University of Crete
Department of Computer Science

FO.R.T.H.
Institute of Computer Science

Voicing detection in spontaneous and real-life recordings from music lessons

(MSc. Thesis)

Sofia-Elpiniki Giannikaki

Heraklion

April 2015

Department of Computer Science
University of Crete

Voicing detection in spontaneous and real-life recordings from music lessons

Submitted to the
Department of Computer Science
in partial fulfillment of the requirements for the degree of
Master of Science

April 2, 2015

Author:

Sofia-Elpiniki Giannikaki
Department of Computer Science

Committee

Supervisor

Yannis Stylianou
Professor

Member

Athanasios Mouchtaris
Assistant Professor

Member

Emmanouil Benetos
Researcher, City University London

Accepted by:

Chairman of the
Graduate Studies Committee

Antonis Argyros
Professor

Heraklion, April 2015

Abstract

Speech is one of the most important abilities that we have, since it is one of the principal ways of communication with the world. In the past few years a lot of interest has been shown in developing voice-based applications. Such applications involve the isolation of speech from an audio file. The algorithms that achieve this are called Voice Detection algorithms. From the analysis of a given input audio signal, the parts containing voice are kept while the other parts (noise, silence, etc) are discarded. In this way a great reduction of the information to be further processed is achieved.

The task of Voice Detection is closely related with Speech/Nonspeech Classification. In addition, Singing Voice Detection and Speech/Music Discrimination can be seen as subclasses of what we generally call Voice Detection. When dealing with such tasks, an audio signal is given as an input to a system and is then processed. The signal is usually analysed in frames, from which features are extracted. The frame duration depends mostly on the application and sometimes on the features being used. Many features have been proposed until now. There are two categories in which the features could be divided, time domain and frequency domain features. In time domain the short time energy, the zero-crossing rate and autocorrelation based features are most often used. In frequency domain cepstral features are most frequently used, due to the useful information about speech presence. To be more specific, in Singing Voice Detection and in Speech/Music Discrimination the state-of-the-art feature are the Mel-Frequency Cepstral Coefficients. It has been reported, that this particular feature provides the best performance in the majority of the cases.

In this thesis an algorithm is developed that performs voice detection in spontaneous and real-life recordings from music lessons. The content of the recordings was such that the proposed algorithm was challenged to discriminate both speech and singing voice from music and other noises. A classic approach for this problem would use MFCCs as the discrimination feature and an SVM classifier for the classification into "speech" or "nonspeech". In our work the methodology of this approach is expanded by preserving the MFCCs as the main feature and incorporating three other features namely, the Cepstral Flux, the Clarity and the Harmonicity. Cepstral Flux is extracted from the Cepstrum, while Clarity and Harmonicity are time-domain autocorrelation-based features. The goal is to improve with these additional features the performance of the system that uses only the MFCCs. So, different combination of the three additional features with the MFCCs were examined and evaluated. A 10-fold cross-validation is applied on segments, which are labelled as "speech" or "nonspeech". The database used for the training and the testing purposes of our algorithm consists of three seminars. Two of them concern traditional

cretan music classes with lira and the third one traditional cretan music classes with lute. Each recording has been carried out under different environmental conditions.
Performance evaluation was conducted using the Detection Error Tradeoff (DET) and Receiver Operating Characteristic (ROC) curves as a visual evaluation tool. Also, the Equal Error Rate (EER), the Efficiency and the Area Under the Curve (AUC) were computed in each case. Each seminar was evaluated separately, as well as all together. A combination of training and testing sets from different seminars was also done, to be able to provide reliable results. It is shown that the use of the additional features significantly enhances the performance of the classic algorithm that uses only the MFCCs from about 0.5% to 20%. Specifically, it is observed that three out of the five combinations stand out, by reducing about 20% the miss probability given a false alarm probability equal to 5%.

Περίληψη

Μία από τις σημαντικότερες ικανότητες που έχει ο άνθρωπος είναι η ομιλία, η οποία αποτελεί και το βασικό τρόπος επικοινωνίας με τον υπόλοιπο κόσμο. Τα τελευταία χρόνια το ενδιαφέρον πολλών έχει επικεντρωθεί στην ανάπτυξη εφαρμογών, οι οποίες βασίζονται στη φωνή. Σε τέτοιου είδους εφαρμογές, μας δίδεται ένα σήμα εισόδου από το οποίο χρησιμοποιούμε μόνο τα κομμάτια που περιέχουν φωνή. Με άλλα λόγια, αναλύοντας το σήμα εντοπίζουμε τα κομμάτια φωνής, τα οποία και κρατάμε, ενώ τα υπόλοιπα (θόρυβος, ησυχία κλπ) τα αγνοούμε. Η διαδικασία αυτή ονομάζεται ανίχνευση φωνής (Voice Detection). Με τη διαδικασία αυτή μειώνεται δραματικά ο όγκος της πληροφορίας που πρόκειται να επεξεργαστούμε, κάτι το οποίο είναι πολύ χρήσιμο.

Η διαδικασία της ανίχνευσης της φωνής σχετίζεται στενά με την ταξινόμηση σε ομιλία και μη ομιλία. Επίσης, τόσο η ανίχνευση τραγουδιού όσο και η διάκριση ομιλίας/μουσικής μπορούν να θεωρηθούν υποκατηγορίες της ανίχνευσης φωνής. Σε όλες αυτές τις περιπτώσεις μας δίδεται ένας σήμα εισόδου το οποίο και επεξεργαζόμαστε. Συνήθως η ανάλυση του σήματος γίνεται σε μικρότερα κομμάτια, από τα οποία εξάγουμε χαρακτηριστικά. Η διάρκεια των κομματιών κυμαίνεται περίπου μεταξύ 0.02 και 3 δευτερολέπτων και ορίζεται ανάλογα με το πρόβλημα που έχουμε κληθεί να λύσουμε. Μπορεί επίσης να εξαρτάται από το είδος των χαρακτηριστικών που θέλουμε να εξάγουμε. Μέχρι τώρα έχουν προταθεί πλήθος χαρακτηριστικών, κάποια από τα οποία είναι εφικτό να παράγουν αποτελέσματα χρησιμοποιώντας μικρά κομμάτια του σήματος. Αντίθετα, υπάρχουν χαρακτηριστικά τα οποία απαιτούν περισσότερη πληροφορία με αποτέλεσμα η διάρκεια των κομματιών να πρέπει να είναι μεγάλη. Τα χαρακτηριστικά μπορούν να χωριστούν σε δύο κατηγορίες, σε αυτά του πεδίου του χρόνου και σε εκείνα του πεδίου των συχνοτήτων. Στο πεδίο του χρόνου ευρέως διαδεδομένα είναι η ενέργεια, ο ρυθμός διέλευσης από το μηδενικό άξονα και χαρακτηριστικά που βασίζονται στην αυτοσυσχέτιση. Από την άλλη, στο πεδίο των συχνοτήτων ένα μεγάλο ποσοστό των χαρακτηριστικών εξάγεται από το Cepstrum (επέκταση του φάσματος). Αυτό συμβαίνει διότι εκεί υπάρχει χρήσιμη πληροφορία για τη φωνή. Συγκεκριμένα, το πιο διαδεδομένο χαρακτηριστικό στην ανίχνευση τραγουδιού και στη διάκριση ομιλίας/μουσικής είναι οι Mel-frequency Cepstral συντελεστές. Υποστηρίζεται ότι το χαρακτηριστικό αυτό δίνει τα καλύτερα αποτελέσματα στην πλειοψηφία των περιπτώσεων.

Στην εργασία αυτή παρουσιάζεται ένας αλγόριθμος ανίχνευσης φωνής πάνω σε πραγματικές καταγραφές από μαθήματα μουσικής. Καθώς η φύση των ηχογραφήσεων είναι τέτοια, στόχος είναι να εντοπίζεται τόσο η ομιλία όσο και το τραγούδι. Ένα κλασικό σύστημα

χρησιμοποιεί τους MFC συντελεστές ως χαρακτηριστικό διαχωρισμού "φωνής"/"μη φωνής" και μία μηχανή διανυσματικής υποστήριξης (Support Vector Machine) για την ταξινόμηση. Βάση ενός τέτοιου συστήματος λοιπόν, ορίζουμε τους MFC συντελεστές ως το κύριο χαρακτηριστικό και προσθέτουμε άλλα τρία, τη ροή του Cepstrum, τη Σαφήνεια και την Αρμονικότητα. Τα δύο τελευταία βασίζονται στην αυτοσυσχέτιση του σήματος στο πεδίο του χρόνου. Ο σκοπός είναι να βελτιωθεί η απόδοση ενός συστήματος, που χρησιμοποιεί μόνο τους MFC συντελεστές. Εξετάζουμε 5 διαφορετικούς συνδυασμούς των χαρακτηριστικών που προαναφέρθηκαν με τους MFC συντελεστές. Έπειτα, εφαρμόζεται ένα 10-fold cross-validation πάνω σε τμήματα του σήματος, για να ταξινομηθούν σε "φωνή" και "μη φωνή". Η βάση που χρησιμοποιήθηκε για την εκπαίδευση και τον έλεγχο του συστήματος αποτελείται από 3 σεμινάρια. Δύο από αυτά σχετίζονται με τη λύρα στην παραδοσιακή κρητική μουσική, ενώ το τρίτο αφορά το λαούτο. Σημειώνεται ότι η κάθε ηχογράφηση έχει πραγματοποιηθεί κάτω από διαφορετικές συνθήκες.

Η απόδοση του αλγορίθμου αξιολογήθηκε βάσει των Detection Error Tradeoff (DET) και Receiver Operating Characteristic (ROC) καμπυλών. Παράλληλα, υπολογίστηκε και το ποσοστό ίσου σφάλματος (Equal Error Rate), το μέτρο Αποδοτικότητας και το εμβαδό της ROC καμπύλης. Πραγματοποιήθηκε αξιολόγηση του κάθε σεμιναρίου χωριστά και όλων μαζί. Επίσης, έγινε συνδυασμός δεδομένων εκπαίδευσης και ελέγχου του συστήματος από δύο διαφορετικά σεμινάρια. Με τον τρόπο αυτό παρέχουμε πιο αξιόπιστα αποτελέσματα. Καταλήγουμε ότι η χρήση των επιπλέον χαρακτηριστικών βελτιώνει αισθητά την απόδοση του κλασικού αλγορίθμου που χρησιμοποιεί μόνο τους MFC συντελεστές από 0.5% εως 20%. Συγκεκριμένα, παρατηρήθηκε ότι τρεις από τους πέντε συνδυασμούς ξεχωρίζουν, μειώνοντας κατά 20% την πιθανότητα του να χάσουμε ένα κομμάτι "φωνής", δεδομένης μιας πιθανότητας ίση με 5%, να χαρακτηρίσουμε ως "φωνή" κάποιο κομμάτι που στην πραγματικότητα δεν είναι.

Acknowledgements

First of all, I would like to thank my supervisor, Professor Yannis Stylianou for giving me the opportunity to work in his team. I would like to thank him for the motivation, the advices and his support during my studies.

I would also like to thank researcher Maria Markaki for her time and help during this thesis.

Lots of thanks to all the members of the laboratory, Maria, George, Olina, Dora and Vero and my closest friends Eleni and Haris for their help and support.

Finally, the persons to whom I would like to say the greatest "thank you" are my family. I thank my parents Suzanne and Stavros, and my brothres and sisters Emili, Melina, Vasilis and Giorgos for their patience, support, encouragement, motivation and for always being by my side in whatever situation I am dealing with.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Speech is one of the most important abilities that we have, since it is one of the principal ways of communication with the world. Our voice helps us to express ourselves and thus others can understand how we feel and what we want. In the past few years a lot of interest has been shown in developing voice-based applications. Such applications involve the isolation of speech from an audio file. This explains the necessity of being able to detect such parts in a given signal. The algorithms that achieve this are called Voice Detection algorithms. From the analysis of a given input audio signal, the parts containing voice are kept while the other parts (noise, silence, etc) are discarded. In this way a great reduction of the information to be further processed is achieved.

1.1 Background

Voice-based applications are widely used in the web, on smart phones and smart homes, in multimedia and generally in telecommunications and networks. We already have the opportunity to use this kind of applications in our daily life. It is a fact that the use of cell phones has increased more than ever before.A cell phone is mostly used while doing something else at the same time. For example when driving a car or walking on the street. In those cases, it is difficult to write something using the keyboard or the touch screen. It is useful to be able to carry out some operations using our voice. Further, many voice browsers and search engines have been developed, that take as input an speech signal. Such applications exist even in security systems, in which for example, voice identification is required for being able to enter a building. Voice-based applications can also be used for making life easier for people with disabilities. An application that works with voice commands can easily be used from a person unable to use its hands. It seems therefore, that the degree of tolerance to errors that we can have varies depending on the application. Of course we want to achieve the optimal performance in whatever application we develop.

However, in the last case where security is an important issue, we obviously demand to get reliable and precise results. Hence in some cases, if the results are not accurate enough the consequences could be significant whereas in other cases, they may not be that important. Another example is Voice over Internet Protocol (VoIP), where the goal is to transmit audio signals over IP networks. Before transmitting the signal, it has to be encoded and compressed to use the least possible bandwidth. One first step is to keep only the parts of the signal needed and discard the rest. After this, the signal can be encoded, compressed and be ready to be transmitted. Furthermore, we often want to process an audio signal, in order to do speaker recognition or identification, speech transcription, etc. The very first thing that we do, is to separate the signal into speech and nonspeech parts. Thus, we reduce the amount of information to be processed, by leaving out the parts being out of our interest. This reduction of the information is for some applications necessary to be done, especially for those used in VoIP. The requirements of applications are increasing and everything has to work faster but still produce reliable results. To be able to discriminate and keep only the needed information from the whole signal is an important step in order to achieve this.

When talking about keeping only the useful information from a given audio signal, we have to explain what we refer to. The kind of information considered as useful depends on the task. In all of the applications discussed earlier, the information that we need is the parts of the signal containing voice. This means that we want to analyse the audio signal and detect those parts. For the analysis we split the signal into smaller parts, which are called frames. The size of the frames depends on the application, but is constant during the analysis. It has to be big enough to be able to collect sufficient information about the audio content and at the same time small enough to be processed fast. An important issue is to choose the most appropriate features in order to get the optimal results. However, the type of features to be used, again varies from task to task. To be able to select the appropriate features, we need to clearly define our problem. As soon as we have decided what features we are going to use, we move on to the analysis. In each frame, the features are extracted and using a classification method a decision is made about the content of this frame. This way we achieve to separate the signal into labelled segments, knowing which of them are useful for further processing.

