



Universidade de Coimbra
Faculdade de Ciências e Tecnologia

Mestrado em Engenharia Electrotécnica e de Computadores

Detecção Automática de Música

Pablo David Young Zubizarreta

ORIENTADOR: FERNANDO MANUEL DOS SANTOS PERDIGÃO

SETEMBRO 2010

ACKNOWLEDGEMENTS

It is not an easy task to be grateful for everything to so many people without forgetting somebody...

First of all I am grateful for my tutor Fernando Santos Perdigão for the chance that he has given to me for doing my final degree's project in the group of Investigation of Signal Processing Laboratory of IT, for all his given help and the professional support transmitted. These acknowledgments would never have been written without the formation offered for Public University of Navarra during these years, I am grateful to the whole educational staff, as well to Coimbra's University which for six months, they all made feel me like in my own house, and always offered me their advices and support.

I am grateful to all my lab's class mates for making these four walls a pleasant and a hard-working place in which I spent such a long time, thanks to each of you.

I am also grateful for the support of my companions of Erasmus that help me to be a better person, in these six unforgettable months of my life, thanks to Miguel for his support, for his understanding and listening to me at any time, to my neighbor Chari for having a perfect word in the perfect moment, although at the end I would never take in account her advice, thanks to Lydia, my particular teacher, for sharing with me those lunches time in the canteen, to Ana, my room-mate, for always having a smile on her face, to Carlos for being always there, to Deivid for these battles of roosters and to all my Erasmus' companions, that somehow or other, they have helped and encouraged me during these months.

I also want to thanks my friends in my childhood, Eki, for being always there, Gontxa, Santi, Calde, Cuetol, Tuto, Pelli, Mane, Chus, Riki, Nachete, Tino, Mow, .. because thanks to them, in some way, I'm as I am.

To my class mates of the UPNA and Pamplona, Edu, Marta, Javi, Risi, Margallo, Yago, Edu and Adria for all those good moments.

Certainly, I want to dedicate this project to my parents, David and Toñi, to my dog, Brownny, to my grandmother, to Germán, to my uncles Congui and Luis, also my uncle Daniel and to all my family in general for the whole support, fondness and confidence that they have given me along all my life, and that undoubtedly, they have been important in the culmination of my studies (and not only for paying the fees at the university) as if it's not because of them, I would not be where I am neither would not be as I'm now

I want to remark and give a special thanks to my parents, for their constant effort, support and for the education that they have given me, helping me all the time, for becoming my principal inducement to put a full stop on the last page of this part of my life.

And of course, thanks very much to HER because, with no doubt, she is the person with whom I have shared most of the moments in my life and the person that deserves to be next to me, enjoying with me, this moment.

And finally, I wish all these persons who have the courage, the will and the time to pass page and do a useful reading of this project.

Certainly; Not all of the listed are here, but the ones that are here, are listed.

To all, thank you.

Pablo David Young Zubizarreta, September 2010.

Summary

This document presents the work done in automatic detection of music events present in audio signals. The signal corresponds to recordings of broadcast radio and TV programs. The aim of this work is the development of algorithms to discriminate musical segments from other sounds. The algorithms require the definition of models for audio classification, in our case Gaussian models. Two models were created, one for music and another for non-music (a background model representing speech in most of the times) and the classification is based on log likelihood ratios.

In the first part of this work several hours of audio recordings have been manually annotated in order to define an audio database. The database includes two sets of audio files, one set for model training and another to test the detection system. The proposed method, despite its simplicity, has proven capable of achieving good results.

Ultimately the intent of this project is to construct a series of algorithms to differentiate, in the best way possible, different audio events that are present in the files, including silence, music, speech and other events.

Table of Contents

1. INTRODUCTION	9
1.1 Problem Definition	10
1.2 Objectives	10
1.3 Organization of the project	11
2. AUDIO CLASSIFICATION	12
2.1 Theoretical basis for the characterization	12
2.1.2 Mel-Frequency Cepstral Coefficients, MFCC	12
2.2. Classification Techniques	17
2.2.1 Introduction.....	17
2.2.2 State of art of speech/music discrimination	17
2.3 Techniques of audio classification	19
3. THE DATABASE	26
3.3 Software use in the labeling	27
3.4 Final database	28
4. CLASSIFICATION AND EVALUATION	33
4.1 Design of the work.	33
4.2 Evaluation.....	40
5. RESULTS AND CONCLUSIONS	43
5.1 Conclusions.....	45
5.3 Future work	47
References.....	48

Figure list

Figure 2.1- Block diagram of the process of calculating the Mel-Frequency Cepstral	13
Figure 2.2- Window function (rectangular)	14
Figure 2.3- Window function (hamming)	14
Figure 2.4- Mel-Hz scale	15
Figure 2.5- Bark scale	15
Figure 2.6- Example calculation Delta coefficients	16
Figure 2.7- Maximum-margin hyperplane and margins for an SVM trained with samples from two classes	20
Figure 2.8-Layer structure of a neural network	21
Figure 2.9-Example of k-NN classification.	22
Figure 3.1-Example of a manually classified Transcriber	26
Figure 3.2- Working environment of Transcriber	27
Figure 3.3- export files to lbl format	31
Figure 3.4- lbl file open with notepad	32
Figure 4.1- Block diagram of the overall algorithm	33
Figure 4.2- Block diagram of the training system	34
Figure 4.3- LLR without applying the filter	35
Figure 4.4- LLR filter applied	36
Figure 4.5- Block diagram of the classifier	37
Figure 4.6- Block diagram of the evaluator	38
Figure 4.7- Block diagram of the General algorithm	39
Figure 4.8- Precision: horizontal arrow. Recall: diagonal arrow.	40

Table list

Table 3.1- treino files of the first database	28
Table 3.2- teste files of the first database	29
Table 3.3- treino files of the final database	29
Table 3.4- teste files of the final database	30
Table 4.1- classification context	41
Table 4.2- nomenclature used in this work	41
Table 5.1- experiment 1	43
Table 5.2- experiment 2	43
Table 5.3- experiment 3	44
Table 5.4- experiment 4	44
Table 5.5- experiment 5	44
Table 5.6- experiment 6	45
Table 5.7- experiment 7	45
Table 5.8- experiment 8	45
Table 5.8- experiment 9	46

1. INTRODUCTION

The society we live in today provides us with increasingly great amount of multimedia content.

All this information has to be ordered and scheduled to be useful, because otherwise, its subsequent location and consultation would be impossible. Normally, this classification of information is done manually by people (as in the case of radio and television broadcasts, press releases, etc.), but there is obviously a growing need to automate, at least in part, this process. In this document we focus on classifying the sound messages (music) of broadcast radio and TV stations.

The sound transmission of messages through various modern communication channels such as internet (radio programs) are often considered as an ordered sequence of sounds made by humans, modern technological devices (synthesizers) and musical instruments. For these reasons, they may be classified or regulated by codes of a “radio language”, which can be defined by four elements:

- Speech
- Music
- Noise or sound effects
- Silence

Although this work will be focused only on discriminating the music part from the others, i.e., identify segments containing music in each audio file, the intention of the general project is to find a set of audio signal characteristics to distinguish other elements of “radio language” as best as possible (speech, silence, jingles, etc.).

1.1 Problem Definition

Due to the enormous amount of information we have to work with regularly, we must have file collections well structured and ordered, as well as content indexed, as this will lead to smooth communications, improved data transmission and increase in storage capacity of our teams.

This is of vital importance in information servers, particularly in the audio information. When sorting files, a common and simple way to distinguish between files that contain only speech and contain only music. This can give us the ability to differentiate radio content. But the following questions arise: how to make this distinction? What characteristics can we rely to make such distinctions in the audio files? In this project we will try to address these questions, clarifying more deeply the methods of discrimination between music and other events in an radio recording and what are the characteristics of the audio used that allow for such discrimination.

1.2 Objectives

The objective of this project is to discriminate, as accurately as possible, the segments of the audio signal that corresponds to music and their initial and final times.

Before doing so, we have to parameterize these signals. Choosing good parameters allow us to better discrimination between music and other elements of the audio signals (no music).

So far, the more successful parameters are the ones that have been used for speech recognition. They are cepstral parameters, namely the Mel-Frequency Cepstral Coefficients or MFCCs, which are briefly described below. We therefore choose these feature parameters for our purpose.

Once the parameterization is done, i.e., once we get the parameters out of each audio signal, we will move to the detection of music events. With pre-trained models and with the parameters previously extracted, we classify each frame (small segment of the audio signal) in terms of music/no-music using a likelihood ratio test. This kind of classification will be discussed in the next chapter.

1.3 Organization of the project

This work is structured in five chapters. The first chapter corresponds to the introduction, where the problem is presented and the objectives to carry out in this work are drawn.

Chapter 2 discusses the background, i.e., there will be a detailed description of the existing systems in this area as well as the parameterization based on the MFCCs. Also we provide a brief description of all features, and how are they extracted from the audio files. However, the bulk of this chapter will be the description of the classification system. We will present the most common classification techniques including the used likelihood ratio technique.

Chapter 3 describes the framework in which experiments are carried out, including the database, its annotation and how to perform the tests.

Chapter 4 contains the experimental part of the work, which performs the classification and evaluation of the measurements.

Finally, Chapter 5 shows the results of the tests. We will analyze the results, draw some final conclusions and possible lines of investigation for future works.

2. AUDIO CLASSIFICATION

This chapter deals with determining the classification system to be used in this project. It shows different possibilities, which are commonly used for this purpose and determines which of them is the one that has been proposed for this work.

Based on scientific papers we will give a detailed description of existing systems better known in the application area.

But before explaining in detail the techniques which are a most appropriate classification for this work we provide a brief explanation of how the audio signal has been parameterized, which is based on the extraction of MFCC coefficients.

2.1 THEORETICAL BASIS FOR THE CHARACTERIZATION OF AUDIO

I show now the parameterization technique by extracting the MFCC coefficients as it is one of the possibilities that nowadays exist at the time to customize the audio signal for proper characterization. This form of parameterization, shown below, is one of the best for the purpose of this project, which is the correct differentiation between segments of music and not music.

2.1.2 Mel-Frequency Cepstral Coefficients, MFCC

The Mel Frequency Cepstral Coefficients (cepstral coefficients in the Mel frequency), [1], are used as a compact representation of the spectrum of an audio signal. They are used in order to represent some characteristics of the human auditory perception system, namely the non-linear frequency resolution.

For that a bank of filters is used with central band frequencies that are located logarithmically (in a Mel scale), which models the human auditory response more accurately than linearly spaced bands. Then a discrete cosine transform (DCT) of the log energies of the filters' outputs are computed.

This allows for more efficient data processing, as it is one of the most widely used techniques in signal processing because it has the possibility to parameterize the signal with a small number of patterns, making it possible to rebuild properly, for instance, in audio compression.

This property of the MFCCs, combined with its robustness to noise and ease of calculation, turns it widely used. Another advantage is that it does not require too much processing time, which is important in the implementation time. The following figure shows the scheme for extracting the MFCCs.

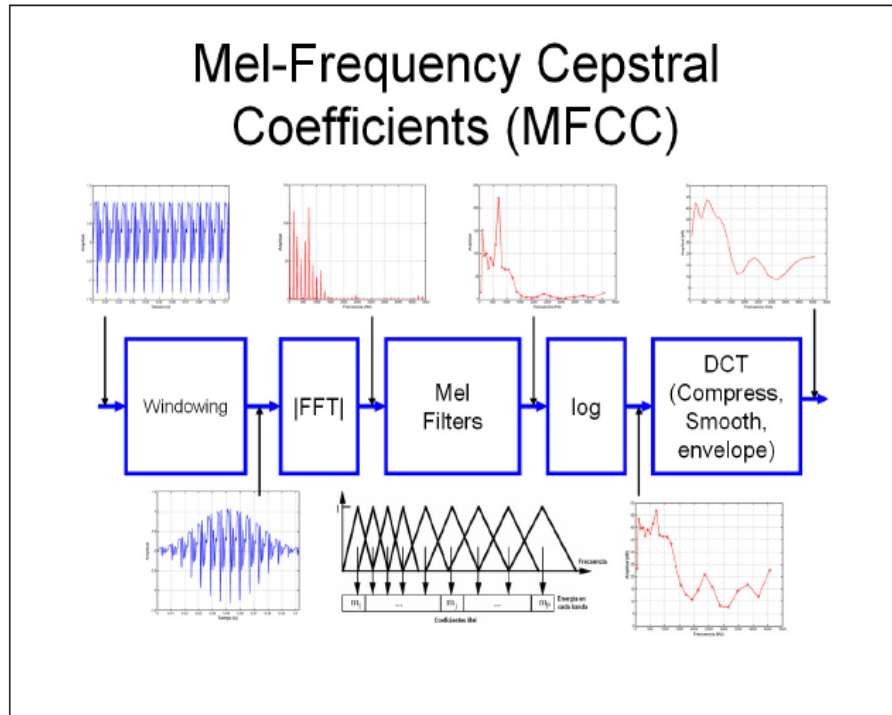


Figure 2.1- Block diagram of the process of calculating the Mel-Frequency Cepstral

Although in our work the MFCC coefficients are extracted directly from the audio signal with an executable (produced in the lab), it is interesting to comment briefly the process that the audio signal follows to obtain their own MFCC coefficients.

