Universidade Federal de Pernambuco

Centro de Informática

Automatic Speaker Verification Using Adapted Gaussian Mixture Models

Final Term Paper

Eduardo Martins Barros de Albuquerque Tenório

March 3, 2015

Declaration

This paper is a presentation of my research work, as partial fulfillment of the requirement for the degree in Computer Engineering. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions.

The work was done under the guidance of Prof. Dr. Tsang Ing Ren, at Centro de Informática, Universidade Federal de Pernambuco, Brazil.

Eduardo Martins Barros de Albuquerque Tenório

In my capacity as supervisor of the candidate's paper, I certify that the above statements are true to the best of my knowledge.

Prof. Dr. Tsang Ing Ren

March 3, 2015

Acknowledgements

I am thankful to my family, for the support and patience during the graduation,
To my adviser, Tsang Ing Ren, for the guidance,
To Cleice Souza, for the previous readings and suggestions,
To Sérgio Vieira, Hector Pinheiro and James Lyons, for clarify many of my questions.

Live long and prosper

Vulcan salute

Abstract

TODO escrever o abstract após terminar tudo (após a conclusão).

Contents

1 Introduction **1**
- 1.1 Speaker Recognition . . . 2
- 1.2 Why Gaussian Mixture Models? . . . 2
- 1.3 Objectives . . . 3
- 1.4 Document Structure . . . 3

2 Speaker Recognition Systems **5**
- 2.1 Basic Concepts . . . 5
  - 2.1.1 Utterance . . . 5
  - 2.1.2 Features . . . 5
- 2.2 Speaker Identification . . . 6
  - 2.2.1 Training . . . 6
  - 2.2.2 Test . . . 6
- 2.3 Speaker Verification . . . 7
  - 2.3.1 Likelihood Ratio Test . . . 7
  - 2.3.2 Training . . . 7
  - 2.3.3 Test . . . 8

3 Feature Extraction **9**
- 3.1 Mel-Frequency Cepstral Coefficient . . . 9
  - 3.1.1 The Mel Scale . . . 10
  - 3.1.2 Extraction Process . . . 11

4 Gaussian Mixture Models **17**
- 4.1 Expectation-Maximization . . . 18
- 4.2 Universal Background Model . . . 19

5 Experiments **21**
- 5.1 Identification . . . 21
- 5.2 Verification . . . 22
  - 5.2.1 Unisex . . . 22
  - 5.2.2 Gender . . . 22

6 Conclusion **23**

A Results for Adapt-GMM-UBM **25**

B Results for C-Adapt-GMM-UBM **27**

C Codes **29**

1. Introduction

The increasing popularity and the intensive usage of computational systems in the everyday of modern life create the need for easier and less invasive forms of user recognition. While enter a hard to memorize password in a terminal and identify a person placing a human to listen to telephone calls are the status quo for respectively authentication and identification, voice biometrics presents itself as a continuing improvement alternative. Passwords can be forgotten and people are biased and unable to be massive trained, but the unique characteristics of a person's voice combined with an automatic speaker recognizer (ASR) outperform any "manual" attempt.

Speech is the most natural way humans have to communicate, being incredibly complex and with numerous specific details related to its producer, *Bimbot et. al.* [1]. Therefore, it is expected an increasing usage of vocal interfaces to perform actions such as computer login, voice search (e.g., Apple Siri, Google Now and Samsung S Voice) and identification of speakers in a conversation and its content. At present, fingerprint biometrics is adopted in several solutions (e.g., ATMs, *Wang & Wu* [2]), authentication through facial recognition comes as built-in software for average computers and iris scan was adopted for a short time by United Kingdom and permanently by United Arab Emirates border controls, *Sasse* [3], *Raisi & Khouri* [4]. These examples indicate a near future where biometrics are common, with people talking to the computer and receiving concise answers, and cash withdrawals allowed via a combination of speaker verification, correctly dictated captcha and other techniques.