1.2 Motivation

The wide use of voice detection algorithms in all the fields we have seen above, lead us to study and develop new or improve existing algorithms and methods, that allow us fast

and, at the same time, efficient and accurate signal processing. When we want to retrieve information from an audio signal, we have to take into account the possible conditions during the recording of the signal. It is not always the case that we record the signal in a perfect environment. To be realistic, in most cases when recording an audio signal, other sounds are also recorded along with our voice. Apart from that, whether the recording contains noise or not, some parts of the signal to be processed will contain useless information. In our case speech parts of a signal are of our interest, so we need to have a way to detect such parts. This makes the task of voice detection very important and one of the most important front-end in audio signal processing.
The tasks of singing voice detection, speech/music discrimination and speech/nonspeech classification could be seen as subclasses of what we generally call voice detection. In singing voice detection, we try to detect the audio parts where someone is singing. On the other hand, when working on speech/nonspeech classification the goal is to discriminate speech from anything else. Singing voice has different properties than speech and this is why we study it separately. Speech/Music Discrimination is an separate task too. It is a case of speech/nonspeech discrimination in a certain environment, which contains music. There are methods, designed in order to discriminate speech from music or to detect singing voice. Also methods has been developed for speech/nonspeech classification. The goal of this thesis is to detect voice, meaning speech and singing voice, in a music environment. As we deal with real life recordings, they contain many types of noise. To efficiently detect voice in such environments we need to develop a method that combines features and/or methods from the tasks described earlier.

1.3 Contribution

This thesis contributes the following:

- presents a voicing detection algorithm that uses combinations of MFCCs, Cepstral Flux, Clarity and Harmonicity, which:
  - is able to discriminate both speech and singing voice from music and other noises.
  - demonstrates robustness in real environmental noises, achieving low scores in the Detection Error Tradeoff curves, on indoors and outdoors recordings.
- compares different combinations of the MFCCs with the other three features and concludes to the use of all features for the best performance.

1.4 Structure of the thesis

In Chapter 2 we will present related work on voicing detection. Four categories will be examined, which are Voice Activity Detection, Speech/Nonspeech classification, Singing Voice Detection and Speech/Music Discrimination. Some methods and the features being used will be described. Thus, we will get a general idea of how we deal with such problem usually.
In Chapter 3 the proposed algorithm will be analysed step by step. We will present the algorithm in pseudo-code, so to have a better overview. Also, a description of the features that have been used will be given.
Then in the Chapter 4, information about the database, which has been used will be given. Finally, the performance of the algorithm will be evaluated in Chapter 5 and results will be presented. We will refer to each combination of the features that has been done separately. In Chapter 6, conclusions will be presented and future work that can be done on voicing detection is going to be discussed in the end.

Chapter 2

Related Work

The task of voicing detection can also be seen as a speech/non-speech classification problem. This thesis aims to detect voice in a music environment and to be more specific, during a music lesson. Consequently, as we have already mentioned, apart form speech it can contain singing voice too. This means that it could also be approached as a speech/music discrimination problem as well as a Singing Voice Detection problem. Methods that solve this kind of problems usually consist of two parts. Firstly, features are extracted from the signal and then a classification takes place. For the first step, we have to study the behaviour of features and choose the ones that are able to discriminate voice from other sounds the best.

In speech processing, when dealing with this kind of problems we analyse the signal in frames. This means that we examine a small part of it each time. The size of the frame depends on the problem and the features being used. Some features need to be computed over a larger period of the signal than others. So the duration of a frame varies from 20ms up to a few seconds (2-3 secs). The first thing that we do when having a task, is to define the goal that we want to achieve. In our case the goal is to detect voice. According to this, we then try to choose the most appropriate features for detecting voice. In most cases more than one features is used for the detection. There are features that detect successfully particular characteristics of voice. However, this is not always guaranteed in the various kinds of noises and environments. For this, it is necessary to combine features properly for being able to demonstrate robustness in noisy environments. This suggests that, the features to be used have to provide complementary information to each other, in order to achieve a good trade-off between low computational cost but still having a good performance. In order to evaluate the performance of an algorithm, most of the times we follow two steps, the training and the testing step. So, we first train the system that was built using a part of the database, mostly more than the half and then test it with the remaining data. Otherwise, the algorithm is directly applied on all the data provided.

As we mentioned above, this thesis is relevant to tasks like Voice Activity Detection (VAD), Speech/Nonspeech classification, Signing Voice Detection and Speech/Music Discrimination. Features are needed that detect not only speech but also singing voice. In addition, the discrimination between voice and music is desired, which can be seen as a specific type of noise. So, we will start by describing some methods and features for VAD and Speech/Nonspeech classification. After this, features for Singing Voice Detection will be analysed and in the end we will refer to features that are used for Speech/Music Discrimination. Also, some classification methods will be mentioned, that are usually used in those tasks.

2.1 Voice Activity Detection and Speech/Nonspeech classification

Voice Activity Detection is defined as a binary classification problem. We have an input frame and the goal is to estimate whether it contains speech or not. An example of the output that an VAD algorithm gives, can be seen in Figure 2.1. It is widely used

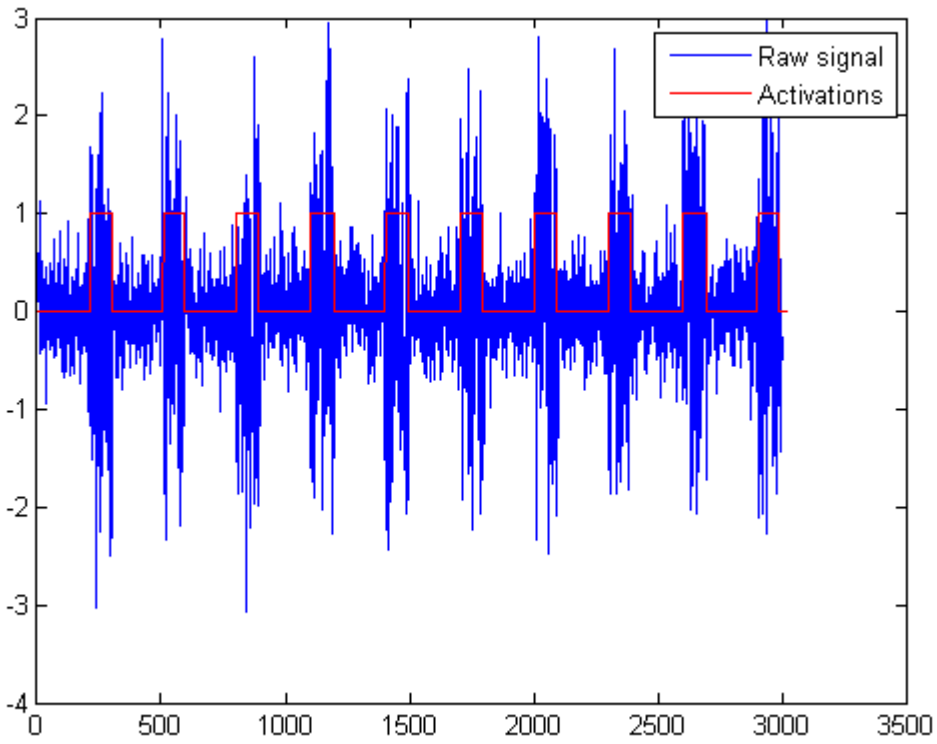


Figure 2.1: Example of Voice Activity Detection, taken from [33].

in many voice-based applications using speech recognition, speech enhancement, speaker recognition and speech coding [1]. Speech/Nonspeech classification is actually very similar. Frames or segments are classified into speech or nonspeech. This is very useful when having large audio signals and we only want to proceed the speech parts. There are many methods proposed for solving this kind of problems. Some of them use simple measures like the frame energy [2], zero crossing rate or autocorrelation function based features, which are computed in time domain. Features extracted in frequency domain are also very often used. For example spectrum based features or Mel-frequency cepstral coefficients. In most cases, a combination of features from both domains is done.

In [2], Moattar, Homayounpour and Kalantari propose an VAD algorithm that uses four short time features. They combine the short-term energy of the frame with the Spectral Flatness Measure (SFM) and the most dominant spectral component as it was proposed in [3]. Apart from these features they use another one, which is called peak-valley difference (PVD). Energy is the most common feature being used in such tasks. However, in noisy conditions and especially in low SNRs energy alone does not work efficiently. As we know, the energy of a frame is computed as the sum of it's squared amplitudes. The computation of the SFM gives an estimation of how noisy the spectrum is. It was observed that apart from SFM, the most dominant frequency component is also good in detecting speech. The computation of the last is very easy and not at all time-consuming. The algorithm was tested on data from TIMIT and Aurora2, after adding different types of noises (white, babble, pink, factory and Vovlo) with 25, 15, 5 and -5 dB Signal to Noise Ratios (SNRs).For the evaluation the average of Silence Hit Rate (HR0) and Speech Hit Rate (HR1) is computed. The scores are between 65%-96% for the various SNR levels independent from the noise type.

The main problem that someone has to overcome in VAD is noise. Many energy-based methods ([6]) do excellent in high quality recordings, but when noise exists the performance is dramatically degraded. Noisy parts of a signal have energy as well as voiced parts do and due to this fact methods like these, produce a significant number of false alarms. Based on this, Chuangsuwanich and Glass propose in [5] the use of two features that are able to measure fundamental attributes of speech: Harmonicity, which is an autocorrelation based feature and Modulation Frequency (MF) that is computed in the frequency domain. Periodicity is one of the main characteristics of speech. By computing the autocorrelation of a speech signal, those repetitions are captured. The two features were evaluated on noisy data (from -5 to 15 dB SNR) where engine, street, background talking and environmental sounds are expected to be encountered. Harmonicity on its own gives an EER equal to 12.93% and the combination of the two reduce the EER to about 4%. An Support Vector Machine (SVM) is used for the classification, but first a Neighbourhood Component

Analysis is applied. Autocorrelation features are also examined by Kristjansson, Deligne and Olsen in [7] for voiced, unvoiced and nonspeech discrimination. In time domain the Maximum Autocorrelation Peak, the Autocorrelation Peak Count and the Maximum LPC Residual Autocorrelation Peak are studied. Also, the Windowed Autocorrelation Lag Energy is introduced as an extension of the Maximum Autocorrelation coefficient. Whereas in frequency domain the Spectral Autocorrelation Peak Valley Ratio (SAPVR) is described. Apart from those, Spectral Entropy and Cepstral Peak are also presented which are well known in this area. The 3 features performing best are the Cepstral Peak, the Maximum Autocorrelation Peak and the Windowed Autocorrelation Peak, which are reported to provided complementary information to the Mel-Frequency Cepstral Coefficients.

Others assume that the most significant information to detect voice in noisy conditions is on the time-varying signal spectrum magnitude. In [4] an adaptive algorithm that estimates the Long-Term Spectral Envelope (LTSE) is proposed. After this, the Long-Term Spectral Divergence (LTSD) between speech and noise periods is computed and compared to an adaptive threshold. The evaluation was conducted on data from TIdigits database, for clean conditions and for SNRs ranging from 20 to -5 dB. The proposed method gives a HR1 of 98.15% while the HR0 is equal to 47.28%. There is a great divergence between the values of HR1 and HR0, so it can be used for application that do not require a low false alarm score, but need a low miss probability.

In the algorithm proposed in [9], three out of five features that are used are based on the autocorrelation function of the signal. These are the Harmonicity, the Clarity and the Prediction Gain, which are computed in time domain. The other two features are Periodicity and Perceptual Spectral Flux, which are in frequency domain. Harmonicity is defined as the relative height of the maximum autocorrelation peak and Clarity as the relative depth of the minimum average magnitude difference function. The Prediction Gain is the ratio of the signal energy to the linear prediction residual signal energy. For the computation of Periodicity, the Harmonic Product Spectrum (HPS) technique is used [8], which is used for pitch detection in noisy environments. Periodicity captures the harmonics of the pitch frequency during voiced and speech-like segments. The Perceptual Spectral Flux is based on the observation that speech has a lower rate of frame-to-frame changes than music does. All the features described above, are combined into one vector and using Principal Component Analysis (PCA) a 1-dimensional feature is obtained, which is used for the discrimination. The results show that the algorithm performs quite well. In addition, each feature is evaluated in terms of P_{miss} (miss probability) for a P_{fa} (false alarm probability) of 3%. The two features performing best are the Clarity and the Harmonicity, achieving a P_{miss} of 7.43 and 10.76 respectively.

Another widely used feature in speech processing tasks, is the MFCCs. As we will see

in the following section, these features perform quite good in Singing Voice Detection. In [1], a voice detection algorithm is proposed using MFCCs with delta and double-delta coefficients. This features does not depend on the energy level of the signal, so it will work well even in cases where energy-based features will fail. The main advantage of the proposed SVM-based VAD is that it works consistently although using different training data. According to the experiments it is concluded that the SVM is easier to adapt to new data sets, as long as there is a short audio sample from the recording environment.
As it has already been mentioned VAD is actually a Speech/Nonspeech classification problem. Consequently, the features that are used for Speech/Nonspeech discrimination are similar to those described earlier. Many speech/nonspeech classification methods have been proposed, especially for endpoint detection, that use short-time energy features and zero crossings rate [12]. But, the efficiency of such features decrease in low SNR conditions due to the fact that noise contains a lot of energy like speech does. In [10] some frequency domain features are proposed like, LPC residual energy and the energy of certain bands in the spectrum, which are similar to those described earlier for VAD. Following, features that are used in Singing Voice Detection (SVD) will be presented.

2.2 Singing Voice Detection

The task of identifying singing voice in an audio signal, does not seem to be hard for humans, even if we are not familiar with the style of the music, the particular singer, or even the language. This does not mean that it is easy to be done automatically. Nevertheless, a method being able to do this can be applied in many applications like singer recognition and identification, audio segmentation, vocal extraction and language detection. Again this task can also be seen as a classification problem. The audio signal is analysed into segments and the goal is to classify each segment as vocal or non-vocal. Many researchers have studied and proposed features that perform well in detecting signing voice ([13], [14], [15], [16], [17]). Techniques and features used in speech detection or recognition can also be applied to singing voice problems due to the similarities that speech and singing voice have.
The feature that is used the most and performs quite good, is the Mel-Frequency Cepstral Coefficients (MFCCs). In [14] it is shown that by using only MFCCs, appropriately parametrised along with their first derivatives is sufficient to achieve results as good as those that we get with more complicated state-of-the-art systems. Rocamora and Herrera [16] studied and compared MFCCs, Perceptually derived LPC (PDLPC), Log Frequency Power Coefficients (LFPC), Harmonic Coefficient (HC) and pitch. Additionally, they combined into one vector spectral features, namely Centroid, Roll-off, Flux, Skewness, Kurtosis

and Flatness. According to this study, MFCCs are the most suitable features for SVD. They also report that the classifier performing best for this task is Support Vector Machines (SVM). However, there are others that prefer using Gaussian Mixture Models (GMMs) as a classifier [17], while using MFCCs, log energies, modulation energy and harmonic coefficient as discrimination features. The evaluation of the proposed method gives an EER of 14%, without applying post-filtering. Again in [13], a signing voice detection method is proposed that uses MFCCs and modulation frequency features and classifies the segments using an SVM. The system was tested on Greek music classifying segments into 4 classes, a) instrumental only, b) voice of target singer without 2nd voice, c) voice of target singer with 2nd voice and d) interjections. The evaluation of the performance of this system gives an EER equal to 12.06%.