First, it performs a windowing of the speech signal through hamming type windows, to avoid if possible, the appearance of high frequency components, due to the discontinuities of the square wave signal.

We see images in the following representation of rectangular windows and Hamming, in the time domain and frequency.

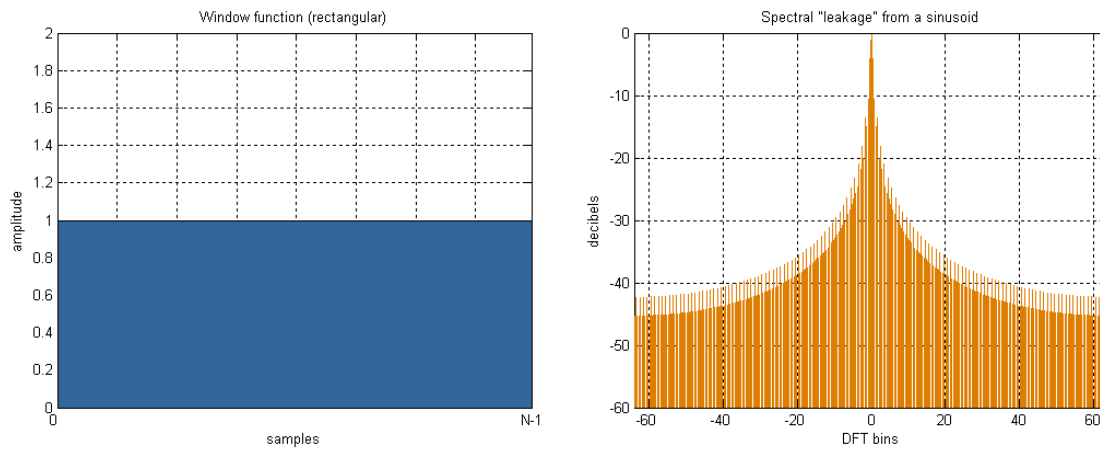


Figure 2.2- Window function (rectangular)

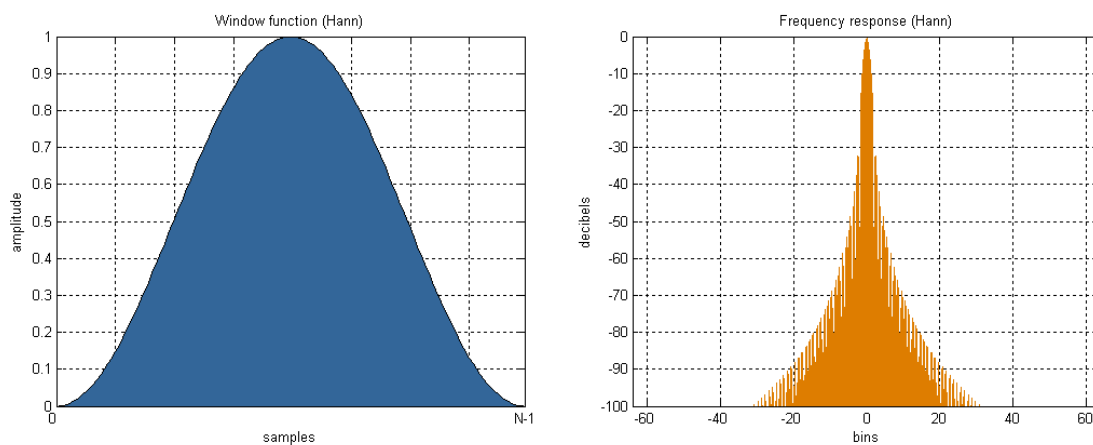


Figure 2.3- Window function (hamming)

As we see in the images, the spectrum of the Hamming window is similar to a delta in the frequency domain, thus introducing distortion in the spectrum of the audio signal is lower than in the case of the rectangular window.

After spending the spectral domain, in our case, through the FFT (Fast Fourier Transform) which is used for digital signal processing, the MFCC coefficients needed are extracted.

Then the resulting signal is filtered by a bank of filters of different frequencies and amplitudes, the aim of this filtering is to give more resolution at low frequencies, as in the human auditory system.

To make an approach to the functioning of the human ear, exhibiting no linear frequency response, filtering is performed by a bank of filters in Mel scale:

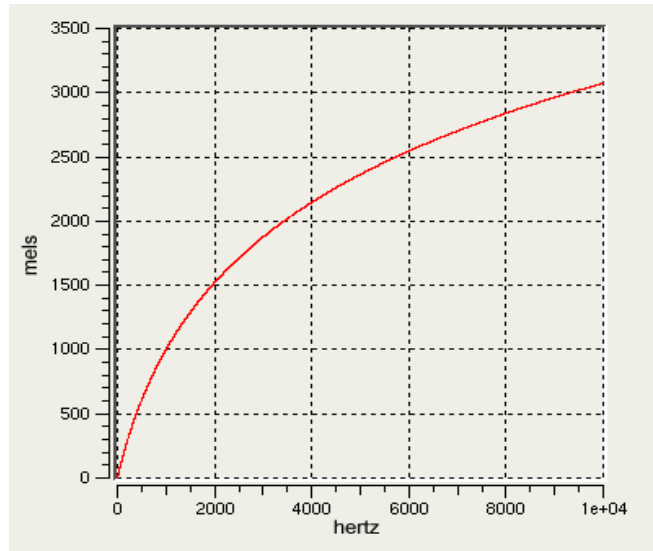


Figure 2.4- Mel-Hz scale

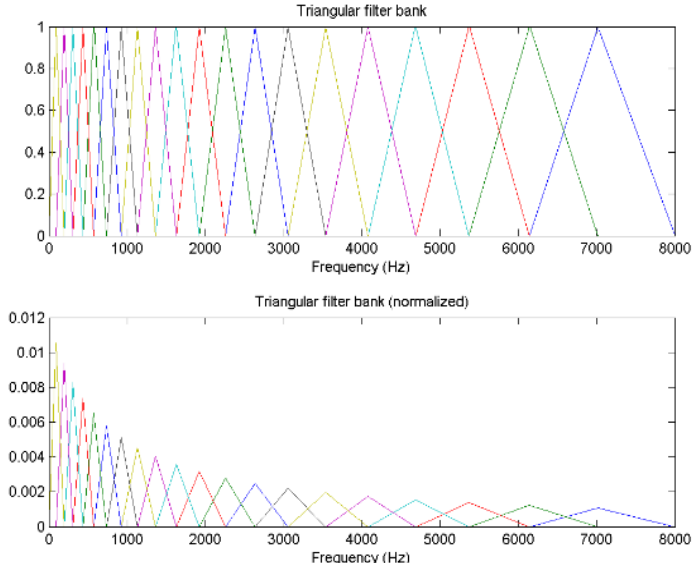


Figure 2.5- Bark scale

From the output of each filter an average energy is calculated thereby obtaining a signal with many energy values as filters. Passing the logarithm of these energies through a DCT (Discrete Cosine Transform), the MFCC coefficients are obtained. For more relevant information, also commonly used speeds (Delta-MFCC) and / or acceleration (Delta-Delta-MFCC).

The following figure shows an example for the calculation of Delta's Coefficients

	t_0	...	t_{n-1}	t_n	t_{n+1}	...
m coef. MFCC	0		C_0	C_0	C_0	
	1		C_1	C_1	C_1	
	⋮					
	m		C_m	C_m	C_m	
m coef. Delta	0			$C_0(t_{n+1}) - C_0(t_{n-1})$		
	1			$C_1(t_{n+1}) - C_1(t_{n-1})$		
	⋮					
	m			$C_m(t_{n+1}) - C_m(t_{n-1})$		

Figure 2.6- Example calculation Delta coefficients

Delta coefficients represent the change in the coefficients MFCC around the instant of time considered. They are called, therefore, Coefficients of First Derivative or Velocity. Similarly, the Delta-Delta are called acceleration coefficients.

The obtained MFCC coefficients are composed of 13 cepstral coefficients, plus another 13 of its first derivative coefficients (coefficients for Delta speed), plus another 13 of its second derivative (Delta-Delta coefficients corresponding to its acceleration). In total, for each frame of the audio signal, we get a vector of 39 MFCC coefficients.

2.2. Classification Techniques

It is obligatory to start by speaking about the existing classifiers up to this moment. For this reason, a detailed description of some existing systems will be given.

2.2.1 Introduction

The automatic classification of audio files turns a need before the abundance of information in the way in which we live and unroll ourselves in our daily life, up to this moment many technologies have developed for the detection of the music in this type of files.

In addition, since the music and the speech are both most important classes of audio, a great number of reserchers have devoted themselves to discriminate against them by means of all kinds of techniques.

These techniques have different approaches looking for the best possible classification. Later we will enunciate some of the proposing authors, in particular Saunders in 1996; Scheirer and Slaney in 1997; Klein, El-Maleh et al in 2000 and Zhang and Kuo in 2001 among others.

2.2.2 State of art of speech/music discrimination

John Saunders, [2], was one of the first authors to propose, in 1996, a technique to discriminate music against speech in real time on broadcast FM (frequency modulation) radio. It consists of the extraction of zero crossing rate (ZCR) and energy. A Gaussian classifier in then applied to a feature vector with statistical parameters taken from ZCR and energy. He reports a precision of classification of 98 %, besides he does not indicate a test database to do the measurements.

Scheirer and Slaney, [3] (1997) presented a complicated approximation to the task. They exploited thirteen features to characterize the different properties of speech and music, and examined three schemes of classification: the multidimensional MAP Gaussian classifier, the GMM classifier, and the nearest-neighbor classifier. They reported an accuracy of more than 90 %.

A comparative view of the value of different types of features in speech music discrimination is provided in Carey, Parris, and Lloyd-Thomas, [4] (1999), where four types of features (amplitudes, cepstra, pitch, and zero-crossings) are compared for discriminating speech and music signals.

Khaled El-Maleh, [5] (2000), combined the line spectral frequencies and zero-crossing-based features for frame-level speech/music discrimination. The classification system operates using only a frame delay of 20 ms, making it suitable for real-time multimedia

applications. The Gaussian classifier and the classifier KNN were evaluated in their work.

An emerging multimedia application, [6], is content based indexing and retrieval of audiovisual data. Audio content analysis is an important task for such an application (Zhang and Kuo 2001).

Stefan, [7] found the extents of modulation of frequency you lower more than 20 critical bands and his diversions standard can form discriminator well for the task, and the features were less sensitive to canalize the quality and the size I shape that MFCC.

Pinquier, [8] presented an original modeling approach that shapes the approximation, called the approximation of differentiated model-maker, to distinguish the speech / music, which characterizes every class with his own spaces of feature and statistical models. According to his report, this system might identify the speech with an accuracy of 99.5 % and musical with 93%.

Mateu Aguilo et al, [9], show the “One-step multiclass” detection, that consists of the following. The audio signal is framed using 30 ms Hamming window and, for each frame, a set of spectral parameters has been extracted. There are two types of parameters: 16 Frequency-Filtered (FF) and a set of the following parameters: zero-crossing rate, short time energy, 4 sub-band energies, spectral flux, calculated for each of the defined sub-bands, spectral centroid, and spectral bandwidth. In total, a vector of 60 components is built to represent each frame. They use a SVM classifier.

2.3 AUDIO CLASSIFICATION SYSTEMS

The classification consists on the identification of the group (class) to which the new attribute belongs to, having in mind the observed characteristics. This is usually done with a supervised systems, which learns from the labeled data. In this case, the process of building an audio classifier consists of two phases:

- Training
- Evaluation or test.

The phase of training tries to extract the characteristics of the segments of each class to discriminate, in our case music and no-music.

The phase of test allows verifying that the classification system discriminates with accuracy the different types of segments present in the audio files.

For this purpose there are several techniques of pattern recognition that have been effective in similar aims (recognition of musical instruments, speaker, etc.). Among the most important are the Support Vector Machines, (SVM), the Artificial Neural Networks, (ANN), and the method K nearest neighbors, (k-NN).

Finally we will dedicate a special attention to the hypothesis testing theory and to the Log-Likelihood Ratio (LLR) between hypothesis. We apply this theory to the Gaussian Mixture Models, (GMM), since they are directly related to the Gaussian models used in the present work.

2. 3.1 Support Vector Machines (SVM)

The machines of vectorial support [10], [11] are a set of algorithms developed by Vladimir Vapnik. They belong to the family of the linear classifiers since they induce linear or hyper flat dividers in spaces of characteristics of very high dimensionality (introduced by functions core or kernel) with an inductive very particular bias (maximization of the margin).

Initially they were used for problems of binary classification, but later on its use has been extended to problems of regression, re-grouping, multi classification, ordinal regression, and now there works for the resolution of the most complex problems (trees and graphs).

A data is seen as a point defined by a p-dimensional vector (a list of p numbers), and what we need to know is how to separate this information with a hyperplane in a (p-1) - dimensions. It is what is called a linear classifier. Of the possible hyperplanes, what we want to obtained is the one with a major separation or margin between the classes of information. This is indicated in the next figure. Therefore the distance needs to be

maximized between the hyperplane and the most nearby information of each one of the classes; it is what is called a classifier of maximum margin.