Current commercial products based on voice technology (e.g., Dragon Naturally Speaking, KIVOX and VeriSpeak) are usually intended to perform either **speech recognition** (*what* is being said) or **speaker recognition** (*who* is speaking). Voice search applications are designed to determine the content of a speech, usually with no concern about who the speaker is or if there is more than one, while computer login and telephone fraud prevention supplement a memorized personal identification code with speaker verification, *Reynolds* [5], not interested on the message spoken. Few applications perform both processes, such as automatic speaker labeling of recorded meetings, that transcribes what each person is saying. To achieve this goal, numerous voice processing techniques have become known in industry and academy, e.g., Natural Language Processing (NLP), Hidden Markov Models (HMMs) and Gaussian Mixture Models (GMMs). Although all of these are interesting state-of-the-art techniques, the subject covered by this paper is a subarea of speaker recognition and only a small subset of these techniques will be unraveled.

1.1 Speaker Recognition

As stated in *Reynolds & Campbell* [6], speaker recognition may be divided in two subareas. The first is **speaker identification**, aimed to determine the identity of a speaker from a non-unitary set of known speakers. This task is also named speaker identification in **closed set**. In the second, **speaker verification**, the goal is to determine if a speaker is who he or she claims to be, not an imposter. As the set of imposters is unknown, this is an **open set** problem. An intermediate task is **open set identification**, when an "unmatched class" is added in order to categorize all unknown speakers.

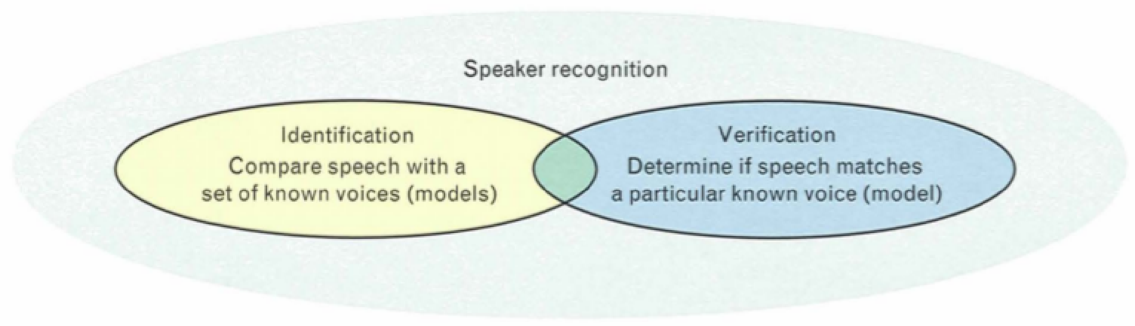
Speaker recognition
Identification
Compare speech with a set of known voices (models)
Verification
Determine if speech matches a particular known voice (model)

Figure 1.1: Speaker identification and speaker verification are different, but not entirely, *Reynolds* [5].

The text inputed may have constraints, such as type (e.g., digits and letters), number of words used (e.g., one word or sentences) and etc.. In **text-dependent** systems the content of the speech is relevant to the evaluation, and the testing texts must belong to the training set (not necessarily be the entire set), *Hébert* [7]. A change in the training inputs demands a complete new training section. **Text-independent** systems have no restrictions to the message in both sets, with the non-textual characteristics of the user's voice (e.g., pitch and accent) being the important aspects to the evaluator. These characteristics are presented in different sentences, usage of foreign languages and even gibberish. Between the extremes in constraints falls the **vocabulary-dependent system**, which constrains the speech to come from a limited vocabulary (e.g., digits) from which test words or phrases are selected (e.g., "two", "one-two-three"), *Reynolds* [5].

1.2 Why Gaussian Mixture Models?

The focus of this paper is on **text-independent speaker verification**, and as independence of the spoken context is a key characteristic of the problem, the most appropriate approach is to consider the training data as a stochastic variable. The best suited distribution to represent random data is the gaussian (or normal), leading to the choice of GMMs to model the ASR.

The verification system is constructed using an Universal Background Model-Gaussian Mixture Model (UBM-GMM) to represent all speakers as a single background and an adaptation of the UBM-GMM for each speaker throught Bayesian Adaptation, *Brown, Lee and Spohrer* [8]. Two types of adaptation are performed on the UBM-GMM, Adapt-GMM-UBM and C-Adapt-GMM-UBM, both described in details in Chapter 4. Finally, a likelihood ratio test is used to evaluate a speech signal and decide if it belongs or not to the claimed speaker (see Chapter 2).