Moreover, in [18], Regnier and Peeters approach the problem from another perspective. It is supposed that singing voice is characterized by harmonicity, formants, vibrato and tremolo. So, features are chosen that are able to describe the particular characteristics. In the proposed approach, frequency modulation and amplitude modulation is used, which describe vibrato and tremolo respectively. In the end, a post processing (e.g median filter) of the segmentation is applied in order to remove short-duration segments. The proposed method is compared to a learning machine approach that uses MFCCs, Δ MFCCs, $\Delta\Delta$ MFCCs, SFM, Δ SFM and $\Delta\Delta$ SFM and a GMM classifier. Before the filtering the learning machine approach is much better than the proposed one, having a difference between 5-14 % in terms of Recall, Precision and Fmeasure. After the filtering, it is noticed that the proposed method gives a better recall whereas the learning approach is still more precise. We observe that the filtering does not help significantly the learning approach as the scores increase only 2%. In opposite, an improvement of about 10% is achieved for the proposed method. Vibrato combined with Harmonic Coefficient is also proposed in [19]. A comparison is made between the proposed system and a classic one based on MFCCs and GMM. It is concluded that, although the results that we get are comparable, the system using MFCCs is still better than the proposed one.

So, the common way to deal with an SVD problem is to extract feature parameters from the input signal and then classify them using a threshold or a statistical classifier. The signal is usually proceeded in frames and the decision is taken whether for each frame or for a block of frames (segment). When using a threshold for the classification, the descriptors being used must be able to clearly discriminate between the different classes. Methods using thresholds apply one threshold on one descriptor or compute more than one descriptors and apply a set of thresholds on them. A combination of several features can be done when statistical classifiers are being used. So the system can be trained and is then able to learn more complex boundaries between the classes. The most often used

classifiers are SVM, GMM, and Hiden Markov Models (HMM). By training the system and combining more features, more time is needed in order to complete the classification, but still the performance is enhanced. If having a task that has no restrictions on time, a statistical classifier could be used achieving better performance.

2.3 Speech/Music Discrimination

In applications of multimedia information retrieval, effective coding for telecommunications and Automatic Speech Recognition (ASR), audio signals need to be segmented and classified, so that each segment can be used in a different way. In the first section, we discussed about detecting voice in a given audio signal. As already said, most of the times the input signal contains noise. The task of discriminating speech from music, can be seen as a special case of voice detection-classification, by considering music as noise. However, music has particular characteristics that are very similar to those of speech, for example the harmonicity. This is why we study Speech/Music discrimination separately from VAD and Speech/Nonspeech Classification. The features being used in such methods, must be able to distinguish speech from music despite the similarities.

Scheirer and Slaney examined thirteen possible features that can be used in a speech/music discrimination system [20]. In their system they used zero-crossing rate, spectral rolloff point, centroid and flux. Also, cepstrum resynthesis residual magnitude is used as well as the 4Hz modulation energy, percentage of low-energy frames and a novel features called pulse metric. Pulse metric determines the amount of "rythmicness" in a 5-second window. Based on the fact that speech tends to have more modulation energy at 4Hz, this energy is computed, using MFCCs and is then used for the discrimination between speech and music. Each one of them was intended to be a good discriminator on their own, but not all of them end up adding value to a multivariate classifier. This proves that when using a combination of several features, even if each feature on his own does well this does not guarantee the optimal performance. The combined features need to provide complementary information to each other, so to achieve the optimal combination. They report that the "best 3" features are the 4Hz energy, the variance of the spectral flux and the pulse metric by giving an total error (frame-by-frame error) of about 5.8%.

In [21] again a spectral feature is proposed, that detects the curved frequency trajectory of the harmonics over a certain period of time. This is based on the observation that in speech, the harmonics are sustained over a certain span of time in which they usually vary in frequency. Whereas music parts of the signal consist of partials with a relatively constant frequency. The spectral of a noise frame does not contain significant peaks that are sustained over time. The results after the evaluation seem to be good, but it is reported

that there is still some room for further improvement. Fu and Wang claim in [22], that cepstrum analysis is a more powerful tool than spectrogram for analysing the details of spectrum. So two novel features are introduced, based again on the differences in the pitch between music (discrete frequencies) and speech (pitch changes continuously) and also the peak values of the real cepstrum. The two features are the Average Pitch Density (APD) and the Relative Tonal Power Density (RTPD). The novel features are combined with: 1) log od variance of RMS, 2) log of variance of spectral centroid, 3) log of variance of spectral flux, 4) 4Hz modulation energy and 5) dynamic range. Evaluation and comparison of this combination to the MFCCs + delta + acceleration shows that the latter have a good ability for discrimination on one of the two data sets, while the combination performs better on the other data set.

The importance of spectrum in all these tasks is obvious as many researches propose features extracted from there. Like in SVD so in Speech/Music discrimination, a feature that is very often used due to its efficiency is MFCCs. That is why, in many works the evaluation is made by comparing the performance of the proposed method to the performance of the MFCCs. Kim, Choi and Lee compare in [24] the efficiency of using the spectrum based Modulation Energy (ME) and the Mel-Cepstrum Modulation Energy (MCME). The experiments show that MCME at 8Hz perform better than 4Hz ME. This proves what Fu and Wang claimed in [22] about cepstrum and spectrogram. Some researchers consider MFCCs as the main feature and try to find other features that can be combined with MFCCs so to get the optimal performance. In [23] two novel features are proposed that can be concatenated with MFCCs. The Delta Cepstral Energy (DCE) is introduced, that measures the energy variation of the signal over time. It is observed that speech has greater energy at low frequencies, unlike music which has also a significant amount of energy at higher frequencies. Of course this depends a lot on the kind of music that we have in our signal. The Power Spectrum Deviation (PSDev) is computed as the standard deviation of filter bank energies in each band in order to discriminate between speech and music. The speech error rate (frame-by-frame) in which we are interested for our work is equal to 6.13% when using MFCCs (12 coefficients), while 6.41% when combining MFCCs with DCE and PSDev. The results seem to be very close, but MFCCs alone still perform better. Moreover, in a recent work [11] a speech discrimination algorithm is presented that uses modulation frequency features. As MFCCs is the state-of-the-art in this kind of tasks, a 12th-order MFCCs was chosen to be combined with the log energy and their first and second differences. Although the general approach for the classification is maximum-likelihood with GMMs, SVM was preferred for the experiments. In order to reduce the dimensionality of the feature subspaces, High Order Singular Value Decomposition (HOSVD) was performed, which is a costly process. The performance of MFCCs+Δ+$\Delta\Delta$ was evaluated,

as well as a fusion of the cepstral with the 21 most relevant features that gave an EER of 4.79% and 4.45% respectively. It seems that the proposed method provides slightly better results than the cepstral features does, apart form the cost of HOSVD and the complexity of deriving the final feature combination.
In general, when dealing with Speech/Music Discrimination tasks the features being used, can be separated into time domain, spectral domain and cepstral domain features. The most commonly used features are the spectral and cepstral domain features and specifically the MFCCs. All the features are estimated in frames of 0.5-5 seconds. The duration depends on the type of the feature and of the specific task working on. As for the classification stage, the classifiers that are usually used are again GMM, HMM, and SVM. It is observed that the performance of an SVM-based system was more consistent or even better than GMMs based on the same cepstral features [1].

2.4 Discussion

As it has been shown above, there are plenty of features that can be used for voice detection and classification. Some features are ideal for detecting specific properties of a speech signal ([21], [20]), but need other features to be able to work successfully in detecting or classifying speech parts of an audio signal. When dealing with voice detection tasks, we have to take into account the various conditions under which the audio signals have been recorded. It is almost impossible to carry out a recording having no additional noise. Besides this, many applications require efficient voice detection regardless the conditions in which they are used, e.g. in cafeterias, in commercial centers, at home or even on the street. Consequently, an algorithm being developed for this purpose needs to show robustness in such recording conditions. Many existing algorithms use energy based feature, but their performance is degraded a lot in noisy conditions. In SVD and Speech/Music Discrimination the most commonly used features are extracted from the frequency domain. A feature that performs quite well and is widely used in both tasks is the MFCCs ([13], [16], [14], [11], [23], [24]). The goal of this thesis is to improve the performance of a classic approach that uses MFCCs as the discrimination feature. So we actually want to use features that provide complementary information to the MFCCs so to improve the performance. We examine here, are Harmonicity and Clarity that are referred to perform good in VAD([9]) and Cepstral Flux that contain useful information for the discrimination of speech and music. Based on the observation made in [1] the SVM classifier will be preferred for the classification in our approach. In the rest of the thesis, a definition of these features will be given and the implementation of the whole algorithm will be described. Furthermore,

we will evaluate each feature combination that was made with the MFCCs and the results will be presented.

Chapter 3

Voicing Detection Algorithm

In this chapter, the features used in the algorithm will be described and their mathematical definition will be given. Also, the algorithm is going to be analysed, explaining each step. A description in pseudo code will also be presented.

3.1 Features

In the previous chapter, features that are usually used in VAD were presented, Speech/Nonspeech classification, SVD and Speech/Music Discrimination. As already mentioned, MFCCs is the state-of-the-art feature in the last two tasks and is also used in the other two. This particular feature is claimed to provide the best performance in the majority of the cases and due to this it is widely used [15], [16], [13], [11], [22], [1]. In this thesis MFCCs are considered as the main feature combined with Cepstral Flux, Harmonicity and Clarity. Each feature is computed separately and then concatenated with the MFCCs, into one global feature vector.

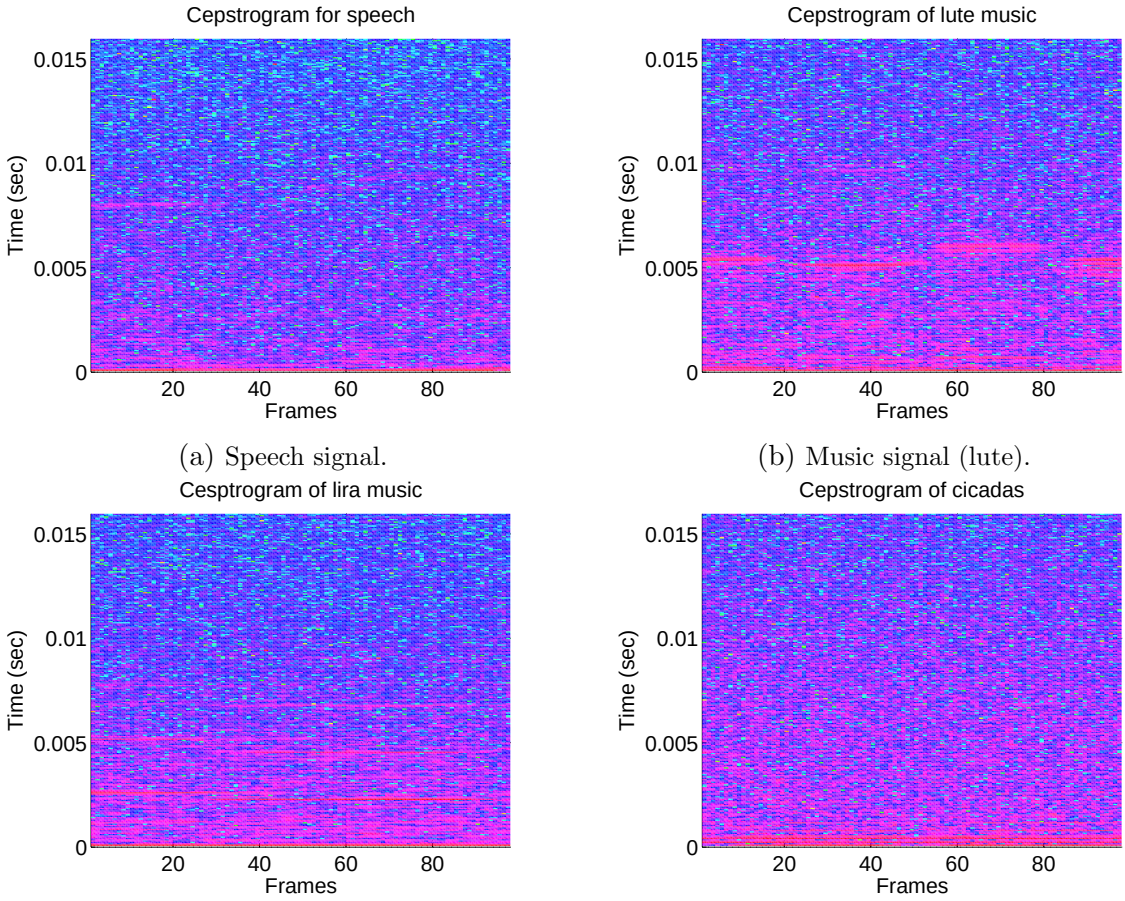
The feature vector based on the MFCCs consists of 12th-order Mel frequency cepstral coefficients containing also the DC component, which is actually the 0th coefficient. The Matlab implementation by Daniel Ellis is used [25]. Specifically, 13 coefficients are derived from 40 mel scale frequency bands for each segment. The magnitude spectrum is obtained through a Fast Fourier Transform (FFT). The energy is computed on each band by processing the magnitude spectrum with a filter bank, whose frequencies are spaced according to the mel scale. After this, the logarithm is taken. Due to the high correlation of the energies of each band, a Discrete Cosine Transform (DCT) is applied to decorrelate the values and then obtain the MFCCs. As it is described in [11], equal-loudness pre-emphasis and cube-root intensity-loudness were applied for the computation of MFCCs according to [30], by setting the parameter 'usecmp' equal to 1. In [16] the authors concluded that the use of delta coefficients can improve the performance. According to this, we use also