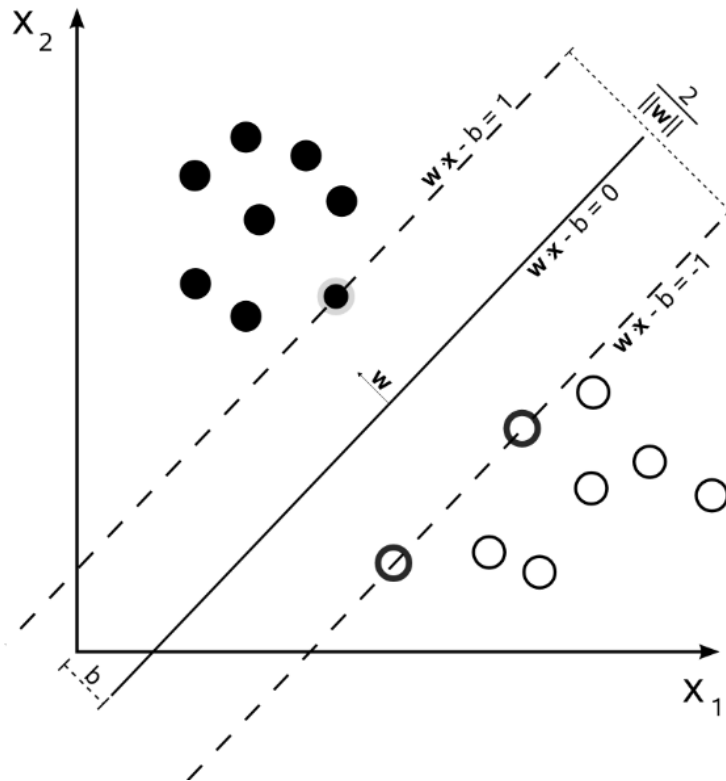


Figure 2.7-Maximum-margin hyperplane and margins for an SVM trained with samples from two classes

2.3. 2 Artificial Neural Networks (ANN)

The Artificial Neural Networks, [12], try to emulate the human capacity of learning and application of the learned to new situations to take decisions, that is to say the memorization and the association.

In other words, a neuronal network is "a new system for the data processing, which basic unit of processing is inspired in the fundamental cell of the nervous human system: the neuron".

The neural networks consist of units of processing that exchange data or information; they are used to recognize patterns (images, time sequences, etc.) and they have aptitude to learn and improve its functioning.

The advantages that this system offers are:

- Adaptive learning: it allows to learn how to do tasks based on one training or in an initial experience.
- Self-organization: the neural networks can create their own organization or Representation of the information that receives by means of a stage of learning.
- Tolerance to failures: the partial destruction of a network leads to a degradation of its structure; nevertheless, some capacities of the network can be retained, even suffering a great damage.
- Real time operation: the calculations neural can be done in parallel; they are designed for it and machines are made by special hardware to obtain this capacity.
- Easy insertion inside the existing technology: it is possible to obtain specialized chips for neural networks which improve its capacity in certain tasks. It will facilitate the modular integration in the existing systems.

A very popular neural network topology is a simple one constituted by neurons interconnected and arranged in layers (Multilayer Perceptron or MLP). The information enters for the input layer, go through the "hidden" layers and they go out by the output layer. The hidden layers can be formed by several layers.

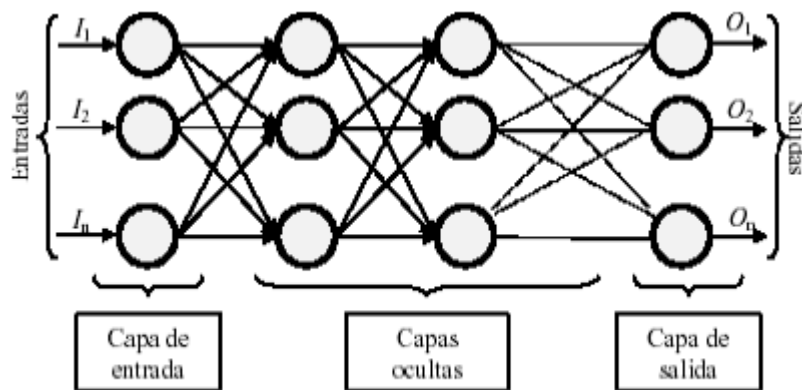


Figure 2.8-Layer structure of a neural network

2.3.3 The *k*-nn method (K nearest neighbors)

In pattern recognition, the *k*-nearest neighbors algorithm (*k*-NN), [13], is a method for classifying objects based on closest training examples in the feature space. *K*-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The *k*-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an

object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of its nearest neighbor.

The neighbors are taken from a set of objects for which the correct classification is known. This can be thought of as the training set for the algorithm, though no explicit training step is required. The k -nearest neighbor algorithm is sensitive to the local structure of the data.

Nearest neighbor rules in effect compute the decision boundary in an implicit manner. It is also possible to compute the decision boundary itself explicitly, and to do so in an efficient manner so that the computational complexity is a function of the boundary complexity.

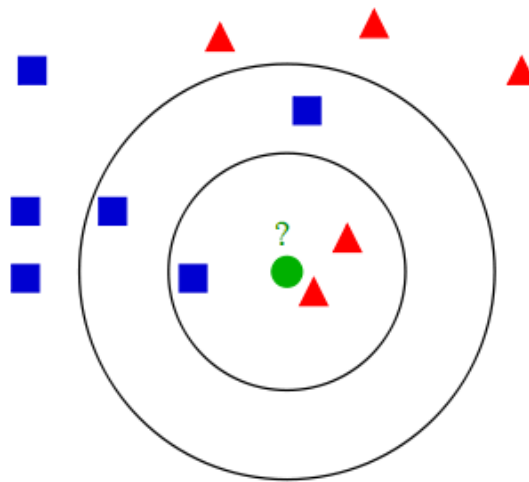


Figure 2.9-Example of k -NN classification.

The test sample (green circle) should be classified either to the first class of blue squares or to the second class of red triangles. If $k = 3$ it is classified to the second class because there are 2 triangles and only 1 square inside the inner circle. If $k = 5$ it is classified to first class (3 squares vs. 2 triangles inside the outer circle).

2.3.4 Gaussian Mixture Models (GMM) and Log-Likelihood Ratio

Gaussian Mixture Models (GMM)

Although on this work we are not going to use this model, it is interesting to comment it briefly, since it is one of the most used methods for classification tasks and discrimination of the music and speech.

The model use of mixture of Gaussian (GMM) for tasks of audio classification is motivated by the Gaussian interpretation of different components that serve to represent different types of audio labels (music, speech, instrumental music ...), and for its aptitude to shape arbitrary functions of density of probability. Said differently, the individual Gaussian components in a GMM have aptitude to shape some acoustic general classes.

The density of mixtures of Gaussian completes components are parameterized by means of the vectors of average, matrix of covariance and weight of mixtures of all the densities.

The GMM can have several different forms depending on the choice of the matrix (or matrices) of covariance. The matrix of covariance can be also complete or diagonal. The diagonal form is most used in GMMs, mainly because the easy way to invert it.

There are several compatible technologies to estimate the parameters of a GMM. The most popular and based method is the estimation of maximum Likelihood, (ML), used in this work for the detection of the music.

The unimodal Gaussian model (the one that we use in our work) represents a distribution of characteristics with only two parameters a mean vector and a matrix of covariance.

In the beginning of this work, GMM or HMM, among other options, have been defined as the models o use. But due to a series of problems and especially to the lack of time this work had to be re-defined in search of other simpler aims, but with valid results and a great deal of reliability.

Log-Likelihood Ratio

The discrimination systems are built around the likelihood ratio test, [14], using diagonal-covariance Gaussian Mixture Models (GMM) for likelihood functions.

Different procedures exist to estimate the parameters of a distribution of probability of the classes. Among these procedures probably the most versatile, as it is possible to apply in great quantity of situations, and therefore more used is known as the "method of maximum likelihood".

The likelihood function corresponds to the product of the individual probabilities of each component, so the log likelihood corresponds to taking logarithms, which transform products into sums and the quotients into subtractions.

The likelihood function allows us to compare models given an observation vector. If the likelihood function has a maximum value for a model, we assume that that observation comes from the class the model represent.

Theoretical explanation

As stated above, Log-Likelihood Ratio tests (LLR) is used to compares the fit of two models, one of which is nested within the other. The test statistic is twice the difference in these log-likelihoods:

$$LLR = -2 \ln \left(\frac{\text{Likelihood for null model}}{\text{Likelihood for alternative model}} \right)$$

It is advisable to know that all the distributions we'll consider belong to the same family, the various distributions in the family differing only through the value of a parameter θ (which may be a vector parameter). For example, we may consider the family of normal distributions $N(\mu, \sigma^2)$, of which each member is fully characterized by the values of μ and σ^2 . The two groups of distributions are then defined respectively by the null hypothesis and the alternative hypothesis. For example, we might want to test:

$$- H_0: \mu = \mu_0$$

against

$$- H_1: \mu \neq \mu_0$$

The hypothesis we'll consider can be indifferently simple or composite. In what follows, it will be convenient to consider that:

- H_0 does not just denote the null hypothesis, but the set of the values of the parameter θ defined by H_0 as well and, by extension, the set of distributions defined by this set of values of the parameter.
- H_1 does not just denote the alternative hypothesis, but the set of the values of the parameter θ defined by H_1 as well and, by extension, the set of distributions defined by this set of values of the parameter.

So the Likelihood Ratio Test (LRT) approach reasons as follows. Suppose that H_0 is true: the distribution that generated the sample belongs indeed to H_0 . We certainly expect the sample to exhibit a large likelihood for the distribution that generated it, and consequently we expect this likelihood to be close to the largest likelihood encountered when scanning through all the distributions in H_0 . Considering the distributions in H_1 will probably change nothing: none of these distributions generated the sample, so none of these distributions is expected to display a large likelihood for the sample. Consequently the largest likelihood in H_0 is not anticipated to be substantially smaller than the largest likelihood observed over the complete set of distributions $H_0 \cup H_1$.

Conversely, suppose that H_0 is false (and therefore that H_1 is true). The distribution that generated the sample belongs to H_1 , and not to H_0 . None of the distributions in H_0 is anticipated to exhibit a large likelihood. The largest likelihood of all is anticipated to be found for a distribution in H_1 because the distribution that generated the sample is in H_1 .

3. THE DATABASE

This chapter defines the database used for this work.

The audio database provided by the Laboratory is called Cision Database. It consists of 6100 audio files (with corresponding MFCC) in which we have all kinds of items because most of all files containing music (the element that we want to detect), long talks, as well as "jingles", silence and other events. No labeling or segmentation was done previously. So, an hard task in this work was to label each file manually. The labeling and segmentation of the audio files was carried out with a program called Transcriber in its free version 1.5.1. (see figure below).

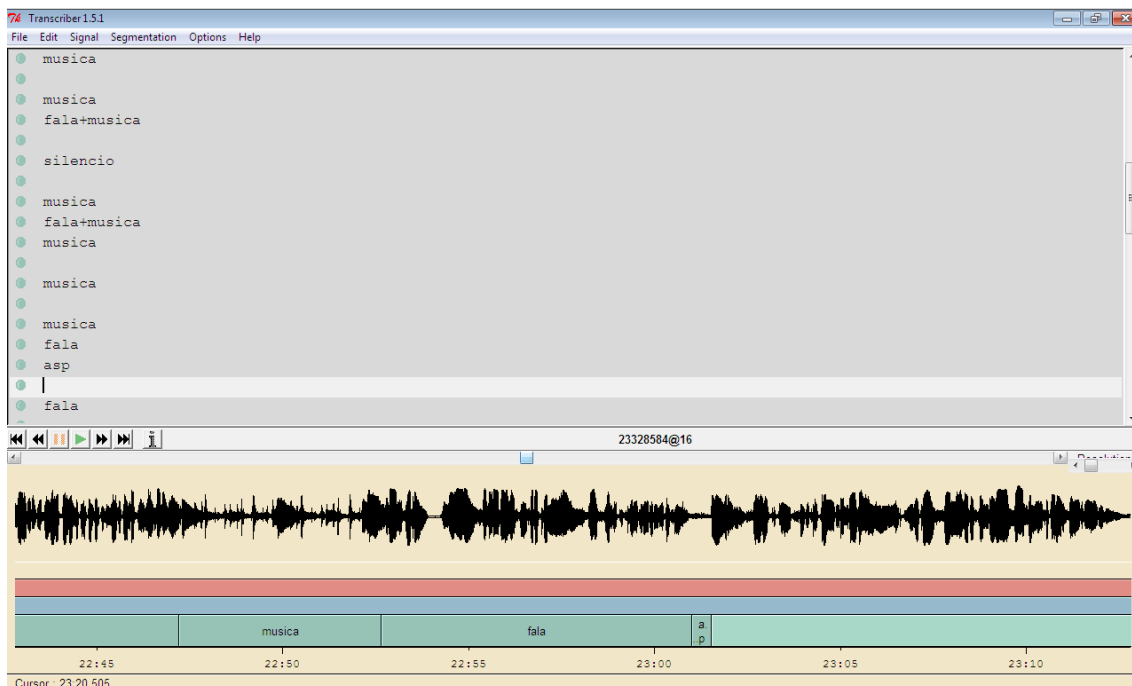


Figure 3.1-Example of a manually classified Transcriber

This means that we need to listen the whole audio file to annotate it. It is a very hard work to do. It is clear that the more files we have tagged audio, more accurate will be the results.

