the y-axis. This is shown in figure 3.3, where S stands for speech and M stands for music. Clearly, the grid has two rows and L columns (L being the length of the feature sequence). In order to grasp the physical meaning of the nodes in the grid, take, as an example, node (O_{d_k}, S), $T_{dmin} \leq d_k \leq L$. This node stands for the case that a speech segment ends at frame index d_k. Following this line of reasoning, a path of K nodes $\{(O_{d_1}, c_1), (O_{d_2}, c_2), \ldots, (O_{d_K}, c_K)\}$,

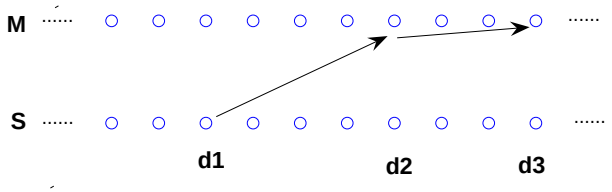


Figure 3.3. A sequence of segments in the dynamic programming grid

corresponds to a possible sequence of segments, where $T_{dmin} \leq d_1 < d_2 < d_k = L$, $T_{dmax} \geq d_k - d_{k-1} \geq T_{dmin}$, $k = 2, \ldots, K$, and $\{c_1, \ldots, c_K\}$ are the respective class labels. We denote the transition to node (O_{d_k}, c_k) from its predecessor in the path, i.e., $(O_{d_{k-1}}, c_{k-1})$, by $(O_{d_{k-1}}, c_{k-1}) \rightarrow (O_{d_k}, c_k)$. This transition can be interpreted as follows: a segment with class label c_{k-1} ends at frame d_{k-1} and the next segment in the sequence starts at frame $d_{k-1}+1$, ends at frame d_k and has class label c_k. We then define a cost function $T(\cdot)$ for the transition $(O_{d_{k-1}}, c_{k-1}) \rightarrow (O_{d_k}, c_k)$ as follows:

$$T((O_{d_{k-1}}, c_{k-1}) \rightarrow (O_{d_k}, c_k)) = p(c_k \mid \{O_{d_{k-1}+1}, \ldots, O_{d_k}\}) \tag{3.3}$$

In other words, the cost of the transition is set equal to the posterior probability of the class label, c_k, given the feature sequence defining the segment $\{O_{d_{k-1}+1}, \ldots, O_{d_k}\}$. Equation (3.3) holds for all nodes in the path, except for the first node (which does not have a predecessor). For the first node, $p(c_1 \mid \{O_1, \ldots, O_{d_1}\})$, stands for the posterior probability of class label c_1 given the first d_1 observations.

Taking into account equations (3.2) and (3.3), for a given sequence of K nodes (segments) and corresponding class labels, the cost function becomes

$$p(c_1 \mid \{O_1, \ldots, O_{d_1}\}) \cdot \prod_{k=2}^{K} T((O_{d_{k-1}}, c_{k-1}) \rightarrow (O_{d_k}, c_k)) = J(\{d_1, \ldots, d_K\}, \{c_1, \ldots, c_K\}, K) \tag{3.4}$$

According to equation (3.4), the value of function $J(\cdot)$ for a sequence of segments and corresponding class labels can be equivalently computed as the cost of the respective path of nodes in the grid. Therefore, the optimal segmentation can be treated as a best path sequence on the grid.

In order to compute the best-path sequence, we need to define how the best predecessor of each node in the grid is chosen. We first turn our attention to the case where a node, (O_{d_k}, c_k) is not the first node in a path ($k \neq 1$). In this case, the node has to be reached from a node, say (O_{d_l}, c_l), such that $d_1 \leq d_l < d_k$ and $T_{dmin} \leq d_k - d_l \leq T_{dmax}$. Following Bellman's optimality principle, if $J(\{d_1, d_2, \ldots, d_l\}, \{c_1, c_2, \ldots, c_l\}, l)$ is the cost of the best path up to node (O_{d_l}, c_l), then the best predecessor of node (O_{d_k}, c_k) is the one that maximizes the product

$$J(\{d_1, \ldots, d_l\}, \{c_1, \ldots, c_l\}, l)\ T((O_{d_l}, c_l) \rightarrow (O_{d_k}, c_k))$$

If (O_{d_1}, c_1) is the first node (segment) in the path, where $T_{dmin} \leq d_1 \leq T_{dmax}$, we also need to compute $p(c_1 \mid \{O_1, \ldots, O_{d_1}\})$. This procedure is repeated for all nodes in the grid and the coordinates of the predecessor for each node are stored. In the end, we turn our attention to the last column of the grid and choose the node with the maximum value as the winner. The winning node is the last node of the best path. Then, we backtrack through the chain of predecessors to reveal the best path.

As it will be presented in the next section, we have chosen to estimate

$$p(c_k \mid \{O_{d_{k-1}+1}, \ldots, O_{d_k}\})$$

by means of a Bayesian Network combiner.

3.4.3 Bayesian Network architecture

As it was explained in Section 3.4.2, a BN has been used in the DPBS for the computation of posterior probabilities. To this end, the BN is trained as a classifier for the binary classification problem of speech versus music. In other words, *given a segment*, the BN is designed as a classifier combiner that *returns the posterior class probability* (whose value "decides" the class label). It is important to emphasize that this classifier structure decides upon the segment as a whole. For example, the results may be different for $O_t O_{t-1}$ and

$O_t O_{t-1} O_{t-2} O_{t-3}$. This led us to design the classifiers using features corresponding to statistics computed over *the whole length of the segment*. The classification scheme consists of two parts. The "individual" classifiers, operating in one-dimensional feature space, and the BN combiner.

3.4.3.1 Individual Classifiers

At a first step, given a segment, a separate statistic is calculated for each one of the five different features. The statistics that we use are shown in Table 3.1. The choice of the statistics was a result of extensive experimentation and was enforced by the nature of the audio signals under study.

Table 3.1. Statistics for each one the five features that have been used

Feature	**Statistic**
Energy	$\frac{\sigma^2}{\mu^2}$
Chroma 1	μ
Chroma 2	$\frac{max}{\mu}$
MFCC 2	σ^2
MFCC 1	μ

As a result, any segment (feature sequence), irrespective of its length, is mapped by means of statistics to a single five-dimensional vector. Each statistic is fed as input to an individual single thresholding classifier, which takes a binary decision, i.e., decides whether the feature statistic has originated from a speech or music segment. The individual decisions are then combined using a BN, which makes the final decision, as described in 3.4.3.2.

3.4.3.2 Bayesian Network Combiner

The idea behind such a procedure is to use very simple (one dimensional) classifiers, and then use a BN as a combiner to boost the overall performance.

In this work, the BN architecture shown in figure 3.4 ([74]) has been used as a scheme for combining the decisions of the individual classifiers described in 3.4.3.1. We will refer to this type of BN as the BNC (Bayesian Network Combiner). Nodes h_1, ..., h_n (also called

hypotheses, rules, attributes or clauses) correspond to the binary decisions of the individual classifiers for the respective segment. Node Y is the output node and corresponds to the true class label. In the BN training step, one has to learn the Conditional Probability Tables

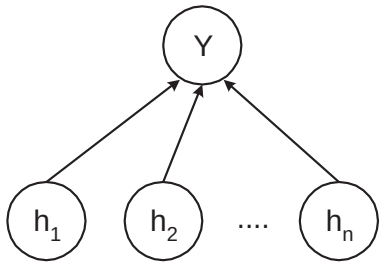


Figure 3.4. BNC architecture

(CPTs) [75] of the BN according to the set:

$$S = \{(h_1(1), \ldots, h_n(1), s(1)), \ldots, \\ (h_1(m), \ldots, h_n(m), s(m)))\} \tag{3.5}$$

where $h_j(i)$ is the result of the classifier $j = 1, \ldots, n$ for input x_i^j, where x_i^j is the feature value presented to the j-th classifier, representing the i-th input pattern, $s(i)$ is the *true label* for $x_i^j, j = 1, \ldots, n$ and m is the total number of training samples. Set S is generated by validating each individual classifier with a test set of length m. In our case, a set of m audio segments with known true class label were used for the training. In general, the CPTs of the BN are learned according to the Maximum Likelihood principle ([75]).

The BN is designed to make the final decision, based on the conditional probability $P_{dec} = P(Y|h_1, ..., h_n)$. The process of calculating P_{dec} is called *inference* and it is, in general, a very time consuming task (see Section B.2.2). However, for the adopted BNC architecture no actual inference algorithm is needed, since the required conditional probability is given directly by the CPT. Another advantage of the specific architecture *is that no assumption of conditional independence among the input nodes (i.e., features) is made* [75].

To summarize, the posterior probability is computed in a three-step process, namely:

1. For any segment, the values of the five statistics are calculated, i.e., x_j, $j = 1, \ldots, 5$.

2. x_j is fed as input to the j-th classifier. Therefore, five binary decisions h_j are extracted.

3. $P_{dec} = P(Y|h_1, ..., h_5)$ is calculated by inferring in the trained BN.

The described BN architecture for probability estimation is presented in figure 3.5. It must be emphasized that training of the classifier scheme has to be performed with a number of speech and music segments, with lengths varying from T_{dmin} to T_{dmax}. However, since the individual classifiers are very simple, this is not much of a problem from a computational point of view.

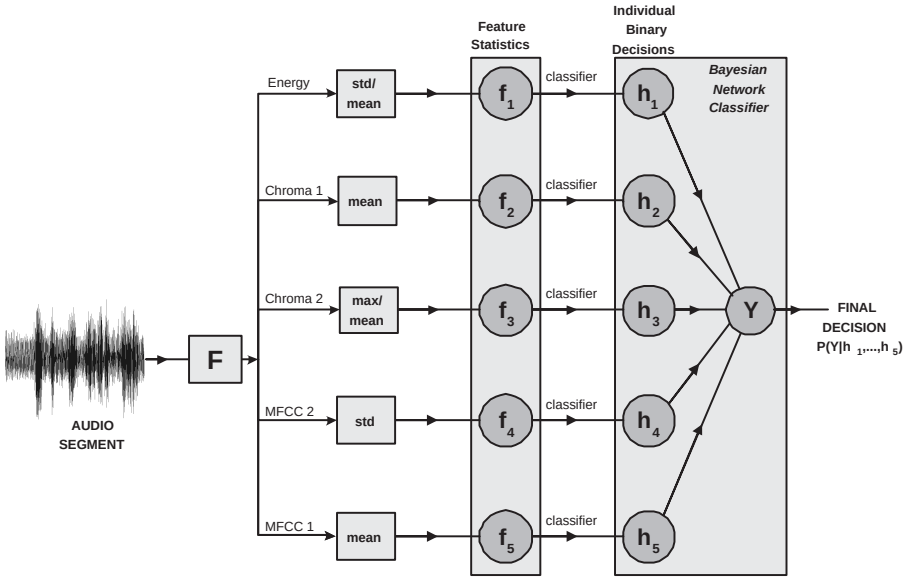


Figure 3.5. BN Architecture for posterior probability estimation

3.4.4 Computational Complexity of the DPBS

We first derive the complexity of the DPBS with respect to the number of required BN inference operations. Parameters T_{dmin} and T_{dmax}, i.e., the minimum and maximum segment length, define the number of predecessors of a node in the grid. A node has threfore $D = 2(T_{dmax} - T_{dmin} + 1)$ predecessors. In addition, the total number of nodes in the grid is $2T$, where T is the length of the feature sequence. Therefore a total of $2T * D$ BN inferences are required, i.e., the number of inferences is $O(T * D)$.

Furthermore, each inference requires 5 thresholding operations and one lookup operation in a CPT of 2^5 entries. As a result, the complexity of the DPBS is also $O(T * D)$ in terms of thresholding and lookup operations. This justifies our choice for the choice of CES as a fast preprocessing stage.

3.5 Post-processing

After the completion of the DPBS step, the audio stream has been segmented and classified. Some of the segments resulted during the CES step and the rest from the DPBS step. In order to further improve the system's accuracy, a post-processing scheme is applied to the segmented data. The post-processing procedure consists of a *boundary correction* algorithm. The idea behind this procedure is to maximize a probabilistic criterion related to the correctness of the boundary's position. This is performed with the following algorithmic steps:

- Let T be the boundary (in seconds) between two segments (speech and music or vise versa). Furthermore, let c_{left} and c_{right} be the labels (i.e., speech or music) of the segments on the left and the right of the boundary T.
- Set $t = T - D$, where D is the searching range, and $i = 0$.
- While $t \leq T + D$ do the following:
  - Let x_{left} be the audio data in the range $[t - D, t]$.
  - Let x_{right} be the audio data in the range $[t, t + D]$.
  - Using the BNC compute the probabilities: $P_{left} = P(Y = c_{left}|x_{left})$ and $P_{right} = P(Y = c_{right}|x_{right})$
  - Set $P_i = P_{left} \cdot P_{right}$.
  - Set $i = i + 1$ and $t = t + 0.050$.
- Calculate $maxPos = \arg\max(P)$.
- Set the new boundary position as follow: $R = T + (maxPos \cdot 0.050 - D)$

The above algorithm locates the boundary that maximizes the product of the probabilities, so that the left and right segments are correctly classified. This boundary correction algorithm, in general, improves the performance of the system if: a) the true boundary is indeed within the search range and b) the initial labels (c_{left} and c_{right}) are correct.

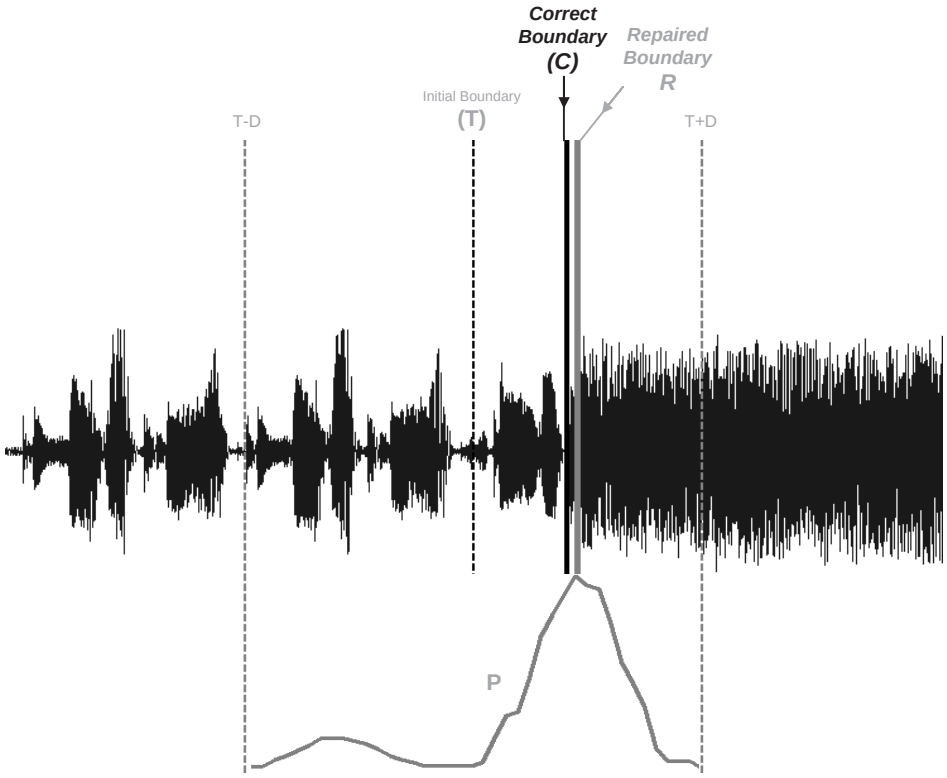


Figure 3.6. Example of the boundary correction algorithm. The initial boundary (T), is used as the center of the search area. The repaired boundary (R) is found by maximizing P, and it is much closer to the real boundary (C).

3.6 Experiments - Results

3.6.1 Data Sets

The following data sets were collected from several Internet radio stations covering a wide range of speakers and some typical musical genres. All recordings were monophonic with a 16kHz sampling rate.

1. Dataset D_1: Consists of 170 minutes of radio recordings that were manually segmented and labelled as music or speech. This resulted in 1100 homogeneous segments of duration of 0.50 to 3.0 seconds. D_1 was used for training and testing the Bayesian Network classifier combiner. *Note that throughout this work, data involving speech over music were considered and labelled as speech.*

2. Dataset D_2: Consists of 60 minutes of radio recordings that were manually segmented and labelled as music or speech. D_2 was used to determine the values of the parameters of the CES.

3. Dataset D_3: Consists of 9 hours of uninterrupted audio recordings from various radio broadcasts. This dataset was divided into six groups according to radio genre (e.g., classical, pop-rock, etc.), in order to test the performance of the system on a genre basis. D_3 was used as the test set for evaluating the performance of the system. The set D_3 was also manually segmented and labelled.

3.6.2 Parameter tuning for the CES

3.6.2.1 Parameter tuning for using CES as a preprocessing stage

The purpose of the CES algorithm within the overall segmentation/classification system is to detect speech and music segments as a fast pre-processing stage. To this end, an exhaustive performance evaluation of the algorithm was meticulously executed on dataset D_2 and the parameter values that **maximize speech and music precision** were determined. The obtained speech and music precision and recall values on the test set D_3 are presented in Table 3.2. In both cases, the precision of the segmentation algorithm is above 98.5% and the recall is almost 54%. This means that about 46% of the audio data will be left unclassified

after the application of the CES algorithm. However, the success rate of the data that have been classified amounts to 98.5%.

Table 3.2. Parameter values for the CES for high class precision.

	T_h	T_{min}	T_{seed}	Precision	Recall
Music	0.3	9.0	2.0	99.5%	45.1%
Speech	0.6	4.0	2.0	98.5%	75.5%

3.6.2.2 Parameter tuning for using the CES as a standalone scheme

Although in this work the CES has been used as a preprocessing method, it could also be used as a stand-alone segmentation system. For the sake of completeness, we also have tested the CES method in such a context. For this purpose, data set D_2 has been used to determine the parameter values. However, for this case the parameters are chosen so that to **maximize the overall accuracy** of the method. In particular, an exhaustive approach was adopted, i.e., each parameter was allowed to vary within a predefined range of values. This parameter estimation process led to the values of Table 3.3. The performance results of CES, when used with those parameter values, are presented and discussed in Section 3.6.4.

Table 3.3. Parameter values subject to maximizing discrimination accuracy over D_2

T_h	T_{min}	T_{seed}
0.50	3.0 sec	2.0 sec

3.6.3 BN-related training and testing issues

In order to train and test the Bayesian Network Classifier, data set D_1 has been used. In particular, 80% of the audio segments of D_1 were used for training and the remaining 20% for testing the BNC, along with the individual classifiers. The results of the classification performances of the individual classifiers and the BNC are displayed in Table 3.4. The best individual classifier (in terms of error rate) is the one based on the 1st MFCC. The

error reduction of the combination scheme, compared to the error of the best classifier, is $e_{red} = 100\frac{|e_{best}-e_{bnc}|}{e_{best}} \simeq 36\%$. The dramatic boosting in performance achieved by the Bayesian Network as a classifier combination scheme is obvious.

Table 3.4. Error rates of the individual classifiers and of the BN combination scheme

	En.	Ch.1	Ch.2	MFCC1	MFCC 2	**BNC**
Music	21%	5%	8.5%	7.5%	14.5%	**3.5%**
Speech	13%	9%	8.5%	3.5%	10.5%	**3.5%**
Overall	17%	7%	8.5%	5.5%	12.5%	**3.5%**

3.6.4 Performance of the overall system and the individual segmenters

The following three schemes were evaluated on dataset D_3:

- The CES, as a standalone discriminator, tuned for maximum overall accuracy.
- The DPBS, as a standalone scheme.
- The overall system without the post-processing step.
- The overall system with the post-processing step (OVERALL2).

Results were recorded for the six radio genres of D_3. The

genre names and respective recording durations, along with the percentage of music and speech data are presented in Table 3.5. In total, almost 70.5% of the audio streams contain music information.

For each method, the average Confusion Matrix was calculated. Each element, $C_{i,j}$, of the confusion matrix corresponds to the percentage of data whose true class label was i and was classified to class j. From C, one can directly extract the recall and precision values for each class:

1. **Recall** (R_i). R_i is the proportion of data with true class label i, that were correctly classified in that class. For example, the recall of music is calculated as $R_1 = \frac{C_{1,1}}{C_{1,1}+C_{1,2}}$.

Table 3.5. Recording Duration per genre

Genre Name	Duration (min)	Music	Speech
POP - ROCK	125	83.02%	16.98%
JAZZ-BLUES	90	67.19%	32.81%
DANCE	85	76.81%	23.19%
NEWS	80	16.17%	83.83%
H. METAL - H. ROCK	80	94.11%	5.89%
CLASSICAL	75	78.64%	21.36%

2. **Precision** (P_i). P_i is the proportion of data classified as class i, whose true class label is indeed i. Therefore, music precision is $P_1 = \frac{C_{1,1}}{C_{1,1}+C_{2,1}}$.

Besides the confusion matrices, the overall accuracy of each segmentation scheme was calculated along with the respective precision and recall values. The overall accuracy, Ac, is the proportion of data that has been correctly classified and it is computed from the confusion matrix according to the equation $Ac = C_{1,1} + C_{2,2}$. The results are displayed in Tables 3.6 and 3.7. The average confusion matrix and respective accuracy (over all genres) for each method is displayed in Table 3.8.

A conclusion drawn from these results is that when CES and DPBS are used independently, as standalone techniques, DPBS offers an enhanced performance compared to CES for most of the genres. The most extreme case is that of genre "News" (table 3.7), where the performance improvement is of the order of 13%. The methods achieve comparable performance for the cases of "Pop-Rock" and "Dance" (table 3.6). This may be explained by the regularity of the music patterns, which makes the problem easier.

Another observation is that combining these two techniques (CES as a preprocessing stage), it only results to an extra gain of the order of 1% with respect to the best individual performance. The obvious question is whether this extra gain really justifies the combination of CES and DPBS. However, the main reason of using CES as a preprocessing stage was primarily of computational nature. As explained previously, the CES algorithm is computationally light. Futhermore, experimentation revealed that on the average, 54% of

Table 3.6. Discrimination results for Pop - Rock, Jazz-Blues, Dance and Classical

Discrimination results for Pop - Rock					
	Precision		**Recall**		
	Music	Speech	Music	Speech	**Overall**
CES	96.6 %	93.8%	98.9%	82.9%	96.2%
DPBS	96.0 %	95.8%	99.3%	80.0%	96.0%
OVERALL	97.2 %	95.1%	99.1%	86.1%	96.9%
OVERALL2	97.5 %	96.5%	99.3%	87.6%	97.4%
Discrimination results for Jazz - Blues					
	Precision		**Recall**		
	Music	Speech	Music	Speech	**Overall**
CES	92.2 %	95.5%	98.1%	83.0%	93.2%
DPBS	99.0 %	92.6%	96.2%	98.0%	96.8%
OVERALL	98.7 %	94.1%	97.0%	97.4%	97.1%
OVERALL2	99.2 %	94.6%	97.3%	98.3%	97.6%
Discrimination results for Dance					
	Precision		**Recall**		
	Music	Speech	Music	Speech	**Overall**
CES	89.8 %	72.0%	92.3%	65.4%	86.1%
DPBS	87.9 %	78.0%	95.2%	56.6%	86.2%
OVERALL	90.3 %	78.8%	94.6%	66.3%	88.0%
OVERALL2	90.1 %	80.6%	95.2%	65.5%	88.3%
Discrimination results for Classical					
	Precision		**Recall**		
	Music	Speech	Music	Speech	**Overall**
CES	91.0 %	100.0%	100.0%	63.5%	92.2%
DPBS	93.6 %	96.6%	99.3%	74.9%	94.1%
OVERALL	93.2 %	99.8%	100.0%	73.1%	94.2%
OVERALL2	93.9 %	99.7%	99.9%	76.1%	94.8%

Table 3.7. Discrimination results for News and Heavy Metal - Hard Rock

Discrimination results for News					
	Precision		**Recall**		
	Music	Speech	Music	Speech	**Overall**
CES	46.8 %	99.0%	95.9%	79.0%	81.7%
DPBS	75.4 %	99.4%	97.0%	93.9%	94.4%
OVERALL	78.4 %	99.4%	97.0%	94.8%	95.2%
OVERALL2	82.7 %	99.4%	97.1%	96.1%	96.3%
Discrimination results for Heavy Metal - Hard Rock					
	Precision		**Recall**		
	Music	Speech	Music	Speech	**Overall**
CES	98.8 %	87.0%	99.2%	81.0%	98.2%
DPBS	99.1 %	86.2%	99.1%	85.3%	98.3%
OVERALL	99.3 %	90.6%	99.4%	88.3%	98.8%
OVERALL2	99.4 %	94.0%	99.6%	90.5%	99.1%

Table 3.8. Average confusion matrix (over all genres) and respective overall accuracies (A) per method.