the delta and delta-delta coefficients. So, we have a (39xN_c) matrix, containing the 39 coefficients, where N_c is the number of elements for each coefficient. After computing the mean and the variance of each coefficient contained in the matrix, we end up with a 78 element feature vector, with whom the following features will be concatenated.
The first feature that is combined with the MFCCs is called Cepstral Flux (CF). It can be seen as an extension of the Spectral Flux (SF), that is used in [9] and [20] . It is defined as the squared absolute value of the frame-to-frame amplitude difference of the real cepstrum. For the computation, we keep in a buffer the cepstrums of the N previous frames. So, considering C_i to be the real cepstrum of the i-th frame, we have:

$$CF_i = \sum_{n=i-N}^{i} |C_i - C_n|^2. \tag{3.1}$$

This feature detects how fast or slow the cepstrum of the signal changes over time. In [20] this feature is used for Speech/Music Discrimination based on the fact that music exhibits more drastic changes from one frame to the next, having a higher rate of change than speech. But this depends on the music style as well. It is possible that the cepstrum of speech changes faster when we compare it for example to the cepstrum of a classical music song. In Figures 3.1a, 3.1b, 3.1c and 3.1d the cepstrums for each frame for a 1 second signal are shown. It is observed that some coefficients for lira and lute maintain similar values for small periods, whereas for speech and cicadas this does not happen. Due to this behaviour, the value of cepstral flux is expected to be small for such periods in the particular music signals, but will produce a peak at the beginning and in the end of this period. As for the speech and the cicadas signal the cepstrum changes constantly from frame to frame. Consequently, the values of cepstral flux corresponding to music will have a larger variance than those for speech and cicadas.
In contradiction to the first two features, the following two are computed in the time domain. Harmonicity, also known as harmonics-to-noise ratio (HNR) is defined as the relative height of the maximum autocorrelation peak in the plausible pitch range [9]. The range of 62.5 to 500 Hz is chosen for human speech, which corresponds to the interval of [2, 16]ms. The lower limit is imposed by the frame length and the fact that it usually chosen so to cover at least about two pitch periods, for a reliable voicing estimation. It is computed as following:

$$h(t) = \frac{r_{xx}(t, k_{max})}{r_{xx}(t,0) - r_{xx}(t, k_{max})}, \quad k_{max} = \underset{2ms \leq k \leq 16ms}{\operatorname{argmax}} \; r_{xx}(t,k). \tag{3.2}$$



(a) Speech signal.

(b) Music signal (lute).

(c) Music signal (lira).

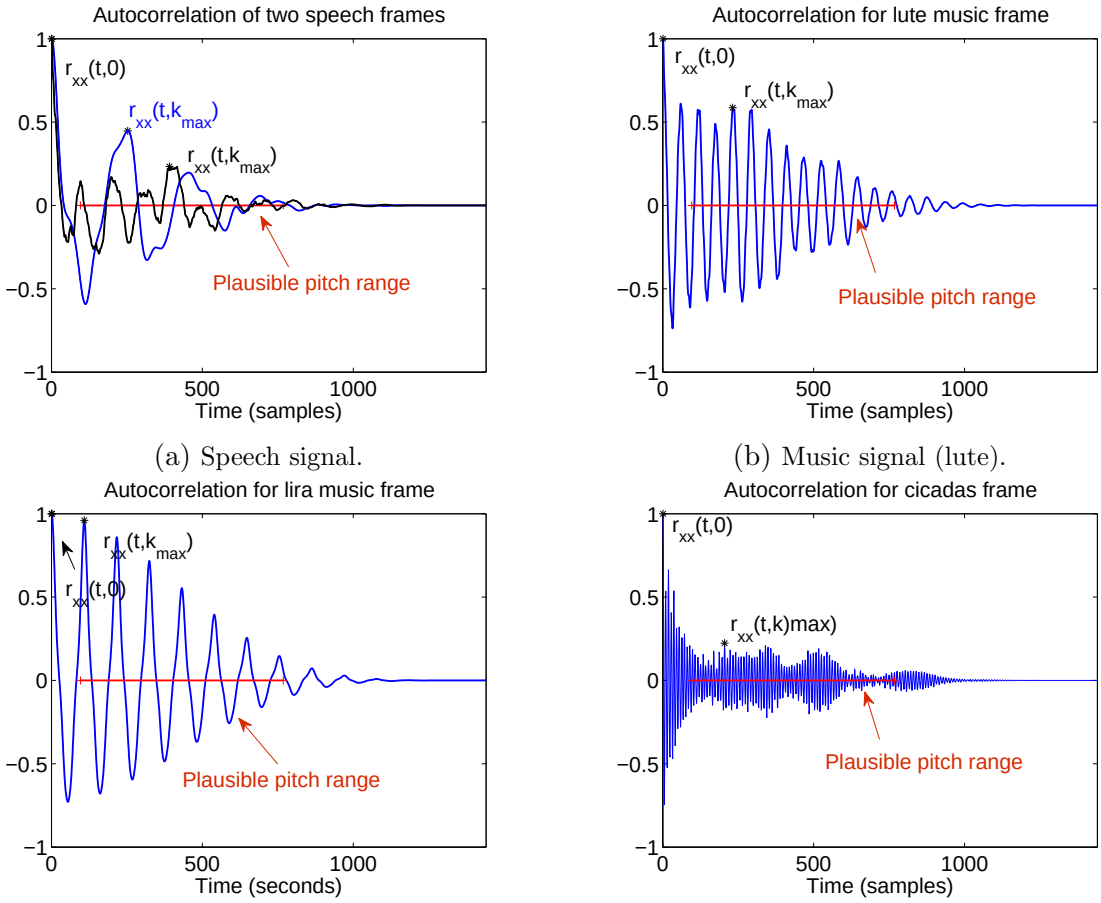
(d) Cicadas signal.

Figure 3.1: Examples of cepstrograms for different signals. The horizontal axis represents the frame number.

The autocorrelation is computed according to the next equation:

$$r_{xx}(t,k) = \sum_{j=0}^{N-1} x(j)w(j)x(j+k)w(j+k) \tag{3.3}$$

where $w(n)$ is a Hanning window and t and k are frame and autocorrelation lag indices respectively. In Figures 3.2a, 3.2b, 3.2c and 3.2d the autocorrelation for different kind of signals can be seen. Four signals that last one second has been processed in small frames, in which the autocorrelation was computed. The mean of all these autocorrelations is plotted. We can see the $r_{xx}(t,0)$ and the $r_{xx}(t,k_{max})$ corresponding to speech, lira, lute and cicadas. We notice that the correlation between these two values differ for each signal. The last feature that we use is also based on the autocorrelation of the frame and is called Clarity. It describes the relative depth of the minimum Average Magnitude Difference



(a) Speech signal.

(b) Music signal (lute).

(c) Music signal (lira).

(d) Cicadas signal.

Figure 3.2: Examples of autocorrelation for different signals. The horizontal axis represents time in samples.

Function (AMDF) valley in the plausible pitch range. It is defined in [9] as

$$c(t) = 1 - \frac{D(t, k_{min})}{D(t, k_{max})} \tag{3.4}$$

where,

$$k_{min} = \underset{2ms \leq k \leq 16ms}{\operatorname{argmin}} D(t,k) \quad \text{and} \quad k_{max} = \underset{2ms \leq k \leq 16ms}{\operatorname{argmax}} D(t,k) \tag{3.5}$$

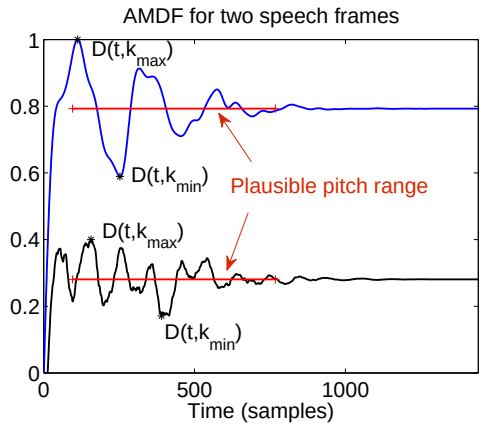
and $D(t,k)$ is the AMDF as mention above. It is costly to compute it from its exact definition, but it has been shown [28] that it can be approximately derived from the autocorrelation as

$$D(t,k) \approx \beta(k)\sqrt{2[r_{xx}(t,0) - r_{xx}(t,k)]} \tag{3.6}$$

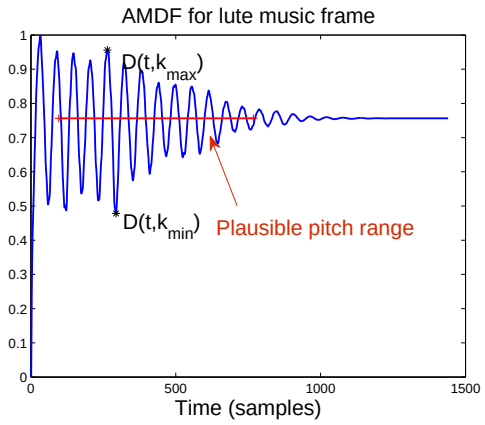
where $\beta(k)$ is a scale factor that varies between 0.6 and 1.0. It was found out that the value of this parameter does not significantly affect the clarity feature. So it was set equal to 0.6. The autocorrelation, rr_{xx}, is again computed as described in (3.3). The subtraction in (3.4) is just for converting the minimum to a maximum. Consequently, speech frames

will have large values whereas non speech frames will present small values. An example of the AMDF for various signals is presented in Figures 3.3a, 3.3b, 3.3c and 3.3d. Like in the previous figures what we see following, is the mean of the AMDFs of the frames for a one second lasting signal. The signals for the examples were chosen so, due to the content of the database that has been used (Chapter 4).

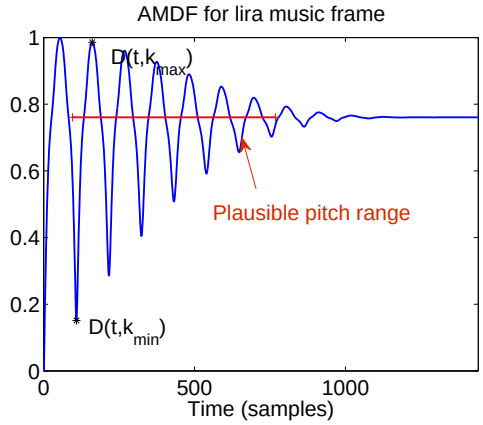
It is important to say that both features, the Harmonicity and the Clarity, are not influ-



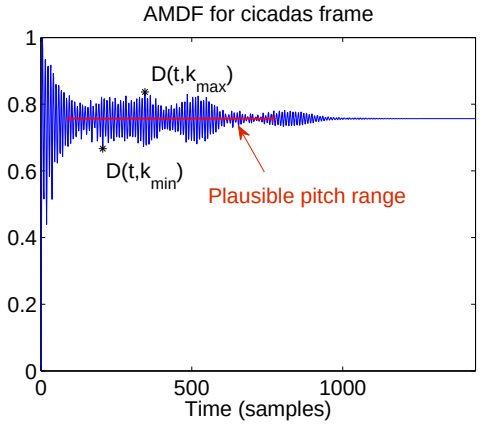
(a) Speech signal.



(b) Music signal (lute).



(c) Music signal (lira).



(d) Cicadas signal.

Figure 3.3: Example of AMDFs for different signals. The horizontal axis represents time in samples.

enced by the energy of the signal due to the way that we compute them. In other words, duplicating a signal's amplitude will give the same value as the original signal will give for both features.

In order to do the classification we used the publicly available SVM^{light} tool [26]. SVM is a binary classifier, that models the decision boundary between two classes as a separating hyperplane. The training set, given as input to the SVM consists of binary training vectors, containing +1 and -1. In our case the positive vectors, labelled as +1, correspond to speech feature vectors and the negative vectors, labelled as -1, correspond to non speech feature

vectors. A model is created that represents the decision boundary, which is mostly not linear, like for example in Figure 3.4. Then the classification is applied on the testing set, according this model. When evaluating a system by training it first, the accuracy of this system depends on how representative the training set is. So, it is very important to choose the appropriate training set. This way the features will be able to discriminate speech from non speech and give reliable results in different cases. In Chapter 5 we will see how results can differ when using different combinations of training and testing sets.

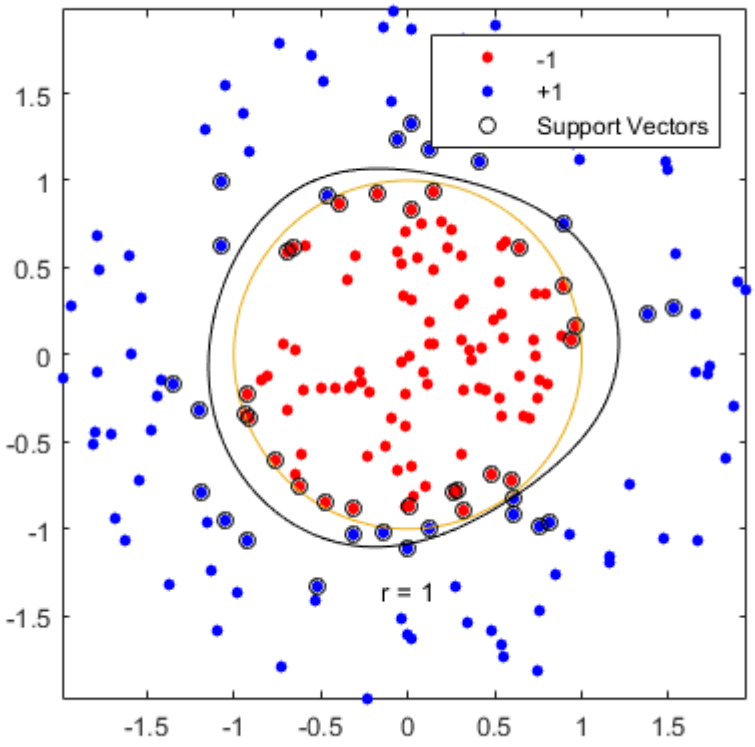


Figure 3.4: Example of SVM classification taken from [32].

3.2 Algorithm description

The algorithm aims at classifying segments of an audio signal into “speech” or “nonspeech”. The class “speech” refers to parts of the signal that contain both speech and singing voice. In opposite the class “nonspeech” refers to all the other possible “noises” (music, noise, silence etc).