All audio files have .WAV extension and also they all have the same sampling frequency which is 16000 Hz, mono sound (mono), 16 bits per sample. This is lower quality than the sound in stereo but in turn takes up less space hard disk.

From the original audio database, 58 audio files were annotated. Initially, the annotation was done in 6 classes:

- speech
- music
- Speech with Music (when a speaker is speaking and sounding background music)
- Silence (No sound nor speech nor music)

- Asp (aspiration event)

Later on the class “music” is taken and all other as taken as “non-music”.

Labeling

Transcriber is a tool for assisting the manual annotation of speech signals. It provides a user-friendly graphical user interface for segmenting long duration speech recordings, transcribing them, and labeling speech turns, topic changes and acoustic conditions. It is more specifically designed for the annotation of broadcast news recordings, for creating corpora used in the development of automatic broadcast news transcription systems, but its features might be found useful in other areas of speech research.

We can see in the figure below the working environment with Transcriber software version 1.5.1. In the green circle we can see the name of the audio file we are annotating. More to the left within the yellow circle we see the labels that we are defining for the audio file. Within the blue circle we can see the audio waveform and in the bottom we can see some markings which corresponds the time for the session and for the events.

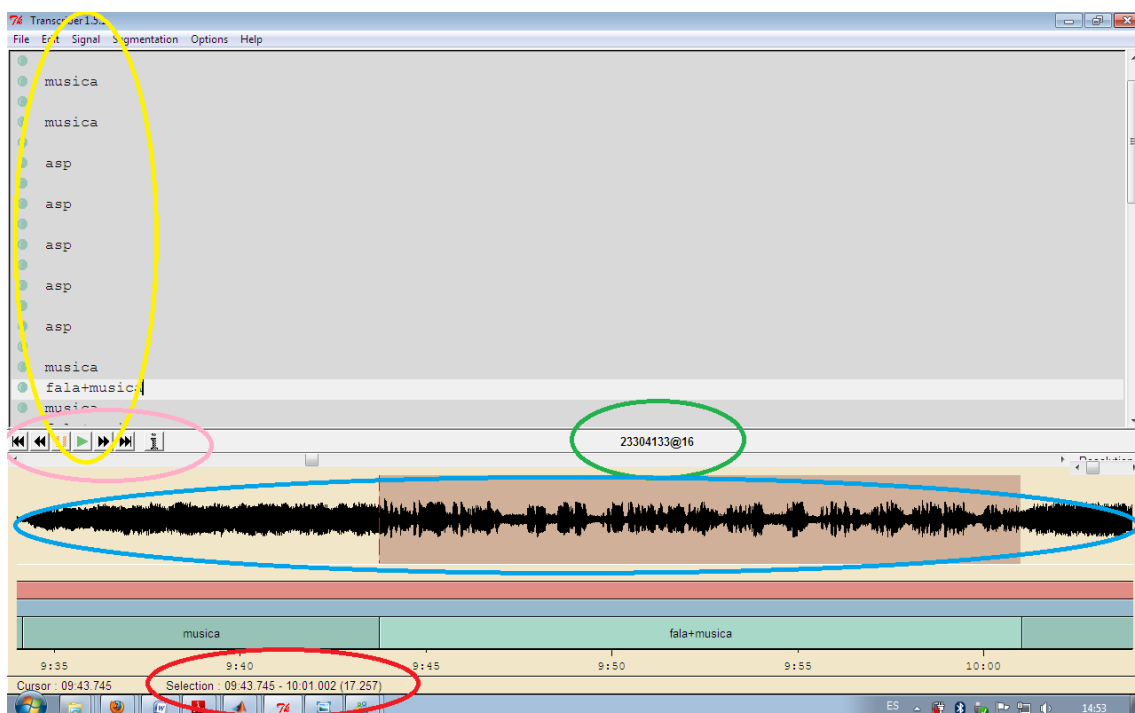


Figure 3.2- working environment of transcriber

The label "fala + music" is shaded, so the red circle indicates the duration of this segment. Finally within the red circle are the buttons "play" "pause", "stop".... to start listening, stop, pause as we like the audio signal.

As we see, the operation of this program is the most practical and simple, although, the annotation is a quite long and boring process.

Final Database

Returning to the database and pointing out that due to the large number of files in the database to use (6100), we discarded the individual treatment of each of the data files.

We chose to label approximately 58 of the files. This corresponds to only had just 30 minutes of music (29:58) and lots of hours of speech (fala). Normally, the databases are divided into two directories, one containing the files used in the training phase (Treino) and another with the files used in the evaluation phase (test).

So our first training database was this:

TREINO (58 FILES)

LABEL	LONG TIME(HH:MM:SS)
Música	29:58
Fala	1:58:46
Fala+Música	28:22

Table 3.1- treino files of the first database

As we wanted to make a correct discrimination of music, we had a problem, we needed more time of music. For that we annotate several other audio files taken from broadcast radio programs: Radio Club de Portugal (CPR) and TSF Radio News for a week in five slots.

For the evaluation database, to find that in the laboratory database, DB Split, was made practically with news segments without music what we did was to recorded in the laboratory early in the morning session of 10 minutes each, Radio Stations Club Portugal (CPR) and TSF Radio News for a week in five slots.

These times were 3 in the morning which, during the 10-minute music composed entirely by the 4 am which alternates music and fala, 5 in the morning in which there are segments of music and fala also 6 in the morning in which almost everything is fala as they are the first news of the day and 8 am which also appears some music.

The end result of the evaluation database was as follows:

TESTE (35 FILES)

LABEL	LONG TIME(HH:MM:SS)
Música	1:11:24
Fala	3:30:48
Fala+Música	17:02

Table 3.2- teste files of the first database

As the two database sets for training and evaluation were a bit unbalanced, the ideal would be to have been 70% of total music labels (of the two databases) in the training database and 30% in database assessment. What we did was to make a mix between the two databases was also necessary to re tag certain files to audio. The definitive database is as follows:

TREINO (40 FILES)

LABEL	LONG TIME(HH:MM:SS)
Música	1:30:5
Fala	4:33:22
Fala+Música	29:10

Table 3.3- treino files of the final database

TESTE (53 FILES)

LABEL	LONG TIME(HH:MM:SS)
Música	39:20
Fala	2:20:25
Fala+Música	29:31

Table 3.4- teste files of the final database

Wav file recordings obtained from radio TSF Radio Club had to be slightly modified for its subsequent handling, this is,

- change the sound from stereo to monaural (mono) ;
- change the sample rate
- extract the MFCC coefficients of these audio signals using the executable called "wave2mfc16KHz".

The label format of the annotation files was the native one: transcriber “. trs”. However, in all Matlab scripts a different label format was used, with extension “. lbl”, which is export from transcriber, as we can see in the picture below.

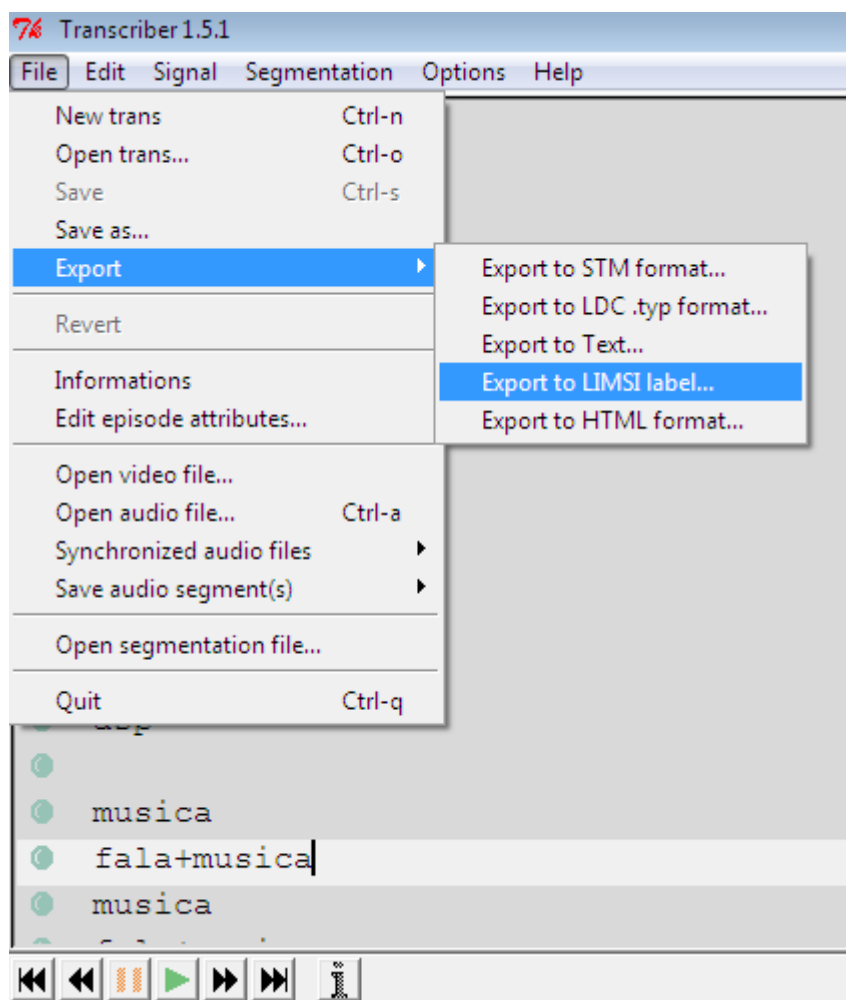
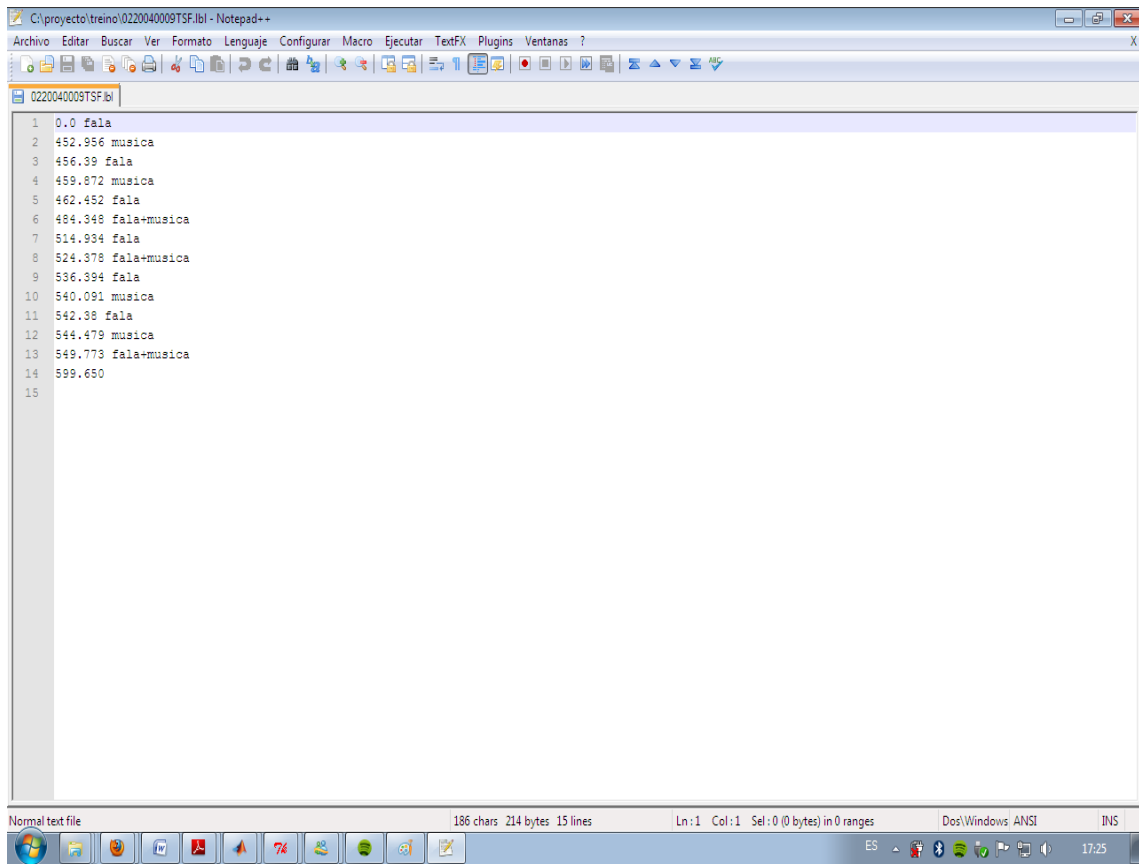


Figure 3.3- export files to lbl format

The file type Lbl has the information as can be observed in the next figure.



The image shows a Notepad++ window titled "C:\proyecto\treino\0220040009TSF.lbl - Notepad++". The window contains a list of 15 lines of text, each representing a time point and an activity. The status bar at the bottom indicates "Normal text file", "186 chars 214 bytes 15 lines", "Ln:1 Col:1 Sel:0 (0 bytes) in 0 ranges", "Dos/Windows ANSI", and "INS".

```
1 0,0 fala
2 452.956 musica
3 456.39 fala
4 459.872 musica
5 462.452 fala
6 484.348 fala+musica
7 514.934 fala
8 524.378 fala+musica
9 536.394 fala
10 540.091 musica
11 542.38 fala
12 544.479 musica
13 549.773 fala+musica
14 599.650
15
```

Figure 3.4- lbl file open with notepad

4. CLASSIFICATION AND EVALUATION

In this chapter we described the system of detection of music and its evaluation.