	CES		**DPBS**		**Overall**		**Overall2**	
	M	**S**	**M**	**S**	**M**	**S**	**M**	**S**
M	69.09	1.59	69.24	1.44	69.34	1.34	69.53	1.15
S	6.74	22.58	4.18	25.14	3.51	25.80	3.17	26.15
	A: 91.67		**Ov. A: 94.38**		**A: 95.15**		**A: 95.68**	

the audio stream is pre-segmented and classified using the CES algorithm (the rest of the data is segmented with the DPBS). Thus, besides a 1% performance gain, employing the CES as a pre-segmntation step leads to a significant reduction in the overall execution time. Finally, the results show that the post-processing step leads to an extra 0.5% performance improvement at only a little extra computational cost.

Inspection of Tables 3.6 and 3.7 also reveals that the worst performance has been reported for the "Dance" genre. This is mainly due to the performance of the CES as a preprocessing stage, which deteriorates when the audio stream consists of drum sounds only, which is quite common in dance music. In order to study this phenomenon more carefully, an additional dataset, D_{drum}, which contains 40 minutes of *only drum sounds* has been created from various radio broadcasts. This dataset was then parsed with CES, tuned for maximum precision. It was observed that the first pass of the CES (tuned for maximum music precision) has correctly pre-classified 25.6% of the 'drums' data as music, while in the general case music recall was 45.1%. This means that it is harder for regions to grow when the audio stream only consists of drums sounds and that it is left up to the DPBS to take the decision. As far as the second pass of the CES is concerned (speech oriented), we would expect that no speech segments are returned at all. However, it was observed that 4.6% of the drum sounds were misclassified as speech, a performance drop compared with the general case reported in Table 3.2. This is the main reason for which the performance on the "Dance" genre decreases to a certain extent. However, given that our study is targeted towards a multitude of genres and that drum sounds in dance music is only a small part of it, the overall system performance is considered satisfactory.

An implementation of the proposed system is publicly available on the Internet at *http : //www.di.uoa.gr/sp_mu*.

3.6.5 Comparison with other methods

This section is an attempt to compare the proposed scheme against methods that have been presented in the literature by other authors. Such a comparison turns out to be a difficult task due to the diversity of data sets that have been used in the literature and the inherent difficulties in reproducing other authors' work. As a result, we have chosen to summarize

in this section the key performance issues of selected papers as presented by the respective authors. It has to be noted that the dataset in this work is significantly larger than datasets used in all previous studies. In addition, we have made an attempt, for the first time, to present results per radio genre (for some well known genres). In terms of response times, the implemented system is comparable with other approaches reported in the literature (e.g., [39]). More specifically :

In [**69**], for training and testing the classifier almost 4500 segments (10 seconds long each) of speech were used, covering several languages and speakers. In addition, approximately 3000 music samples of 10 seconds length were also included in the experiments. The music data was a diverse selection of several musical genres like classical, jazz, African and Arabian. In total, 10 hours of audio data was collected. Each sample either contained speech or music. The classification task was carried out, using Gaussian Mixture Models (GMMs). The reported experiments showed that, depending on the adopted features, the error rate ranges from 1.2% to 6%. It has to be noted that the assumption of homogeneous audio segments of quite a long duration (i.e., 10 s) lead once more to a simplified version of the problem.

In [**70**], for training and testing purposes, almost 13 thousand audio files were obtained from the World Wide Web and were manually labelled as speech, music or other. The duration of each file ranged from 0.5 seconds to 7 minutes, the average duration was 48 seconds and the total duration of the audio data was more than 170 hours. Each audio file contained either speech of music. A limited number of files were non-homogeneous, i.e., contained both speech and music parts. In such cases, during the manual labeling stages, the dominant label was chosen for the whole file. The authors used the "One vs All" and "One vs One" classification schemes for this three-class problem and also present a simple way to combine the results of the two schemes in order to boost performance. The best overall accuracy is around 82%. The reported performance cannot be directly compared with other methods methods, because three classes are treated and each audio file is considered to be homogeneous, an assumption that simplifies the problem of speech/music discrimination.

In [**71**], almost 20 minutes of audio data were used for training and testing purposes. The authors reported results for different feature sets and binary classification methods. On a short-term basis the overall accuracy was around 80%. When a mid-term window was used (1 s long), the accuracy rose to approximately 95.9%. In [**39**] results are reported for four

artificially created datasets (40 minutes total audio duration). The reported performance varies in the range $93\% - 96\%$. The origin of datasets poses an inherent difficulty in comparing this method with other approaches in the literature.

[**72**] works on a frame-level basis. A binary (speech/music) classification decision is taken separately for each short-term frame. The dataset consists of 240 audio recordings, each of which is 15s long (total recording duration is 1 hour). Part of the dataset is used for training purpses. An accuracy of 88%, on frames sampled at 20msec intervals, is reported. When a smoothing technique is applied, the performance rate reaches 93%.

In [**37**] the total speech duration in the audio corpus was 3 hours and 9 minutes, which was subdivided by the segmentation algorithm into about 800 segments (over-segmentation); 97% of these segments were correctly classified as speech. The total music duration in the audio corpus was 52 min, which was subdivided by the segmentation algorithm into about 400 segments (over-segmentation); 92% of these segments were correctly classified as music.

3.7 Conclusions

This chapter presented a multi-stage speech/music discriminator that combines two different approaches: a computationally efficient region growing technique along with an optimal, yet more computationally demanding dynamic programming scheme. The system was tested on 9 hours of audio recordings stemming from a variety of radio broadcasts and its overall accuracy approximates 96%. For some genres, e.g., Hard-Rock, the performance rockets up to 99%. These results compare very favourable with previously obtained results, although a direct comparison is not possible due to the lack of standard datasets. Furthermore, in all previous works there lacks a study on a genre basis.

Chapter 4

Music Tracking in movies

Music tracking in audio streams can be defined as the problem of *locating the parts of an audio stream that contain music, possibly overlapping with other types of audio.* In the literature, the term "music tracking" is often used interchangeably with the term "music detection". We have chosen to use the term "music tracking", in an analogy with the speech processing literature [76] where "speaker tracking" refers to the task of deciding which parts of the speech signal refer to a specific speaker. In addition, the term "detection" does not comprise the meaning of localization of the event of interest.

The problem of music tracking in audio streams has recently attracted a lot of attention, mainly in the context of audio content characterization applications. Intelligent browsing of audio streams, automatic audio content annotation/ indexing, querying audio streams by audio example and copyright management are some of the tasks that can benefit from efficient music tracking algorithms.

In the general case, music tracking is a hard task, because music is frequently mixed with other audio types. This is more apparent in the case of audio streams from movies, due to the diversity of sound sources involved in a film's soundtrack. *In the present work, no assumptions concerning the types of audio to be encountered in the stream have been made. This was the most important challenge of this task, along with the need for a computationally efficient method.*

In the following paragraphs, a computationally efficient method for tracking music in audio streams from movies is presented. The audio stream is first mid-term processed with a fixed length moving window and four features are extracted per window. Each feature is

fed as input to a simple classifier, which produces a soft output for the binary problem of music vs. all other types of audio. The soft outputs are then combined to yield a measure of confidence that quantifies whether the segment corresponds to music or not. At a final step, thresholding is applied to filter out segments, for which the confidence measure is low. The proposed approach has been tested with audio streams from various movies and its performance was measured both on a mid-term segment basis as well as on an event detection basis. Reported results demonstrate that the method exhibits high performance even when music is mixed with other types of audio in the stream.

4.1 Previous works

Related work in the field ([77, 78, 79]) has so far treated the problem of music tracking as a binary classification task on a short-term frame basis; the audio stream is first divided into a sequence of short-term frames, by means of a moving window technique, and a separate classification decision is taken for each short-term frame for the binary problem of music vs. other types of audio. A post-processing stage is also employed in most cases in order to smooth the results and produce longer segments. It can be stated that emphasis has so far been given on selecting a feature set that provides high discrimination performance on a short-term basis using standard classifiers, i.e., kNN or GMM based ones. Comparative studies of features can be found in [77] and [78]. The work in [77] deals with the task of music tracking in TV productions and proposes that using a feature that captures the shape of spectrograms of music signals on a short-term basis (the "Continuous Frequency Activation" feature) along with a single thresholding classifier is sufficient to yield satisfactory performance. In [78] emphasis is given on detecting pure music and music mixed with speech in artificially created datasets, where music is mixed with speech at varying music to background signal ratios. Finally, the work in [78] deals with the related, yet simpler problem, of music detection in audio streams from user-generated video clips, i.e., the authors have developed a system that answers whether a video clip contains music or not. The features used in [79] evolve around the assumption that a music signal exhibits certain harmonic and rhythm-related properties.

4.2 Proposed method: General

The method presented in this work is different in the following aspects:

- It treats the problem on a **mid-term segment** basis, i.e., the audio stream is processed with highly overlapping mid-term segments, in order a) to avoid classification decisions on a short-term basis and b) to exploit the fact that there exists certain context dependency among successive short-term frames. To this end, four features that are related to the properties of music signals are extracted per mid-term segment.
- The proposed classifier functions on a mid-term segment basis for the binary problem of "music vs. all other types of audio" is a simple combiner of histogram-based weak learners. Note that this type of approach is independent of the features used and can be considered as a general framework for the task at hand. Furthermore, the computational complexity is kept low, around 10% of the duration of the audio recording (measured in seconds).
- No assumptions concerning the types of audio, which are likely to be encountered in an audio stream are made. Moreover, the performance of the proposed approach is tested on an audio corpus where a multitude of audio events is encountered.

4.3 Feature extraction

At a first step, the audio signal is mid-term processed with a moving window technique. In particular, the mid-term window length is equal to 3 secs, while a 2.5 secs overlap exists between successive windows. The goal is to extract four features per mid-term window. Each feature is a statistic, computed over a sequence of short-term features comprising the mid-term window. The specific choice of the short-term features and the related statistics are the result of extensive experimentation, which has indicated that this choice leads to high discrimination performance for the music vs all classification task. In particular, the following four features / statistics have been used:

1. **1^{st} chroma based feature:** This feature is described in Section 2.4.6 and it experiences higher values for music segments.

2. **2^{nd} chroma based feature:** This second chroma feature is a measure of the degree of variation of each chroma element over successive short-term frames (Section 2.4.6), and therefore it has lower values for music signals. Both chroma features have been calculated using the window lengths described in Section 2.4.6.

3. **Minimum Entropy of Energy:** Entropy of energy is a measure of abrupt changes in an audio signal (see Section 2.3.3). 50 msecs short-term windows have been adopted for the feature sequence calculation, while 10 sub-frames were used in each frame (for the computation of the entropy as described in Section 2.3.3). The adopted statistic for this feature is the *minimum value* over all short-term frames of the mid-term window. Experiments have indicated that this feature exhibits higher values for music segments. This is something expected, since low values of H correspond to abrupt signal changes in a small time duration, while it is obvious that for music signals such changes occur less frequently.

4. **Non-zero Pitch ratio:** To compute this feature, the mid-term window is first broken into non-overlapping short-term frames, 50 msecs long. From each short-term frame the pitch is extracted by means of a standard autocorrelation-based pitch detection method ([80]). This particular pitch tracker has been chosen because of its computational simplicity. Once all pitch values have been extracted, the *non-zero pitch ratio*, i.e., the percentage of frames with non-zero pitch, is employed as a statistic. This feature can be considered as a measure of the harmonicity of the audio signal. Our experiments have indicated that music segments tend to exhibit high values for this feature.

4.4 Music tracking

Let $\underline{x} = [x_1\ x_2\ x_3\ x_4]^T$ be the feature vector that has been extracted from a mid-term segment and let ω_1 and ω_2 stand for the class of music segments and non-music segments respectively. Our next goal is to estimate $p(\underline{x} \mid \omega_1)$ and $p(\underline{x} \mid \omega_2)$. To this end we assume that the x_i s are statistically independent. Therefore

$$p(\underline{x} \mid \omega_1) = \prod_{k=1}^{4} p(x_k \mid \omega_1)$$

and

$$p(\underline{x} \mid \omega_2) = \prod_{k=1}^{4} p(x_k \mid \omega_2)$$

In order to estimate $p(x_k \mid \omega_1)$, $k = 1, \ldots, 4$ and $p(x_k \mid \omega_2)$, $k = 1, \ldots, 4$ we resort to a simple histogram lookup operation. For each feature, the histogram of values for each class is known and it is generated during the training stage (two histograms per feature). This is practically equivalent to estimating the pdf of each feature by means of Parzen approximation with rectangular windows. Details of the training stage can be found in Section 4.5.

At a second step, the following log-likelihood ratio is computed:

$$\begin{aligned} R(\underline{x}) = & \quad \log \tfrac{p(\underline{x}|\omega_1)}{p(\underline{x}|\omega_2)} = \log p(\underline{x} \mid \omega_1) - \log p(\underline{x} \mid \omega_2) \\ = & \quad \textstyle\sum_{k=1}^{4} \log p(x_k \mid \omega_1) - \sum_{k=1}^{4} \log p(x_k \mid \omega_2) \end{aligned} \tag{4.1}$$

This technique is common in speaker tracking [76]. $R(\underline{x})$ can be considered as a soft output, i.e., a measure of confidence that $\underline{x}$ has been extracted from a music segment. When the histogram lookups yield comparable results, the log-likelihood ratio will be close to zero, indicating a case of uncertainty. On the other hand, positive values are in favor of music and negative values indicate other types of audio.

To proceed, let $\mathbf{P} = \{P_k; P_k = R(\underline{x}_k), k = 1, \ldots, L\}$ be the sequence of soft decisions for all mid-term segments, where L is the number of segments. $\mathbf{P}$ is then processed by means of a median window, 7 mid-term frames long, to remove spurious values. A hard threshold, T_h, is then applied and all mid-term segments with P value exceeding T_h are kept as music segments. In the end, short segments (shorter than 3 seconds) are filtered out. After extensive experimentation, the recommended threshold value was chosen to be equal to 0.1. Figure 4.1 presents the P-sequence for a part of an audio stream from the movie "Pink Floyd - The Making of 'The Dark Side of the Moon"'. The solid horizontal line indicates the position of the threshold.

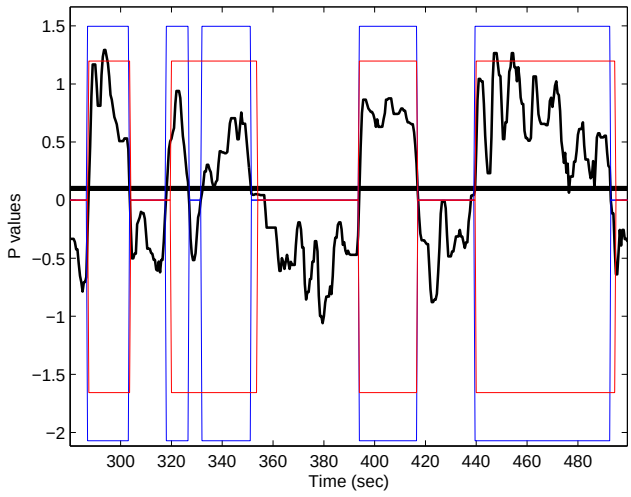


Figure 4.1. Sequence of soft decisions for a part of an audio stream. Horizontal line represents the threshold (0.1). Red rectangles represent the true music segments, while blue rectangles represent the detected music segments.

4.5 Experiments

4.5.1 Datasets

Two distinct datasets have been used, one for training, i.e., for generating histograms, and one for testing the proposed method. Details of the datasets are given below. It has to be emphasized that for both datasets a manual annotation stage was necessary. In the case of audio streams from movies, manual annotation is also a challenging task because humans tend to perceive music boundaries differently, especially in the case where music is mixed with other audio events. To deal with this problem, the manual annotation task was carried out independently by three individuals. The annotation results were merged by adopting the rule that part of a recording contains music only when the labels of the annotated data of all three individuals coincide.

1. **Training Dataset** In order to generate the histograms of values for each feature per class, 4000 homogeneous audio segments have been manually extracted and labeled from more than 30 movies. Half of the segments were used for populating the histograms of each class. Care was taken so that, in the case of music, a large portion of the segments contained music mixed with other types of audio, e.g., speech and en-

vironmental sounds. Similarly, the non-music segments contain several types of audio that are frequently encountered in movies, e.g., speech, gunshots, explosions, environmental sounds, machine sounds, screams, beatings, etc. The average duration of the segments in each class was approximately 3 seconds. Figure 4.2 shows the feature histograms per class for all four features.

2. **Testing Dataset** In order to evaluate the performance of the proposed music tracking system, audio streams from eight movies have been manually annotated. The audio streams were "ripped" directly from the movie DVDs and were channel averaged and resampled to 16 KHz. In Table 4.1, the details of this dataset are given. It can be seen that the total duration of the audio streams is 2.5 hours. Music duration is approximately 39 minutes (almost 26% of the total recording time), and the total number of music segments is 140.

4.5.2 Evaluation Results

The proposed system was first evaluated on a mid-term basis. Since the mid-term step is 0.5 secs, a correct classification decision on a mid-term segment (3 secs long) is considered to be valid only for the first 0.5 seconds of the segment. As a result, if M successive mid-term segments yield the same classification decision, then the length of the resulting homogeneous segment is $M \times 0.5$ secs. As it is usually the case, precision and recall, as defined below, were used as the performance evaluation measures on a mid-term basis:

- Music Precision: The proportion of audio data that was classified as music and was indeed music.
- Music Recall: The proportion of music data, that was correctly classified as music.

The system's performance has also been measured using another pair of performance measures that refers to the event detection ability of the algorithm:

- Music Detection Precision: The number of detected music segments, that were indeed music, divided by the total number of detected music segments.
- Music Detection Recall: The number of correctly detected music segments divided by the total number of **true** music segments.

Table 4.1. Audio streams from movies, used for testing the proposed method: Movie title, genre, audio duration (D, in minutes), music duration (MD, in minutes) and number of music events (#S).

Movie Title	Genre ([81])	D	MD	#S
Harry Poter	Adventure / Family / Fantasy	10	2.51	15
The Aviator	Biography / Drama	5	0.73	5
The Bear	Adventure / Family / Drama	20	4.39	5
U-571	Action / Drama / War	15	2.47	15
Billy Eliot	Comedy / Drama	25	6.55	30
Kill bill 1	Action / Crime / Drama / Thriller	25	2.71	5
The Phantom of the Opera	Drama / Musical / Romance / Thriller	20	8.24	25
Pink Floyd - The Making of 'The Dark Side of the Moon'	Documentary / Music	30	11.52	40
Total	-	**150**	**39.1**	**140**

Note that by "correctly detected music segments", we mean the detected segments that overlap with a true music segment. The values of the two kinds of measures may differ significantly. In Figure 4.3, an example of music detection is given (for an audio stream with four music segments). In this case, the detection precision is 100% (all three detected segments are indeed music segments), while the detection recall is 75% (three out of four music segments have been detected). Furthermore, the precision of classified data is

$$Pr = \frac{T/2 + T + T/8}{T/2 + 1.2T + T/8} = 89\%$$

and the recall is equal to

$$Re = \frac{T/2 + T + T/8}{T + T + T/2 + T/2} = 54\%$$

.

In Figures 4.4 and 4.5 the results of the classification and detection process are respectively presented (precision, recall and F1 measure) for different values of the threshold.

In both cases, the threshold affects the precision and the recall rates. As expected, higher values of the threshold lead to higher precision and lower recall (and vise-versa). Depending on the demands, this parameter can therefore be used accordingly. If, for example, a specific multimedia application requires very high detection precision rates, the largest threshold value (T=1) can be used, and therefore achieve a precision rate over 95%, while the recall rate will drop to almost 75%. In this work, threshold value $T = 0.1$ maximizes the F1 measure of the detection performance. For this value, the performance (classification and detection) is presented, in detail, in Table 4.2.

Table 4.2. Classification and detection performance for threshold value $T = 0.1$.

	Precision	**Recall**	**F1 Measure**
Classification	89%	83%	86%
Detection	91%	90%	90.5%

4.5.3 Computational complexity

Excluding the feature extraction stage and following equation (4.1), the cost of a classification decision per mid-term segment is equal to the cost of 8 histogram lookups plus the computational cost of 8 logarithms, 6 additions, 1 substraction and 1 threshold comparison. Assuming that the costs for addition, substraction and threshold comparison are practically equal and taking into account that an audio stream which is N seconds long, will yield approximately $\frac{N}{0.5} = 2N$ mid-term segments (mid-term step is 0.5 secs), then the computational cost for all classification decisions is $8 * 2 * N$ histogram lookup operations plus $8 * 2 * N$ logarithms plus $8 * 2 * N$ additions. In other words, the overall classification cost is equal to $16 * N * (lookup\ cost + log\ cost + addition\ cost)$ in terms of computational burden. According to our experiments, the proposed system spends approximately 1.5% of the total recording time on this stage, on a standard PC platform where the implementation has been carried out in the Matlab programming environment.

Concerning the feature extraction stage, it can be stated that it takes approximately 8.5% of the recording length of the audio stream (measured in seconds) to be completed. Overall, the proposed method requires around 10% of the length of the audio recording to complete execution.

4.6 Conclusions

This chapter has presented a robust and computationally efficient music tracker in the context of audio streams from movies. The system takes classification decisions for the binary problem of music vs. all other types of audio on a mid-term segment basis. To this end, a feature extraction stage yields four feature values per mid-term segment. These are statistics which are computed over short-term features on each mid-term segment. The classifier thus combines the soft-outputs of four histogram-based weak learners. The performance of the method has been measured both on mid-term segment basis as well as on an event detection basis. The results indicate that the method is robust when music is in the background of other audio events, while computational complexity is kept quite low (around 10% of the recording time). The proposed music tracker can be used in an overall system for multi-class audio classification, as a pre-processing stage. For the particular case of violence detection, it can be used in order to exclude audio areas from the more computationally demanding classification-segmentation algorithms.

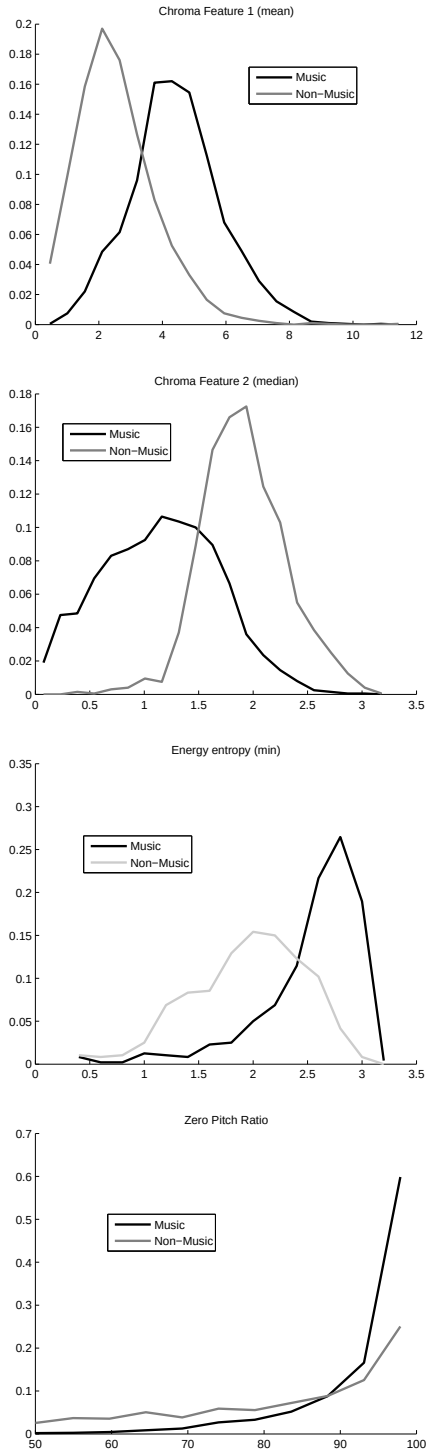


Figure 4.2. Histograms of all four features for both classes (Music vs Non-Music)

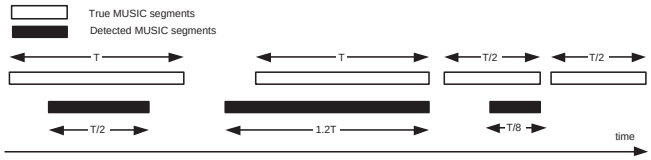


Figure 4.3. Music tracking example

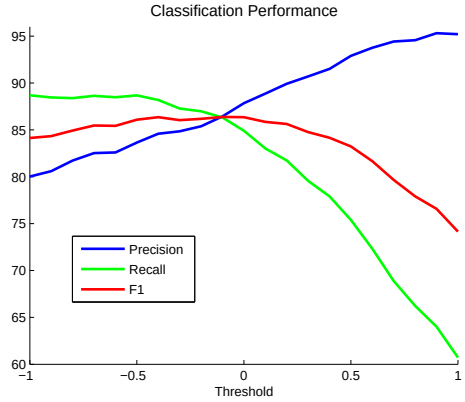


Figure 4.4. Classification Performance

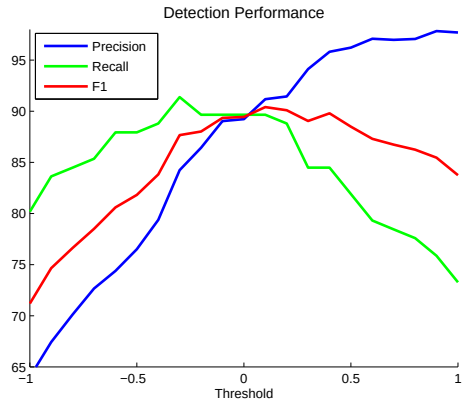


Figure 4.5. Detection Performance

Chapter 5

Audio Segmentation

The purpose of audio segmentation is to *locate changes* in the content of the audio signals; in other words, to detect changes among acoustically homogenous audio regions. It is an important preprocessing step in any audio characterization system. Music information retrieval, video segmentation and audio characterization in security surveillance systems are some notable applications of high current interest. In such systems, besides accuracy, computational time is also of paramount importance, especially when a real-time or almost a real-time operation is desirable.