1.3 Objectives

This study is aimed to implement an ASR and analyze the following:

- False detection and false rejection rates using a DET curve, *Martin et. al.* [9].
- Accuracy for different sizes of mixtures and features.
- Accuracy of Adapt-GMM-UBM and C-Adapt-GMM-UBM.

1.4 Document Structure

Chapter 2 contains basic information about speaker recognition, as well as the basic architecture of a speaker verification system. The feature extraction process is explained in Chapter 3, from the reasons for its use to the chosen technique (Mel-Frequency Cepstral Coefficient, MFCC). In Chapter 4 the theory of GMM, UBM, Adapt-GMM-UBM and C-Adapt-GMM-UBM is presented. Experiments are described in Chapter 5, as well as its results. Finally, Chapter 6 concludes the study. Furthermore, important contents are presented in the appendices. Appendix A shows the results and curves for Adapt-GMM-UBM, Appendix B for C-Adapt-GMM-UBM and Appendix C contains the codes, written in Python 3, used to perform the study.

2. Speaker Recognition Systems

The process of speaker recognition lies on the field of pattern classification, with the speaker's utterance (a speech signal) as input for a classifier. This decision may be, given a speech signal $\boldsymbol{Y}$ produced by a speaker $\mathcal{S}$ and a set $\boldsymbol{\mathcal{S}} = \{\mathcal{S}_1, ..., \mathcal{S}_S\}$ of enrolled users, "identify $\mathcal{S}$ as $\mathcal{S}_i$ if $i = \arg\max_j P(\mathcal{S}_j|\boldsymbol{Y})$". This is a case of speaker identification and the output is a $\mathcal{S}_i$ from $\boldsymbol{\mathcal{S}}$. Another type of decision is "accept $\mathcal{S}$ as $\mathcal{S}_i$ if $P(\mathcal{S}_i|\boldsymbol{Y}) \geq \alpha$", where $\mathcal{S}_i$ is the claimed identity of $\mathcal{S}$. This is a speaker verification decision. Both are covered in this chapter.

2.1 Basic Concepts

Before explain the architecture of both types of recognizers, the elucidation of some basic concepts is necessary.

2.1.1 Utterance

An utterance is a piece of speech produced by a speaker. It may be a word, a statement or any vocal sound. The terms *utterance* and *speech signal* sometimes are used interchangeably, but from herenow speech signal will be defined as an utterance recorded, digitalized and ready to be processed. An example is shown in Fig. 2.1.

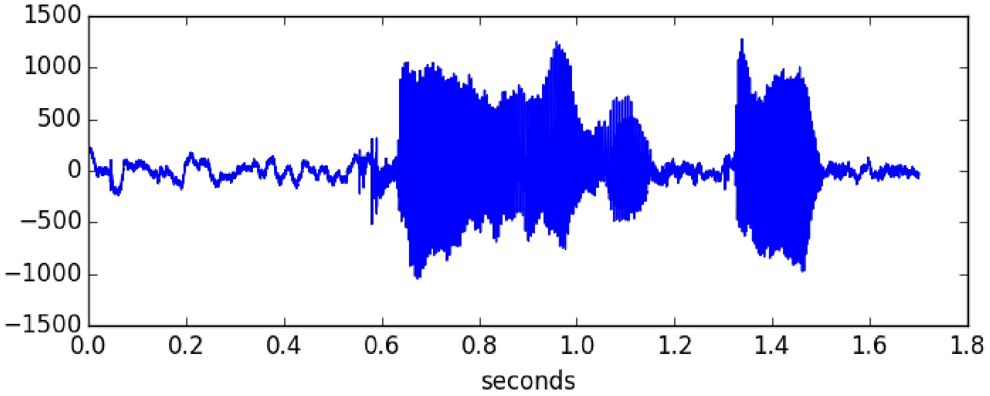

Figure 2.1: Speech signal for utterance "karen livescu", *Woo, Park & Hazen* [10].