4.1 Design of the work.

The classification system used in this project is based on the Log-Likelihood Ratio described previously.

In the following figure there appears the scheme of the global system, which consists of three algorithms; each of them has a specific function which we will explain later, step by step.

Block diagram of the overall algorithm

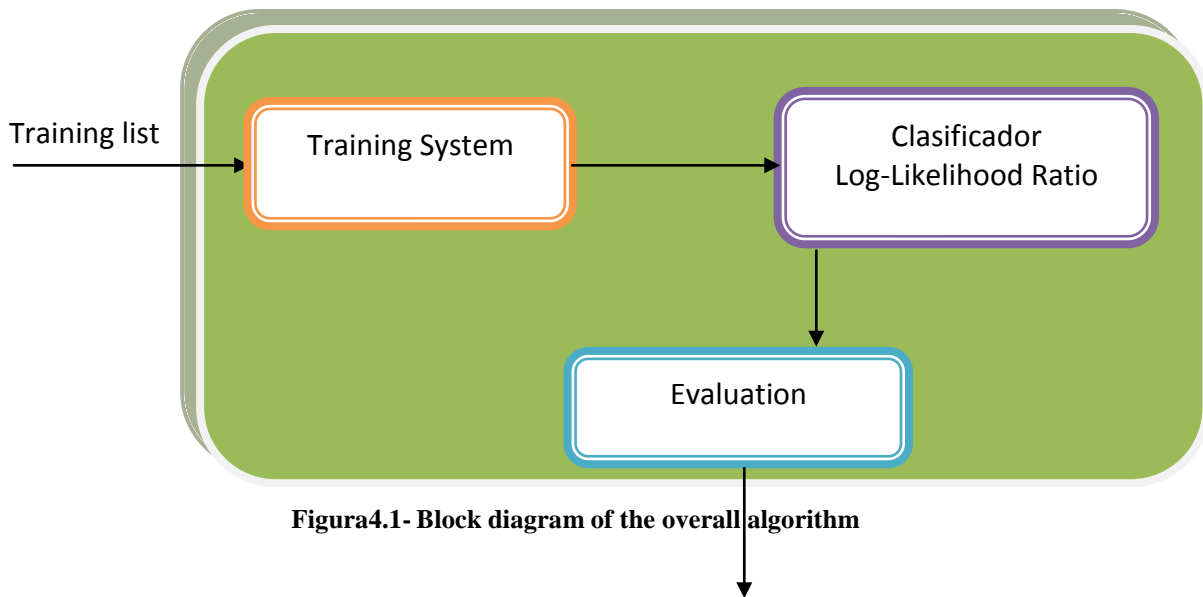


Figura4.1- Block diagram of the overall algorithm

Measures: Acc, Precision, Recall FNR

The first algorithm used in this work, called Training System, is composed by a series of functions implemented in MATLAB (note that all the work of implementation of these three algorithms have been done with the software MATLAB).

This algorithm consists in the phase of training, in which the models are created. This corresponds to collecting all labels from all files for the class under training (music or

non-music) and the computation of the averages and the matrices of covariance with the corresponding MFCC vectors.

For it, we use two principal functions:

- **Calc_mu_cov**: it is a function that calculates the average and the matrix of covariance for the audio files of a database for a given label. In our case, we are interested in the label music. This function invokes another two, one call **getlbl** which opens the list of the files of training that are with an extension .lbl and creates an array of the labels that contains this list (treino.txt) in which it indicates us the initial time (ti) and the final time (tf) for every label as well as its name (labname). The other one is called “**getMFCframes**” and its aim is to read the frames of the corresponding MFCC file from "frame_ini", the beginning of every label, up to "frame_fin", when it finishes.

- **Train**: This function trains our classifier given a list of files and labels, for which its means and matrices of covariance are calculated, for example lablist = {'music', 'non-music'}, finally the average of these means and matrices of covariance of the list of files are stored as files .mat for its use in the following algorithm, this function does not return anything.

In the following figure it is possible to estimate better the explained in this first algorithm:

Training System

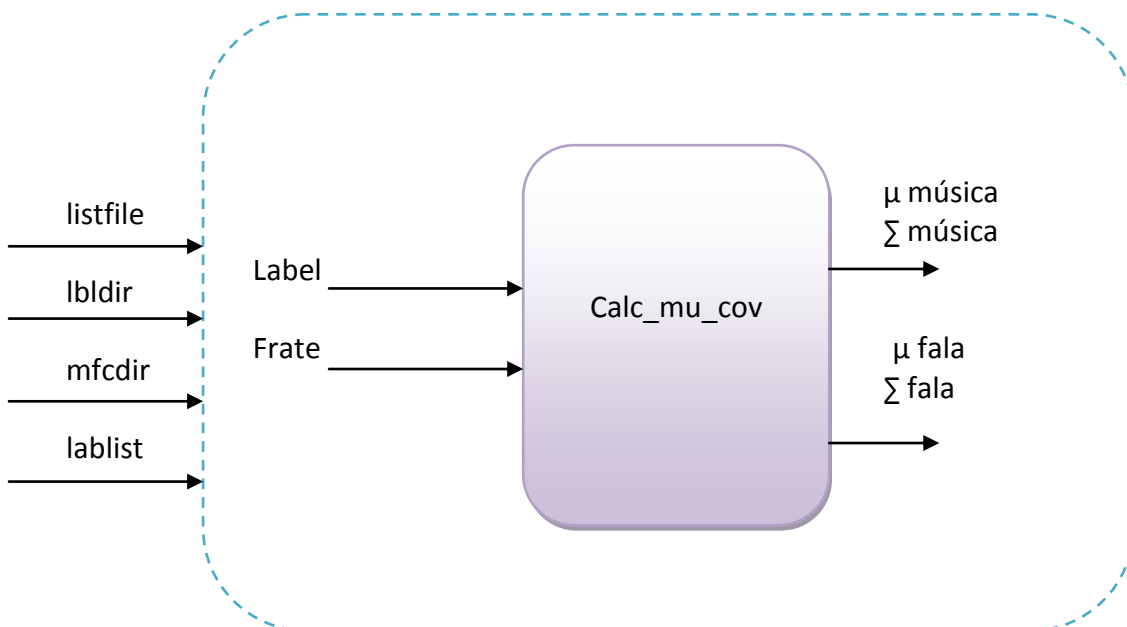


Figure 4.2- Block diagram of the training system

The mean vectors have dimensions 39x1, because we extract 39 MFCC parameters,

$$X = \begin{bmatrix} x_1 \\ \dots \\ x_n \end{bmatrix}.$$

On the other hand the covariance matrices have dimensions of 39x39.

The second algorithm corresponds to the classification or test. The log-likelihood ratio between the two models (music/non-music) is computed, given the respective Gaussian models (mean vectors and matrices of covariance) and an observation vector. Finally, it is assigned a class “music” to the input vector (observation) if the computed likelihood ratio is above a pre-defined threshold. Otherwise the class will be “non-music”.

In this phase we use two functions:

- **Loglike:** it calculates LLR (log likelihood ratio) for the hypotheses H0 and H1 given a defined model with averages μ_0 and μ_1 and the matrices of covariance C_0 and C_1 .

The equation used in this algorithm is:

$$LLR(\mathbf{x}) = (\mathbf{x} - \mathbf{u}_1)^T \mathbf{C}_1^{-1}(\mathbf{x} - \mathbf{u}_1) - (\mathbf{x} - \mathbf{u}_0)^T \mathbf{C}_0^{-1}(\mathbf{x} - \mathbf{u}_0)$$

where T means transposition. Once applied this equation to all the frame vectors of an audio segment, we see, in the figure below, that the LLR values do not discriminate well. So, we need to apply an average filter. We do this as a mean of length 2 seconds. The result is seen in Figure ???. The problem is the delay or lag introduced, which we will solve in the next function.

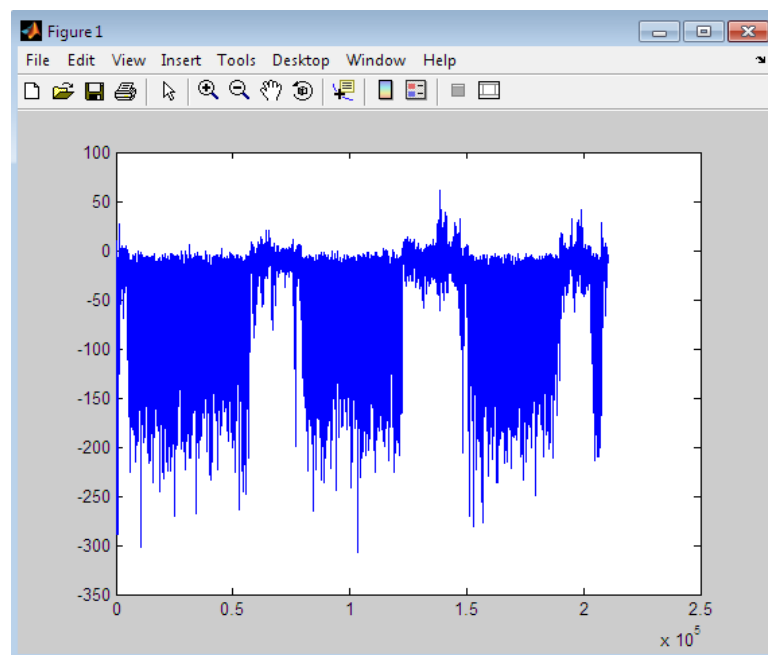


Figura4.3- LLR without applying the filter

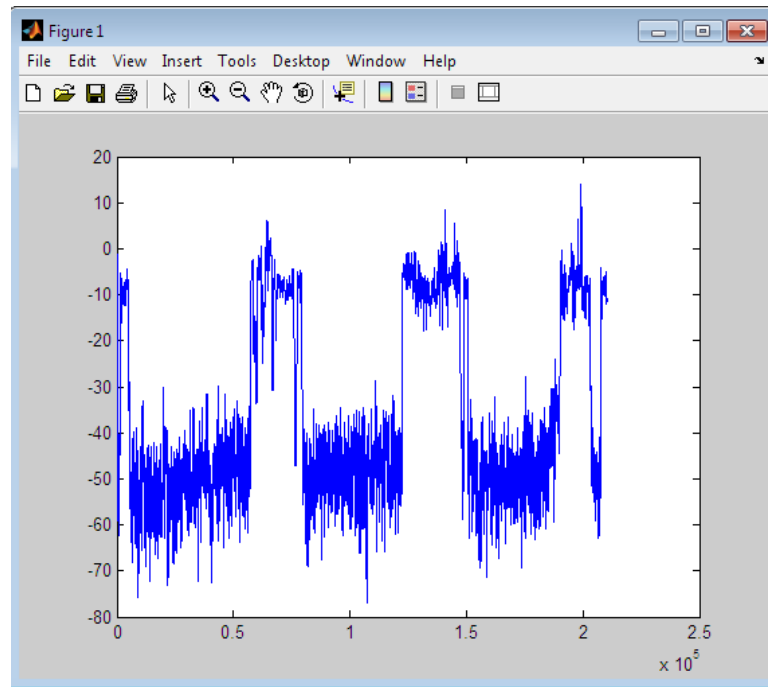


Figure 4.4- LLR filter applied

For this example, belonging to the training database, it is evident what are the segments of music and with no music..

We cannot overlook the utilization of the function **readHTK** inside **Loglike** since this function helps us to do a reading frame to frame of the coefficients MFC to be able like that to do a correct use of Log-Likelihood Ratio's technology.

to the next pass is the binarization of the filtered LLR. The aim is to obtain a vector that will be only represented by two binary numbers (0 or 1). The idea to do this work is simple, we must only to give a threshold.

All the values superior to the threshold will belong to a certain class and those which are below will belong to other class. The used threshold is -20, therefore all the values superior to this threshold will be classified as music and binary digit 1, otherwise will be classified as no-music and binary digit 0.

For it we use MATLAB's function:

- **Classify**: which classifies LLRi of agreement with the threshold Th, this function returns the classification, cl, and the indexes of the raises (up) and of the descents (down) in cl.

Let's say that:

- If $LLR_i(k) > th$, it classifies it as a 1, with a delay (delay) that is due to the filter applied in the previous function **Loglike**

- If $LLR_i(k) \leq th$, it classifies like 0, also with a lag (delay);

This lag (delay) is provoked by the filter in LLRi's calculation and for default it is of 200 frames since it is a filter of average of length 2 sg and at this work we have done it with a frame rate of 100 frames per second.