In this chapter, a novel approach to audio segmentation is presented. The problem of detecting the limits of homogenous audio segments is treated as a **binary classification** task. Each audio frame is classified as **"segment limit" vs "non-segment limit"**. For each frame, a spectrogram is computed and eight feature values are extracted from respective frequency bands. Final decisions are taken based on a classifier combination scheme. The algorithm has very low complexity with almost real time performance. The algorithm has been evaluated on real audio streams from movies and it achieves 85% accuracy rate. Moreover, it introduces a general framework to audio segmentation, which does not depend explicitly on the number of audio classes.

5.1 Introduction

In general, audio segmentation approaches can be categorized into supervised and unsupervised techniques. Supervised approaches, e.g., [82], use a group of *a-priori known* audio

classes and audio segmentation is performed via a classification task, by assigning audio frames in the respective classes. Unsupervised techniques treat audio segmentation as a hypothesis test by detecting changes in the audio signal, given a specific observation sequence ([33], [35], [34]). Another differentiation between audio segmentation methods is that, depending on the task, the definition of homogeneity may vary. For example, the notion of homogeneity is different when dealing with a speech-music segmentation task, than with speaker change detection.

In this chapter, the supervised approach rationale is followed, albeit using a completely different viewpoint, compared to previously developed techniques. Since all it is required is to detect content "changes" in the audio stream, we focus on this task *directly*, instead of solving another problem first (i.e., a classification task) and trying to infer our desired goal from it. Using this path, no a-priori assumption on the number of audio classes is required, which in a general audio stream cannot be easily determined. The segmentation task is treated as a binary classification problem: non-segment limit vs segment limit (the term "segment" refers to a part of an audio stream with homogenous segment). In turns out that the proposed method has substantially lower computational demands, without sacrificing accuracy, compared to previously derived techniques.

The proposed algorithm first computes eight feature sequences, from eight corresponding frequency bands of the spectrogram of the audio stream. After extensive experimentation, we found that these eight bands are sufficient to encode and monitor activity of different types of audio signals. For each band, transition activity is first measured on a frame basis. In the sequel, for each frequency sub-band, a binary classification problem is defined: non-segment limit vs segment limit, on a frame basis. Then, a simple histogram-based classifier is employed for each frequency band (binary sub-problem) and final results are obtained by combining individual outputs. In order to train the binary classifiers, only audio streams with known segment limits are used and, as stated before, there is no need for any assumption concerning the class of the individual segments: we only need to know that the segments are of homogenous content. Segment limits are finally detected by computing local maxima in the output of the combiner.

5.2 Feature Extraction

At a first stage, the audio stream is divided into non overlapping 100 msec frames and the spectrogram, $S_{t,f}$, of the signal is computed (t is the frame index and f is the frequency bin index). At a next step, eight frequency sub-bands are defined according to the following frequency limits (in Hz):

$$fb = \{0, 150, 400, 800, 1500, 2500, 4000, 5500, 8000\}$$

.

The bands have been selected after extensive experimentation. For each sub-band, the normalized spectral energy is then calculated:

$$E_i(t) = \frac{\sum_{f=fb(i-1)}^{fb(i)} S(t,f)}{\sum_{f=0}^{Fs} S(t,f)}, i = 1 \ldots, 8$$

.

Each E_i is then smoothed using a fixed averaging window (1.0 sec long). Finally, for each E_i, a sequence of energy changes is computed, using a long-term window of length SW, i.e. the following *distance function* is computed:

$$D_i(t) = |\frac{\sum_{\tau=t-SW/2}^{t-1} E_i(\tau)}{SW/2} - \frac{\sum_{\tau=t+1}^{t+SW/2} E_i(\tau)}{SW/2}|$$

.

SW has been selected to be equal to 3 seconds (30 frames). $D_i(t)$ expresses the degree of change of E_i around the t-th short-term frame. In Figure 5.1, an example of E_1 and D_1 is presented, for a short audio stream containing two segments. It can be observed that the real segment limit (vertical line) lies in short distance to the local maxima of D_1.

5.3 Classifiers Architecture

5.3.1 Individual Binary Classifiers

For each sequence $D_i, i = 1, \ldots 8$, a separate one-dimentional binary classification problem is defined: "Non - Segment Limit" (class ω_1) vs "Segment Limit" (Class ω_2), on a frame

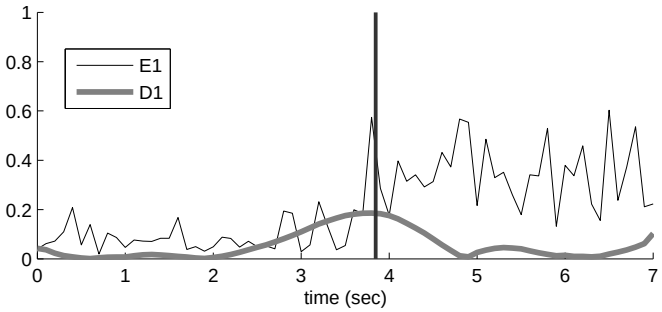


Figure 5.1. E_1 and D_1 for an audio stream.

basis. In order to train the binary classifiers, 25 hours of audio streams of known segment limits were used: the values of D_i, which correspond to frames within a 0.5 second interval (tolerance), before and after the real segment limits, were used to populate class ω_2 and all other vales of D_i were used to populate class ω_1. As an example, consider the 4-segment audio stream of figure 5.2, which is used for training the first binary classifier and for which the segment limits are known. The values of D_1 in an interval of ± 5 frames (± 0.5 sec) around the true segment limits (**bold** areas of D_1) are used to populate class ω_2 and all other values of D_1 are used for ω_1.

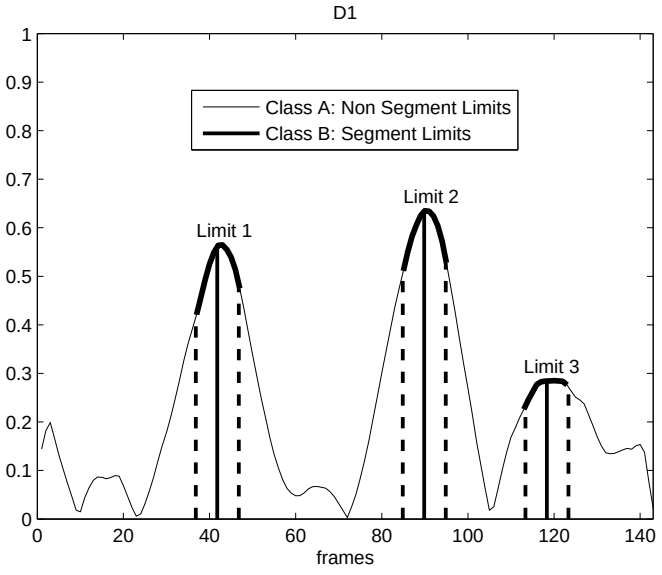


Figure 5.2. The class population process.

The same process is repeated for all available pre-segmented audio streams and for all eight frequency bands. It must be emphasized the audio segments that constitute the audio

streams are of a homogenous content. More details about those homogenous audio segments can be found in Section 5.6.1. When the training datasets for both classes are populated, the respective histograms are calculated and used to estimate the two pdfs underlying the two classes. In other words, for each binary classification sub-task i, $P(D_i|\omega = \omega_1)$ and $P(D_i|\omega = \omega_2)$ are estimated. This comprises the training phase. In figure 5.3 the histograms for the two classes of the 1st classification task are presented.

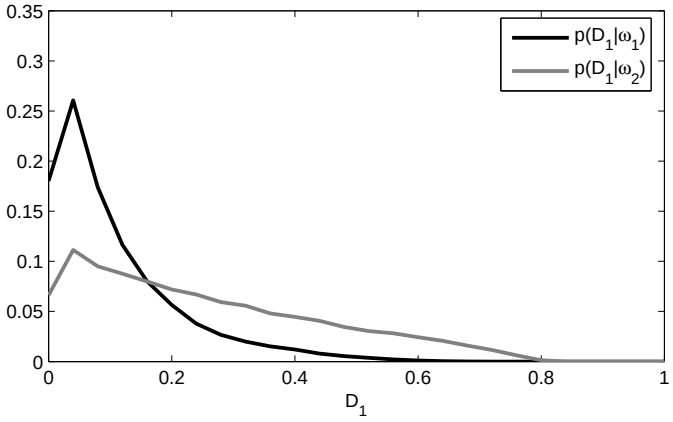


Figure 5.3. Histograms for the 1st binary classification task. $P(D_1|\omega = \omega_1)$ is the estimated probability that the value of the first distance function is D_1 for a non-segment limit, while $P(D_1|\omega = \omega_2)$ is the estimated probability that the value of the first distance function is D_1 for a segment limit.

In the classification stage, given an unknown frame, j, the following measure is computed: $CM_i(j) = \frac{P(\omega=\omega_2|D_i(j))}{P(\omega=\omega_1|D_i(j))}$, where $D_i(j)$ is the distance function of the i-th sub-band and the j-th frame. $CM_i(j)$ is the (soft) output of the i-th classifier associated with the jth frame. This is **a measure of confidence** that the j -th frame is a segment limit in the i-th sub-band sequence.

5.3.2 Combination Rules

Classifier combination aims at boosting the performance of the individual classifiers ([52]). In the current chapter, three rules have been implemented for combining the classifiers' soft outputs. The simpler combination scheme is the rule of the **arithmetic average**; the average value, $CM(j)$ is computed as: $CM(j) = \frac{\sum_{i=1}^{8} CM_i(j)}{8}$

The second combination method is the Global Weighted Average rule (GWA); a different weight W_i is assigned to the soft output of each classifier, i.e., $CM(j) = \frac{\sum_{i=1}^{8} W_i \cdot CM_i(j)}{\sum_{i=1}^{8} W_i}$. To choose the weights W_i, the overall performance of the segmentation method was tested, using each individual classifier, and the overall accuracy was used as a weight in each case.

The third method follows the one suggested in [83], where each classifier's accuracy, in local regions in the feature space, is estimated and then the decision of the most locally accurate classifier is used (Classifier Selection). In this chapter, the Local Weighted Average (LWA) method is used to assign to each classifier **a weight that depends on its soft-output**. A histogram W_i of each classifier's accuracy, for different soft-output values $CM_i(j)$, is trained, by testing the segmenter's performance using the individual classifiers. In other words, the local weight $W_i(CM_i(j))$ is an estimate of the i-th classifier's (local) accuracy, when the output value is $CM_i(j)$. The combined output of the LWA method is computed by: $CM(j) = \frac{\sum_{i=1}^{8} W_i(CM_i(j)) \cdot CM_i(j)}{\sum_{i=1}^{8} W_i(CM_i(j))}$.

In order to estimate the local (or global) accuracy, the segmenter (using each individual classifier) is tested on audio streams with known segment limits. In particular, using each individual classifier i, the segmentation process is first applied on the audio stream. Note that in order to estimate the weights, the **whole** segmentation process is applied, i.e., after the calculation of the combined output CM, the detection process described in Section 5.4 is also applied, in order to detect the possible segment limits. In the sequel, the segmentation correctness is checked, on a frame basis. In details, for each frame j the following steps are executed (after the segmentation process):

1. Find the closest *real* segment limit L_R and compute its distance from the current frame: $D_R = |L_R - j|$.

2. Find the closest *detected* segment limit L_S and compute: $D_S = |L_S - j|$.

3. Define a tolerance $D_T = 0.5sec$ and compute:

$$r(j) = \begin{cases} 0, & \text{if } D_R \geq D_T \text{ and } D_S \geq D_T \\ 1, & \text{if } D_R < D_T \text{ and } D_S \geq D_T \\ 2, & \text{if } D_R \geq D_T \text{ and } D_S < D_T \\ 3, & \text{if } D_R < D_T \text{ and } D_S < D_T \end{cases}$$

If $r(j) = 0$ frame j is neither near a true or a detected limit (true negative decision). In the case that $r(j) = 1$, the current frame is near a true limit but not a detected limit (that can be counted as a false negative decision), while if $r(j) = 2$, this obviously means that a false alarm has occurred. Finally, if $r(j) = 3$, the j-th frame is both in the area of a detected and a real limit (true positive decision).

4. Using $r(j)$, define a correctness sequence c, which specifies whereas the segmentation process is correct, on a frame basis:

$$c(j) = \begin{cases} 1, & \text{if } r(j) = 0 \text{ or } r(j) = 3 \\ 0, & \text{otherwise} \end{cases}$$

Obviously, the overall accuracy of the segmentation process can be computed directly from c: $A = \frac{\sum_{j=1}^{L} c(j)}{L}$, where L is the total number of frames. This measure is used as a global weight in the GWA method. On the other hand, in the LWA method, as explained above, our purpose is to compute the accuracy for specific values of CM and use it as a weight. Towards this end, we define a set of bins for the CM sequence. For each bin value CM_b there is a respective c_b subsequence of c, which is composed by the frames whose CM values belong to the b-th bin. Therefore, the (locally estimated) accuracy is computed according to the equation: $A_b = \frac{\sum_{j=1}^{L_b} c_b(j)}{L_b}$, where L_b is the length of c_b (and CM_b.) This process is repeated for several audio streams with known segment limits, and the weights are computed by averaging the respective local accuracies. In figure 5.4, an example of the weights for the 1st and 8th classifiers is presented. For example, the fact that the weight of the 1st classifier for the first CM bin is almost 0.88, means that a soft decision of the 1st classifier that belongs to that CM bin is accurate 88% of the time.

Figure 5.5 is an example of the results obtained by the three combination rules for the case of a 4-segment audio stream with known segment limits (vertical lines). The first eight sub-figures show the outputs of the individual classifiers (CM_i), and the last sub-figure shows the combined output (for all three combination schemes).

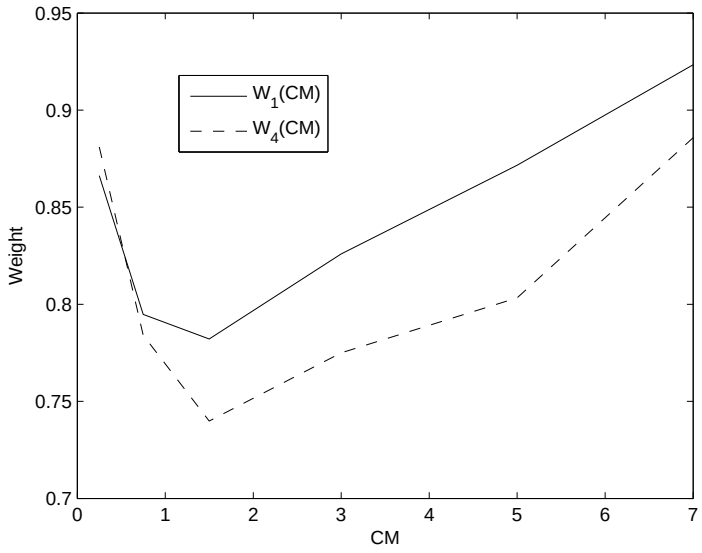


Figure 5.4. Weights for the 1st and the 8th classifier in the LWA method.

5.4 Detection of segment limits

All combination methods in 5.3.2 lead to a fused soft output, which can be interpreted as the overall certainty measure that a frame belongs to a segment limit. High values of this quantity are interpreted as an indication that the probability of the respective frame being a segment limit is also high. Therefore, a local maxima detection algorithm has been applied to the resulting CM sequence. At a second stage, the local maxima are post-processed by means of a global thresholding algorithm.

5.4.1 CM Maxima Detection

For estimating the local maxima of the resulted CM sequence (and therefore the detected segment limits) the following algorithm has been implemented:

- Step 1: Detect all elements i that satisfy both:

$$\frac{\sum_{j=1}^{maxWin} CM(i-j)}{maxWin} < CM(i)$$

and

$$\frac{\sum_{j=1}^{maxWin} CM(i+j)}{maxWin} < CM(i)$$

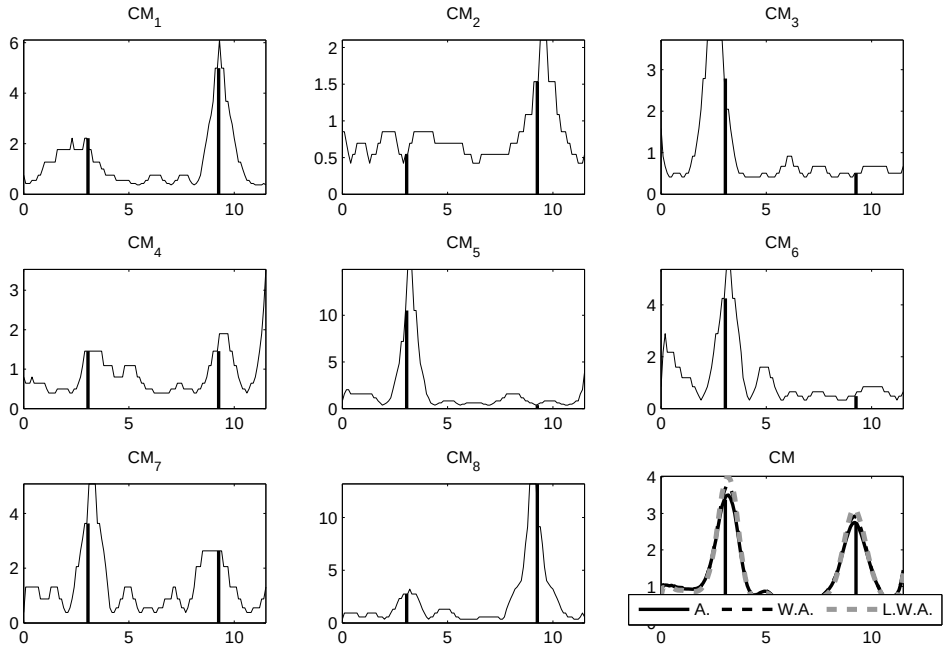


Figure 5.5. Individual and combined CMs.

. $maxWin$ is a user-defined parameter. In other words, a frame i is detected if the average CM values of the windows (of length $maxWin$) on the right and on the left of i are smaller than $CM(i)$.

- Step 2: Divide the detected elements of step 1 in "groups of neighbors": the distance of two successive elements of the same group should be $\leq \frac{maxWin}{2}$.
- Step 3: The element with the maximum value of each group of neighbors is the detected local maximum, and therefore the detected segment limits.

In Figure 5.6, an example of the maxima estimation algorithm is presented, for a sequence with three local maxima.

5.4.2 Thresholding

Detected maxima are then post-processed by applying a thresholding criterion. In particular, a global, user-defined threshold T is used. A thresholding example is presented in Figure 5.7, where $T = 1$.

5.5 Random Segmentation

For comparison reasons, in this paragraph we describe the theoretical performance of a random segmenter. As in classification methods, where the performances are compared to the process of random selection of a class, we compare the proposed segmentation methods to the random segmenter, which places the segment limits in random positions uniformly distributed in the audio stream. In the sequel, we are presenting the performance of this random selection of segment limits. Let:

- d: average segment duration (true segments)
- N: number of true segments
- m: average segment duration (random segmenter)
- M: number of segments (random segmenter)
- L: signal length
- tol: error tolerance

In figure 5.8 a general example of random segmentation is presented. Dotted lines represent (randomly) estimated segment limits, while solid lines represent the real segment limits. Obviously, the average segment duration (of the real segments) is related to the number of true segments and the total signal length, according to the equation: $d = \frac{L}{N}$. Similarly, for the average duration of the (randomly) detected segments the following equation is true: $m = \frac{L}{M}$. It has to be noted that the only parameter of the random segmenter is the average duration of its segments m.

A random segment limit is correct if it lies in a distance from the closest true limit that is shorter than tol. The probability that a random limit is correct is obviously $P_{cor} = \frac{2 \cdot tol \cdot N}{L}$. In addition, the precision and recall of the random segmenter are computed according to the equations:

$$Precision = \frac{\#correctly\ detectedlimits}{\#total\ detected\ limits}$$

$$Recall = \frac{\#correctly\ detected\ limits}{\#total\ real\ limits}$$

As it has been mentioned above, the number of real segment limits is N, the number of detected segment limits is M and obviously the number of correctly detected limits is $M \cdot P_{cor} = M \cdot \frac{2 \cdot tol \cdot N}{L}$. Therefore we have:

$$Precision = \frac{2 \cdot tol \cdot N}{L} = \frac{2tol}{d}$$

$$Recall = \frac{2 \cdot tol \cdot M}{L} = \frac{2tol}{m}$$

The equations above are valid under the assumption that $2tol \geq d$ and $2tol \geq m$. In the extreme case that $2tol \leq d$ (or $2tol \leq m$) the Precision (or Recall) is 100%.

5.6 Experiments

5.6.1 Datasets

In order to train and test the proposed method, 300 homogeneous audio segments have been recorded from more than 30 movies, covering a wide range of audio classes (speech, different genres of music, gunshots, screams, fights, environmental sounds, etc.). These segments have been used for generating **phantom** audio streams with **known** segment limits, both for training and testing. As explained in Section 5.1, during the segmentation stage we make no assumptions about the content audio classes that exist in the stream. The only restriction is that the segments involved in the training process are homogenous.

More specifically three sets were formed:

- S_1 comprises 150 of the homogeneous segments and it has been used for **training** the individual classifiers and computing the weights of the combining rules. It contains 300 audio streams of 70 segments each (total duration: 25 hours).
- S_2 has been generated from the remaining 150 homogeneous segments and it is used for **testing** purposes. It has the same size as S_1.

- S_3 has been used for evaluating the methods for specific class transitions (e.g. music to speech). This dataset is described in detail in Section 5.6.2.3.

In addition, for testing purposes S_4 has been formed using 20 uninterrupted audio streams from movies (real data), which have been manually segmented. The total duration of S_4 is approximately 300 minutes.

5.6.2 Performance on Phantom Data

5.6.2.1 Performance for Different Thresholds

As a first step, the proposed method has been tested for different values of the threshold, using S_1. Figure 5.9 presents the performance for different threshold values, for the case of 1 sec of tolerance. As expected, the precision rate grows with the threshold value, while the recall rate is decreased. As a result of this fine tuning experiment, the threshold parameter T in the final system was set equal to 1.

5.6.2.2 Performance for Different Tolerances

For the above threshold value, the method was then tested on a tolerance basis, (in the range 100 msecs to 1 sec), using dataset S_2. In figure 5.10, the F1 measure is presented, for all three combination rules, along with the random segmenter. The dotted line represents the performance of the random segmenter. On average (i.e., for all tolerances in the range), the GWA method achieves 0.40% improvement compared to the averaging method, while the LWA method achieves an improvement of 1.1%.

5.6.2.3 Performance for Different Genres of Transitions

An interesting information for a general audio segmentation system is the performance for specific audio transitions (e.g., a class change from music to speech). Using dataset S_3, we have measured the performance of all three segmentation methods for specific class transitions. In particular, S_3 has been formed from the same audio segments as S_2; these segments were firstly divided into three content classes, namely: music, speech and other environmental sounds. Then, for all possible combinations (e.g. music-speech) 300 streams of 70 segments (of the combined classes) have been formed. In table 5.1 the performances of

	Average		
	Music	Speech	Envir.
Music	**84.77%**	-	-
Speech	87.26%	59.94%	-
Envir.	88.53%	86.84%	84.01%
	GWA		
	Music	Speech	Envir.
Music	84.52%	-	-
Speech	**87.76%**	59.81%	-
Envir.	88.55%	**88.12%**	83.90%
	LWA		
	Music	Speech	Envir.
Music	84.55%	-	-
Speech	87.66%	**59.99%**	-
Envir.	**89.37%**	87.55%	**84.02%**
	Random		
	Music	Speech	Envir.
Music	26.26%	-	-
Speech	29.69%	34.16%	-
Envir.	29.77%	34.26%	34.35%

Table 5.1. Performance (F1 measure) of all three methods for specific class transitions.

all three methods (and that of the random segmenter) are presented. The tolerance of the specific experiments was 0.5 seconds. It is obvious that the performances of all methods drop when the audio stream is composed of speech segments. This happens because all methods, generally, lead to over-segmentation of speech segments and therefore the precision rate is decreased. Note that over-segmentation for speech data is expected, since the proposed methods have been trained for segmentation of several classes and speech usually contains more abrupt signal changes than any other audio classes.

	Tolerance = 0.5	Tolerance = 1
Av.	75.3%	84.1%
GWA	75.0%	84.4%
LWA	76.0%	85.0%
Random	37.5%	73.5%

Table 5.2. Performance on dataset S_4.