The description of the algorithm in pseudo code is given below:

Feature extraction for each segment

- - Define frame and step size
  - Initialize feature vectors and CF buffer
  - Compute number of frames Nfr
- For k from 1 to Nfr
  - Compute the autocorrelation of the frame
  - Compute Clarity, Harmonicity and Cepstral Flux
  - Compute MFCCs, deltas and double deltas
- - Take median and variance of features
  - Subtract mean and divide with standard deviation
  - Combine all the features into one vector

Classification

- Define K equal to 10 (10-fold cross-validation)
- For k from 1 to K
  - Split data into training and testing sets
  - Train the SVM classifier
  - Test the remaining data
- Store the results

The two classes were defined so, due to the content of the database as we will see in the next chapter. It consists of 3 seminars on traditional Cretan music. Consequently, the audio signals may contain speech, singing voice, music and other types of noise too. A listener might search the recordings under different aspects. He might be interested in those instances where music is played. On the other hand, he could be interested in the parts where the teacher speaks, or a discussion is made between the teacher and the participants. Therefore, it is very useful for the listener being able to get the important parts without consuming much time. In this thesis, we focus on the second case where voice is in our interest. Either way, classification must be applied in order to discriminate the different parts. When choosing to apply short-term classification (classifying each frame separately), the information that is considered is local. This makes the classification prone to errors, having many changes from one class to the other. For this reason, classification is usually smoothed by splitting the signal into segments and assigning the same class to the whole segment. The segments consist of many frames and so long-term information

is introduced [29]. In [31], Marolt presents a probabilistic approach of segmentation of recordings containing music. He emphasizes the low priority of temporal accuracies of the boundaries and states that a few seconds is a sufficient accuracy when determining the boundaries. So, 3 second long chunks are classified into one of the classes defined. According to this work, we define the length of a segment to be 3 seconds. Additionally, the duration of the segment to be classified has also to do with the content of the audio signals being processed. We assume that the parts in which we are interested can last at least 3 seconds. For example, when the teacher explains something during the seminar he will not say it only using one sentence, but more. Thus, the goal is to classify these segments according to the classes defined previously ("speech" and "nonspeech").
For the analysis, each segment is split into smaller frames, in which the feature extraction takes place. The decision to be made is then addressed to the whole segment. The length of each frame is 30ms and there is an overlap of 20ms between them. For each frame a Hanning window is applied and then the features are extracted. The Clarity and the Harmonicity are computed using the autocorrelation of the current frame, according to the equations (3.4) and (3.2) respectively. For the computation of the Cepstral Flux, we use a buffer of size N = 15 frames. So, for each frame the feature CF is computed as the squared sum of the difference between the current frame and the 15 previous frames, divided by the length N of the buffer (Eq. 3.1). This produces one value per feature for each frame. As already described, the MFCCs are extracted using the implementation of Daniel Ellis [25]. The frame size is 30ms and an overlap of 20ms is defined similar to the other features. Also, the deltas and double deltas are computed for more efficiency. Then, the median and variance are computed for these three vectors (MFCCs, deltas and double deltas) and put into the feature vector to be later used. This produces an MFCC based vector of 78 elements.
Using only one value for the Clarity, one for the Harmonicity and one for the Cepstral Flux to represent each segment, does not provide enough information for the classification. What actually happens is a smoothing of the real feature vector and there is a large possibility to miss classify a segment. Through the feature extraction previously applied, a vector is produced for each feature, which contains as many elements as the number of frames are. Since the length of a segment is defined to be 3 seconds and the frame size 30ms with a step of 10ms, the total number of frames is 298. This is computed using the following formula:

$$\text{Number of frames} = \frac{\text{Length of segment} - \text{Length of frame}}{\text{Step size}} + 1 \tag{3.7}$$

In order to avoid this kind of smoothing, more than one value is required for representing

one segment. After trial and error, it was found out that it is enough to use 12 values, for the representation of each segment. So, we down sample the feature vector down to 12 from the 298 values that we initially had. This means that we have to group together every 24 frame values. We then take the median and the variance of those 12 values to end up with 24 values for each feature per segment. For the current application the standard score, also known as z-score was computed for each feature vector separately. According to its definition, for each vector the mean was subtracted and then a division was applied by its standard deviation. Doing this, the mean is going to be 0 and the standard deviation equal to 1. By computing the z-score for a vector, the amplitudes will change but the "envelope" of it will remain the same. We have seen that the MFCCs are often used for speaker recognition due to the fact that they vary from speaker to speaker. The thing in which they differ in this case is the amplitude (energy) of the MFCCs. But when dealing with speech in general, the structure of them is similar regardless the speaker. Normalizing the MFCCs in the way described previously, the values of the amplitude change, but this does not concern us since speaker recognition is not in our interest. The correlation between the coefficients and their structure do not change and this is very important for our task. Relying on this fact we use this features to discriminate voice from other sounds. After this, we concatenate all the previously computed feature vectors into one, which will be used for the classification of the segments. So a 150 element vector is finally extracted for each segment, 78 values for the MFCCs, 24 for Cepstral Flux, 24 for Harmonicity and 24 for Clarity.

After we have finished with the feature extraction for the given data, the classification must be performed. An SVM classifier needs to be trained and then tested. As described in 3.1, the training and testing vectors are defined to be binary, containing +1 for class "speech" and -1 for "nonspeech". So a 10-fold cross validation is applied on the database , which will be described in the next chapter. This means that the data are split into 10 disjoint subsets, from which one is kept for the testing and the other 9 subsets are used to train the classifier. A description will be given in Chapter 5 (Evaluation), of the various data combination that have been made for the training and testing. As it is mentioned in the manual of SVM^{light}, the RBF kernel is selected. During the training the support vectors are created, according to which the testing data will be classified. The evaluation of the results produced will take place in Chapter 5 presenting plots with the Detection Error Trade-off and Receiver Operating Characteristic curves, as well as tables with the scores (Equal Error Rate, Efficiency and Area Under the Curve) and the corresponding diagrams.

Chapter 4

Description of the database

As it has been mentioned before, the goal of this thesis is to develop an efficient algorithm that is able to detect the voicing parts of an audio signal. In order to do that we used a database, obtained from the Phonogramm Archiv in Vienna [27], which will be described following. The database consist of three different seminar recordings that have been carried out in Crete. As we will see there are different environmental settings as well as different instruments. Although the recordings are available in high resolution video, we extracted and used only the audio track for this thesis. The length of the recordings varies from 31 minutes to 62 minutes. In each seminar there is a person(renowned master) that teaches how a specific musical instrument is applied on a certain repertoire of Cretan folk music. In Table 4.1 some basic information can be seen about the recordings.

	Video 1	Video 2	Video 3
Instrument	Lira	Lira	Lute
Duration	62m	52m	31m
Location	Houdetsi	Meronas	Meronas
Date	April 2011	August 2011	August 2011
Environment	Class room	Class room	Outdoors

Table 4.1: Basic information about seminars

We considered two classes, speech and nonspeech. Consequently, silence, music and noise parts are labelled as nonspeech. We chose to do so because we are just interested in detect the voicing parts of the signal and keep them. Although we name the first class speech, singing voice is also included in this class.

4.1 Video 1: Lira seminar

The first seminar was held in a class room, in a small village in Crete called Houdetsi in April of 2011. The video was recorded on tape with a Sony DSR-PD150P camera, using the internal microphones of the camera. It was compressed just for the uploading, probably with Adobe Premiere. The teacher and the students play a traditional Cretan instrument called lira. It is a stringed instrument, which is played with a bow and is similar to a violin. The video camera, on which the microphone is, is placed between the teacher and the students. So there is a distance between the microphone and all the persons being there. When someone speaks, we can hear something like an echo, probably due to the size of the room. There are parts in which the teacher only speaks and nobody plays the instrument, but there are also parts where one or more speak and at the same time we can hear someone playing. In parts where the students play all together, we notice that they are not synchronized at all. A small proportion of the parts of the signal that contain speech correspond to singing voice. After finishing the annotation of the audio signal, it was observed that the most parts that were annotated as "Speech", contained a lot of noise. This means, that by evaluating our method on this data the results will be reliable, due to the rough recording conditions. If we get good results by testing the algorithm on these data, we expect to get even better results when testing it on less noisy data.

4.2 Video 2: Lira seminar

The next two seminars were held in the village Meronas, which is also in Crete, in August 2011. The first one was recorded in a class room and the instruments that we hear are again liras. For the recording a HDR-FX7E Sony camera was used. The internal microphones were used in this case too. The video was compressed so to achieve a 720p format. The Final Cut Pro was used for this purpose, using an export to Mpg4 with the following settings:
Quality: 50%
Compression: H.264
Data rate limited to 2 Mbps
Sound Mpeg-4 AAC Stereo Automatic 256 kbps
Frame size: 1280x720 (HDTV 720p)
The camera is closer and although it is not right in front of the teacher, we can hear him better than in the first seminar. It is worth mentioning that in this particular seminar the teacher is singing more often than in the other seminars. The windows of the classroom

were open during the seminar so noise from outside (cicadas) was also recorded. Still, it was noticed that when someone speaks there is not much noise, e.g. by others playing the instruments. So the parts of the signal that were annotated as "Speech" are clearer. That is why the results that we will get by evaluating the method with these data will not correspond to the general case and the method is probably going to be overestimated.

4.3 Video 3: Lute seminar

The last seminar was about playing the lute. The lute is also a stringed instrument and is played like a guitar so no bow is used. It is important to mention that during the whole seminar cicadas can be heard, which produce a noise in high frequencies. The microphone is in distance but close enough so we can understand what is said, although the sound of the cicadas is loud. The teacher is trying to teach how a particular song is been played with the particular instrument. He asks from each student to play on his own and sometimes he sings the notes at the same time in order to help them. There are also parts in the signal where they play all together with the teacher giving the rhythm. So again, there are parts here in which someone speaks and others play, or only music or speech can be heard.

To be able to use these data and validate the results of the algorithm, each seminar was segmented and annotated manually. The free available audio editor *Wavesurfer* was used for this purpose. As already described, two classes were considered, "speech" and "nonspeech". The manual segmentation was not performed according to the definition that was given in Section 3.1, about the duration of a segment. Parts of continuous speech and singing voice where labelled as "speech", whereas the remaining parts where labelled as "nonspeech". After finishing with the annotation of all the seminars, these large chunks where split into 3 second lasting segments with a Matlab script. Segments with durations less than 3 seconds were discarded, as they do not agree with the predefined conditions of the algorithm. In the following chapter, the results will be presented and the performance of the algorithm will be evaluated.

Chapter 5

Evaluation

In this chapter, the results will be presented and the algorithm will be evaluated. The detection error trade-off (DET) curve is used as the evaluation tool. The DET curve shows the miss probability (P_{miss}) as a function of the false alarm probability (P_{fa}) on a normal deviate scale. The P_{miss} refers to the probability of the algorithm to miss classify a voice segment. In opposite, P_{fa} is the probability of classifying a segment that does not contain voice as a voice segment. Depending on the application, one of the two errors can be considered less significant. So the threshold can be adjusted properly to get the desired result. The number of false alarms can be reduced by defining a higher threshold, at the cost of increasing the number of missed voice segments. The lower this curve is the better the system is. We also compute the Equal Error Rate (EER), which corresponds actually to the threshold for which P_{miss} is equal to P_{fa}. Another measure that was used for the evaluation is the Receiver Operating Characteristic(ROC) curve. The True Positive rate (TP_r) is plotted against the False Positive rate (FP_r) at various thresholds. The TP_r is computed as the number of "speech" segments that were correctly classified over the total number of "speech" segments. Similarly, the FP_r is the ratio of the number of "nonspeech" segments that were classified as "speech" over the total number of "nonspeech" segments. The TP_r and FP_r are also called Sensitivity and Fall-out respectively. The two previous evaluation tools are are visual tools. We also use three other measures for the evaluation. The Efficiency and the Area Under the Curve (AUC) are computed. The Efficiency is equal to the number of the segments that were classified correctly (both "speech" and "nonspeech"), over the total number of segments that were processed. And finally by saying AUC, the area under the ROC curve is meant. Computing the AUC we measure the possibility, that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. This is because, as mentioned earlier the ROC shows the TP_r against the FP_r.

5.1 Evaluating the performance on each seminar separately

First we will see how the algorithm performed on each seminar separately. In Figure 5.1 we can see the DET curve that we get, when using only the data from the first Lira seminar. This means that both training and testing data are from the same seminar. Due to this, the training and testing sets will be similar, since they were recorded under the same environmental conditions. With dashed lines the DET curve when using only CF, Clarity and Harmonicity on their own are represented. Whereas the solid lines correspond to the DET curves of the feature combinations, except the black one that corresponds to the MFCCs. The goal of this thesis is to improve the performance of a classic voice detection/discrimination system that uses the MFFCs. By observing the following plot, it is clear that all the feature combinations, except the MFCCs-Clarity, perform better that the MFCCs alone.

In the description of this seminar we mentioned that most of the parts that contain

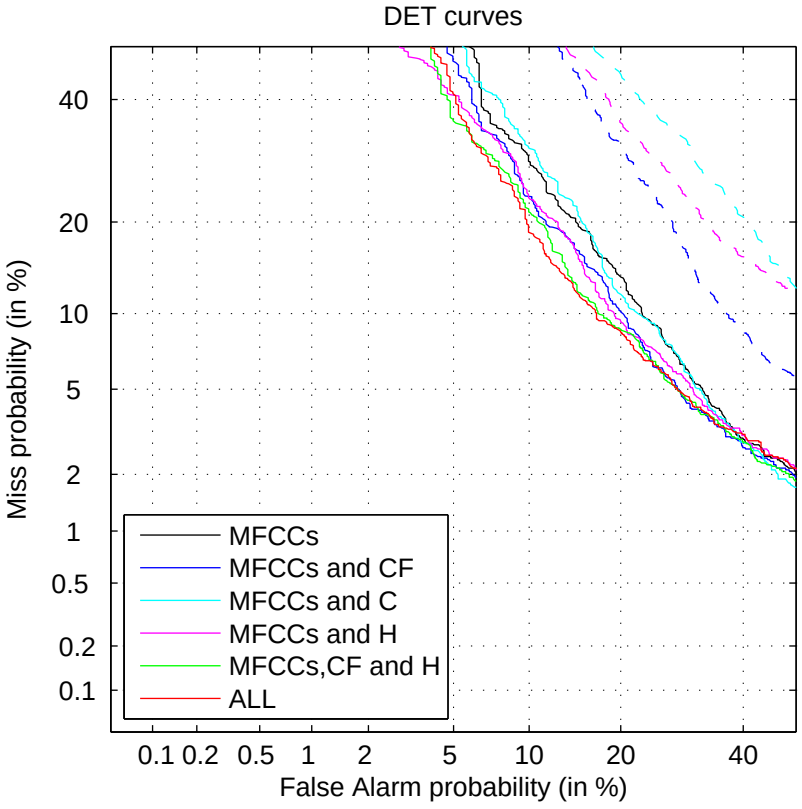


Figure 5.1: DET curves for seminar 1 Lyra.

speech do also contain music. This makes the classification harder. In opposite, after the annotation of the second Lira seminar we noticed that the speech parts are clearer than in the first. Most of the times, nobody played music while someone was talking. This

difference has a great influence on the performance of the algorithm. Comparing Figures 5.1 and 5.2, we can see that the DET curves for the second seminar take much lower values.