Classifier (filtering and binarization)

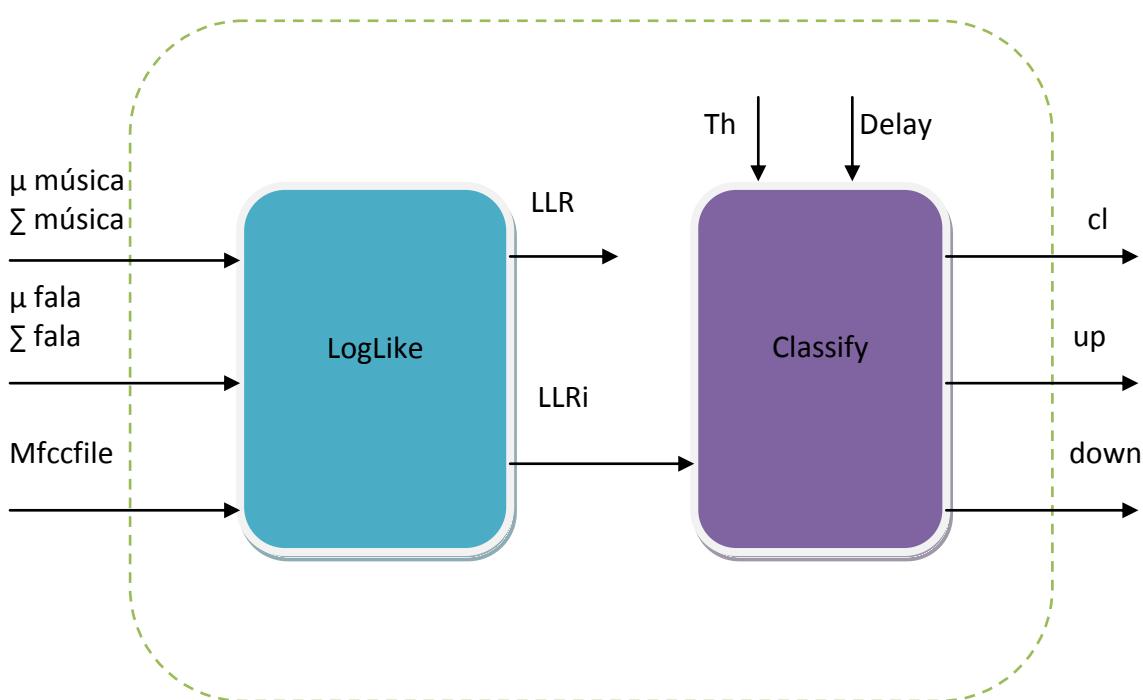


Figura4.5- Block diagram of the classifier

Finally, the third algorithm corresponds to the evaluation, for this we use the function:

- **CompareLabelsFor1File**: compares a reference transcription with the result of the classifier, for a given label `tstlab= "music"`. In this function, we call the result given by the classifier as `"hyp"` (hypothesis). Notice that:

`hyp==1` \Rightarrow class = "music"
`hyp==0` \Rightarrow class \neq "music"

Therefore, for each audio frame, t , there are four possible classification results: two positive (true) and two negative (false), according to the values of the the reference label. They are:

- CA ("Correctly Accepted") if `reflab(t) = "music"` and `hyp(t)=1`
- CR ("Correctly Rejected") if `reflab(t) \neq "music"` and `hyp(t)=0`
- FR ("Falsely Rejected") if `reflab(t) = "music"` and `hyp(t)=0` (or miss)
- FA ("Falsely Accepted") if `reflab(t) \neq "music"` and `hyp(t)=1`
(or `false_alarm`)

The output of this function is the accumulated values of CA, CR, FR, FA, for each frame of the audio file. There is also another parameter to input to this function that defines a **tolerance** number (of frames, `ftol`) to apply to the reference labels, in order to alleviate the problem of wrongly attributed initial and final temporal marks. It is usual to define this tolerance value correspond to 1 second (100 frames in our case). The tolerance parameter affects only the counts of FR and FA.

Evaluation



Figura4.6- Block diagram of the evaluator

All the implemented algorithms that have been explained in this chapter, are invoked by a global function called **do_test in()**. It is necessary to introduce a list with the files of evaluation set (test set) in the format .lbl.

General algorithm

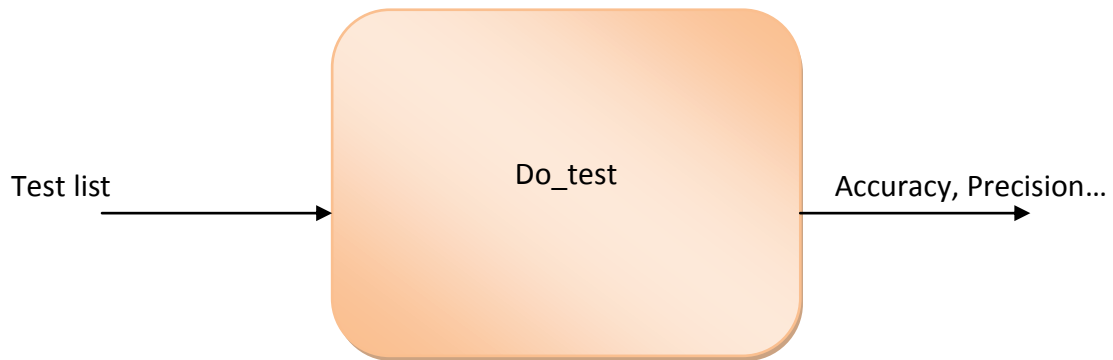


Figura4.7- Block diagram of the General algorithm

4.2 Evaluation

The evaluation system uses mainly two performance measures, called “Precision” and “Recall”. **Precision** and **Recall**, [15], are two widely used statistical classification measures. **Precision** can be seen as a measure of exactness or fidelity, whereas **Recall** is a measure of completeness.

In a statistical classification task, the precision for a class is the number of true positives (i.e. the number of items correctly labeled as belonging to the positive class) divided by the total number of elements labeled as belonging to the positive class (i.e. the sum of true positives and false positives, which are items incorrectly labeled as belonging to the class). Recall in this context is defined as the number of true positives divided by the total number of elements that actually belong to the positive class (i.e. the sum of true positives and false negatives, which are items which were not labeled as belonging to the positive class but should have been).

In a classification task, a precision score of 1 for a class A means that every item labeled as belonging to class A does indeed belong to class A (but says nothing about the number of items from class A that were not labeled correctly) whereas a recall of 1 means that every item from class A was labeled as belonging to class A (but says nothing about how many other items were incorrectly also labeled as belonging to class A).

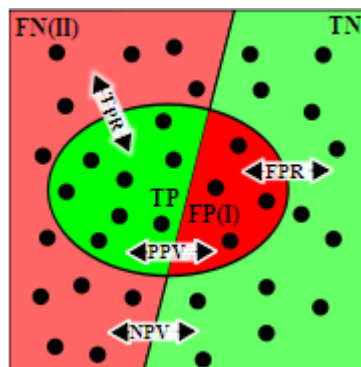


Figura4.8- Precision: horizontal arrow. Recall: diagonal arrow.

In traditional statistical hypothesis testing, the tester starts with a null hypothesis and an alternative hypothesis, performs an experiment, and then decides whether to reject the null hypothesis in favour of the alternative.

A positive result is one which accepts the null hypothesis (“music”). Doing this when the null hypothesis is true a true positive results. Doing this when the null hypothesis is false results in a false positive. A negative result is one which rejects the null hypothesis. Doing this when the null hypothesis is false a true negative results; doing this when the null hypothesis is true results in a false negative.

In the context of classification tasks, the terms true positives, true negatives, false positives and false negatives are used to compare the given classification of an item (the class label assigned to the item by a classifier) with the desired correct classification (the class the item actually belongs to).

We can see all that a bit clearer in the following figure:

		REFERENCE	
		MUSIC	NO MUSIC
HYPOTHESIS	MUSIC	TP (true positive)	FP (false positive)
	NO MUSIC	FN (false negative)	TN (true negative)

Table4.1- classification context

This corresponds to our previous nomenclature:

Standard Nomenclature	Nomenclature used in this work
TP (True Positive)	CA (Correctly Accepted)
FP (false positive)	FA (Falsely Accepted)
FN (false negative)	FR (Falsely Rejected)
TN (true negative)	CR (Correctly Rejected)

Table4.2- nomenclature used in this work

We can summarize the situation in the following way:

- CA ("Correctly Accepted") occurs if the classifier classifies a frame as music and the label says "music" in that frame;
- CR ("Correctly Rejected") occurs if the classifier classifies a frame as not music and the label says something (but not "music");
- FR ("Falsely Rejected") occurs if the classifier classifies a frame as not music and the label says "music".
- FA ("Falsely Accepted") occurs if the classifier classifies a frame as "music" and the label says something (but not "music").

Whith these counts we can evaluate the performance measures Precision and Recall:

$$Precision = \frac{CA}{(CA + FA)}$$

$$Recall = \frac{CA}{(CA + FR)}$$

Two other measures are often used in classification results: the accuracy and the false negative rate, which are evaluated according to the following equations:

$$Accuracy = \frac{(CA + CR)}{Total\ Frames}$$

$$FNR(False\ Negative\ Rate) = \frac{CR}{(CR+FA)}.$$

5. RESULTS AND CONCLUSIONS

This chapter presents the results obtained with the classification system. The results are discussed and future work is proposed.

The main results are drawn in the following tables. We made a series of tests by changing the tolerance (Tolerance) and threshold (Th).

Tolerance=0 seg; Th= -20	
Precision	73,34%
Recall	66,4%
Accuracy	99,02%
FNR	99,58%

Table5.1- experiment 1

Tolerance=1seg; Th = -20	
Precision	76,57%
Recall	72,64%
Accuracy	99,23%
FNR	99,65%

Table5.2- experiment 2

Tolerance=2seg ; Th = -20	
Precision	79,96%
Recall	78,28%
Accuracy	99,4%
FNR	99,71%

Table5.3- experiment 3

Tolerance=0; Th= -25	
Precision	59,9%
Recall	78,09%
Accuracy	98,74%
FNR	99,1%

Table5.4- experiment 4

Tolerance=1seg ; Th= -25	
Precision	61,99%
Recall	84,9%
Accuracy	98,95%
FNR	99,17%

Table5.5- experiment 5

Tolerance=2seg ; Th= -25	
Precision	64,16%
Recall	90,9%
Accuracy	99,12%
FNR	99,24%

Table5.6- experiment 6

Tolerance=0 ; Th= -30	
Precision	41,5%
Recall	82,38%
Accuracy	97,73%
FNR	97,99%

Table5.7- experiment 7

Tolerance=1sg ; Th= -30	
Precision	42,52%
Recall	89,14%
Accuracy	97,93%
FNR	98,07%

Table5.8- experiment 8

Tolerance=2seg ; Th= -30	
Precision	43,57%
Recall	95,06%
Accuracy	98,11%
FNR	98,15%

Table5.9- experiment 9

5.2 Conclusions

Obviously, the results improve when we increase the tolerance. The Precision and Recall also increases because falsely accepted (FA) and the falsely rejected (FR) counts diminish.

On the other hand, when the threshold increases, the number of falsely accepted (FA) frames reduces considerably. However, the FR number increases. This means that Precision rate increases and Recall rate decreases. In the other way, if we decrease the threshold, the reverse occurs.

The results, although not excellent, are quite good if we compare it with the ones presented in the literature. We can say that the present work corresponds to a preliminary and exploratory work in area of music/speech discrimination. The main result of this work may be the algorithms and scripts produced to do the classification of audio files, as well as an annotated database.

5.3 Future work

The audio classification is a research field where much remains to be done. In this case, much better results probably could be obtained. A key improvement would be better models of music and no-music, for instance, GMM with dozens of Gaussians. More robust classifiers, such as SVMs, could also be used with advantage.

The study of features more adequate for recognition of musical styles, could also be used. It would take a broader database and more complex parameterization.

References

- [1] J. Jensen, M. Christensen, D. Ellis, S. Jensen, 2009. Quantitative Analysis of a Common Audio Similarity Measure, *IEEE Trans. Audio Speech and Lang.*, Vol. 17, NO. 4, May 2009.
- [2] Saunders, J. 1996. Real-time discrimination of broadcast speech/music. In *Proc. IEEE ICASSP '96*, Atlanta, GA, pp. 993–996.
- [3] Scheirer, E. and M. Slaney. 1997. Construction and evaluation of a robust multifeature speech/music discriminator. *Proc. IEEE ICASSP '97*, Munich, Germany, pp. 1331–1334.
- [4] Carey, M. J., E. S. Parris, and H. Lloyd-Thomas. 1999. A comparison of features for speech, music discrimination. *Proc. IEEE ICASSP '99*, Phoenix, AZ, pp. 1432–1435.
- [5] El-Maleh, K., M. Klein, G. Petrucci, and P. Kabal. 2000. Speech/music discrimination for multimedia applications. *Proc. IEEE ICASSP 2000* 6:2445–2448.
- [6] Zhang, T. and J. Kuo. 2001. Audio content analysis for online audiovisual data segmentation and classification. *IEEE Trans. on Speech and Audio Processing* 9(4):441–457.
- [7] Stefan Karneböck, "Discrimination between speech and music based on a low frequency modulation feature", European Conference on Speech Communication and Technology, September 3-7, 2001, Allborg, Denmark, pp.1891-1894,
- [8] Julien Piquier, Christine Sénac and Régine André-Obrecht, "Speech and Music Classification in Audio Documents", ICASSP 2002
- [9] A Hierarchical Architecture for Audio Segmentation in a Broadcast News Task, Mateu Aguilo, Taras Butko, Andrey Temko, Climent Nadeu, Department of Signal Theory and Communications, TALP Research Center Universitat Politècnica de Catalunya, Barcelona, Spain; Proceedings of the Iberian SLTech 2009
- [10] "A tutorial on Support Vector Machines for Pattern Recognition". Christopher J. C. Burges. Kluwer Academic Publishers, Boston