5.6.3 Performance for Real Audio Streams

In Table 5.2, the overall performance of the method for dataset S_4 is presented. It is observed that in this case the performance of the LWA rule is the best of the three, which is in line with the figure 5.10.

5.6.4 Computational Complexity

The average execution time of the proposed algorithm has been measured to be *at most equal to* 1% of the input data length. For example, for a 2-hour audio stream, the execution time was less than 2 minutes. This makes the method almost a real time one. This experimental result refers to a Matlab implementation of the proposed method, applied on a standard Windows workstation.

5.6.5 Comparison with existing methods

Related work has so far focused on speech related tasks, thus making a direct comparison with the proposed method a hard task. A comparative study of unsupervised techniques can be found in [34], where experiments were performed on the CHIL Isolated Acoustic Event Dataset. This is a corpus mainly consisting of speech sounds recorded in a multi-speaker environment (including some other types of audio events like applause and laughter). The work in [34] suggests that, for a 1 sec tolerance around segment limits, methods [33], [34] and [35] exhibit comparable performance in terms of the F-measure ($0.76 - 0.85$) and that the best execution time is reported in [34] (7.8% of recording time). Our method achieves comparable performance in terms of the F-measure, irrespective of the audio type and, in addition, the execution time is significantly reduced, i.e., 1% of the recording time. Moreover,

our method does not need to adopt any assumptions about the number of the involved audio classes and hence perform a separate training for each one of them. This is a very welcome feature, since in a general audio stream the exact number of classes is not known.

Finally, the comparative study in [82] (a supervised method), indicates that on the TDT-3 Mandarin audio corpus (a speech oriented corpus recorded from radio broadcasts) [82] slightly outperforms [33], but execution time is around 22% of the recording time.

5.7 Conclusions

In this chapter, the problem of detecting homogeneous audio segments has been treated as a classification task using classifier combination schemes. The system has been tested both, on phantom as well as real audio streams from movies. For the case of the LWA combiner and for real audio streams, F1 measure was almost 85%, for a tolerance of 1 second. Furthermore, the method has a very low computational complexity, since the execution time does not exceed 1% of the input data length, using MATLAB code. Finally, the presented algorithm, provides a general framework for audio segmentation, which does not explicitly depend on the number of audio classes.

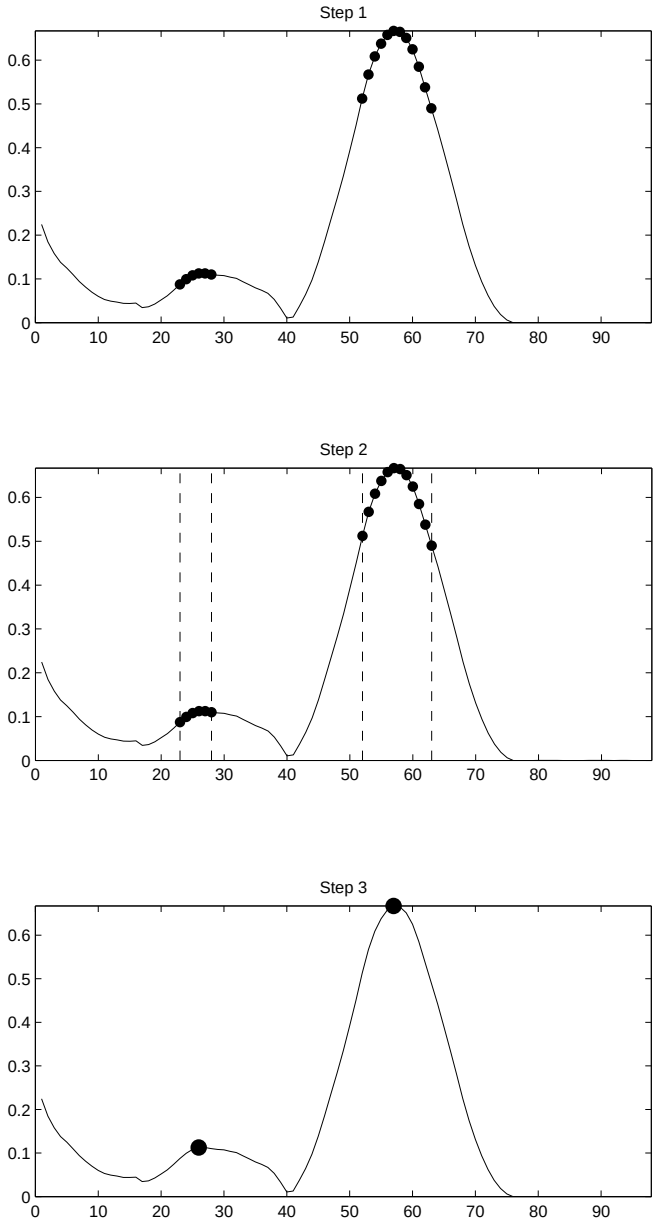
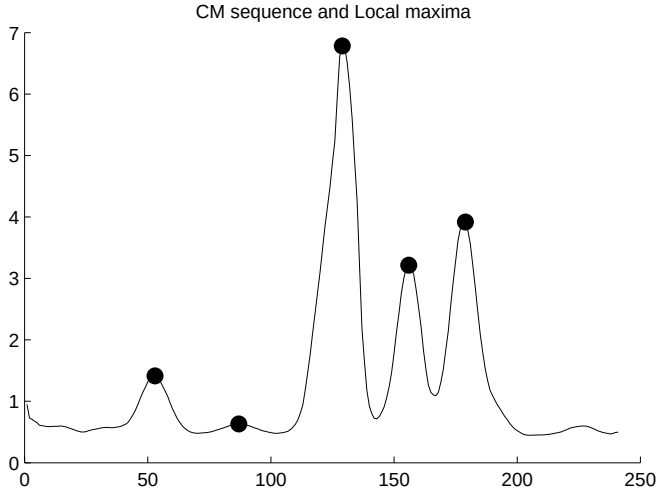
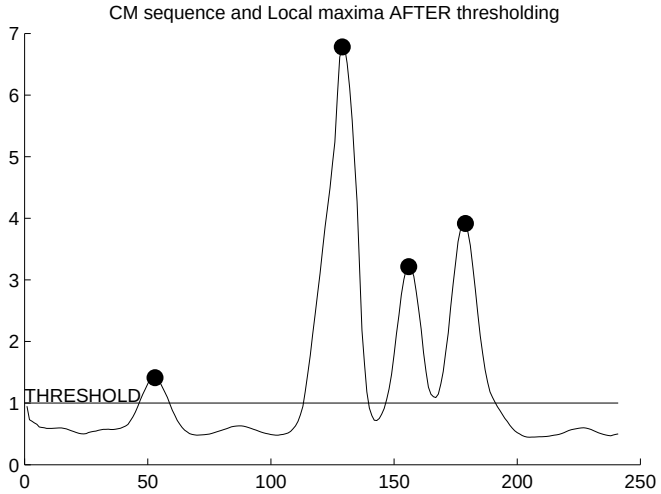


Figure 5.6. Maxima detection example: At a first stage, each "maximum candidate" i is detected, if $CM(i)$ is larger than the average value of the $maxWin$-long areas on the left and on the right of i. Secondly, the neighbor maximum candidates are grouped and finally the maximum value of each group is kept.



(a) Maxima Detection



(b) Thresholding

Figure 5.7. Change detection example: (a) is the result of maxima detection in the CM sequence and (b) is the result after the thresholding procedure.

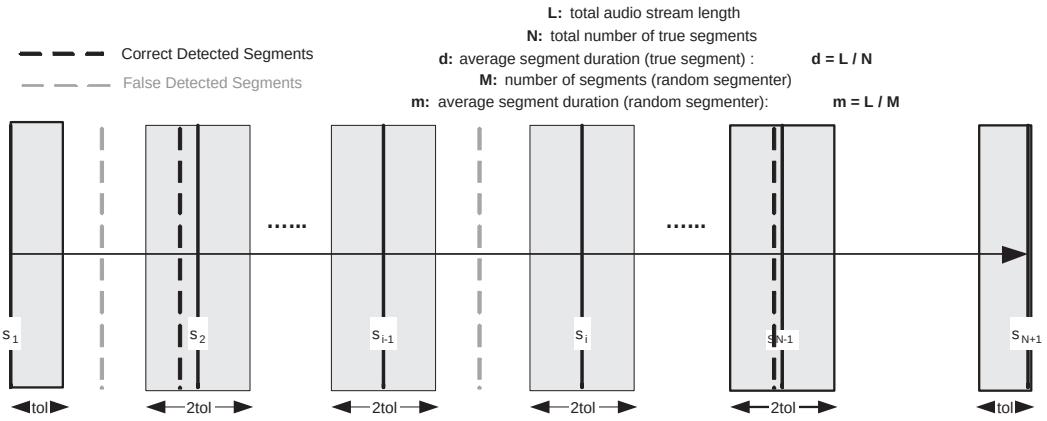


Figure 5.8. Random Segmentation

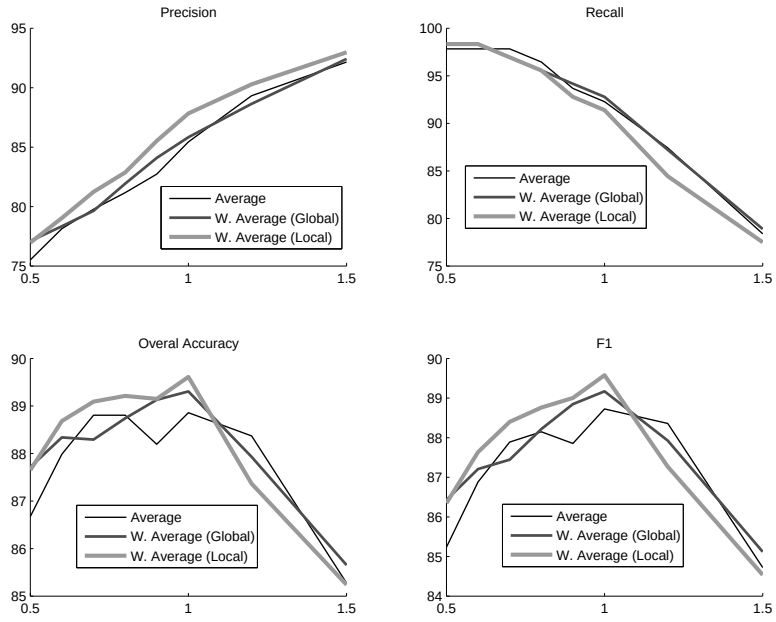


Figure 5.9. Performance vs threshold parameter T for 1 sec tolerance

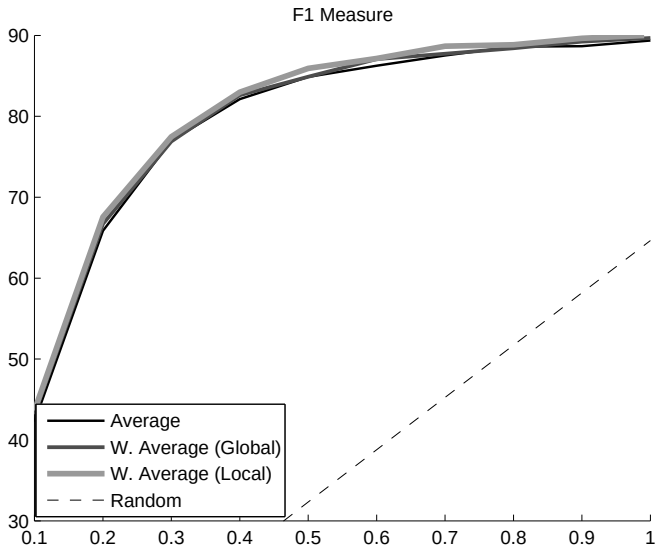


Figure 5.10. F1 measure (varying tolerance), for the phantom dataset S_2

Chapter 6

Multi-class audio classification

In this chapter, a multi-class classification algorithm for audio segments recorded from movies is presented, focusing on the detection of violent content, for protecting sensitive social groups (e.g. children). Towards this end, twelve audio features have been used, stemming from the nature of the signals under study. In order to classify the audio segments into seven classes (three of them violent), Bayesian Networks have been used in combination with the One Versus All classification architecture. The overall system has been trained and tested on a large data set (5000 audio segments), recorded from more than 30 movies of several genres. The experimental results verified that the proposed method can be used as an accurate multi-class classification scheme, as well as, as a binary classifier for the problem of violent - non violent audio content classification.

6.1 Introduction

The task of detecting violence is difficult, since the definition of violence itself is ambiguous. One of the most widely accepted definition of violence is: "behavior by persons against persons that intentionally threatens, attempts, or actually inflicts physical harm" ([84]). In video data, most violent scenes are characterized by specific audio signals (e.g. screams and gunshots). The literature related to violence detection is limited and, in most of the cases, it examines only visual features ([49], [50]).

In [44] the audio signal is used as additional information to visual data. In particular, a single audio feature, namely the energy entropy, is used in order to detect abrupt changes

in the audio signal, which, in general, may characterize violent sounds. However, the usage of energy entropy as a feature for violent detection can only be used in combination with other audio or visual features, since it only detects abrupt changes and it could therefore lead to the classification of a non violent impulsive noise (e.g. a door closing) as violent. In [85], a film classification method is proposed that is mainly based in visual cues, since the only audio feature adopted is the signal's energy. A more detailed examination of the audio features for discriminating between violent and non-violent sounds was presented in [57]. In particular, seven audio features, both from the time and frequency domain, have been used, while the binary classification task (violent and non violent) was accomplished via the usage of Support Vector Machines.

6.1.1 Class Definitions

This thesis focuses on more audio features in order to detect violence in audio signals but also to give a more detailed characterization of the content of those signals. Therefore, facing the problem as a binary classification task (violent/non-violent) would not be adequate. In addition, such a treatment of the problem would be insufficient in terms of classification accuracy. For example the sound of a non-violent impulsive sound (e.g. a thunder or a door closing) is more similar to a gunshot (violent) than to speech (non violent). It is therefore obvious, that the binary approach would lead to the grouping of distinct sounds, which is undesirable. Thus, we treat the problem as a multi-class audio classification problem.

In particular, we have defined seven classes (3 violent and 5 non-violent), motivated by the nature of the audio signals met in most movies. The non-violent classes are: *Music*, *Speech*, *Others1*, and *Others2*. The later two non-violent classes are environmental sounds met in movies. These sounds have been divided into two sub-categories according to some general audio characteristics. In particular, "Others1" contains environmental sounds of low energy and almost stable signal level (e.g. silence, background noise, etc). "Others2" contains environmental sounds with abrupt signal changes, e.g. a door closing, an airplane landing, a car accelerating, etc. This definition of the environmental audio classes is not only based on the two classes' content differentiation, but also on the significant differences in the adopted feature representation. For example, in Figure 6.1 the histograms of the 2nd

adopted feature is presented (see details for the adopted audio features in Paragraph 6.2.1), for the two environmental classes. It is obvious, that for the audio samples of the "Others2" class, the particular audio feature has significantly lower values.

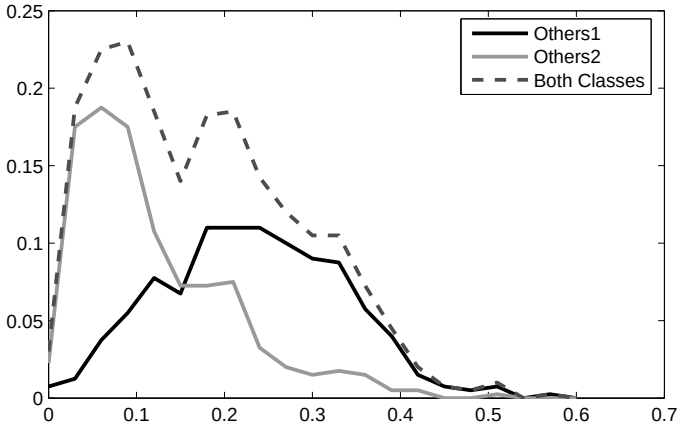


Figure 6.1. Histograms of the 2nd audio feature for the two environmental (non-violent) classes. If a unique class for environmental sounds would have been used, this would have led (for the specific feature) in a non-homogenous histogram.

As far as the *violent*-related content is concerned , the following classes have been defined: *Shots*, *Fights* (beatings) and *Screams*. A detailed description of all seven audio classes is presented in Table 6.1.

6.2 Proposed method

For each audio segment, a number of audio features and respective statistics is calculated, leading to a 12-D feature vector. Next, each class is modelled by a separate Bayesian Network (BN) classifier. Each BN is used as an estimator of the probability that the input audio sample belongs to the respective class. At a final step, the maximum BN probability determines the "winner" class. In the following paragraphs a more detailed description of the adopted methods is presented.

Table 6.1. Classes Definitions and Descriptions

	Class Name	Class Description
1	Music	Music from film soundtrack and shorter music effects.
2	Speech	Speech segments from various speakers, languages and emotional states. Also, several levels of noise, since speech is usually mixed with other. types of audio classes (especially in films).
3	Others 1	Environmental sounds of low energy and almost stable signal level (e.g. silence, background noise, wind, rain, etc)
4	Others 2	Environmental sounds with abrupt changes in signal energy (e.g. a door closing, a sound of a thunder, an object breaking, etc).
5	Gunshots	Sounds from several types of guns. Contains both short abrupt and continuous gunshots.
6	Fights	Sounds from human fights - beatings.
7	Screams	Sounds of human screams.

6.2.1 Audio Features

At a first step, 12 audio features are extracted for each segment on a short-term basis, i.e., each segment is broken into a sequence of non-overlapping short-term windows (frames), and for each frame a feature value is calculated. This process leads to 12 feature sequences. In the sequel, a statistic is calculated for each sequence, leading to a 12-D feature vector for each audio segment. The features, the statistics and the window lengths adopted are presented in Table 6.2. For more detailed descriptions of those features, refer to Chapter 2.

Table 6.2. Window sizes and statistics for each of the adopted features

	Feature	**Statistic**	**Window (msecs)**
1	Spectrogram	σ^2	20
2	Chroma 1	μ	100
3	Chroma 2	*median*	20 (mid term:200)
4	Energy Entropy	*min*	20
5	MFCC 2	σ^2	20
6	MFCC 1	*max*	20
7	ZCR	μ	20
8	Sp. RollOff	*median*	20
9	Zero Pitch Ratio	–	20
10	MFCC 1	max/μ	20
11	Spectrogram	*max*	20
12	MFCC 3	*median*	20

6.2.2 Classification Method

6.2.2.1 Multiclass Classification Scheme

In order to achieve multi-class classification, the "One-vs-All" (OVA) classification scheme has been adopted. This is one of the simplest but most accurate approaches for the multi-class classification task ([86]). It is based on decomposing the K-class classification problem into K binary sub-problems. In particular, K binary classifiers are used, each one trained to distinguish the samples of a single class from the samples in the remaining classes, i.e.

each class is opposed to all the others. For example, for the present audio classification task, one of the single binary classifiers is trained to distinguish a speech signal for non-speech signals. In the current work, we have chosen to use Bayesian Networks (BNs) for building those binary classifiers. As described below, the BNs are used to determine the probability that a sample belongs to one of the classes.

6.2.2.2 Binary Classifiers

In this paragraph, a description of the Binary Classifiers, that compose the OVA architecture, is presented. At a first step, the 12 feature values $v_i, i = 1 \ldots 12$ described in Paragraph 6.2.1, are grouped into three 4D separate feature vectors:

$$V^{(1)} = [v_1, v_4, v_7, v_{10}] \tag{6.1}$$

$$V^{(2)} = [v_2, v_5, v_8, v_{11}] \tag{6.2}$$

$$V^{(3)} = [v_3, v_6, v_9, v_{12}] \tag{6.3}$$

In the sequel, for each one of the 7 binary sub-problems, three k-Nearest Neighbor classifiers are trained on the respective feature space. In particular, each kNN classifier KNN_i^j, $i = 1 \ldots 7$ and $j = 1 \ldots 3$ is trained to distinguish between class i and all i' (not i), given the feature vector $V^{(j)}$. This process leads to three binary decisions for each binary classification problem. Thus, a 7x3 matrix R is defined as follow:

$$R_{i,j} = \begin{cases} 1, & \text{if the sample was classified in class } i, \text{ given the feature vector } V^{(j)} \\ 0, & \text{if the sample was classified in class } \textit{not } i, \text{ given the feature vector } V^{(j)} \end{cases} \tag{6.4}$$

Let us consider, the following result matrix:

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{6.5}$$

For example, the fact that $R_{1,1} = 1$, indicates that KNN_1^1 (i.e. the KNN classifier of the first binary sub-problem - music vs non-music - that functions on the feature space of the $V^{(1)}$ feature vectors) decided that the input sample is music. On the other hand, the other two kNN classifiers of the same binary sub-problem decided that the input sample is non-music. Also, for the sixth binary sub-problem (i.e. fights vs non-fights) all three kNN classifiers decided in favor of the fights class. The emerging subject here is to decide to *which class the input sample will be classified, according to R*. An obvious approach would be to apply a majority voting rule for each binary sub-problem. Though, in the current work BNs have been adopted: each binary subproblem has been modelled via a BN which combines the individual kNN decisions to produce the final decision, as described in the sequel.

In order to classify the input sample to a specific class, the kNN binary decisions of each subproblem (i.e. the rows of matrix R) are fed as input to a separate BN, which produces a probabilistic measure for each class. In this work, the BN architecture shown in figure 6.2, has been used as a scheme for combining the decisions of the kNN individual classifiers. This is similar to the combiner used in Paragraph 3.4.3.2 (BNC). Discrete nodes $R_{i,1}$, $R_{i,2}$ and $R_{i,3}$ correspond to the binary decisions of the kNN individual classifiers for the i-th binary sub-problem and are called hypotheses of the BN, while node Y_i is the *output* node and corresponds to the true binary label. Y_i, like the elements of R, is 1 if the input sample really belongs to class i, and it is 0, otherwise.

In the BN training step, the CPTs of each BN i are learned according to the set ([75]):

$$S^{(i)} = \{(R_{i,1}^{(1)}, R_{i,2}^{(1)}, R_{i,3}^{(1)}, s_{i,1}), \ldots, (R_{i,1}^{(m)}, R_{i,2}^{(m)}, R_{i,3}^{(m)}, s_{i,m})\} \tag{6.6}$$

where m is the total number of training samples, $R_{i,j}^{(k)}$ is the result of j-th kNN classifier ($j = 1, \ldots, 3$) for the j-th feature vector of the k-th input sample ($k = 1, \ldots, m$), and $s_{i,k}$ is

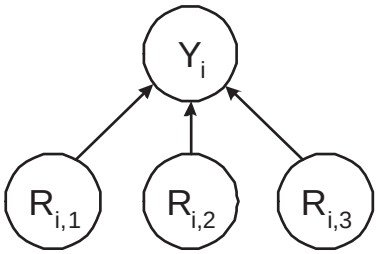


Figure 6.2. BNC architecture

the *true binary label* for the k-th input sample and for the i-th binary subproblem. In other words, $s_{i,k}$ is defined as follow:

$$s_{i,k} = \begin{cases} 1, & \text{if the } k\text{-th sample's true class label is } i \\ 0, & \text{if the } k\text{-th sample's true class label is } i' \text{ (not } i) \end{cases} \tag{6.7}$$

Each set $S_{(i)}$ is generated by validating each individual kNN classifier (with results $R_{i,j}$) with a test set of length m, in our case a set of m audio segments with known true class label.

Each BN i, makes the final decision for the i-th binary subproblem, based on the conditional probability

$$P_i(k) = P(Y_i(k) = 1 | R_{i,1}^{(k)}, R_{i,2}^{(k)}, R_{i,3}^{(k)}) \tag{6.8}$$

This is the probability that the input sample's true class label is i, given the results of the individual kNN classifiers. Like with the BN scheme in Section Paragraph 3.4.3.2, no actual inference algorithm is needed, since the required conditional probability is given directly by the CPT, which has been calculated in the training phase. Also, *no assumption of conditional independence between the input nodes is made*, like e.g. in the Naive Bayesian Networks. After the probabilities $P_i(k)$, $i = 1, \ldots, 7$ are calculated for all binary subproblems, the input sample k is classified to the class with the largest probability, i.e.

$$WinnerClass(k) = \arg\max_i P_i(k)$$

Note that the above combination scheme is used as a classifier, though it can also be used as a probability estimator. Therefore, the proposed method can be included in a joint segmentation / multi-class classification system, like the one proposed in Chapter 3 for the binary problem of speech-music discrimination.

6.3 Experimental results

6.3.1 Datasets and System Training

In order to train and test the proposed system, seven datasets D_i, $i = 1 \ldots 7$ consisting of 200 minutes of movie recordings have been compiled. Almost 5000 of audio samples have been extracted and manually labelled as "music", "speech", "others", "shots", "fights" and "screams", i.e. almost 800 samples per class. The duration of those audio segments varies from 0.5 to 10 seconds. The data was collected from more than 30 films, covering a wide range of genres (e.g. drama, adventure, horror, and war). Some of the films were chosen not to contain violent content, and were therefore used only for populating the non-violent classes. As described in Paragraph 6.2.2, the adopted multiclass classification technique is the One Vs All scheme. It is therefore obvious that, in order to train the binary sub-classifiers used in the OVA scheme, one must create (from the original datasets) other seven datasets D'_i, each one containing audio samples from all other classes, than i. For example the dataset D'_4 contains audio segments that are *not* labelled as "shots".