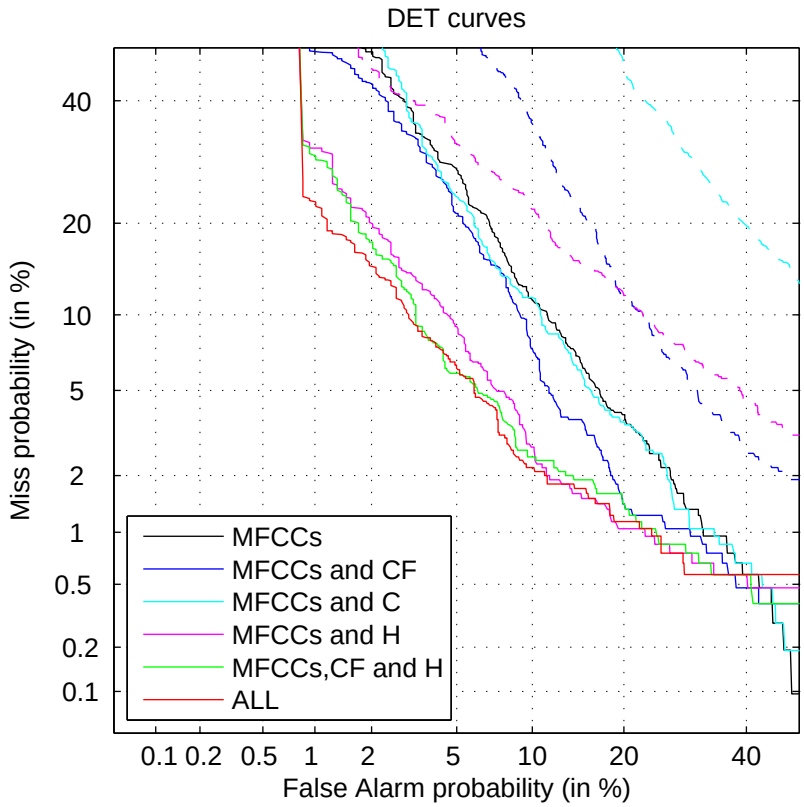


Figure 5.2: DET curves for seminar 2 Lyra.

Further, in Figures 5.2 and 5.3 the improvement of the results is more visible. Using MFCCs and Harmonicity, MFCCs, CF and Harmonicity or all features together a significant P_{miss} and P_{fa} reduction is achieved. It is remarkable that for a P_{fa} of 5%, the P_{miss} decreases about 20% for any of the previously mentioned combinations. Exactly the same thing can also be seen in Figures 5.5 and 5.6, where the ROC curves are shown. The curves that we get by using the combinations mentioned just before, are better than the black one (black corresponds to MFCCs). In other words the Area Under the Curve is getting bigger which is our goal. It is obvious from Figures 5.1-5.3, that the performance of using CF, Harmonicity and Clarity alone is worse. So it is not worth comparing these results with those from the combinations. This is why the ROC curves are not presented for those cases.

The third seminar, which is on lute was help and recorded outdoors. We referred to the noisy conditions, due to the cicadas in the previous chapter. The results agree with our expectations and it can be seen that the algorithm does not perform as good as in the second seminar, which is clear observing Figures 5.2 and 5.3.

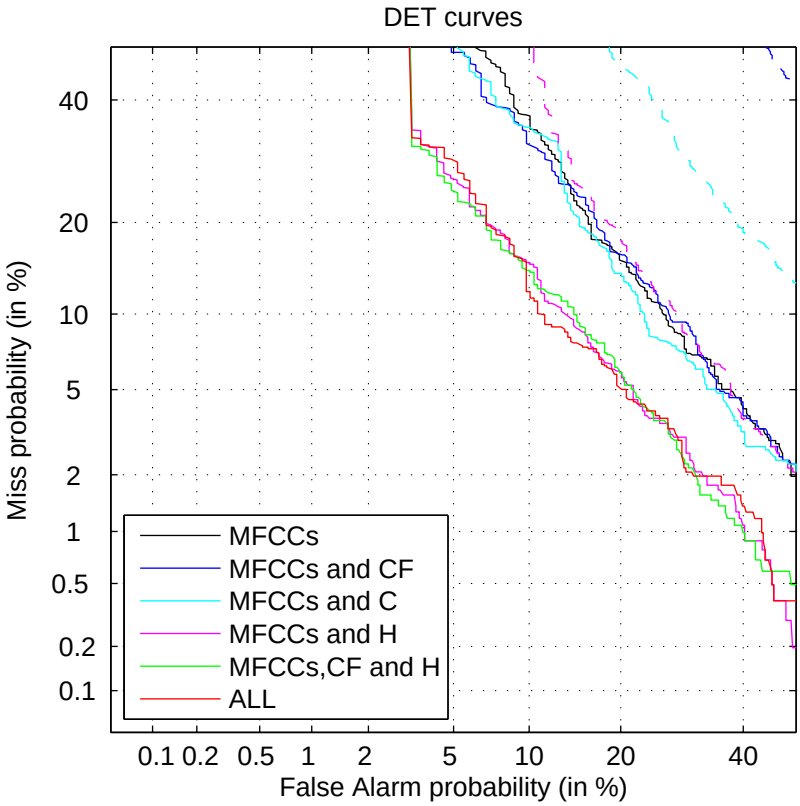


Figure 5.3: DET curves for seminar 3 Lute.

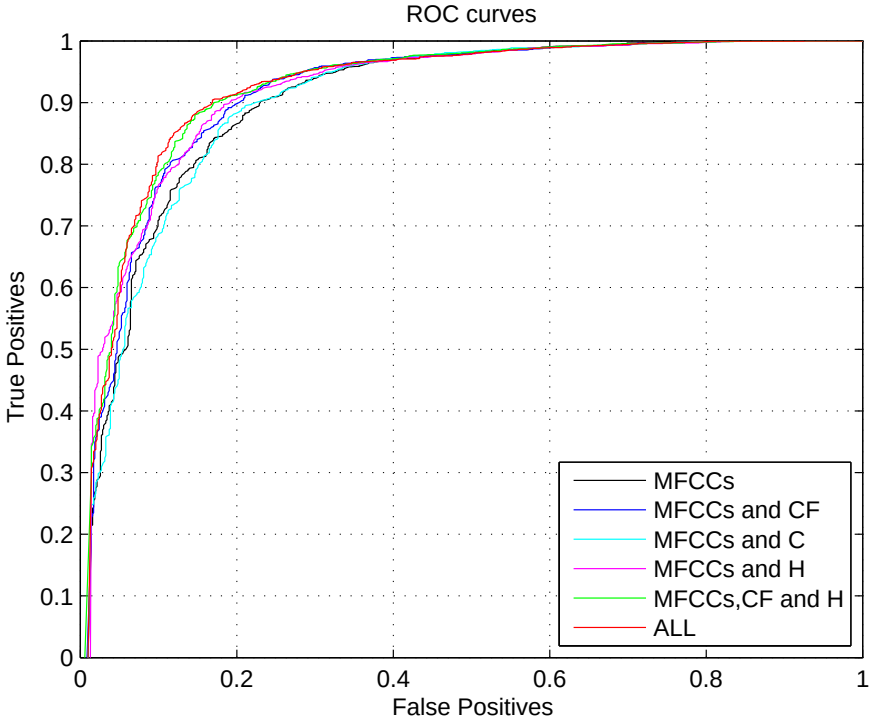


Figure 5.4: ROC curves for seminar 1 Lyra.

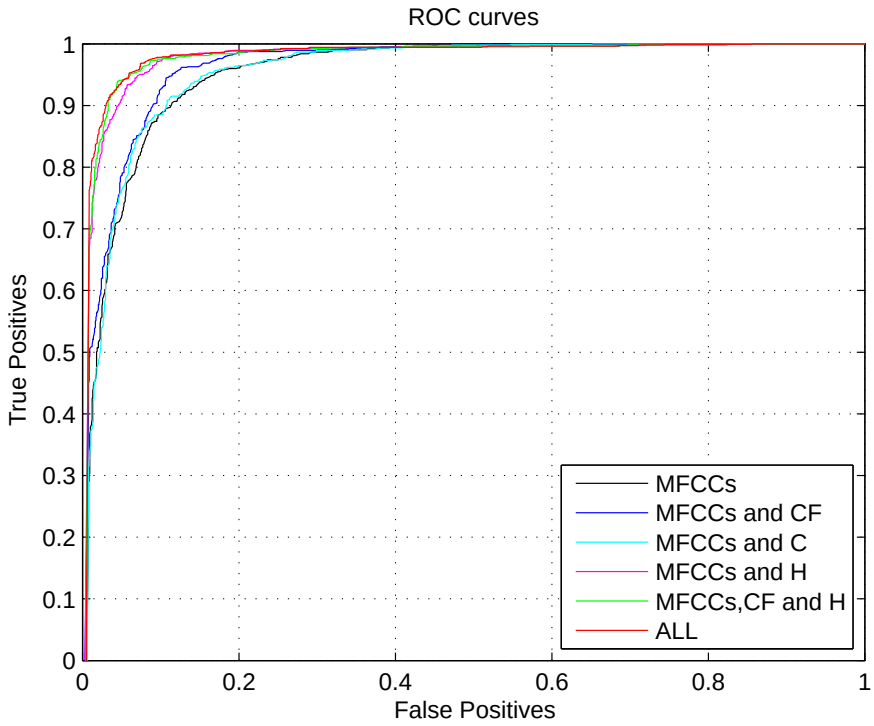


Figure 5.5: ROC curves for seminar 2 Lyra.

Analysing Figures 5.1 - 5.6 we see that whichever feature we combine with MFCCs, the algorithm performs better than in the case when we use only MFCCs. However, we notice that the combinations that perform best are MFCCs and Harmonicity (pink curve), MFCCs, CF and Harmonicity (green curve) and all the features together (red curve). The same applies, as we will see in the next section, when we use different data for the training and testing.

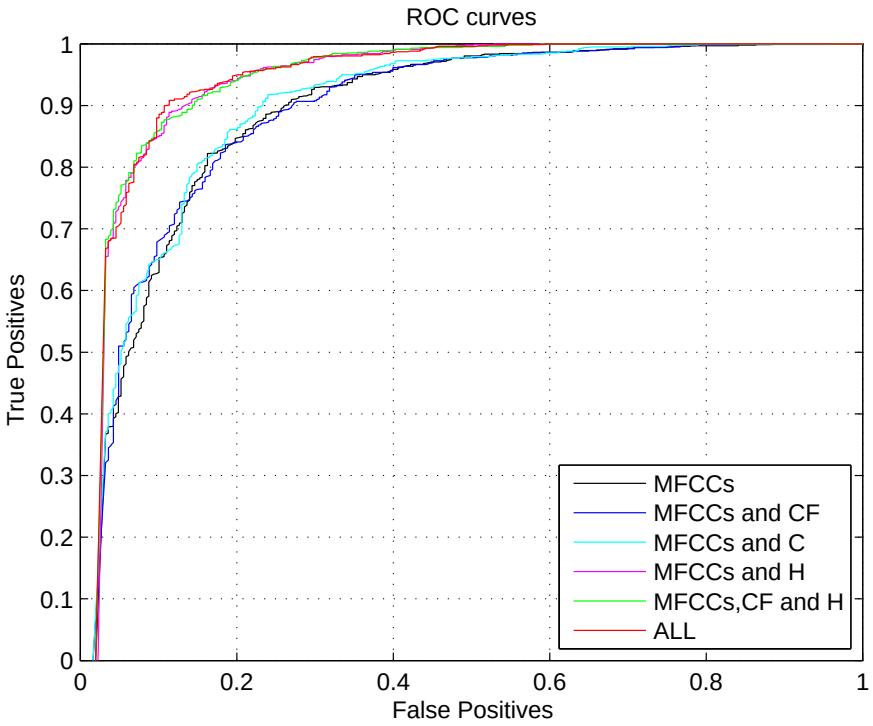


Figure 5.6: ROC curves for seminar 3 Lute.

5.2 Evaluation when using data from seminar 1 for training and seminar 2 for testing

In Chapter 3, we mentioned that when using different combinations of data for training and testing the results can differ. As described in Chapter 4, the conditions under which each seminar took place are different. In the first one, the microphone is not placed close to the teacher and due to the size of the room, an echo can be heard when someone talks. In the second seminar the microphone is closer to the teacher and the participants. Moreover, it was observed that the "speech" parts of this recording are much clearer in the second seminar than in the first.