[11] SVM (Support Vector Machines), *Wikipedia*

[12] Redes neuronales , *Ana Bollella*

[13] *K*-nn (K nearest neighbors), *Wikipedia*

[14] Likelihood Ratio Test, *Wikipedia*

[15] Precision and Recall, *wikipedia*



Detecção Automática de Música

Universidade De Coimbra
Faculdade De Ciências e Tecnologia
Mestrado Integrado em Engenharia Electrotécnica e
de Computadores

**Pablo David Young
Zubizarreta**



Summary

- INTRODUCTION
- AUDIO CLASSIFICATION
- THE DATABASE
- EVALUATION
- RESULTS AND CONCLUSIONS

Introduction

Audio Classification
The Database
Evaluation
Results and Conclusions

Problem Definition

Objetives

multimedia
content



To order
and to
classify

Manual Search (lot of work)

Introduction

Audio Classification
The Database
Evaluation
Results and Conclusions

Problem Definition

Objetives

multimedia
content



Automatic
organization

Speech

Music

Noise or sound
effects

Silence

Easier Search

Introduction

Audio Classification
The Database
Evaluation
Results and Conclusions

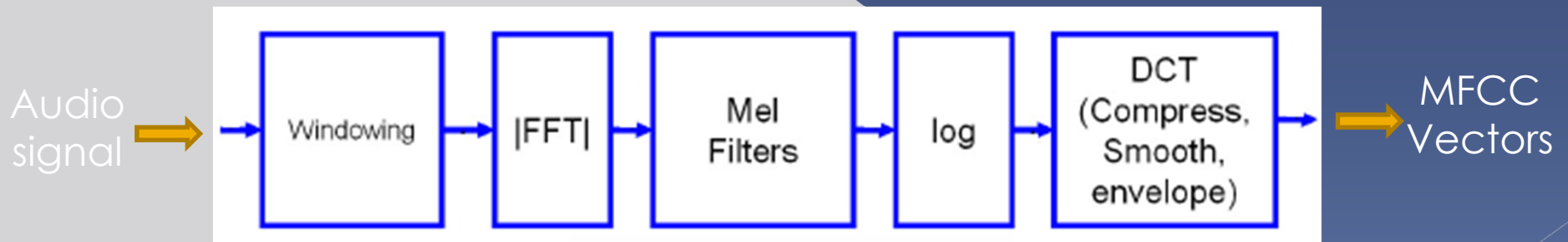
Problem Definition Objectives

- Annotation of audio recordings (several hours) in order to define an audio database for training and testing.
 - The database includes two sets of audio files, one set for model training and another to test the detection system.
- Development of algorithms to discriminate music from other sounds.
 - The signal corresponds to recordings of broadcast radio and TV programs.

Introduction
Audio Classification
The Database
Evaluation
Results and Conclusions

Parameterization
Modelos
Classifier

- The parameterization technique is MFCC (Mel Frequency Cepstral Coefficients).
 - > Nowadays, is one of the best possibilities to parameterize the audio signal.
 - > MFCC's are used to represent some characteristics of the human auditory perception system, namely the non-linear frequency resolution.



A MFCC vector contains 39 coefficients.

Introduction
Audio Classification
 The Database
 Evaluation
 Results and Conclusions

Parameterization
 Modelos
 Classifier

13 MFCC
 coefficients



first derivative
 coefficients
 (coefficients for
 Delta speed)



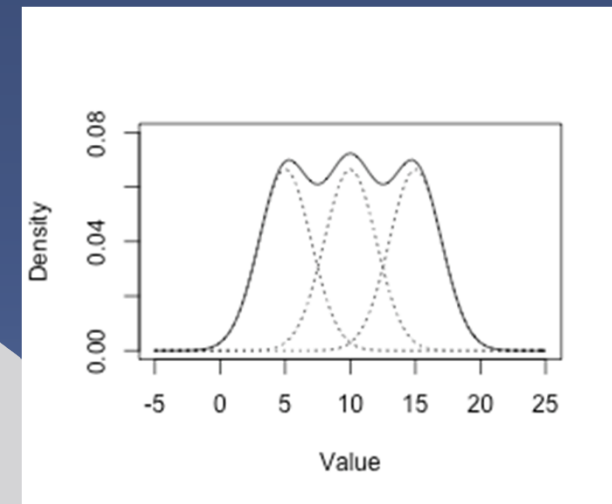
		t_0	...	t_{n-1}	t_n	t_{n+1}	...
m coef. MFCC	0			C_0	C_0	C_0	
	1			C_1	C_1	C_1	
	⋮						
	m			C_6	C_6	C_6	
m coef. Delta	0				$C_0(t_{n+1}) - C_0(t_{n-1})$		
	1				$C_1(t_{n+1}) - C_1(t_{n-1})$		
	⋮						
	m				$C_6(t_{n+1}) - C_6(t_{n-1})$		

Introduction
Audio Classification
The Database
Evaluation
Results and Conclusions

Parameterization
Models
Classifier

◎ Basic techniques of pattern recognition :

- Support Vector Machines (SVM)
- Neural Networks (NN)
- Gaussian Mixture Models (GMM)



Gaussian Mixture Models (GMM)

- Individual Gaussian components in a GMM have ability to shape some acoustic general classes.
- The unimodal Gaussian model represents a distribution of characteristics with only two parameters:
 - mean vector (39×1)
 - Covariance matrix (39×39)
- Two models needed:
 - Music model
 - Non-music model



Discriminated using a Likelihood Ratio

Log-Likelihood Ratio

➤ LLR



$$LLR(\mathbf{x}) = (\mathbf{x} - \mathbf{u}_1)^T \mathbf{C}_1^{-1}(\mathbf{x} - \mathbf{u}_1) - (\mathbf{x} - \mathbf{u}_0)^T \mathbf{C}_0^{-1}(\mathbf{x} - \mathbf{u}_0)$$

➤ Classification:

➤ $LLR > \text{Threshold} \Rightarrow \text{hypothesis} = \text{"music"}$

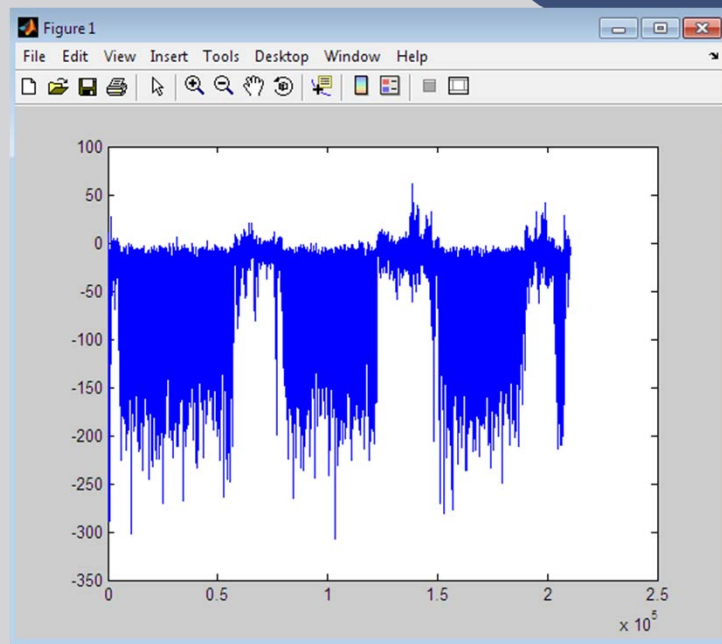
➤ $LLR \leq \text{Threshold} \Rightarrow \text{hypothesis} = \text{"non-music"}$

➤ Every frame;

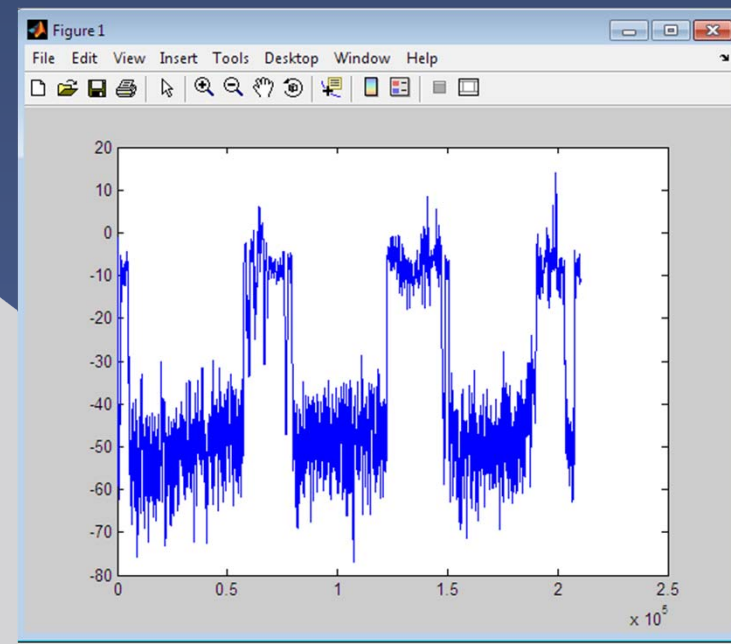
➤ Smoothing: moving average over two seconds

Introduction
Audio Classification
The Database
Evaluation
Results and Conclusions

Parameterization
Modelos
Classifier



LLR unfiltered



LLR filtered

○ For each audio frame, t , there are four possible classification results:

- CA ("Correctly Accepted") if $\text{reflab}(t) = \text{"music"}$ and $\text{hyp}(t)=1$
- CR ("Correctly Rejected") if $\text{reflab}(t) \neq \text{"music"}$ and $\text{hyp}(t)=0$
- FR ("Falsely Rejected") if $\text{reflab}(t) = \text{"music"}$ and $\text{hyp}(t)=0$ (or miss)
- FA ("Falsely Accepted") if $\text{reflab}(t) \neq \text{"music"}$ and $\text{hyp}(t)=1$

Introduction
Audio Classification
The Database
Evaluation
Results and Conclusions

Parameterization
Modelos
Classifier

		Reference	
		Music	No Music
Hypothesis	Music	CA	FA
	No Music	FR	CR

These counts will be used to evaluate the classification system.

- The evaluation system uses four performance measures:

➤ Precision



$$Precision = \frac{CA}{(CA + FA)}$$

➤ Recall



$$Recall = \frac{CA}{(CA + FR)}$$

➤ Accuracy



$$Accuracy = \frac{(CA + CR)}{Total\ Frames}$$

➤ FNR (False Negative Rate)



$$FNR(False\ Negative\ Rate) = \frac{CR}{(CR+FA)}$$

Introduction
Audio Classification
The Database
Evaluation
Results and Conclusions

Introduction
Labeling(Transcriber)
Database

- The audio database is Cision Database(DB):
 - 6100 audio files (.WAV) → frate= 16 kHz with mono sound
 - 6100 MFCC files (.MFC)
- Radio recordings:
 - 35 audio files

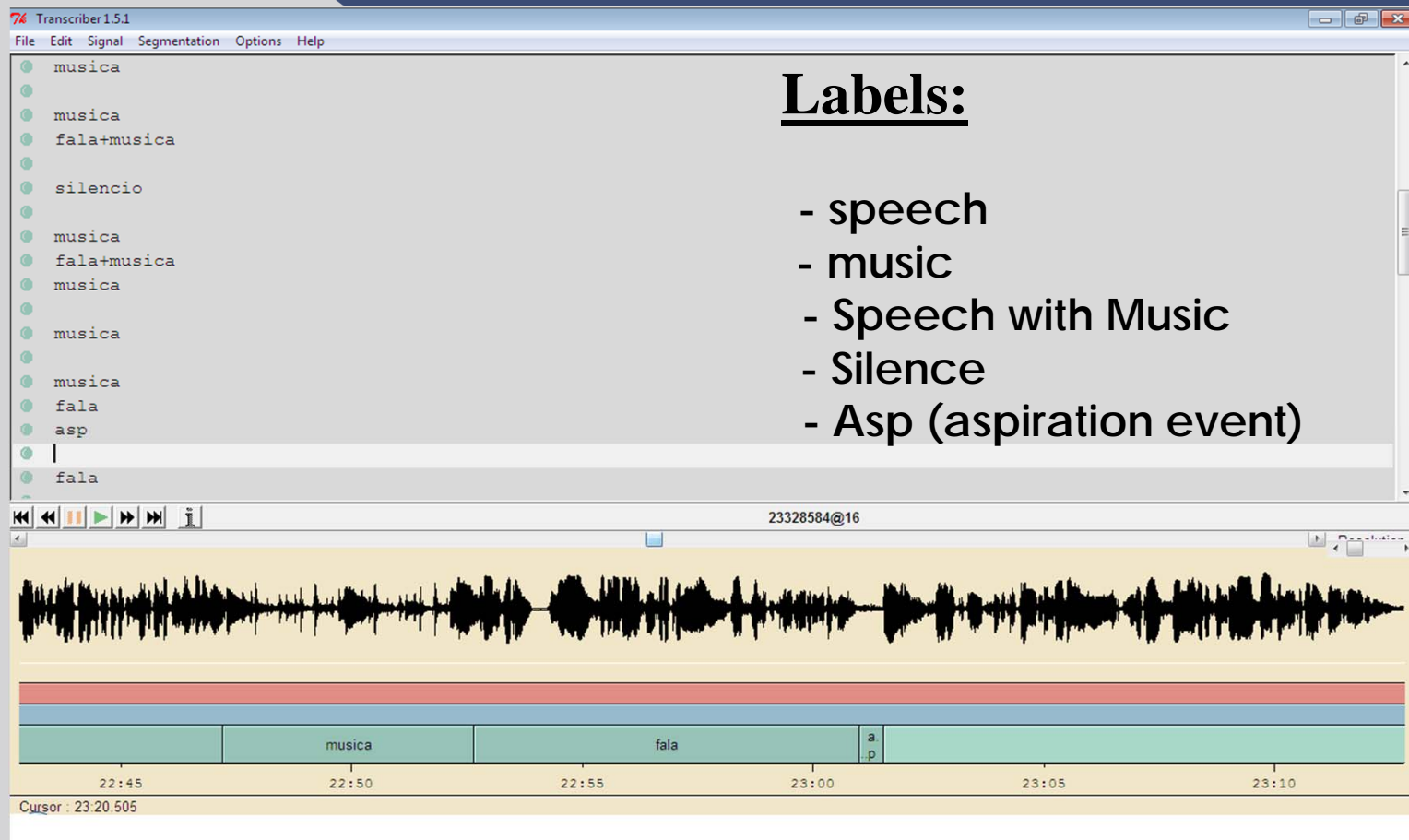


Introduction
Audio Classification
The Database
Evaluation
Results and Conclusions