After the datasets D_i and D'_i, D_i, $i = 1 \ldots 7$ have been created, 20% of the audio samples are used for populating the individual kNN classifiers. At a second step, the BNs are trained, via the validation of the respective kNN classifiers, as described in Paragraph 6.2.2. Towards this end, 60% of the datasets are used. The remaining 20% of the audio data is used for testing the final system.

6.3.2 Overall System Testing

In order to test the overall classification system, hold-out validation has been used. Therefore, each of the datasets D_i and D'_i were randomly separated as explained above and experiments were executed for different selection of the subsets. In total, 100 iterations were executed. The normalized average confusion matrix (C) is presented in Table 6.3. For example $C_{2,2}$ is the percentage of the speech data that was indeed classified as speech, whereas $C_{7,1}$ is the percentage of "Screams" segments that were classified as "Music".

The diagonal of C is also the recall R_i of the classification results, i.e. the proportion of data with true class label i, that were correctly classified in that class. On the other hand,

Table 6.3. Average Confusion Matrix

	Classified						
True ↓	Mu	Sp	Ot1	Ot2	Sh	Fi	Sc
music	68.22	2.36	13.60	1.76	3.27	3.83	6.95
speech	1.66	81.96	6.38	4.75	0.23	2.08	2.95
others1	4.59	1.90	70.24	11.20	5.44	2.52	4.11
others2	2.00	3.15	15.21	59.83	10.30	8.57	0.94
shots	1.26	0.19	3.00	6.66	79.10	9.68	0.11
fights	1.70	2.23	0.89	11.81	26.38	52.29	4.71
screams	9.18	3.44	4.00	1.29	2.20	7.86	72.04

the precision of each class $Pr_i, i = 1 \ldots 7$ (i.e the proportion of data classified in class i, whose true class label is indeed i) is defined as:

$$Pr_i = \frac{C_{i,i}}{\sum_{j=1}^{7} C_{ji}}$$

.

The recall and precision values of each class are presented in Table 6.4. The overall classification accuracy (i.e. the percentage of the data that were correctly classified) of the proposed method is 69.1%.

Table 6.4. Recall and Precision per Class

	Mu	Sp	Ot1	Ot2	Sh	Fi	Sc
RECALL:	68.2	82.0	70.2	59.8	79.1	52.3	72.0
PRECISION:	77.0	86.1	62.0	61.5	62.3	60.2	78.5

The percentage of 69.1% refers to the classification accuracy of the multi-class classification problem. Though this is a high performance rate according to the nature of the problem, one may prefer to use the proposed classification scheme as a binary classifier. For example, the confusion between "Shots" and "Fights" is quite large ($CM_{5,6} = 9.68$ and $CM_{6,5} = 26.38$). This means that a large amount of data that should have been classified as "Shots" was classified as "Fights" (and vise versa), but in both cases the content can be also characterized as violent. In general, binary classification could be achieved by classifying

each sample with class label 1, 2, 3 or 4 as "Non-Violent" and the samples with class labels 5, 6 or 7 as "Violent". It is obvious that the recall and precision values for the violent class would therefore be computed using the following equations:

$$Re_{violence} = \frac{\sum_{i=5}^{7}\sum_{j=5}^{7} C_{ij}}{\sum_{i=5}^{7}\sum_{j=1}^{7} C_{ij}} \tag{6.9}$$

$$Pr_{violence} = \frac{\sum_{i=5}^{7}\sum_{j=5}^{7} C_{ij}}{\sum_{i=1}^{7}\sum_{j=5}^{7} C_{ij}} \tag{6.10}$$

Applying equations 6.9 and 6.10 given the computed confusion matrix, the violence recall was found equal to 84.8% and the violence precision equal to 83.2%. This means that the overall binary classification accuracy was almost 84%.

For comparison reasons, the method has also been tested using the majority vote rule as a combiner of the individual binary decisions. In Table 6.5 the overall accuracy of the multi-class classification task and the recall and precision of the binary problem (violent vs non-violent content) are displayed. It can be seen that the proposed combination rule has a higher accuracy by 1.9%.

Table 6.5. Overall accuracy of the multi-class classification task, violence recall and violence precision for the two combination methods (BN combiner and majority vote combiner)

	Ov. Accuracy	V. Recall	V. Precision
BN combiner	69.1%	83.2%	84.8%
Maj. Vote combiner	67.2%	84.1%	81.1%

6.3.3 Examples of using the proposed scheme for audio stream segmentation and classification

One of the main advantages of the proposed method is that it produces a **probabilistic** measure (see equation 6.8) for each one of the audio classes, based on the decision of individual classifiers. This probabilistic measure can be used in a segmentation-classification scheme for audio streams from movies. In this section, some examples of this probabilistic measures for audio streams from movies are presented. Towards this end, the proposed multi-class

classification method has been applied on overlapping mid-term windows of the stream, in a similar way to the music tracking method presented in Chapter 4. In particular, mid-term windows of 2 seconds have been used, while the overlap was equal to 50%.

In Figure 6.3 an example of the above process is shown for an audio stream that consists of three parts: music, speech and gunshots. Furthermore, the particular stream was selected so as the transitions between the three content classes is not abrupt (i.e., the speech part "fades in" when the music part "fades out", etc). The BN outputs for the three particular classes show that they can be used for segmentation-classification of this audio stream. Furthermore, it is obvious that the probabilistic measures follow the transition between the different classes.

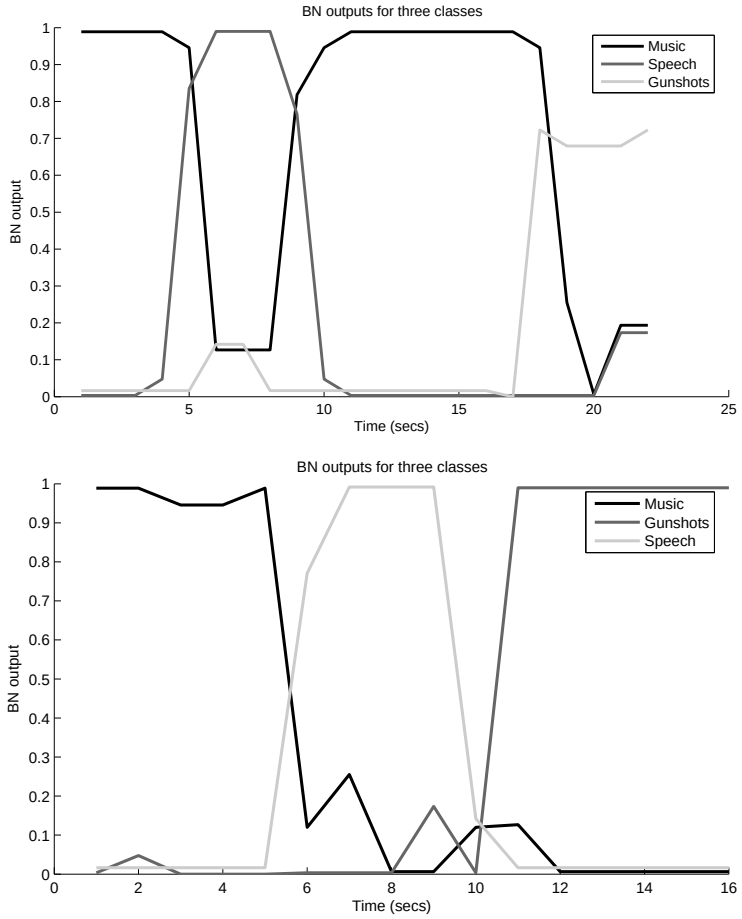


Figure 6.3. Examples of applying the multi-classification algorithm on a mid-term basis for an audio stream that contains music, speech and gunshots. Each line corresponds to the BN output of the respective binary classification subproblem.

6.4 Conclusions and future work

In this chapter, we proposed a multi-class audio classification system for movies, with respect to violent content. In total, seven audio classes were adopted (three of them violent). Exhaustive examination resulted in a number of audio features stemming from the nature of the signals in the specific classification problem. The classification scheme was generally based on the One Versus All architecture. Each class was modelled using a Bayesian Network which was used as an estimator of the respective class probability, given the input sample. To extract the above probability, each BN was used as a combination scheme for classifying a set of three audio feature vectors into the classes of each binary sub-problem of the OVA architecture.

The proposed scheme was tested using more than 3 hours of audio recordings from more than 30 movies, covering a wide range of genres. The overall performance of the multi-class classification system was found to be equal to 69.1%. This is a high classification performance, taking into account the number of classes and the fact that some classes are quite similar (i.e. the classes "Shots" and "Fights"). Finally, the proposed system could also be used as a binary classifier for the "Violent" - "Non Violent" problem. In this case of binary classification, almost 15% of the violent data was incorrectly classified (false negative rate), while less than 17% of the non-violent data were classified as violent (false alarm rate). The overall binary classification error is therefore almost 16%.

To sum up, the proposed method can be used both as a multi-class audio classification system, but also as a binary classifier, resulting (as expected) in different performance rates. For example, one could use the system for blocking violent content in movies with a high performance rate (binary problem), while more detailed semantic information could be obtained from the seven-class classification results, with an error rate of almost 30%.

In the future, new features could be examined and used, in order to achieve boosted performance of the classification task. On the other hand, more classes could be added in the classification problem, in order to have a more detailed description of the audio data. Also, in the proposed combination scheme, more input nodes could be added and separate types of individual classifiers could be used (i.e., support vector machines). This could boost the overall classification performance. Furthermore, an audio segmentation algorithm could

be implemented and combined with the audio classification scheme. Such a segmentation scheme could make direct use of the BN output probabilities for segmenting an audio stream to homogenous segments.

Finally, the audio classification system could be combined with synchronized *visual* cues for increased classification performance. Towards this end, the BNs could be expanded by adding visual-based individual decisions, and they could provide a type of "dynamic weighting" between the two media (audio and visual). The combination of decisions based on different media would be achieved through the usage or training data or through empirical knowledge. For example, a BN node that combines the audio-based and the visual-based individual decisions for the "gunshots vs non-gunshots" binary subproblem could use a CPT that "trusts" more the audio-based decision. This means, that the corresponding CPT could be trained using the empirical knowledge that gunshots are usually detected through the audio information, while the visual detection could only function as complementary.

Chapter 7

Speech Emotion Recognition

Besides extracting information regarding events, structures (e.g., scenes, shots) or genres, a substantial research effort of several multimedia characterization methods has focused on recognizing the **affective** content of multimedia material, i.e., the **emotions** that underlie the audio-visual information ([28], [24], [29]). Automatic recognition of emotions in multimedia content can be very important for various multimedia applications. For example, recognizing affective content of music signals ([25], [26]) can be used in a system, where the users will be able to retrieve musical data with regard to affective content. In a similar way, affective content recognition in video data could be used for retrieving videos that contain specific emotions. In this chapter, emphasis is given on affective content that can be retrieved from the speech information of movies. This approach can also help in detecting oral violence in movies, based on the emotional recognition results. This is very important, since oral violence is quite often present in films and it may sometimes be more harmful for children than physical violence. Note that emotion recognition in movies is a difficult task, since both audio and visual channels are more complicated in movies than in similar studio-acted databases. In [87], a method has been proposed, for detecting fear-type emotions in movies, while the binary classification performance (fear vs normal) reached 70%. In this thesis, attention has been paid to all types of emotions, since violent situations may be related to other emotion categories, apart from fear.

In this chapter, a complete framework for emotion recognition of movies is presented, exploiting information that resides in the speech data. First, a fast and accurate speech tracking technique is proposed based on bayesian combination of individual thresholding

decisions, using four speech-specific audio features. The detected speech segments are then fed to the emotion recognition stage. The latter is based on a two-dimensional representation of the emotions in speech (Emotion Wheel). The goal is twofold. First, to investigate whether the Emotion Wheel offers a good representation for emotions associated with speech signals. Second, three regression approaches have been adopted, in order to predict the location of an unknown speech segment in the Emotion Wheel. Each speech segment is represented by a vector of ten audio features. The results indicate that the Emotion Wheel is a good representation of emotions of speech segments and that the resulting architecture can estimate emotion states of speech segments from movies, with sufficient accuracy. Finally, a possible scheme to extract affective content from uninterrupted audio streams from movies is investigated.

7.1 Previous Works

The most common approach to affective audio content recognition, so far, is to apply well-known classifiers (Hidden Markov Models, etc.) for classifying signals into an *a-priori known number of distinct* categories of emotions, e.g., fear, happiness, anger ([28], [23]). One drawback of such techniques is that, in many cases, the emotions of multimedia content cannot easily be classified in specific distinct categories. For example, a speech segment from a horror movie may contain both fear and disgust feelings. In addition, the level of categorical taxonomy of emotion is subjective, i.e., the number of classes is an ambiguous subject. For example, the state of happiness can be further divided into pleasure and excitement.

An alternative way to emotion analysis is the dimensional approach, according to which, emotions can be represented using specific dimensions that stem from psychophysiology ([27], [26], [88], [89]). In [27], Valence-Arousal representation is used for affective video characterization. Towards this end, visual cues, such as motion activity, and simple audio features, e.g., signal energy are used for modelling the emotion dimensions.

7.2 Proposed Method - General

In this chapter, the problem of speech emotion recognition is treated as a regression task: 10 audio features are mapped to the Valence and Arousal dimensions using several regression methods. The contribution of this work is focused on the following:

1. A computationally efficient algorithm for tracking speech in audio streams from movies is presented. The proposed algorithm achieves a precision rate of 95%.

2. In order to find the emotional state of the detected speech segments from movies, we propose a 2-D representation (Arousal-Valence). To investigate whether the Arousal-Valence representation (Emotion Wheel) is appropriate for speech signals, several humans have manually annotated speech segments using this representation. If the Emotion Wheel is a good representation, then the differences in annotation by separate humans should be relatively small and the respective perceptions should be, on average, in good agreement.

3. An extensive experimentation has led to the final selection of certain audio features and then the regression problem of mapping the feature space to the emotional plane is defined.

4. Three regression schemes are evaluated using the annotated data, and the performance errors are compared to the error of the human annotation.

5. An overall scheme for emotion recognition of large audio streams is proposed, that combines: a) the novel speech tracking algorithm, b) a segmentation algorithm that detects homogenous speech segments and c) the proposed method for emotion recognition of these speech segments.

The chapter is organized as follows: in Section 7.3, the speech tracking algorithm is described. In Section 7.4, we present the proposed dimensional representation of emotions of speech segments, along with the regression methods that map 10 audio features to the emotional space. The overall scheme that extracts speech emotional states from audio streams is presented in Section 7.5. The experimental results of the proposed algorithms is described in Section 7.6 and finally the conclusions are drawn in Section 7.7.

7.3 Speech Tracking

The first step of the overall movie emotion classification framework is the tracking of speech. Obviously, this stage has to achieve high levels of precision, since non-speech audio data that is detected as speech can lead to ambiguous and misleading decisions of the speech emotion recognition stage. The speech tracking algorithm, which is described in the sequel, is based on four audio features oriented to the speech versus non-speech binary classification task and it uses a probabilistic approach of combining four individual decisions.

7.3.1 Speech Features

The features used in the speech tracking stage stem from the nature of the speech signals. For the feature extraction step, the audio samples are divided into non-overlapping short-term windows of 20 msecs. For each frame, an audio feature is computed, leading to a feature sequence. Then, for each audio feature sequence a specific statistic is extracted (e.g., the standard deviation of the sequence). The following features and respective statistics have been used:

1. The maximum value of the 2nd MFCC.
2. The standard deviation by mean ratio ($\frac{\sigma^2}{\mu}$), of the zero crossing rate.
3. The standard deviation by mean ratio ($\frac{\sigma^2}{\mu}$), of the spectral centroid.
4. A feature based on the variation of the coefficients of the chroma vector. This is the 2nd chroma related feature described in [65] by the authors, where it has been used for tracking music in audio streams. This feature is a measure of the degree of variation of each chroma element over successive short-term frames. In speech segments, the degree of variation of each chroma element is high over successive frames (and low for music).

Although other features may be used, we have found that the previous features are sufficiently informative and can be used individually, leading to simple and computationally efficient rules. This is of paramount importance when dealing with the large amounts of data associated with movies.

7.3.2 Individual Thresholding Decisions

The base of the speech tracking algorithm consists of four individual binary classifiers. In particular, one threshold is computed for each audio feature for the speech vs non-speech classification task. Towards this end, 1500 speech segments and 1500 non-speech segments, recorded from several movies, have been used. The non-speech class was carefully selected and is composed by several types of sounds recorded from movies: music, environmental sounds, gunshots, cars, etc.

Each threshold is computed using a criterion that is based on the speech precision rate and not on the overall performance. For the specific binary classification task, speech **precision** is the proportion of data classified as speech, that is indeed speech, while speech **recall** is the proportion of speech data that was finally classified as speech. As mentioned before, precision is adopted as it is more critical for the specific classification problem. Therefore, the threshold values are estimated so that to maximize the precision rate (on the training data) and subject to the limitation that the recall rate must be at least 40%. This process is described in Figure 7.1 for the threshold of the 2nd feature: in the first diagram, the histograms of the two classes (speech and non-speech) are presented, while in the 2nd part of the figure we plot the speech precision and recall rates for different threshold values. The vertical solid line represents the selected threshold value, which maximizes the speech precision rate and satisfies the limitation that the recall rate is maintained over 40%. The particular threshold value ($T = 0.588$) leads to a recall rate of 44.5% and a precision rate of 90.5%.

7.3.3 Combining Thresholding Decisions

The thresholding process described in Section 7.3.2 leads to four binary decisions for the speech vs non-speech classification problem. In particular, let:

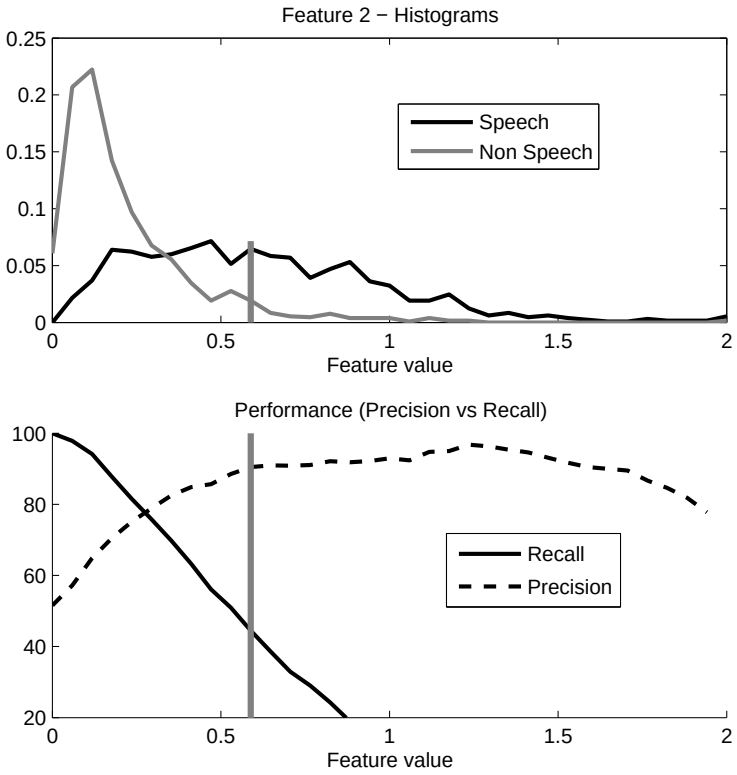


Figure 7.1. Threshold estimation for the 2nd feature: The selected threshold leads to maximum speech precision for a low bound of speech recall (at least 40%).

$$r_{i,j} = \begin{cases} 1, & \text{if the } i\text{-th sample is classified as} \\ & \text{speech, given the } j\text{-th threshold,} \\ & \text{j} = 1, \ldots, 4 \\ 0, & \text{if the } i\text{-th sample was classified as} \\ & \text{non-speech given the } j\text{-th threshold,} \\ & \text{j} = 1, \ldots, 4 \end{cases} \tag{7.1}$$

Furthermore, let ω_1 be the speech class and ω_2 be the non-speech class. The final (combined) decision is taken based on the probability that the true class label (let c) is speech, given the individual decisions $r_{i,j}$. This probability is computed according to the Bayes rule:

$$P(c = \omega_1 | r) = \frac{P(r | c = \omega_1) \cdot P(c = \omega_1)}{P(r)} \tag{7.2}$$

In order to compute the probabilities of the right hand side in the above equation we

evaluate all individual classifiers using the training dataset. In this way, $P(r|c = \omega_1)$ is obtained from the histogram of the individual decisions r for the speech class, $P(r)$ is estimated by counting the number of times the combination of the individual binary decisions r appears (for both classes), and $P(c = \omega_1)$ is 1/2.

7.3.4 Speech Tracking of Audio Streams

The segment-level classification schema described in Sections 7.3.2 and 7.3.3 is used for tracking speech in large audio streams. Towards this end, the following steps are in order executed:

1. The feature sequences (not the associated statistics) described in Section 7.3.1 are computed for the whole audio stream. The four feature values are computed for all non-overlapping short-term windows of 20 msecs. This leads to *four feature sequences* for the whole audio stream.

2. Then the speech tracking operation is takes place every 0.1 seconds. Towards this end, a set of equally spaced points t_k is defined on the audio stream. The distance between two successive points is 0.1: $|t_k - t_{k-1}| = 0.1$. For each point, t_k, a 2-sec mid-term segment is defined, using t_k as a center; in other words, the segment's limits are $t_k - 1$ and $t_k + 1$. For each such segment, the *statistics* of the respective feature sequences, described in Section 7.3.1, are calculated. Then, the probability that the segment is speech is computed, according to the process described in Sections 7.3.2 and 7.3.3. This step leads to *a sequence of probability values* P_k (one value for each 0.1 sec point). This process of the P sequence generation (i.e. steps 1 and 2) is displayed in Figure 7.2.

3. **P** is smoothed using a 1-second averaging window.

4. A probabilistic threshold T_P is used for the final decision: the points for which **P** is larger than T_P are classified as speech. Then successive points are merged to form the final speech segments.

5. Finally, detected speech segments that are shorter than 0.5 second are rejected.

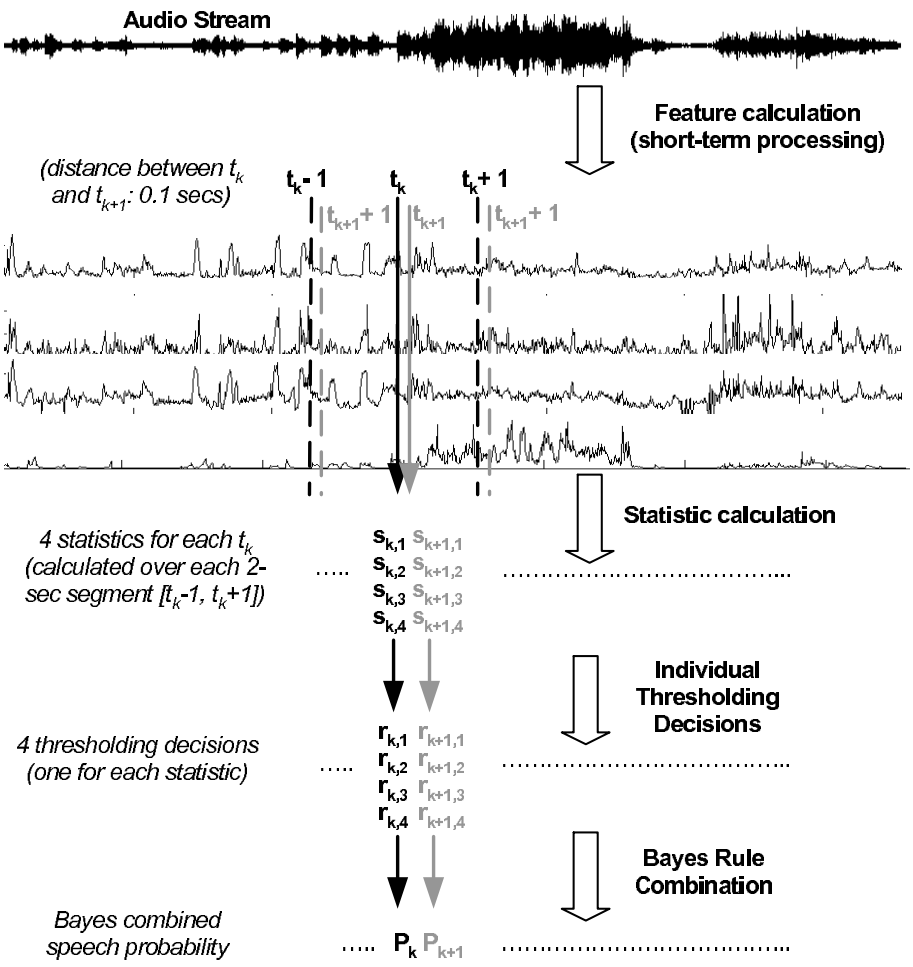


Figure 7.2. Calculation of the speech probability sequence (P) for an audio stream.