What we did in this case is to use data from the first seminar for the training and from

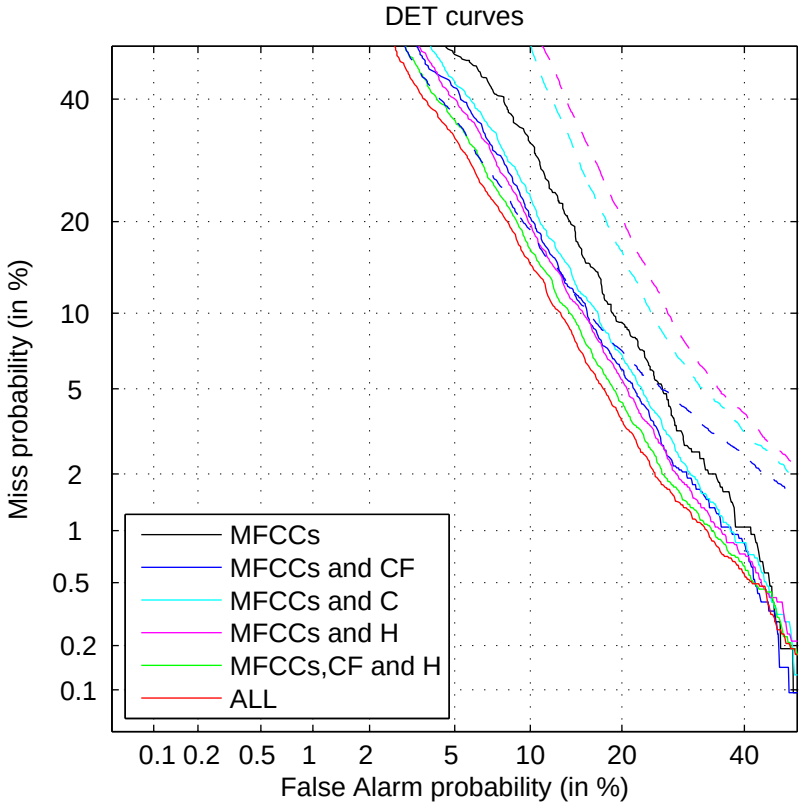


Figure 5.7: DET curves using data from seminar 1 for training and Lyra 2 for the testing.

the second one for the testing. The results of this data combination can be seen in the next two figures (5.7 and 5.8). It reasonable to compare Figure 5.7 with Figures 5.1 and 5.2. We notice that the DET curves in Figure 5.7 are better than those in 5.1. This behaviour was expected, as the training data were noisy whereas the testing data were not. So the algorithm was trained in rough conditions and then tested in simpler and clean conditions. Consequently the classification was easier to be applied. Comparing now Figures 5.7 and 5.2 we see that the DET curves of 5.7 are slightly worse than those in 5.2. Again, this

result is not surprising us. Since the training was performed on different data than the testing, the classification will obviously be more difficult and the possibility to miss classify a segment is bigger. This experiment confirms the importance of choosing the appropriate data for the training stage of a system (Chapter 3.1). The accuracy of the algorithm is affected of how representative the training data are. For this reason, we mixed all the data available and then tested the performance of the algorithm. The results can be seen following.

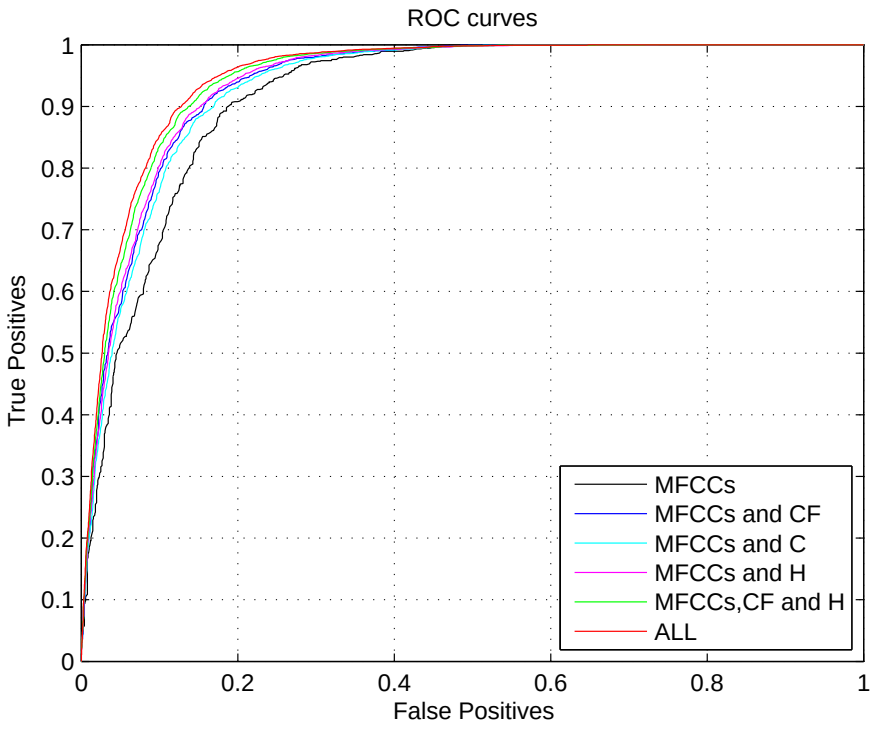


Figure 5.8: ROC curves using data from seminar 1 for training and seminar 2 for the testing.

5.3 Evaluation of the performance when using all data available

Here, the results can be seen in Figures 5.9 and 5.10 in the case where all data were used. The data of all three seminars are put into one set and both training and testing data are then derived from this set. The results produced in this way are more reliable than those presented previously. By using all the data, not all but many possible conditions are taken into account, under which a recording could take place. Thereby, the system will be trained in a way, where is could perform good whether the conditions are easy(relative clean data) or not(noisy data). The type of data on which the algorithm is tested is also important. If the testing data are clean, the scores will be better than in the general case. To be closer to the general case, noisy data need also to be contained in the testing set.

The difference between the performance of MFCCs and Harmonicity, MFCCS,CF and

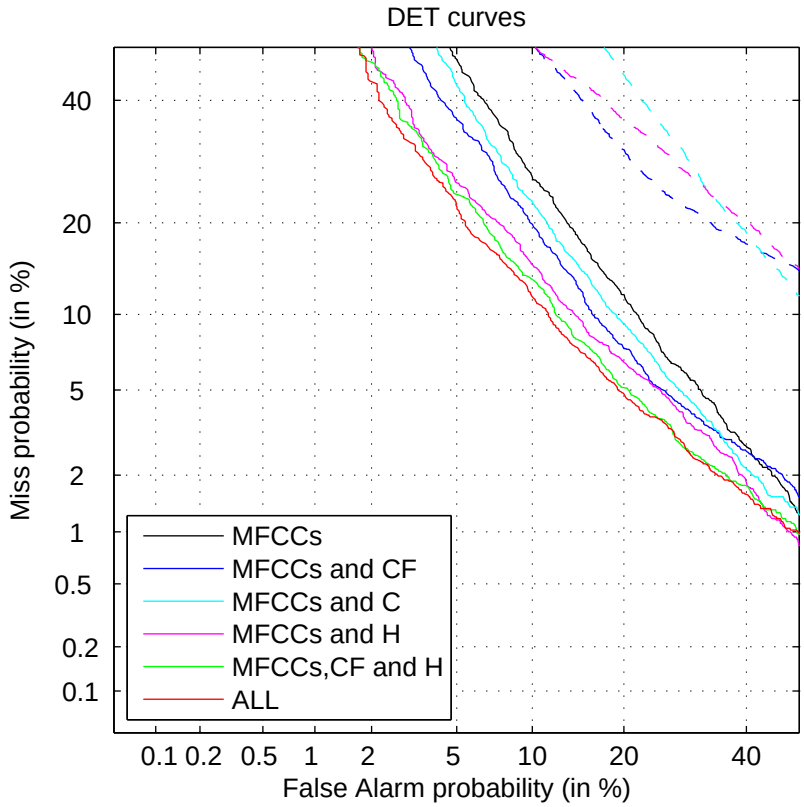


Figure 5.9: DET curves using all data.

Harmonicity and all the feature is again visible in Figures 5.9 and 5.10. The results that we derive from those three combination stand out from the others. Following, tables are presented for the Equal Error Rate (EER), the Efficiency and the Area Under the Curve (AUC) for the cases described above. Since only three out of the five combinations show a significant improvement, we computed EER, Efficiency and AUC just for those cases.

In Table 5.1, the EER can be seen for each data set used with the feature combinations mentioned earlier. The mean of those measures are reported and their variance in brackets. Since the value of the variance is very small and close to zero not the exact value is written. Then in Tables 5.2 and 5.3, the Efficiency and the AUC is presented. Plots are also presented, as it is a more convenient way to compare the performance in each case.

Observing Figure 5.11 we clearly see the improvement that is achieved through the fea-

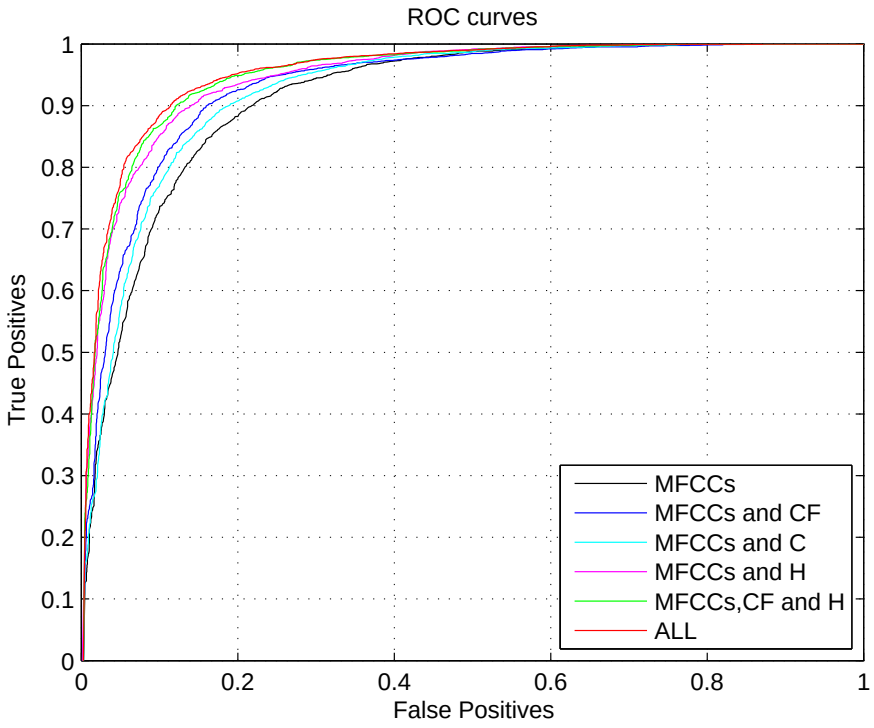


Figure 5.10: ROC curves using all data.

ture combinations. The value of the EER differs a lot between the blue bar (using only MFCCs) and the other three bars.

Seminars → Features ↓	Seminar 1 (Lira)	Seminar 2 (Lira)	Seminar 3 (Lute)	Seminars 1 and 2	All seminars
MFCCs	$0.104(< 10^{-3})$	$0.097(< 10^{-3})$	$0.1082(< 10^{-3})$	0.126(0)	$0.1139(< 10^{-3})$
MFCCs and Harmonicity	$0.0994(< 10^{-3})$	$0.0475(< 10^{-3})$	$0.0764(< 10^{-3})$	0.0986(0)	$0.0949(< 10^{-3})$
MFCCs, CF and Harmonicity	$0.0934(< 10^{-3})$	$0.0466(< 10^{-3})$	$0.0679(< 10^{-3})$	0.0748(0)	$0.0894(< 10^{-3})$
All features	$0.0911(< 10^{-3})$	$0.046(< 10^{-3})$	$0.0715(< 10^{-3})$	0.0681(0)	$0.0849(< 10^{-3})$

Table 5.1: Mean and variance of Equal Error Rates (EERs) for the feature combinations on the various data.

In the next two Figures (5.12 and 5.13), we see that the values are much higher for the three combination than for the MFCCs. It is remarkable to see the difference between the

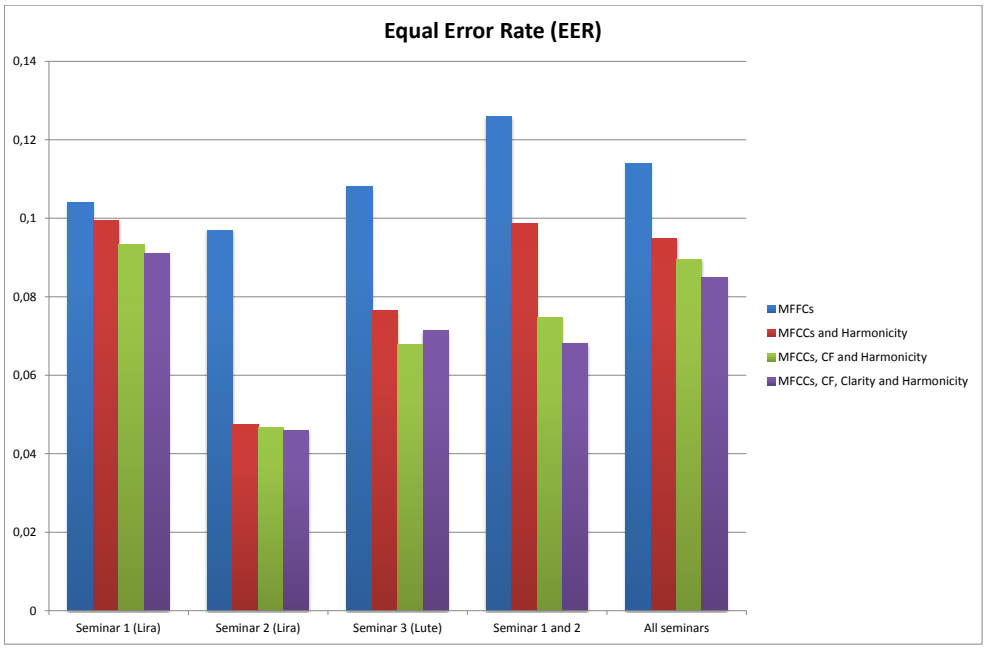


Figure 5.11: EER scores for all the tested data.

results that we get from seminar 2 and from seminar 1. In Chapter 4, we said that the parts containing speech in seminar 2 were relatively clean, whereas in seminar 1 this does not hold.

In seminar 2, it is easier for the classifier to distinguish between the two classes, since

Seminars → Features ↓	Seminar 1 (Lira)	Seminar 2 (Lira)	Seminar 3 (Lute)	Seminars 1 and 2	All seminars
MFCCs	$0.8957(< 10^{-3})$	$0.9074(< 10^{-3})$	$0.8921(< 10^{-3})$	0.8529(0)	$0.8735(< 10^{-3})$
MFCCs and Harmonicity	$0.9(< 10^{-3})$	$0.9545(< 10^{-3})$	$0.9291(< 10^{-3})$	0.8992(0)	$0.8974(< 10^{-3})$
MFCCs, CF and Harmonicity	$0.9063(< 10^{-3})$	$0.9584(< 10^{-3})$	$0.9328(< 10^{-3})$	0.9151(0)	$0.9059(< 10^{-3})$
All features	$0.9043(< 10^{-3})$	$0.9588(< 10^{-3})$	$0.9291(< 10^{-3})$	0.9181(0)	$0.9085(< 10^{-3})$

Table 5.2: Mean and variance of Efficiency for the feature combinations on the various data.

the various parts (speech and nonspeech) differ a lot. This explains the high values of Efficiency and AUC and low ones of EER for the second seminar. In the last three Figures (5.11, 5.12, 5.13) the degree by which the data being used for the training and testing of the system affect the results can be visually seen. For seminar 2, the EER scores are lower than in the other cases. As for the Efficiency and the AUC the difference can be seen clearly too . In the fourth case, by combining the data from seminar 1 and 2, we notice that although the results are better than in the first case, they are slightly worse than in the

second case. This is not strange, because as described earlier, not the same conditions hold for both training and testing data. As we can see in Tables 5.1, 5.2 and 5.3, the variances of all metrics are equal to zero. This happens due to the fact that the validation was only performed once for this particular case. Concerning the last case, putting together all the available data, it is reasonable to expect that the results will be an average of those in the others cases. What actually happens in this case is that the training set consist of data from all seminars and as well as the testing set. Figures 5.11, 5.12 and 5.13 confirm this estimation as the results really seem to be averaged.