Introduction
labeling(Transcriber)
Database

Labels:

- speech
- music
- Speech with Music
- Silence
- Asp (aspiration event)



Introduction
Audio Classification
The Database
Evaluation
Results and Conclusions

Introduction
Labeling(Transcriber)
Database

TREINO (53 FILES)

LABEL	LONG TIME(HH:MM:SS)
Música	1:30:5
Fala	4:33:22
Fala+Música	29:10

TREINO (40 FILES)

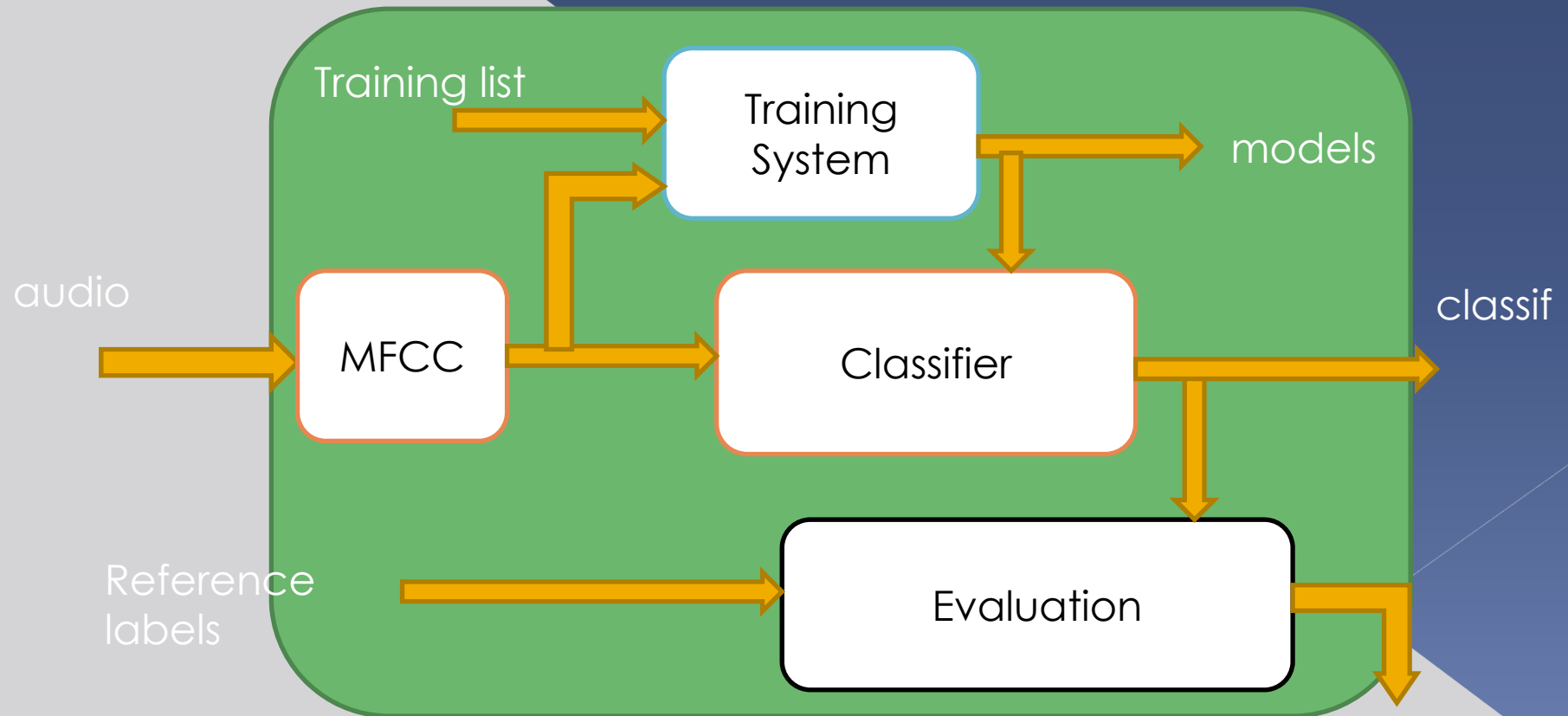
LABEL	LONG TIME(HH:MM:SS)
Música	39:20
Fala	2:20:25
Fala+Música	29:31

58 files – Cision
35 files – radio recordings
~ 50 hours of audio

Introduction
Audio Classification
The Database
Evaluation
Results and Conclusions

Diagrams and algorithms
Tolerance

Block Diagram of the Global System



Measures: Acc, Precision, Recall, FNR

Introduction
Audio Classification
The Database
Evaluation
Results and Conclusions

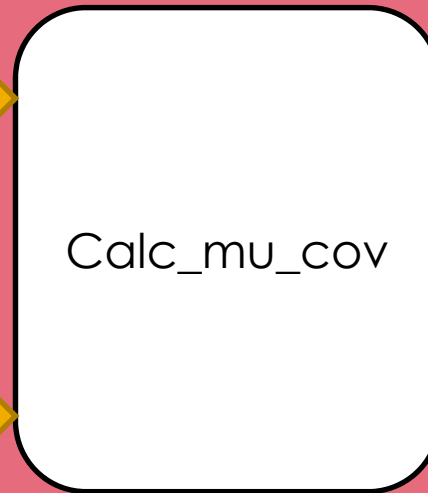
Diagrams and algorithms
Tolerance

Training System

Training list



MFCC



μ música
 Σ música



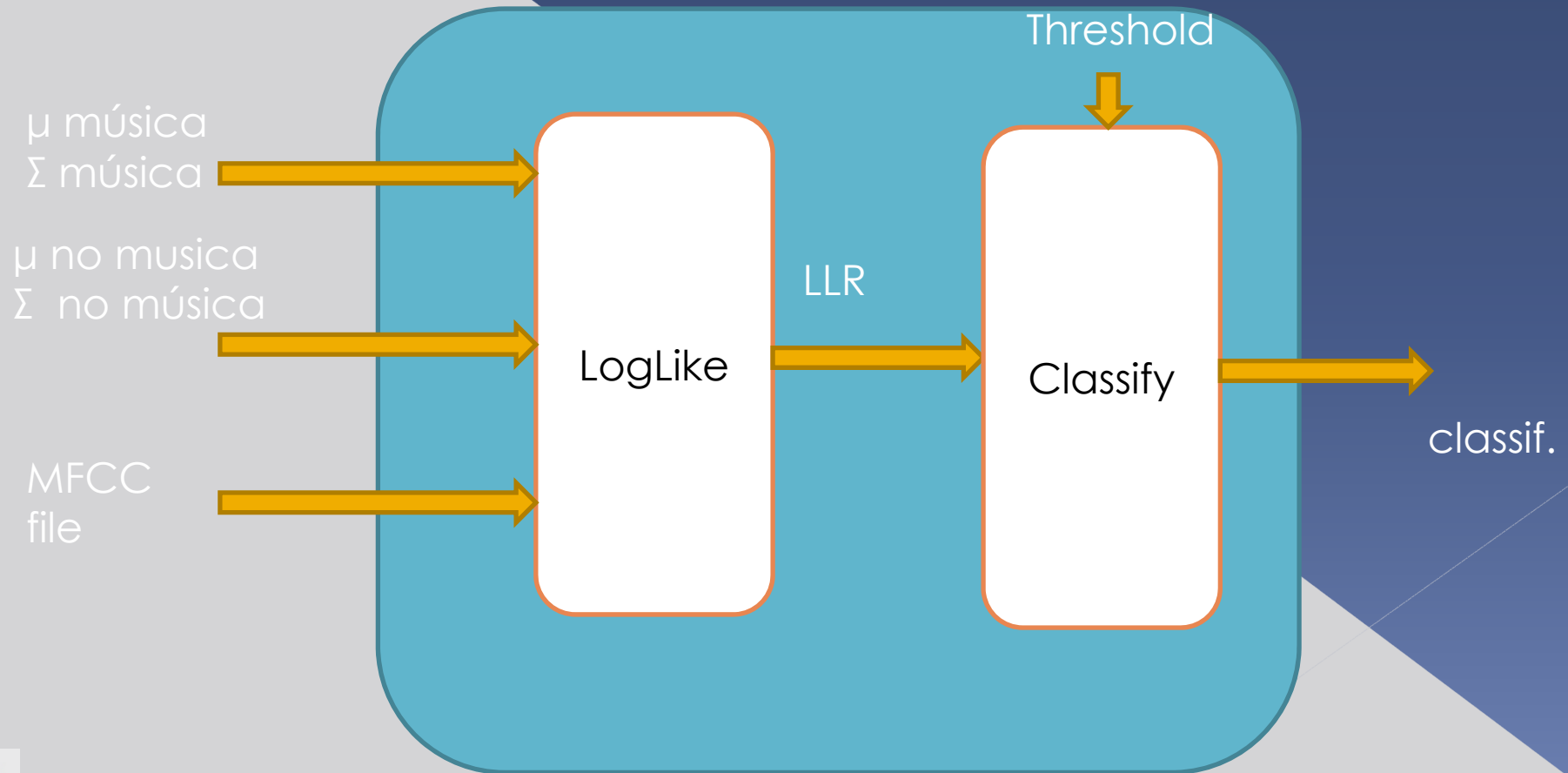
μ no musica
 Σ no música



Introduction
Audio Classification
The Database
Evaluation
Results and Conclusions

Diagrams and algorithms
Tolerance

Classifier System



Introduction
Audio Classification
The Database
Evaluation
Results and Conclusions

Diagrams and algorithms
Tolerance

Evaluation System

references



classif



tstlab



ftol



Compare Labels



CA,CR,FA,FR

Acc, Precision, Recall, FNR

- Tolerance:
 - to alleviate the problem of wrongly attributed initial and final temporal marks (in the reference labels).
 - It is usual to define this tolerance value correspond to 1 second (100 frames in our case).
- The tolerance parameter affects only the errors: FR and FA.

Introduction
Audio Classification
The Database
Evaluation
Results and Conclusions

Results
Conclusions
Future Work

Tolerance=0 seg; Th=-20	
Precision	73,34%
Recall	66,4%
Accuracy	99,02%
FNR	99,58%

Tolerance=1seg; Th = -20	
Precision	76,57%
Recall	72,64%
Accuracy	99,23%
FNR	99,65%

Tolerance=2seg; Th = -20	
Precision	79,96%
Recall	78,28%
Accuracy	99,4%
FNR	99,71%

Introduction
Audio Classification
The Database
Evaluation
Results and Conclusions

Results
Conclusions
Future Work

Tolerance=0; Th= -25	
Precision	59,9%
Recall	78,09%
Accuracy	98,74%
FNR	99,1%

Tolerance=1seg; Th= -25	
Precision	61,99%
Recall	84,9%
Accuracy	98,95%
FNR	99,17%

Tolerance=2seg; Th= -25	
Precision	64,16%
Recall	90,9%
Accuracy	99,12%
FNR	99,24%

Introduction
Audio Classification
The Database
Evaluation
Results and Conclusions

Results
Conclusions
Future Work

Tolerance=0 ; Th=-30	
Precision	41,5%
Recall	82,38%
Accuracy	97,73%
FNR	97,99%

Tolerance=1sg ; Th=-30	
Precision	42,52%
Recall	89,14%
Accuracy	97,93%
FNR	98,07%

Tolerance=2seg ; Th=-30	
Precision	43,57%
Recall	95,06%
Accuracy	98,11%
FNR	98,15%

Introduction
Audio Classification
The Database
Evaluation
Results and Conclusions

Results
Conclusions
Future Work

- The results, although not excellent, are quite good if we compare it with the ones presented in the literature.
- The present work corresponds to a preliminary and exploratory work in area of music/speech discrimination:
- Algorithms and scripts produced to do classification of audio files.
- Annotated database.

Introduction
Audio Classification
The Database
Evaluation
Results and Conclusions

Results
Conclusions
Future Work

- A key improvement would be better models of music and no-music, for instance, GMM.
- More robust classifiers, such as SVMs or ANN.
- The study of features more adequate for recognition of musical styles.

OBRIGADO

FIM