In figure 7.3 an example of the above process is presented for a short audio stream. In this specific example, the probability threshold is set equal to 0.5. In Section 7.6.1 detailed experimental results of the performance of the proposed algorithm are presented. Finally, the selected probability threshold value of the speech tracking algorithm is 0.7. As it is reported in the section discussing the experiments, the precision rate of this speech tracking algorithm reached 95% for this threshold value.

It must be emphasized that the proposed speech tracking algorithm detects speech regions in the audio streams, which are not necessarily homogenous, e.g., may contain different speakers in succession, the same speaker at different emotion states, etc. Furthermore, the emotion recognition algorithm, described in the sequel, functions on mid-term speech segments of homogenous content. It is obvious that an algorithm for segmenting the detected speech regions into homogenous speech segments needs to be applied directly after the speech

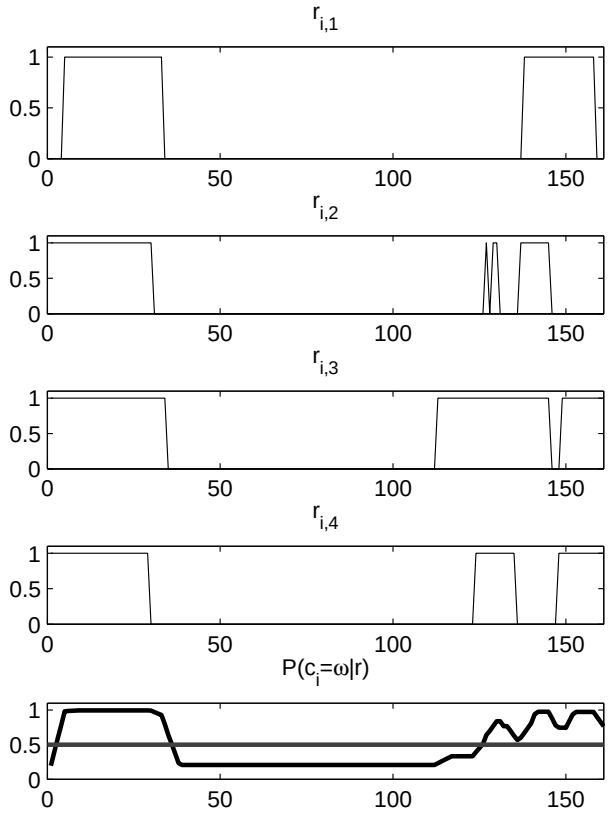


Figure 7.3. A speech tracking example: the first four sub-plots present the individual binary decisions for the respective features. The last sub-plot presents the computed speech probability. The horizontal solid line represents the probabilistic threshold T_P

tracking algorithm. Details on how these parts are finally embedded are given in Section 7.5.

7.4 Emotion Recognition of Speech Segments

7.4.1 2-D Emotional Representation

Dimensional emotion representation ([27], [29]) is based on some psychological understandings. In particular, the emotion plane is viewed as a continuous 2-D space, where each point corresponds to a separate emotion state. The two dimensions of this plane are valence (V) and arousal (A). Valence varies from -1 (unpleasant) to 1 (pleasant) and therefore it can be

characterized as the level of pleasure. Arousal, on the other hand, represents the intensity of the affective state and it ranges from -1 (passive, calm) to 1 (active). Each emotional state can be understood as a linear combination of these two dimensions. Anger, for example, can be conceptualized as an unpleasant emotional state (negative V-values) with high intensity (positive A-values). In Figure 7.4 a scheme of the 2-D emotional representation is presented (usually called "Emotion Wheel" - EW), along with some basic emotional states and their (approximate) positions in the plane.

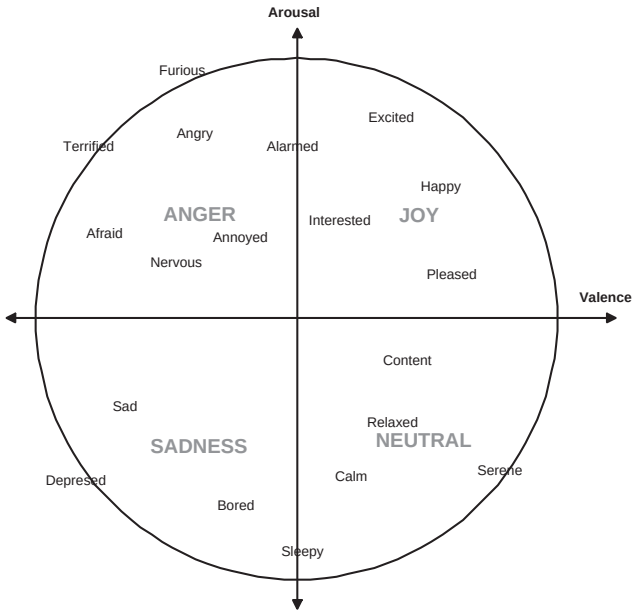


Figure 7.4. 2-Dimensional Affective Representation

7.4.2 Emotional Data Collection

In order to evaluate the representation discussed in the previous Section (7.4.1) and, also, to train and test the proposed emotional recognition method, we have manually selected 2000 speech audio samples (i.e., midterm segments) from more than 30 films. The films were selected to cover a wide range of genres (e.g. horror, comedy, etc). The average duration of the segments is 2.5 seconds. The manual annotation of speech emotion was accomplished by 50 humans. In particular, each human randomly listened to a number of speech segments. For each speech segment he/she selected a point in the emotion plane, according to the estimated emotion. It has to be emphasized that, each time, the users were prompted with

a random audio sample and the same sample could appear in a later annotation. In this way, we evaluated the level of disagreement among annotations, of *the same segment*, by *the same user*. The manually annotated data was therefore used for three purposes, namely:

1. Train (and test) the proposed automatic emotion recognition methods. Towards this end, for each audio sample i, if the number of annotations N_i was larger than 4 (i.e., at least 5 humans have annotated this sample), the average annotated coordinates were used as the final coordinates (ground truth). In other words, for each sample i, with user-annotated coordinates: $xs_{ij}, i = 1, \ldots, N_i$ and $ys_{ij}, i = 1, \ldots, N_i$, the ground-truth emotion coordinates were $x_i = \frac{\sum_{j=1}^{N_i} xs_{ij}}{N_i}$ and $y_i = \frac{\sum_{j=1}^{N_i} ys_{ij}}{N_i}$.

2. Evaluate the level of disagreement among the different users. Suppose that A_j (length L_j) is an array that contains the indices of the audio segments that have been annotated at least once by user j, and also have been annotated by at least 5 users in total. We have decided to use the average normalized Euclidian distance of the decisions of this user from the respective average decisions, as a measure of disagreement:

$$D_j = \frac{1}{L_j} \cdot \sum_{i \in A_j} \frac{\sqrt{(xs_{ij} - x_i)^2 + (ys_{ij} - y_i)^2}}{\sqrt{x_i^2 + y_i^2}} \tag{7.3}$$

3. Evaluate the level of disagreement for the annotation decisions of the same user. Towards this end, we detect the audio segments, which have been annotated at least twice by user j. For each one of those audio segments, we calculate the average user decision (i.e., average emotion coordinates) and then the average normalized distance of all decisions from that average value. Finally, DS_j is computed by averaging normalized distances for all audio segments. Therefore, DS_j is a measure of (normalized) deviation of the $j-$th user's own annotation decisions. We will refer to this as "self-error".

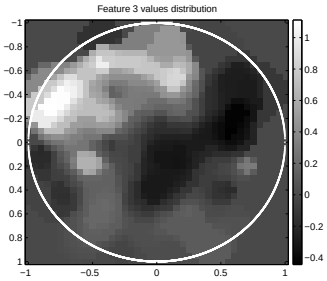
7.4.3 Audio Features

For each audio segment, 10 features and respective statistics are extracted. In particular, a short-term processing is applied: each audio segment is broken into non-overlapping short-term windows (frames) and for each frame a feature value is calculated. Then, for the extracted feature sequence, a statistic is computed (e.g., standard deviation). This statistic

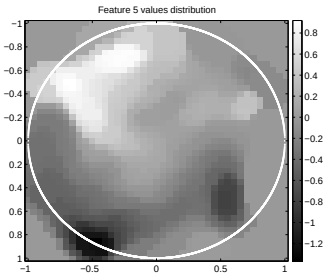
is the final feature value that characterizes the whole segment. The following features / statistics have been used ([52], [65]):

1. The average value of the 3rd MFCC.
2. The maximum value of the 2nd MFCC.
3. For each 20 mseconds frame, the FFT is computed and the position of the maximum FFT value is kept. Then, the maximum value of that sequence is the final feature for the audio segment.
4. This feature is also based on the position of the maximum FFT bins, though, this time the adopted statistic is the standard deviation of the sequence.
5. The Zero Crossing Rate is first calculated on a short-term basis (20 mseconds). The adopted statistic is the standard deviation to average ratio ($\frac{\sigma^2}{\mu}$).
6. The median value of the Zero Crossing Rate sequence.
7. The $\frac{\sigma^2}{\mu}$ ratio of the Spectral Centroid sequence.
8. The $\frac{max}{\mu}$ ratio of the pitch sequence. The pitch was calculated using the autocorrelation method.
9. The $\frac{\sigma^2}{\mu}$ ratio of the pitch sequence.
10. The 2nd chroma-based feature, described in [65], which is a measure of variation of chroma elements over successive short-term frames.

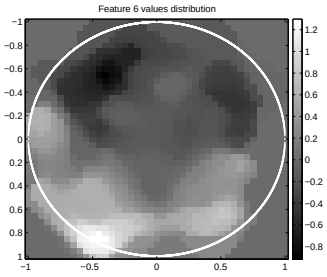
The features and statistics have been selected after extensive experimentation for the specific recognition task. Note that the adopted set of features and statistics is not the same as in the speech tracking method, since the goal of this audio analysis task is totally different. Furthermore, most of the features have a physical meaning for the specific problem. For example, the $\frac{\max}{\mu}$ ratio of the pitch sequence (8-th feature) shares high values for audio segments generally characterized as "anger", "excitement" and "alarmed", since speech under such emotional states exhibits large pitch variations. In Figure 7.5, we give four examples of feature distribution in the 2-D emotional plane. For example, it can be observed that, indeed, the 8-th feature is higher (brighter areas) for the high-arousal areas (case (d)).



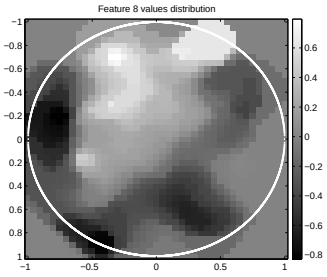
(a) 3rd feature



(b) 5-th feature



(c) 6-th feature



(d) 8-th feature

Figure 7.5. Examples of features distribution in the 2-D emotion space. Brighter values represent higher feature values.

7.4.4 Regression

As explained in Section 7.4.3, for each audio sample a 10−D feature vector is computed. Furthermore, each speech segment, i, is represented using two continuous values (x_i, y_i), which express the respective position in the EW. Therefore, we need to train two regression models that map the 10 features (for each speech segment) to the corresponding emotion dimensions, i.e. two functions $f_1, f_2 : \mathbb{R}^{10} \rightarrow \mathbb{R}$. For a set of speech segments with known emotional coordinates, the training data is described by sets: $X = \{x_i\}$ and $Y = \{y_i\}$ and the respective feature vectors $\mathbf{F} = \{\overline{F_i}\}$, $i = 1 \ldots K$, where K is the total number of training samples (i.e., the number of samples that have been annotated by at least 5 humans). Given those training sets and an audio segment described by a 10−D feature vector F_{test}, we need to estimate the emotion wheel coordinates of that audio segment: x' and y'. We have used the following regression methods:

7.4.4.1 k-Nearest Neighbor

We have chosen the kNN rule in its regression mode ([90]), since it is a simple and efficient way to estimate the values of an unknown function, given a number of training points. Towards this end, we form the subsets $N_1 \subset X$ and $N_2 \subset Y$, composed by those elements whose respective feature vectors (of $\mathbf{F}$) are the k-nearest to F_{test}. The kNN estimation is then applied for both dimensions, according to the following equations:

$$x' = \hat{f}_1(\overline{F}_{test}) = \frac{1}{k} \sum_{x \in N_1} x \tag{7.4}$$

$$y' = \hat{f}_2(\overline{F}_{test}) = \frac{1}{k} \sum_{y \in N_2} y \tag{7.5}$$

7.4.4.2 Support Vector Machine Regression

In the recent years, Support Vector machines (SVMs) have been widely used for classification tasks, and also have been extended to regression and probability density function estimation problems ([52], [91], [92], [93]). In this work, two SVM regression models have been adopted, one for each emotion coordinate. We have selected to use linear epsilon insensitive cost, while the gaussian kernel's bandwidth was set equal to 10 and the bound on the lagrangian multipliers equal to 3 ([94], [52]). These parameters were set after extensive experimentation.

7.4.4.3 Continuous Bayesian Network Classifier

Apart from classification applications, Bayesian Networks (BNs) have been used for solving regression problems ([95], [96]). In this work, the BN architecture shown in figure 7.6 has been adopted. The response nodes (X and Y) model the emotion coordinates, while the 10 feature values are modelled as explanatory variables. All nodes are continuous and a Gaussian distribution has been adopted. Given a set of feature observations $\{f_1, f_2, \ldots, f_{10}\}$, X and Y are predicted, by computing the average values of the probabilistic conditional densities $p(X|f_1, f_2, \ldots, f_{10})$ and $p(Y|f_1, f_2, \ldots, f_{10})$. It has to be noted that the specific architecture contains no assumption of independence between the feature nodes (this would be the case if, e.g., a Naive Bayesian scheme was adopted).

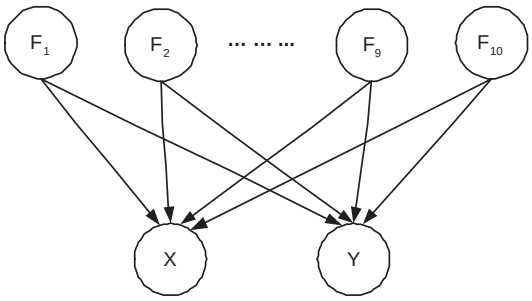


Figure 7.6. Continuous Bayesian Network for Regression

7.4.5 Regression performance measures

In order to evaluate the performance of the regression algorithms for a set of (test) samples, we compute the following error measure, which is the average distance between the real and estimated coordinates, normalized by the distance of the real coordinates from the $(0, 0)$ point of the EW:

$$E = \frac{1}{K} \cdot \sum_i \frac{\sqrt{(x_i - x_i')^2 + (y_i - y_i')^2}}{\sqrt{x_i'^2 + y_i'^2}} \tag{7.6}$$

In addition, in order to evaluate the respective regression performance for each one of the two dimensions (i.e. Valence - Arousal), the R^2 statistic ([97]) was used:

$$R_X^2 = 1 - \frac{\sum_i (x_i - x_i')^2}{\sum_i (x_i - \overline{x})^2} \tag{7.7}$$

$$R_Y^2 = 1 - \frac{\sum_i (y_i - y_i')^2}{\sum_i (y_i - \overline{y})^2} \tag{7.8}$$

We have selected to use the error measure defined in Equation 7.6, because of its physical meaning in the 2-D emotion plane: in particular, E expresses the 2-D error as a proportion of the average distance of all true points from the center of the emotion wheel. If, for example, $E = 1$, this means that the 2-D regression error is (on average) equal to the distance of the true points of the emotion wheel. On the other hand, the R^2 statistic is widely used in the bibliography in regression problems.

7.5 Emotion Recognition of Audio Streams From Movies

The proposed speech tracking scheme (described in Section 7.3), along with any one of the three emotion recognition schemes (described in Section 7.4) can be embedded in an overall method for analyzing the affective content of uninterrupted audio streams. The proposed overall scheme for affective recognition of audio streams is presented in Figure 7.7 and it can be divided in the following steps:

1. Speech Tracking: The algorithm described in 7.3 is used to detect all speech areas of the audio stream.

2. Segmentation: For each speech area, a segmentation algorithm is applied, in order to detect homogenous speech segments. In particular, the algorithm proposed in [36] has been used for detecting signal changes between successive segments.

3. Emotion Recognition: The method described in Section 7.4 is applied to each one of the detected speech segments. At the end of this step, a pair of emotion wheel coordinates is extracted for each speech segment.

These are the basic steps of the emotion recognition method for audio streams. The emotion coordinates of all speech segments can then be grouped to a predefined number of clusters, according to a clustering algorithm, e.g., the k-means ([52]). The resulted clustered coordinates of the emotion wheel are an overall representation of the affective content of the audio stream. Therefore, they can be used for discriminating films based on their speech-emotional content.

7.6 Experiments

7.6.1 Speech Tracking

In order to evaluate the performance of the speech tracking algorithm, a dataset that contains uninterrupted audio streams has been used. Those audio streams have been recorded from several movies but also radio stations. Furthermore, the speech segments of those streams have been manually annotated in order to being used as ground truth. The total duration of this dataset is 2 hours.

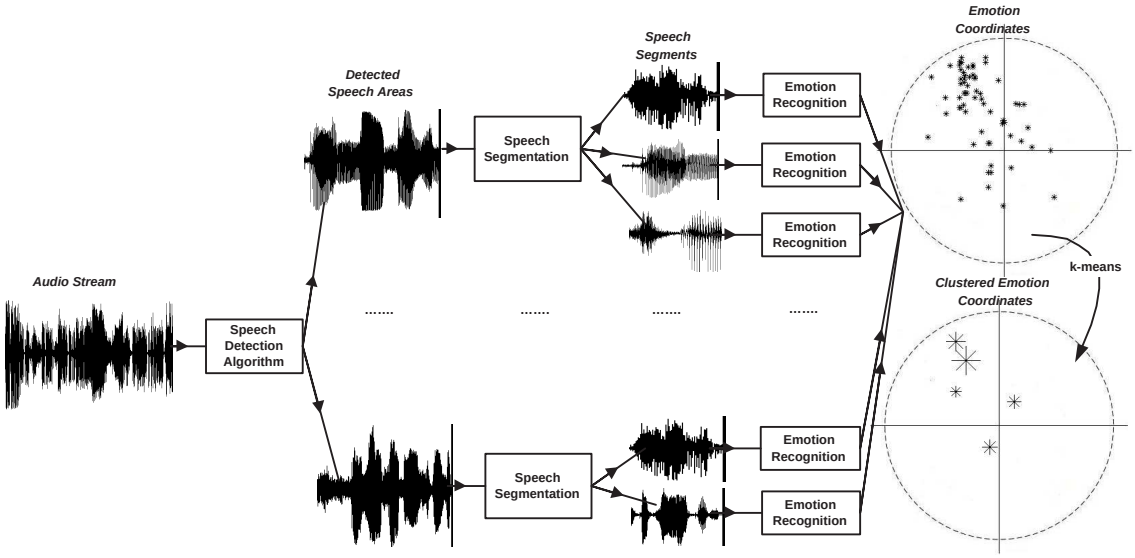


Figure 7.7. Overall Scheme

The algorithm has been evaluated on the above audio streams for different values of the probability threshold (T_P). In Figure 7.8 the results of these experimental procedure are presented. As expected, the speech classification precision rate grows with the threshold value, while the recall rate is reduced with the threshold value. The $F1$ measure reaches 89% for a threshold value around 0.55. However, as explained in Section 7.3, it is more important for the speech tracking stage to achieve high precision rates. Therefore, in the final emotion recognition system, the threshold of the speech tracking stage has been chosen to be 0.7. For this value, the speech precision rate reaches 95%, while the recall rate is 75%.

7.6.2 Emotion Representation Evaluation

As discussed in Section 7.4.1, D_j and DS_j correspond to the $j-$th user's normalized distance from the average decisions and normalized distance from the same user's mean decision. These two quantities are used to evaluate the 2D emotion representation itself. In particular, the average error of the users' annotation decisions has been found to be equal to 0.75. In other words, it is (on average) equal to 75% of the sample's mean true distance from the center of the EW. This means that the users' annotations are in good agreement. This finding is indicative that the EW offers a good affective representation for speech segments.

Finally, the self-user error DS was found equal to 0.56. In other words, the user's "self-

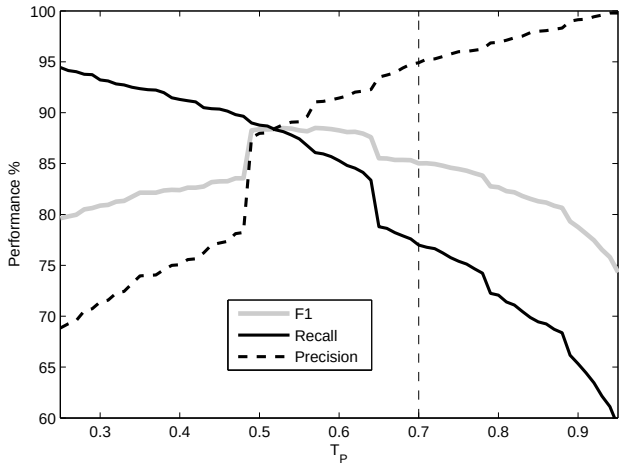


Figure 7.8. Speech Tracking performance for different probability thresholds. Maximum $F1$ measure (89%) appears for threshold values around 0.55, but in this work the selected threshold is 0.7, for which the speech precision rate reaches 95%.

error" is, on average, almost half of the distance of the mean user's decision from the center of the EW. This means that the agreement between different annotations of the same user, for the same sample, was high.

7.6.3 Speech Segment Emotion Recognition Evaluation

For testing and training the proposed emotion recognition method on speech segments, the K audio samples (i.e., the number of speech segments annotated by at least 5 humans) have been used. After the completion of the annotation procedure, K was equal to 400. For training all three regression schemes, 60% of the samples were used, while the remaining samples were used for testing purposes. For the final experiments, cross-validation has been used. In particular, 1000 repetitions of random sub-sampling validation have been executed. For comparison purposes, and in order to have a worst case scenario, we have computed the same performance measures for the random estimator of emotion coordinates, i.e., by selecting randomly (x, y) in the EW.

The emotion recognition results are shown in Table 7.1. It can be seen that the SVM and BN methods perform similarly, while the kNN approach has a slightly lower performance. Furthermore, in all cases, the R^2 measure for Valence is lower than for Arousal. This

means that estimating the "pleasantness" of an emotion is a harder task, than estimating its intensity. Finally, in all cases, the error is comparable to the human annotation error, which indicates that the regression methods lead to a high performance for the given data.

	E	R_X^2	R_Y^2
User	0.75	-	-
kNN	0.92	0.21	0.34
SVM	0.87	0.23	0.36
BN	0.88	0.23	0.35
Random	2.3	-3	-2.2

Table 7.1. User performances and emotion recognition results for speech segments

7.6.4 Examples of Emotion recognition of uninterrupted audio streams: The emotional signature

The algorithm described in Section 7.5, extracts the (clustered) points in the emotion wheel for large audio streams. These points can provide with a sufficient description the affective content of the respective audio streams. We will therefore refer to the clustered points as the "**emotional signature**" of the corresponding audio stream. In this Section, we have applied the overall algorithm to audio streams recorded from five types of videos: news, commercials, films that contain oral violence, documentaries and sportcasting videos. In particular, three audio streams from each genre have been used. In Figures 7.9, 7.10, 7.11, 7.12 and 7.13 the results of the overall method is presented for all five genres, along with the respective comments.

In addition to the taxonomical conclusions described in the figures, we propose some possible applications that may use the above emotion signatures:

1. Violence detection in multimedia content. Several methods have been proposed in the past for detecting violence in movies and videos, in order to protect sensitive population groups, such as children ([50], [44], [56]). Speech emotional information could also be used to complement visual information (e.g. people fighting) and audio events (e.g. gunshots, explosions, etc.), in order to detect oral violence. For example,

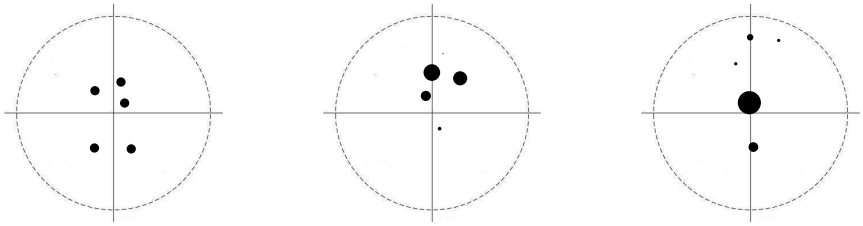


Figure 7.9. Emotion signatures for the audio streams from news: In most cases the Valence is neutral, while the Arousal can be both positive and negative (obviously that depends on the speaker).