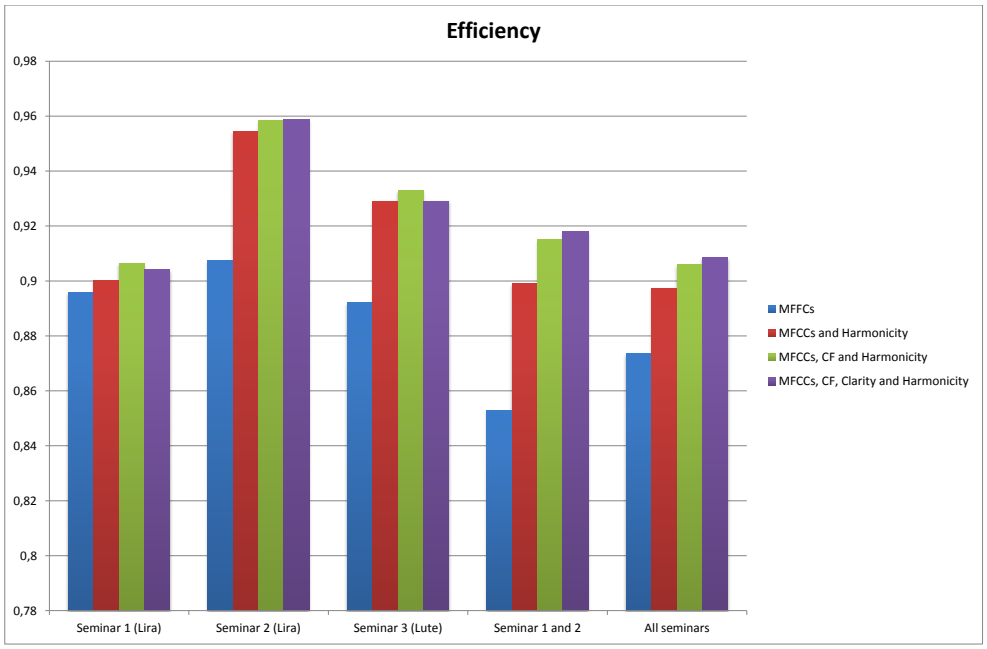


Figure 5.12: Efficiency scores for all the tested data.

Seminars → Features ↓	Seminar 1 (Lira)	Seminar 2 (Lira)	Seminar 3 (Lute)	Seminars 1 and 2	All seminars
MFCCs	$0.9081(< 10^{-3})$	$0.9577(< 10^{-3})$	$0.8921(< 10^{-3})$	0.9187(0)	$0.9194(< 10^{-3})$
MFCCs and Harmonicity	$0.9217(< 10^{-3})$	$0.981(< 10^{-3})$	$0.9441(< 10^{-3})$	0.9555(0)	$0.9481(< 10^{-3})$
MFCCs, CF and Harmonicity	$0.9262(< 10^{-3})$	$0.982(< 10^{-3})$	$0.9462(< 10^{-3})$	0.9682(0)	$0.9538(< 10^{-3})$
All features	$0.9257(< 10^{-3})$	$0.9826(< 10^{-3})$	$0.9441(< 10^{-3})$	0.9698(0)	$0.9556(< 10^{-3})$

Table 5.3: Mean and variance of Area Under the Curve for the feature combinations on the various data.

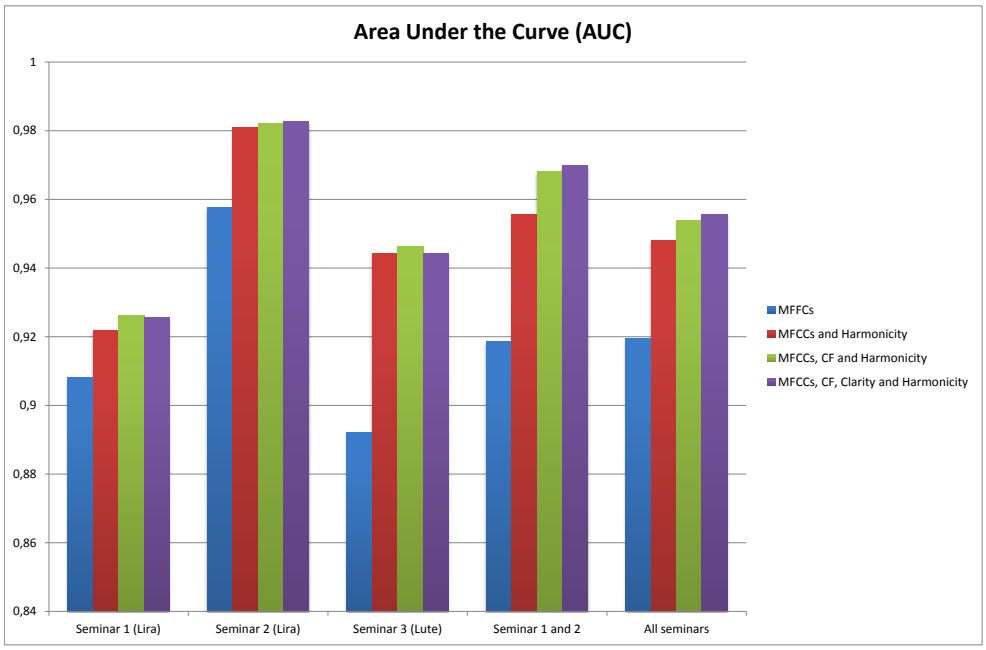


Figure 5.13: AUC scores for all the tested data.

Chapter 6

Conclusions and future work

In this thesis, we worked on voice detection in musical environment. An algorithm was implemented in order to detect the presence of voice in spontaneous and real-life recordings from music lessons. The developed algorithm is based on a classic system which extracts MFCCs from the input signal and classifies segments into "speech" and "non-speech". Three additional features were used to achieve an improvement of the results given by the classic system. These features are the Cepstral Flux, the Harmoniicty and the Clarity, which are computed during the signal analysis.
Various combinations of those three features with the MFCCs are examined and evaluated. We conclude that only three combinations are worth of discussing and comparing with the state-of-the-art MFFCs. Those are the MFCCs with the Harmonicity, the MFCCs with the Cepstral Flux and Harmonicity and the MFCCs combined with all three features. Although the results produced by the combinations mentioned before are very close to each other, the use of all features performs best.
We have seen the importance of using representative enough data for both the training and testing stage. In Chapter 5, observing the figures presented, the difference between the results can be seen when using different combination of training and testing sets. To provide reliable results, we note that the training set needs to contain data recorded in as many as possible environmental conditions (very noisy but also less noisy). This way we are able to guarantee that the algorithm will perform as good as the results show, in the majority of the cases.

Extending this work, it would be interesting to build a system that takes an input audio signal and returns it segmented. This means that after the processing and the classification of the small segments (3 seconds) that our algorithm performs, continuous segments would be grouped together. So, the access to parts of the signal according to its content will be easier. Also, in some application there is the need to know with a high

precision the boundaries of each segment. So another extension could be to develop a method or using an existing one, in order to determine precisely the beginning and the end of each segment given as an output from our algorithm. Studying more features and test various combination for improving even more the results by providing more robustness in environmental noises could also be done.

Finally, it would be quite interesting to be able to run such a system in real time. Of course this needs to be fast but accurate enough, which is difficult in some cases. Developing such a system means, fast feature extraction and classification too. Consequently, the features to be used need to carefully be chosen by having low computational cost. In the case of a real time system, the segments to be classified must last less than 3 seconds, otherwise the system will not provide results often enough. Even reducing the duration to 1 second would possibly not be enough. In this case, we have to deal with the question of, which is the ideal length of a segment. It has to be short enough so the processing will not take too long, but big enough to be able to get useful and representative information of the underlying signal at the same time.

Bibliography

[1] Tomi Kinnunen, Evgenia Chernenko, Marko Tuononen, Pasi Fränti, and Haizhou Li. Voice Activity Detection Using MFCC Features and Support Vector Machine. In *Interspeech*, 2007.

[2] M.H. Moattar, M.M. Homayounpour, and Nima Khademi Kalantari. A New Approach for Robust Realtime Voice Activity Detection Using Spectral Sattern. In *ICASSP*, 2010.

[3] M.H. Moattar and M.M. Homayounpour. A Simple but Efficient Real-Time Voice Activity Detection Algorithm. In *EUSIPCO*, 2009.

[4] Javier Ramirez, José C. Segura, Carmen Benitez, Angel de la Torre, and Antonio Rubio. A New Adaptive Long-Term Spectral Estimation Voice Activity Detector. In *Eurospeech*, Geneva, 2003.

[5] Ekapol Chuangsuwanich and James Glass. Robust Voice Activity Detector for Real World Applications Using Harmonicity and Modulation Frequency. In *Interspeech*, Florence, Italy, 28-13 August 2011.

[6] A. Sangwan, M.C. Chiranth, H.S. Jamadagni, R. Sah, V. Prasad, and V. Gaurav. VAD Techniques for Real-Time Speech Transmission on the Internet. *IEEE High Speed Networks and Multimedia Communications*, pages 46–50, 2002.

[7] Trausti Kristjansson, Sabine Deligne, and Peder Olsen. Voicing Features for Robust Speech Detection. In *Interspeech*, Lisbon, Portugal, 2005.

[8] L. R. Rabiner and R. W. Schafer. *Theory and Applications of Digital Speech Processing.* 1st ed. Upper Saddle River, NJ: Prentice-Hall, 2010.

[9] Seyed Omid Sadjadi and John H. L. Hansen. Unsupervised Speech Activity Detection Using Voicing Measures and Perceptual Spectral Flux. *IEEE Signal Processing Letters*, 20:197–200, 2013.

[10] Wo-Ho Shin, Byoung-Soo Lee, Yun-Keun Lee, and Jong-Seok Lee. Speech/Non-Speech Classification using Multiple Features for Robust Endpoint Detection. *IEEE Acoustics, Speech, and Signal Processing*, 3:1399–1402, Istambul, 5-9 June 2000.

[11] Maria Markaki and Yannis Stylianou. Discrimination of Speech from Nonspeech in Broadcast News Based on Modulation Frequency Features. *Speech Communication, Perceptual and Statistical Audition*, 53:726–735, 9 March 2010.

[12] M. H. Savoji. A robust algorithm for accurate endpointing of speech. *Speech Communication*, 8:45–60, 1989.

[13] Maria Markaki, Andre Holzapfel, and Yannis Stylianou. Singin Voice Detection using Modulation Frequency Features. *Workshop on Statistical and Perceptual Audition*, Brisbane, Australia, 21 September 2008.

[14] Berhard Lehner and Reinhard Sonnleitner adn Gerhard Widmer. Towards Lightweighted, Real-time-capable Singing Voice Detection. *International Society for Music Information Retrieval*, 2013.

[15] Kyogu Lee and Markus Cremer. Automatic labeling of training data for singing voice detection in musical audio. *Signal Proccessing, Patter Recognition and Applications*, 2009.

[16] Martín Rocamora and Perfero Herrera. Comparing audio descriptors for singing voice detection in music audio files. *11º Simpósio Brasileiro de Computação Musical (SBCM07)*, 1 September 2007.

[17] Wu Chou and Liang Gu. Robust Singing Detection in Speech/Music Discriminator Design. *IEEE Acoustics, Speech, and Signal Processing*, 2:865–868, Salt Lake City, UT, 7-11 May 2001.

[18] L. Regnier and G. Peeters. Singing Voice Detection in Music Tracks using Direct Voice Vibrato Detection. *IEEE Acoustics, Speech, and Signal Processing*, pages 1685–1688, Taipei, Taiwan, 19-24 April 2009.

[19] H. Lachambre and R. André-Obrecht adn J. Pinquier. Singing Voice Characterization for Audio I. *EUSIPCO*, Poznań, 2007.

[20] Eric Scheirer and Malcolm Slaney. Construction and Evaluation of a Robust Multifeature Speech/Music Discriminator. *IEEE Acoustics, Speech and Signal Processing*, 2:1331 – 1334, 21-24 April 1997.

[21] R. Sonnleitner, B. Niedermayer, and G. Widmer J. Schlüter. A simple and effective specterl feature for speech detection in mixed audio signals. *Conference on Digital Audio Effects*, York, UK, 17-21 September 2012.

[22] Zhong hua Fu and Jhing-Fa Wang. Robust Features for Effective Speech and Music Discrimination. *Proceedings of the 20th Conference on Computational Linguistics and Speech Processing*, pages 209–215, Taipei, Taiwan, 2008.

[23] Omer Mohsin Mubarak, Eliathamby Ambikairajah, and Julien Epps. Novel Features for Effective Speech and Music Discrimination. *IEEE Engineering of Intelligent Systems*, Islamabad, 2006.

[24] Bong-Wan Kim, Dae-Lim Choi, and Yong-Ju Lee. Speech/Music Discrimination using Mel-Cepstrum Modulation Energy. *10th International Conference, Text Speech Dialogue*, pages 406–414, Pilsen, Czech Republic,3-7 September 2007.

[25] http://www.ee.columbia.edu/∼dpwe/resources/matlab/rastamat.

[26] http://svmlight.joachims.org.

[27] http://www.phonogrammarchiv.at.

[28] L. R. Rabiner and R. W. Schafer. Theory and applications of digital speech processing. *Prentice Hall Press*, 1st ed. Upper Saddle River, NJ, USA, 2010.

[29] W. H. Tsai and H. M. Wang. Automatic singer recognition of popular music recordings via estimation and modeling of solo vocal signal. *IEEE Transactions on Audio Speech and Language Processing*, pages 330–341, January 2006.

[30] Hermansky H. Perceptual linear predictive (PLP) analysis of speech. *JASA 87(4)*, 1738-1752, 1990.

[31] Marolt M. Probabilistic segmentation and labeling of ethnomusicological field recordings. *ISMIR - International Conference on Music Information Retrieval*, pp. 75-80, 2009.

[32] www.mathworks.com.

[33] blogs.mathworks.com.