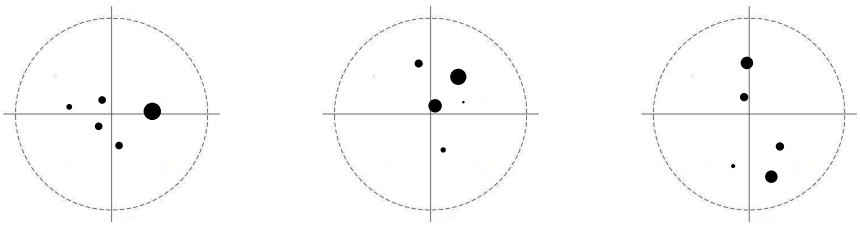


Figure 7.10. Emotion signatures for the audio streams from commercials: In general two kinds of areas are dominant: the first lies in the upper positive semicircle of the emotion wheel (large Arousal) and has positive or neutral Valence values (i.e. excitement and happiness), while the second has negative Arousal values and positive Valence values (this indicates a feeling of calmness). Both the excitement-happiness and the calmness feelings are quite often present in most commercials.

audio streams that contain large proportions of areas with Valence values smaller than -0.5 may possibly be characterized as unsuitable for children.

2. Audio-based search and retrieval. The proposed emotion recognition method could also be used in a system that retrieves multimedia content using the emotional signature as a searching criterion.

3. Automatic characterization of sport broadcasts. The emotion signatures could be used for retrieving sport events with particular affective labels. Furthermore, if the emotion recognition is applied on a midterm basis, one could extract specific parts of a sports events (e.g. a goal in a football game), according to the sportcaster's emotional state.

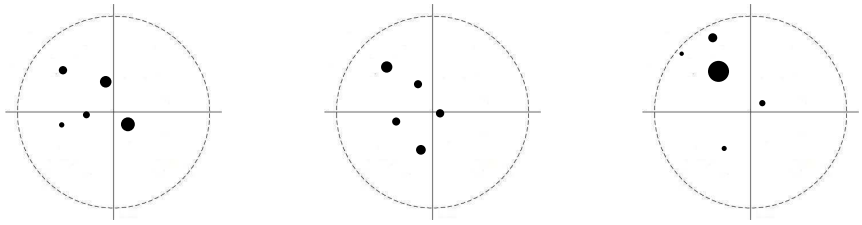


Figure 7.11. Emotion signatures for the audio streams from violent films: In this case almost all clusters have negative Valence. Arousal, on the other hand is both negative and positive, which indicates anger (or fear) and sadness in general.

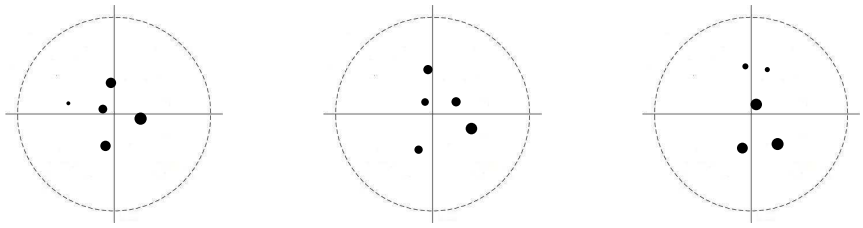


Figure 7.12. Emotion signatures for the audio streams from documentaries: This category shares similar emotional signatures with the news category, which is expected.

Of course, more research is required towards this direction.

7.7 Conclusions

A system for recognizing emotional content in speech from movies has been proposed. First, a computationally efficient speech tracking algorithm has been presented, that reaches a precision performance of 95%. Then, we propose using a general audio segmentation scheme for detecting homogenous speech segments. The major contribution of this chapter is the novel dimensional approach for emotion recognition of speech segments. We have described this task as a regression problem of mapping ten feature values to the two-dimensional emotion plane (Emotion Wheel). Three regression methods have been implemented and evaluated for this purpose. Besides testing the emotion recognition methods, we have also focused on evaluating the emotional representation itself. The Emotion Wheel has been found to be a good representation of the affective content of speech segments, since the corresponding manual

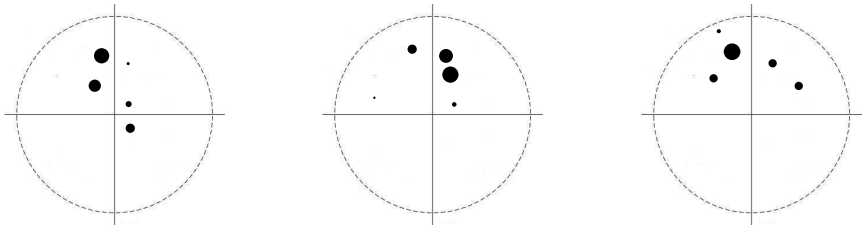


Figure 7.13. Emotion signatures for the audio streams from sportcasting videos: In this case, all clusters have positive Valence values, which is something expected for speech signals from sportcasting videos. Also, both positive and slightly negative (close to zero) Arousal values are present, which indicates emotional states like: excited, alarmed, happy and slightly angry.

annotations performed by several humans were in good agreement. Moreover, experiments have shown that the regression performance of all three methods was high, since the error was comparable to the average error of the manual (human) annotations. This means that the proposed audio features can be successfully mapped to the emotion plane. Finally, we have demonstrated how to extract emotional information from uninterrupted audio streams from movies.

Chapter 8

Conclusions and Future Directions

8.1 Conclusions

This thesis investigated methods for content-based characterization of multimedia data based on the audio information. At first, a novel speech-music discrimination algorithm, based on dynamic programming and Bayesian Networks, has been proposed and evaluated on radio broadcasts. The novelty of the method is located on the fact that the problem is treated as a task of maximizing a product of posterior probabilities, which is solved by means of dynamic programming.

The remaining chapters proposed algorithms for audio-based characterization of *video* data. First, the problem of locating the parts of audio streams from movies that contain music is treated as a mid-term classification task. This method can be used for extracting music-related information from movies, but also as a preprocessing stage in an overall movie characterization system.

In the sequel, the problem of detecting audio segments of homogenous content has been investigated. The problem of locating changes in the content of an audio stream has been treated as a soft-output binary classification task. The algorithm is not computationally demanding, since experiments have indicated that the execution time is almost 1% of the input audio data length. Moreover, the method introduces a general framework to audio segmentation, which does not depend explicitly on the number of audio classes. The proposed algorithm has been evaluated on real audio streams from movies, and the experiments showed an overall performance of 85%.

In the context of sequential segmentation-classification, the next part of this thesis focused on recognizing the content of a homogenous audio segment from movie. Towards this end, a profound investigation has led to the definition of seven audio classes. It has to be noted that focus has been given in defining classes of *violent* content. In order to solve the multi-class classification problem, the "One Vs All" multiclass classification method has been used, in combination with Bayesian Networks. The method has been evaluated on audio segments from more than 30 movies, and experiments indicated that the overall performance reached almost 70% for the seven-class classification task. Furthermore, the proposed system can also be used as a binary classifier for the "Violent" - "Non Violent" problem. In this case almost 15% of the violent data was incorrectly classified, while less than 17% of the non-violent data were classified as violent. The overall binary classification error was therefore almost 16%. Finally, an important advantage of the proposed method is that it produces a **probabilistic** measure for each one of the audio classes, which can be used in a segmentation-classification scheme for audio streams from movies.

Besides recognizing distinct audio classes, this thesis has also focused on recognizing emotions that underlie the speech information. A complete framework for speech emotion recognition of movies has been presented. Experiments have shown that the Emotion Wheel offers a good representation for speech emotions from movies. Furthermore, three regression models have been evaluated for the emotion recognition task. Finally, a possible scheme for extracting affective content from uninterrupted audio streams from movies is investigated.

With the exception of the speech-music discrimination algorithm, all other parts of this thesis can be used for *characterizing the content of a movie*, based on the audio information. The examined algorithms can be combined, as shown in Figure 8.1. At a first stage the music tracking algorithm proposed in Chapter 4 and the speech tracking algorithm (which is part of the emotion recognition system proposed in Chapter 7 are used, in order to detect music and speech areas of the audio stream, with high precision rate. Then, for the detected speech areas, the rest of the system described in Chapter 7 is used, in order to recognize the speech emotional states. All audio areas that have been left unclassified by the music-speech tracking stages are then fed as input to the general audio segmentation algorithm (described in Chapter 5). In this way, segments of homogenous content are detected. Each of those segments, is then classified using the multi-class classification algorithm presented

in Chapter 6.

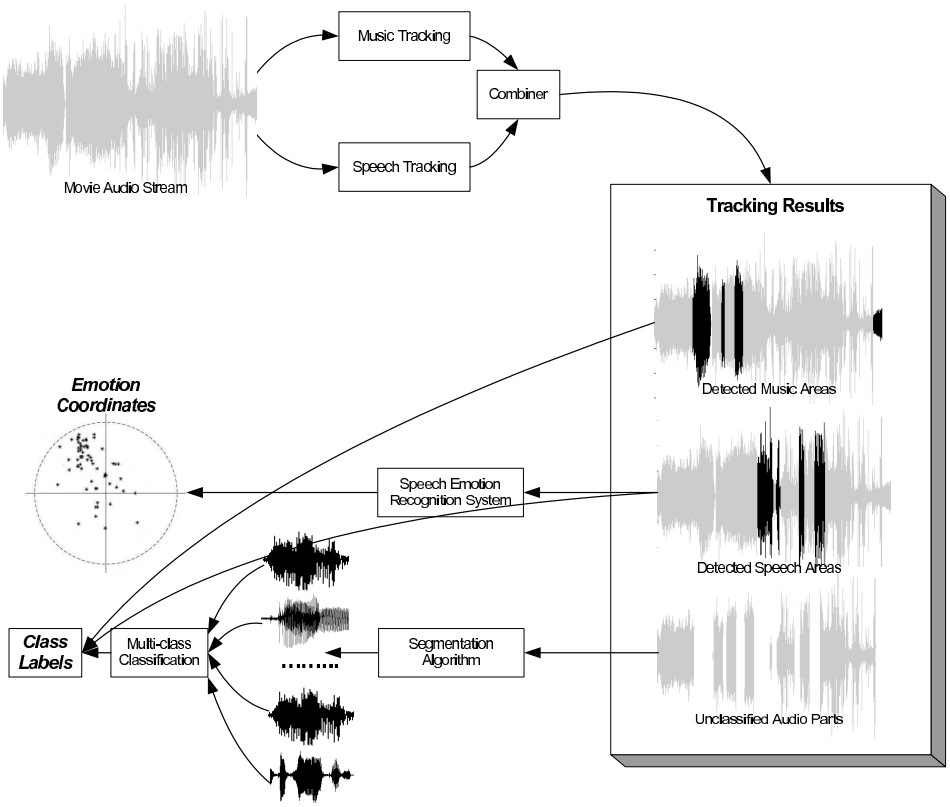


Figure 8.1. Overall architecture of a movie characterization system, based on audio information.

8.2 Future Directions

As already referred in Chapter 6, a challenging and promising research issue, is the development of a **joint segmentation - classification** method for the multi-class problem. Towards this end, the Bayesian probability provided by the proposed multiclass-classification algorithm may be used by a maximization algorithm in a similar way as in the binary case of speech-music discrimination (described in Chapter 3).

Furthermore, the Bayesian Network classifier could be expanded with more nodes. Those nodes could represent other types of individual classifiers (e.g. Support Vector Machines). Apart from implementing other types of individual classifiers, some individual decisions based on **other types of media** (e.g. image, text) can be added. For example, let the 6-th BN classifier described in Chapter 6, i.e., the classifier trained on the binary problem of "Fights

Vs Non-Fights". For the particular binary subproblem, the BN could be enriched by visual-based decisions (e.g., decisions based on motion characteristics, or features that stem from human modelling techniques). A possible BN-scheme for the particular binary sub-problem is shown in Figure 8.2, where the binary decision based on the audio information (node Y_6) is combined with a binary decision about the existence of fights based on visual modelling (node V). Nodes $V_1 \ldots V_N$ are individual decisions based on different visual characteristics. Note that, by using Bayesian Networks, some of the conditional probability tables (CPTs) can be empirically estimated (i.e., no training is required). In the particular example, if both the visual and the audio decisions have decided for the existence of "fights" then the overall probability for the fights class is 0.99. If the visual node's state is 1 (i.e., fights have been detected using the visual cues) while the audio node's state is 0, then the probability that the output node is 1 is equal to 0.75. If, on the other hand, a fight has not been detected by the visual part of the BN, but only by the audio part, then the corresponding probability is significantly lower. This stems from the experience that a fight event can be more effectively detected by the visual cues, and therefore more confidence should be given if the fight is detected by the visual part of the BN.

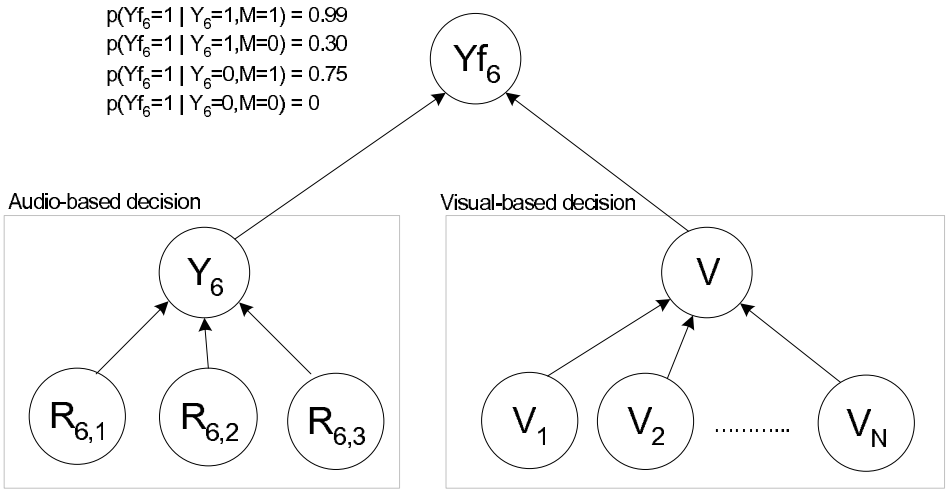


Figure 8.2. Example of a possible BN-scheme for combination of audio and visual decisions for the binary sub-task of "fights vs non-fights". $V_1 \ldots V_N$ are nodes that correspond to the individual decisions based on different *visual* characteristics. Node Y_6 corresponds to the binary decision for the "fights vs non-fights" subproblem using the audio information.

Another promising future issue is to enhance the performance of the overall movie charac-

terization system, by implementing **music classification** algorithms. Often music tracks in movies reveal particular semantic meanings, e.g., the emotional tension of a scene in a horror film. Apart from that, the musical genre itself may provide us with important information about the content of a movie.

Finally, a challenging task will be to develop methods for movie (or generally video) search, based on audio analysis. The audio class-specific probabilistic measures described in Chapter 6, the speech emotional representation presented in 7, along with other types of audio analysis (e.g. music recognition), can be used for creating an **audio-based movie indexing scheme**. Such a system could let the users execute particular queries, e.g., "search for movies which are composed by at least 70% of music track, while speech does not exceed 2%". In addition, *clusters* of similar movies in terms of acoustic labelling could be populated.

Appendix A

Format of audio files

All audio files used for training and evaluating the proposed methods of this thesis are stored in uncompressed WAV files. The sample resolution is 16 bits, the selected sampling frequency is 16000 Hz, while a single audio channel has been used (mono). The particular sampling frequency is common in most speech recognition applications. Though, in order to classify several audio types (which is the purpose of this particular thesis) a more detailed investigation of *how the sampling frequency affects the signal quality* is required.

The main reason that the above sampling rate has been selected is that different multimedia resources use different sampling rates. For example, CD quality music is sampled at 44100 Hz, while most radio broadcasts available through the WWW have a sampling frequency of 22050 or 16000 Hz. Furthermore, the different types of video files use different (or none) compression schemata for the audio information. Generally, the audio quality dramatically varies for the available video data. The adopted sampling frequency of 16000 Hz is a minimum prerequisite for the multimedia resources used in the experiments of this thesis.

So the purpose of the current Section is to investigate the SNR of the sampled audio data, for different sampling rates and for different content classes (compared to the high-quality data). Though, for the reasons described before, it was *not possible to obtain a sufficient number of high quality audio data for every class*. In order to solve this problem, the following procedure has been followed:

2 hours of high quality (i.e., 44100 Hz) audio data has been recorded from 5 movies. The data has been divided to 1000 mid-term segments (duration 2 to 5 seconds each). For each

of those segments, the following steps have been executed:

1. Each segment was downsampled to the following rates: 8000, 16000, 22050 and 32000 Hz.

2. For each of the downsampled segments, the SNR (compared to the initial high quality signal) has been computed.

3. Each of the resampled segment is broken into non-overlapping short-term windows and the zero crossing sequence is calculated. In addition, the average value of this sequence is computed (see Paragraph 2.3.2).

The next step was to **estimate the relationship between the adopted feature and the corresponding SNR**. In Figure A.1, an example of the relationship between the computed feature (average ZCR) and the SNR of the corresponding segment is presented, for the sampling rate of 16000 Hz. There is an obvious relationship between the two quantities. In particular, the SNR decreases for higher values of the average ZCR. Furthermore, the above relationship has been estimated using *polynomial interpolation*. The same process has been applied for all other sampling frequencies, and results are displayed in Figure A.2.

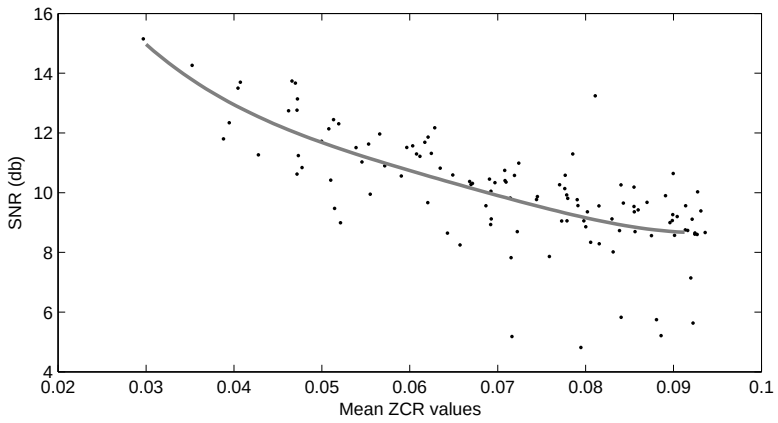


Figure A.1. Relationship between SNR and mean ZCR values, for the audio segments downsampled to 16000 Hz. The solid line represents the polynomial estimation of this relationship. It is obvious that the SNR is higher for low values of the adopted audio feature.

The estimated polynomial functions that map the mean ZCR of an audio segment to its sampling frequency-related SNR have been used for estimating the SNR for the several

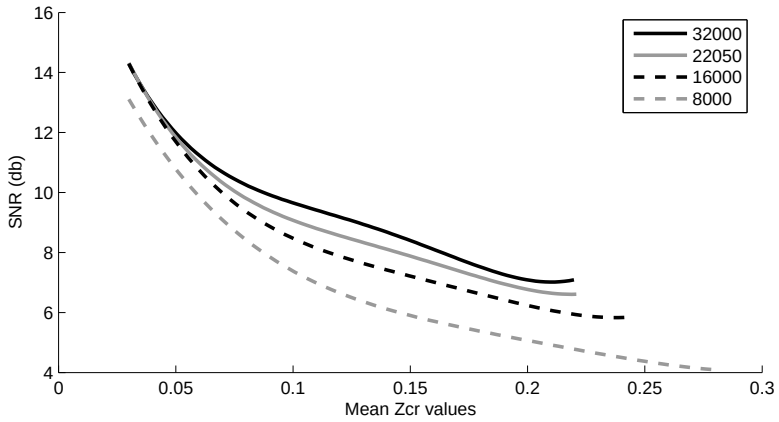


Figure A.2. Estimated relationship between the mean ZCR values and the corresponding SNRs, for all of the adopted (down)sampling frequencies.

Table A.1. SNRs (in dB) for different sampling rates and for all adopted classes.

Class	32KHz	22KHz	16KHz	8KHz
Music	10.1	9.7	9.2	8.1
Speech	9.3	8.8	8.2	7.0
Others1	10.9	10.6	10.3	9.2
Others2	9.9	9.5	9.0	7.9
Gunshots	8.6	8.1	7.5	6.3
Fights	9.4	8.9	8.3	7.1
Screams	8.5	8.0	7.4	6.1
Average	**9.5**	**9.1**	**8.6**	**7.4**

audio classes adopted in 6. The average estimated SNRs for each class and for each sampling frequency are displayed in Table A.1.

Appendix B

Bayesian Networks Basics

B.1 Probability Theory - Basics

B.1.1 Discrete Random Variables

Random variables represent the outcome of a random experiment and are usually represented by a capital Roman letter, such as X. *Sample space* is the set of all possible outcomes of the experiments and it is usually denoted by the Greek letter Ω. If Ω is finite or countably infinite, then X is called *discrete* random variable.

A *probability distribution pr* (for discrete random variables) is the probability that a unidentified random variable is equal to a particular element of Ω. *pr* satisfies the following equations:

$$pr(x) \geq 0, x \in \Omega \tag{B.1}$$

and

$$\sum_{x \in \Omega} pr(x) = 1 \tag{B.2}$$

In Figure B.1, an example of a probability distribution of a random variable is given, where $\Omega = \{1, 2, 3, 4, 5\}$.

Furthermore, let E be a subset of Ω (also called "event"). Then the probability of E is defined as:

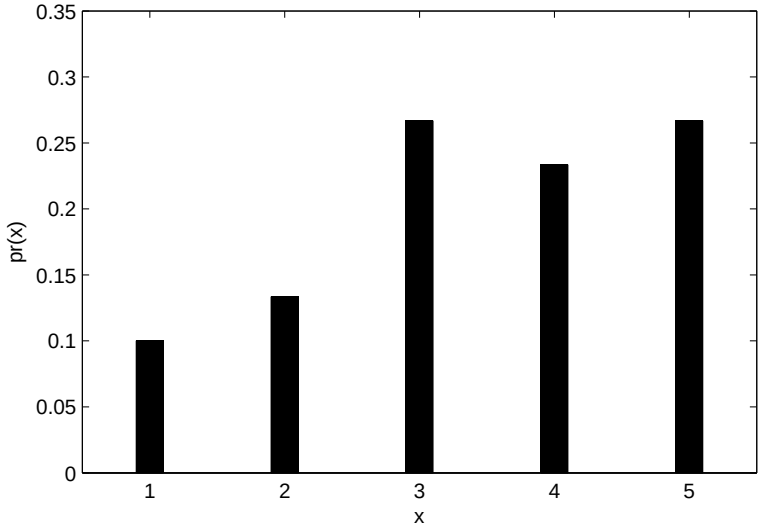


Figure B.1. Example of a probability distribution of a discrete variable.

$$P(E) = \sum_{x \in E} pr(x) \tag{B.3}$$

Obviously, the probability of a single element of Ω is: $P(\{x\}) = pr(x)$.

Furthermore, the *joint probability* of two events A and B is denoted as $P(A, B)$. The unconditional probability $P(A)$ is also called *marginal* probability ([98]). This is the probability that event A occurs regardless of another event B. Suppose that B is an event of a random variable Y, then the marginal probability $P(A)$ can be computed by summing all joint probabilities over all n outcomes of Y $(y_i, i = 1, \ldots, n)$:

$$P(A) = \sum_{i=1}^{n} P(A, y_i) \tag{B.4}$$

This procedure is usually called *marginalization*. If the variable Y is binary (i.e., only two possible outcomes of Y exist: B and B') then:

$$P(A) = P(A, B) + P(A, B') \tag{B.5}$$

$P(A|B)$ symbolizes the *conditional probability* of the event A, *given* the event B. Note that, in the general case, the conditional probability is defined using the following equation:

$$P(A|B) = \frac{P(A, B)}{P(B)} \tag{B.6}$$

An immediate observation from the definition of the conditional probability is the following equation, which is widely known as the *product rule*:

$$P(A,B) = P(A|B) \cdot P(B) = P(B|A) \cdot P(A) \tag{B.7}$$

Conditional probabilities $P(A|B)$ and $P(B|A)$ are *not* equal, in the general case. In particular, the two conditional probabilities are related according to the following equation, which is widely known as the Bayes' Rule:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \tag{B.8}$$

According to the Bays' rule, the two conditional probabilities are equal, if the corresponding prior probabilities (i.e., $P(A)$ and $P(B)$) are also equal. Finally, if the Bayes' Rule is combined with the marginilization procedure (equation B.5), then the result is the following alternative form:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B|A) \cdot P(A) + P(B|A') \cdot P(A')} \tag{B.9}$$

B.1.2 Independence

Statistical independence means that a particular event does not effect the probability of another event. The detailed definition of probabilistic independence is given below:

Two events X and Y are *independent* if and only if:

1. $P(X|Y) = P(X)$ and $P(Y|X) = P(Y)$ and $P(X) > 0$ and $P(Y) > 0$, **or**:
2. At least one has 0 probability.

Another approach to probabilistic independence can be given through the following theorem: "Two events X and Y are independent if and only if: $P(X,Y) = P(X) \cdot P(Y)$."

In addition, two events A and B are *conditionally independent*, given a third event C if and only if: $P(A,B|C) = P(A|C) \cdot P(B|C)$. Note, that conditional independence does not imply statistical independence. Furthermore, two events can be independent, but not conditionally independent, given a third event. Also, note that if X and Y are not conditionally

independent given C, then:

$$P(A, B|C) = P(A|B, C) \cdot P(B|C) = P(B|A, C) \cdot P(A|C)$$

B.2 Bayesian Networks

Bayesian Networks (BNs) are directed acyclic graphs (DAGs) *that encode conditional probabilities* among a set of random variables ([99], [52]). Each node of the graph corresponds to a separate random variable and the arcs of the graph encode the probabilistic dependence of the random variables (nodes). BNs are actually a sub-category of Graphical Models, i.e. graphs which nodes represent random variables, while the lack of arcs represents conditional independence. Undirected Graphical Models are known as Markov Random Fields (MRFs), while BNs are directed Graphical Models without circles.

In the case of discrete random variables, for each node (random variable) A, with parents B_1, ..., B_k *a conditional probability table* (CPT) $P(A|B_1, ..., B_k)$, is defined. In Figure B.2, a simple BN is presented. The BN consists of 5 binary nodes. The local conditional probabilities of each node are represented using the conditional probability tables. For example, if the state of node A is 1, then the probability that node C is 1, given that evidence, is 0.95, i.e., $P(C = 1|A = 1) = 0.95$). Note that, node A has no parents (i.e., it is a root node), and therefore the CPT reduces to unconditional probability. In that case, prior probabilities have to be specified.

A very important property of BNs is that CPTs can determine the full joint distribution, i.e., the joint probability over all nodes of the BN. This is widely known as the "Chain Rule", or "Recursive Factorization". So, let a BN over $U = \{A_1, A_2, \ldots, A_n\}$; then the joint probability distribution $P(U)$ is the product of all conditional local distributions:

$$P(U) = P(A_1, A_2, \ldots, A_n) = \prod_{i=1}^{n} P(A_i|parents(A_i)) \tag{B.10}$$

B.2.1 BNs and conditional independence

The structure of a BN can help in determining conditional independencies ([100], [101]). Towards this end, the "d-separation" graphical criterion can be used. A path p is "d-

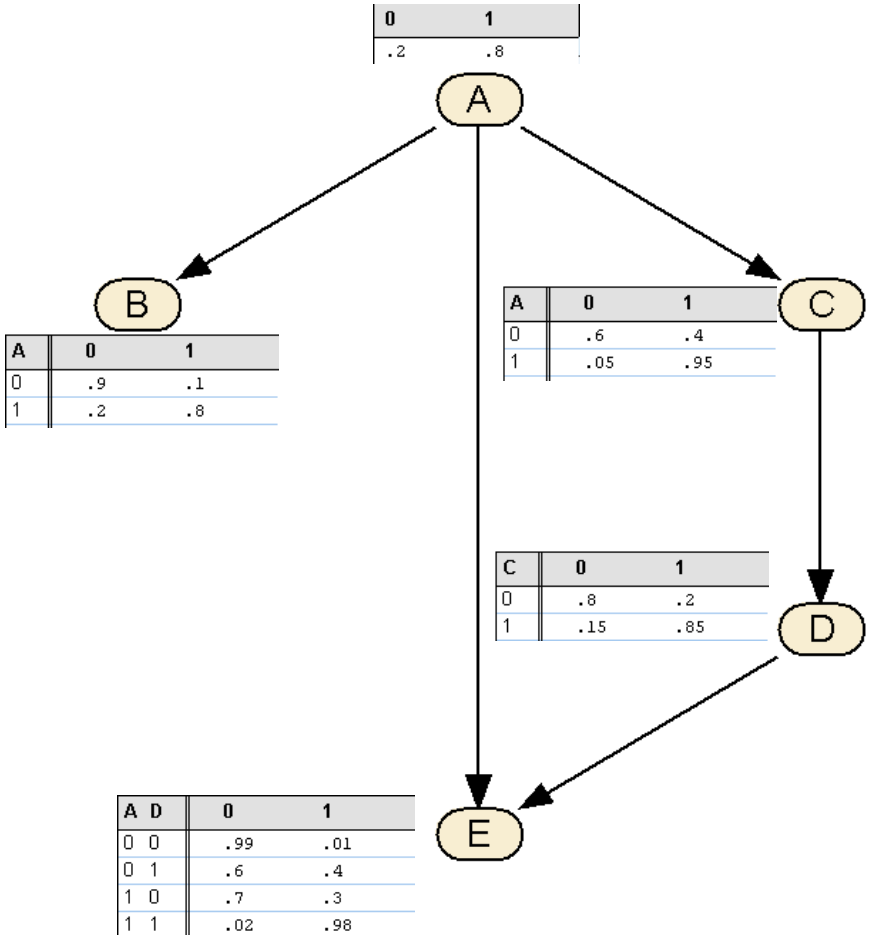


Figure B.2. A simple Bayesian Networks. All nodes correspond to binary random variables. The CPTs of each node are also presented.

separated" (or blocked) by a set of nodes Z iff:

- p contains a chain $i \Rightarrow j \Rightarrow k$ or a fork $i \Leftarrow j \Rightarrow k$ such that the middle node j is in Z, **or**:
- p contains an inverted fork $i \Rightarrow j \Leftarrow k$ such that neither the middle node j or any of its descendants are in Z.

Otherwise, the path is called *active*.

This criterion is used for finding conditional independencies as follow: **"X and Y are independent given evidence Z (or a set of variables) iff every undirected path from $X \Rightarrow Y$ is "blocked" or "d-separated" by Z"**

B.2.2 BN inference

Inference is the procedure of answering probabilistic queries using the BN structure. Suppose that values e have been *observed* on a set of variables E. The purpose of inference is to compute the posterior probability that node V is equal to v, given the observed variables ($P(V = v|E = e)$). Observed nodes are called *evidence*. The most obvious solution to the problem of inference is to sum irrelevant variables by applying marginalization in the Bayes' Rule. In the general case, this has been proved to be an NP-hard problem ([102]).

Though, some more efficient algorithms have been presented for solving the problem in restricted types of BNs. For example, a message passing algorithm has been proposed for exact inference in polytrees (i.e., single connected networks), which manages to solve the problem in a *linear* complexity in the number of nodes ([100]). Apart from that, *approximate inference* algorithms have also been proposed. Stochastic sampling is the general idea among such methods, according to which a set of random selected samples is selected and query probabilities are approximated by the frequencies of the sample (simple counts). Some alternative methods to stochastic sampling are the Gibbs sampling ([103]), Hybrid Monte Carlo sampling ([104]) and Metropolis sampling [101].

B.2.3 BN training

The process of estimating the BN topology (i.e. the topology of the graph structure) and the CPTs is called BN training. This can be achieved through using expert knowledge or raw data. The simplest case of BN training is when the structure is known and training data has no missing values. In that case, the goal is simply to estimate the values of each conditional probability table, which is achieved by maximizing the contribution to the likelihood function of each node independently. In the case of discrete nodes this is equivalent to a simple *counting* process. In the case the data is incomplete or the structure is unknown, learning is computationally intractable. Especially the process of learning an unknown structure is an NP-hard problem [75].

Bibliography

[1] Michael S. Lew, Nicu Sebe, Chabane Djeraba, and Ramesh Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2(1):1–19, 2006.

[2] H.M. Blanken, A.P. de Vries, H.E. Blok, and L. (Eds.) Feng. *Multimedia Retrieval.* Springer Verlag, 2007.

[3] John R. Smith and Shih F. Chang. Visualseek: A fully automated content-based image query system. In *ACM Multimedia*, pages 87–98, 1996.

[4] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Qian Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: the qbic system. *Computer*, 28(9):23–32, 1995.

[5] Smeulders Arnold W. M., Worring Marcel, Santini Simone, Gupta Amarnath, and Jain Ramesh. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380, 2000.

[6] Kunio Kashino, Takayuki Kurozumi, and Hiroshi Murase. A quick search method for audio and video signals based on histogram pruning. *IEEE Trans. on Multimedia*, 5:348–357, 2003.

[7] Ziyou Xiong, Xiang S. Zhou, Qi Tian, Yong Rui, and Huangm Ts. Semantic retrieval of video - review of research on video retrieval in meetings, movies and broadcast news, and sports. *Signal Processing Magazine, IEEE*, 23(2):18–27, 2006.

[8] A. Kokaram, N. Rea, R. Dahyot, M. Tekalp, P. Bouthemy, P. Gros, and I. Sezan. Browsing sports video: trends in sports-related indexing and retrieval work. *Signal Processing Magazine, IEEE*, 23(2):47–58, 2006.

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

[10] Knees Peter, Pohle Tim, Schedl Markus, and Widmer Gerhard. A music search engine built upon audio-based and web-based similarity measures. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 447–454. ACM, 2007.

[11] Foote Jonathan. An overview of audio information retrieval. *Multimedia Syst.*, 7(1):2–10, 1999.

[12] Wold Erling, Blum Thom, Keislar Douglas, and Wheaton James. Content-based classification, search, and retrieval of audio. *IEEE MultiMedia*, 3(3):27–36, 1996.

[13] Tzanetakis George and Cook Perry. Music analysis and retrieval systems for audio signals. *J. Am. Soc. Inf. Sci. Technol.*, 55(12):1077–1083, 2004.

[14] Asif Ghias, Jonathan Logan, David Chamberlin, and Brian C. Smith. Query by humming: musical information retrieval in an audio database. In *In ACM Multimedia*, pages 231–236, 1995.

[15] Lu Lie, You Hong, and Zhang Hong J. A new approach to query by humming in music retrieval. In *2001 IEEE International Conference on Multimedia and Expo*, 2001.

[16] Cheng Yang. Efficient acoustic index for music retrieval with various degrees of similarity. In *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*, pages 584–591, 2002.

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

[18] Changsheng Xu, N. C. Maddage, and Xi Shao. Automatic music classification and summarization. *Speech and Audio Processing, IEEE Transactions on*, 13(3):441–450, 2005.

[19] N. Scaringella, G. Zoia, and D. Mlynek. Automatic genre classification of music content: a survey. *Signal Processing Magazine, IEEE*, 23(2):133–141, 2006.

[20] Antti Eronen. *Automatic Musical Instrument Recognition, Master of Science Thesis*. Institute of Signal Processing, Department of Information Technology, Tampere University of Technology, Finland, 2001.

[21] S. Furui. Cepstral analysis technique for automatic speaker verification. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 29:254–272, 1981.

[22] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn. Speaker verification using adapted gaussian mixture models. In *Digital Signal Processing*, 2000.

[23] A. Bonafonte A. Nogueiras, A. Moreno and J. B. Mariρo. Speech emotion recognition using hidden markov models. In *in Proc. Eurospeech*, pages 2679ὐ–2682, 2001.

[24] E.; Tsapatsoulis N.; Votsis G.; Kollias S.; Fellenz W. Cowie, R.; Douglas-Cowie and J. Taylor. Emotion recognition in human-computer interaction. *Signal Processing Magazine, IEEE*, 18:32–80, 2001.

[25] L. Lu, D. Liu, and H. Zhang. Automatic mood detection and tracking of music audio signals. *IEEE Trans. on Audio, Speech & Language Processing*, 14:5–18, 2006.

[26] H.H Yi-Hsuan Yang; Yu-Ching Lin; Ya-Fan Su; Chen. A regression approach to music emotion recognition. *IEEE Transactions on Audio, Speech & Language Processing*, 16:448–457, 2008.

[27] A. Li-Qun Xu Hanjalic. Affective video content representation and modeling. *IEEE Transactions on Multimedia*, 7:143–154, 2005.

[28] Y. Wang and L. Guan. Recognizing human emotional state from audiovisual signals. *Multimedia, IEEE Transactions on*, 10:936–946, 2008.

[29] A. Hanjalic. Extracting moods from pictures and sounds: towards truly personalized tv. *Signal Processing Magazine, IEEE*, 23:90–100, 2006.

[30] C. Cotsaces, N. Nikolaidis, and I. Pitas. Video shot detection and condensed representation. a review. *Signal Processing Magazine, IEEE*, 23(2):28–37, 2006.

[31] Alexander G. Hauptmann and Michael A. Smith. Text, speech and vision for video segmentation: The informedia project. In *AAAI Fall Symposium, Computational Models for Integrating Language and Vision*, pages 10–12, 1995.

[32] J. S. Boreczky and L. D. Wilcox. A hidden markov model framework for video segmentation using audio and image features. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 6, pages 3741–3744, 1998.

[33] S. Chen and P.S. Gopalakrishnan. Speaker, environment and channel change detection and clustering via the bayesian information criterion. In *DARPA Proc. Speech Recognition Workshop*, 1998.

[34] T.N. Sainath, D. Kanevsky, and G. Iyengar. Unsupervised audio segmentation using extended baum-welch transformations. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2007.

[35] M. Omar, U. Chaudhari, and G. Ramaswamy. Blind change detection for audio segmentation. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, 2005.

[36] Theodoros Giannakopoulos, Aggelos Pikrakis, and Sergios Theodoridis. A novel efficient approach for audio segmentation. In *19th International Conference on Pattern Recognition, 2008 (ICPR08)*.

[37] C. Panagiotakis and G. Tziritas. A speech/music discriminator based on rms and zero-crossings. 7(1):155–166, 2005.

[38] T. Zhang and J. Kuo. Audio content analysis for online audiovisual data segmentation and classification. *IEEE Transactions On Speech And Audio Processing*, 9(4):441–457, 2001.

[39] J. Ajmera, I. McCowan, and H. Bourlard. Speech/music segmentation using entropy and dynamism features in a hmm classification framework. *Speech Commun.*, 40(3):351–363, 2003.

[40] A. Pikrakis, T. Giannakopoulos, and S. Theodoridis. A speech/music discriminator of radio recordings based on dynamic programming and bayesian networks. *Multimedia, IEEE Transactions on*, 10(5):846–857, 2008.

[41] T. Zhang Y. Li and D. Tretter. An overview of video abstraction techniques. *Technical Report HPL-2001-191, HP Laboratory, 2001*, 2001.

[42] Ba Tu Truong and Svetha Venkatesh. Video abstraction: A systematic review and classification. *ACM Trans. Multimedia Comput. Commun. Appl.*, 3(1):3, 2007.

[43] Ying Li, Shih-Hung Lee, Chia-Hung Yeh, and C. C. J. Kuo. Techniques for movie content analysis and skimming: tutorial and overview on video abstraction techniques. *Signal Processing Magazine, IEEE*, 23(2):79–89, 2006.

[44] Jeho Nam and Ahmed H. Tewfik. Event-driven video abstraction and visualization. *Multimedia Tools Appl.*, 16(1-2):55–77, 2002.

[45] Wei Chai. Semantic segmentation and summarization of music: methods based on tonality and recurrent structure. *Signal Processing Magazine, IEEE*, 23(2):124–132, 2006.

[46] Wei Chai and Barry Vercoe. Music thumbnailing via structural analysis. In *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, pages 223–226, 2003.

[47] Namunu C. Maddage, Changsheng Xu, Mohan S. Kankanhalli, and Xi Shao. Content-based music structure analysis with applications to music semantics understanding. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 112–119, 2004.

[48] Mark A. Bartsch and Gregory H. Wakefield. Audio thumbnailing of popular music using chroma-based representations. *IEEE Transactions Multimedia*, 7(1), 2005.

[49] A. Vasconcelos, N.; Lippman. Towards semantically meaningful feature spaces for the characterization of video content. In *International Conference on Image Processing, 1997*, pages 25–28.

[50] N. V. Lobo A. Datta, M. Shah. Person-on-person violence detection in video data. In *IEEE International Conference on Pattern Recognition, Canada, 2002.*

[51] Jeho Nam, Masoud Alghoniemy, and Ahmed H. Tew K. Audio-visual content-based violent scene characterization. In *in IEEE International Conference on Image Processing*, pages 353–357, 1998.

[52] S. Theodoridis and K. Koutroumbas. *Pattern Recognition, Third Edition.* Academic Press, Inc., Orlando, FL, USA, 2008.

[53] Ν. Καλουπτσίδης. *Σήματα, Συστήματα και Αλγόριθμοι.* Δίαβλος, 1993.

[54] E. Scheirer and M. Slaney. Construction and evaluation of a robust multifeature speech/music discriminator. In *ICASSP '97: Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, page 1331, Washington, DC, USA, 1997. IEEE Computer Society.

[55] T. Giannakopoulos A. Pikrakis and S. Theodoridis. Gunshot detection in audio streams from movies by means of dynamic programming and bayesian networks. In *33rd International Conference on Acoustics, Speech, and Signal Processing (ICASSP08).*

[56] Theodoros Giannakopoulos, Aggelos Pikrakis, and Sergios Theodoridis. A multi-class audio classification method with respect to violent content in movies, using bayesian networks. In *IEEE International Workshop on Multimedia Signal Processing (MMSP07).*

[57] Theodoros Giannakopoulos, Dimitrios Kosmopoulos, Andreas Aristidou, and Sergios Theodoridis. Violence content classification using audio features. In *4th Hellenic Conference on Artificial Intelligence (SETN06).*

[58] Kim Hyoung-Gook, Moreau Nicolas, and Thomas Sikora. *MPEG-7 Audio and Beyond: Audio Content Indexing and Retrieval.* John Wiley & Sons, 2005.

[59] H. Misra and et al. Spectral entropy based feature for robust asr. In *ICASSP, Montreal, Canada, 2004*, 2004.

[60] T. Giannakopoulos A. Pikrakis and S. Theodoridis. A computationally efficient speech/music discriminator for radio recordings. In *2006 International Conference on Music Information Retrieval and Related Activities (ISMIR06)*.

[61] T. Tolonen and M. Karjalainen. A computationally efficient multipitch analysis model. 8, 2000.

[62] A.P. Klapuri. Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. In *IEEE Transactions on Speech and Audio Processing*, volume 11, pages 804 – 816, 2003.

[63] R.N. Shepard. Circularity in judgments of relative pitch. In *Journal of the Acoustical Society of America, Vol. 36, pp. 2346-2353*, 1964.

[64] G.H. Wakefield. Mathematical representation of joint time-chroma distributions. In *Proceedings of the International Symposium on Optical Science, Engineering and Instrumentation (SPIE), Denver, Colorado*, 1999.

[65] Theodoros Giannakopoulos, Aggelos Pikrakis, and Sergios Theodoridis. Music tracking in audio streams from movies. In *IEEE International Workshop on Multimedia Signal Processing 2008 (MMSP08)*.

[66] L.L. Beranek. *Acoustic Measurements*. Wiley, New York, 1949.

[67] J. Saunders. Real-time discrimination of broadcast speech/music. In *Proceedings of the Acoustics, Speech, and Signal Processing (ICASSP96)*, pages 993–996.

[68] G. Williams and D. Ellis. Speech/music discrimination based on posterior probability features. In *Proceedings of Eurospeech, Budapest, 1999*, pages 687–690, 1999.

[69] M.J. Carey, E.S. Parris, and H. Lloyd-Thomas. A comparison of features for speech, music discrimination. In *ICASSP '99: Proceedings of the Acoustics, Speech, and Signal Processing, 1999. on 1999 IEEE International Conference*, pages 149–152, Washington, DC, USA, 1999. IEEE Computer Society.

[70] P.J. Moreno and R. Rifkin. Using the fisher kernel method for web audio classification. In *ICASSP '00: Proceedings of the Acoustics, Speech, and Signal Processing, 2000. on IEEE International Conference*, pages 2417–2420, Washington, DC, USA, 2000. IEEE Computer Society.

[71] K. El-Maleh, M. Klein, G. Petrucci, and P. Kabal. Speech/music discrimination for multimedia applications. In *ICASSP '00: Proceedings of the Acoustics, Speech, and Signal Processing, 2000. on 2000 IEEE International Conference*, pages 2445–2448, Washington, DC, USA, 2000. IEEE Computer Society.

[72] N. Casagrande, D. Eck, and B. Kegl. Frame-level audio feature extraction using adaboost. In *ISMIR*, pages 345–350, 2005.

[73] A. Papoulis and S. Unnikrishna Pillai. *Probability, Random Variables and Stohastic Processes, 4th edition.* McGraw-Hill, NY, 2001.

[74] V. Pavlovic A. Garg and T.S. Huang. Bayesian networks as ensemble of classifiers. In *IEEE International Conference on Pattern Recognition, pp. 779–784, Quebec City, Canada, August 2002.*

[75] D. Heckerman. A tutorial on learning with bayesian networks. *Microsoft Research, MSR-TR-95-06.*

[76] F. Bimbot, J. F. Bonastre, C. Fredouille, G. Gravier, Magrin I. Chagnolleau, S. Meignier, T. Merlin, Ortega J. Garcia, Petrovska Delacretaz, and Reynolds. A tutorial on text-independent speaker verification. *EURASIP Journal on Applied Signal Processing*, 4:430–451, 2004.

[77] K. Seyerlehner, T. Pohle, M. Schedl, and G. Widmer. Automatic music detection in television productions. In *Proc. of the 10th Int. Conference on Digital Audio Effects (DAFx-07), Bordeaux, France*, September 10 - 15.

[78] T. Izumitani, R. Mukai, and K. Kashino. A background music detection method based on robust feature extraction. In *Proc. of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008), Las Vegas, Nevada, USA*, March 30 - April 4 2008.

[79] K. Lee and D. Ellis. Detecting music in ambient audio by long-window autocorrelation. In *Proc. of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008), Las Vegas, Nevada, USA*, March 30 - April 4.

[80] L. Rabiner. On the use of autocorrelation analysis for pitch detection. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on*, 25(1):24–33, 1977.

[81] The internet movie database.

[82] Chung-Hsien Wu. and Chia-Hsin Hsieh. Multiple change-point audio segmentation and classification using an mdl-based gaussian model. *IEEE Transactions on Audio, Speech and Language Processing*, 14:647–657, 2006.

[83] Kevin Woods, Jr. W. Philip Kegelmeyer, and Kevin Bowyer. Combination of multiple classifiers using local accuracy estimates. *IEEE Transactions Pattern Analysis Machine Intelligence*, 19:405–410, 2997.

[84] J.A. Roth A.J. Reiss. *Understanding and Preventing Violence*. National Academy Press, Washington, DC, USA, 1993.

[85] Zeeshan Rasheed and Mubarak Shah. Movie genre classification by exploiting audio-visual features of previews. In *In Proceedings 16th International Conference on Pattern Recognition*, pages 1086–1089, 2002.

[86] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *J. Mach. Learn. Res.*, 5:101–141, 2004.

[87] C. Clavel, I. Vasilescu, L. Devillers, G. Richard, and T. Ehrette. Fear-type emotion recognition for future audio-based surveillance systems. *Speech Commun.*, 50(6):487–503, 2008.

[88] Michael Grimm, Kristian Kroschel, Emily Mower, and Shrikanth Narayanan. Primitives-based evaluation and estimation of emotions in speech. *Speech Commun.*, 49(10-11):787–800, 2007.

[89] S. Reiter B. Schuller C. Cox E. Douglas-Cowie M. Wollmer, F. Eyben and R. Cowie. Abandoning emotion classes - towards continuous emotion recognition with modelling of long-range dependencies. In *Proc. 9th Interspeech*, pages 597ů–600, 2008.

[90] N. Tishby A. Navot, L. Shpigelman and E. Vaadia. Nearest neighbor based feature selection for regression and its application to neural activity. *Advances in Neural Information Processing Systems*, 2005.

[91] Michael E. Mavroforakis and Sergios Theodoridis. A geometric approach to support vector machine (svm) classification. *IEEE Transactions on Neural Networks*, 17(3):671–682, 2006.

[92] Vladimir Vapnik, Steven E. Golowich, and Alex Smola. Support vector method for function approximation, regression estimation, and signal processing. *in Advances in Neural Information Processing Systems*, 9:281–287, 1997.

[93] Alex J. Smola and Bernhard Scholkopf. A tutorial on support vector regression. *in Statistics and Computing*, 14:199–222, 2004.

[94] Vladimir Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, New York, 1995.

[95] Geoffrey Holmes Eibe Frank, Leonard Trigg and Ian H. Witten. Technical note: Naive bayes for regression. *in Machine Learning, Kluwer Academic Publishers*, 41:5–25, 2000.

[96] Antonio Fernandez and Antonio Salmeron. Extension of bayesian network classifiers to regression problems. In *in Proc. of the 11th Ibero-American conference on AI: Advances in Artificial Intelligence*, 2008.

[97] A. Sen and M. Srivastava. *Regression analysis, theory, methods and applications.* M. Springer, 1990.

[98] Charles M. Grinstead and Laurie J. Snell. *Introduction to Probability.*

[99] Finn V. Jensen. *Bayesian Networks and Decision Graphs.* Springer, 2001.

[100] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference.* Morgan Kaufmann, 1988.

[101] J Pearl. Evidential reasoning using stochastic simulation of causal models. *Artif. Intell.*, 32(2):245–257, 1987.

[102] G. F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence*, 42(2-3):393–405, 1990.

[103] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[104] W. R. Gilks. *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC, 1995.