2.1.2 Features

The raw speech signal is unfit for usage by an ASR. For a correct processing, the unique features from the speaker's vocal tract are extracted, what reduces the number of variables the system needs to deal with (leading to a simpler implementation) and performs a better evaluation (prevents the curse of dimensionality). Due to the stationary properties of the speech signal when analyzed in a short period of time, it is divided in overlapping frames of small and predefined length, to avoid "loss of significancy" in the features, *Davis & Mermelstein* [11], *Rabiner & Schafer* [12]. This extraction is executed by the MFCC algorithm, explained in details in Chapter 3.

2.2 Speaker Identification

In speaker identification, the objective of the system is to assign an identity from a set of enrolled speakers to the so far unknown speaker, using a speech signal produced by him or her. This system contains a set $\boldsymbol{\mathcal{S}}$ and is fed with features $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_T\}$ extracted from the speech $\boldsymbol{Y}$ produced by the speaker $\mathcal{S}$, the one to be identified.

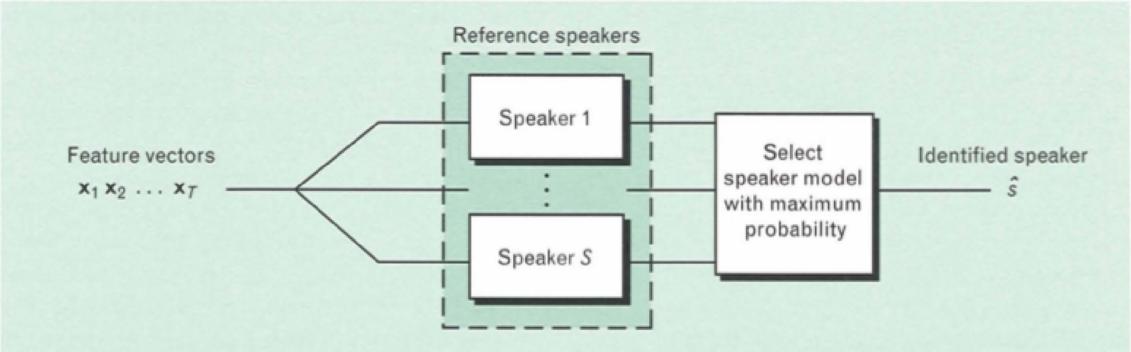

Figure 2.2: Speaker-recognition system for identification, *Reynolds* [5].

The system is the closed set speaker identification described in Section 1.1, and it classifies any speaker, be it an enrolled one or not. Unenrolled speakers are incorrectly identified as the speaker which model returns the highest probability. The equation to determine the identity of $\mathcal{S}$ is

$$\mathcal{S}_i \text{ if } i = \arg\max_j P(\mathcal{S}_j|\boldsymbol{X}), \tag{2.1}$$

and as all speakers have the same prior probability, Eq. 2.1 is reduced to

$$\mathcal{S}_i \text{ if } i = \arg\max_j p(\boldsymbol{X}|\mathcal{S}_j). \tag{2.2}$$

2.2.1 Training

The use of $\mathcal{S}_j$ to represent the speaker is inaccurate, because it maintains the problem in a high level of abstraction. To present a more concrete solution, it is necessary to use a model λ_j for each $\mathcal{S}_j$. Each λ_j is trained independently until a stop condition is fulfilled.

2.2.2 Test

Once all λ_j's are trained, the system if able to be used for classification. Eq. 2.2 is then redefined as

$$\mathcal{S}_i \text{ if } i = \arg\max_j p(\boldsymbol{X}|\lambda_j), \tag{2.3}$$

and the unknown speaker receives the identity of the enrolled speaker for which $\boldsymbol{X}$ maximizes the **likelihood** of λ_j (see Fig. 2.2).

2.3 Speaker Verification

In speaker verification, $\mathcal{S}$ claims to be a particular $\mathcal{S}_i$ from $\boldsymbol{\mathcal{S}}$. The strength of this claim resides on how similar the features $\boldsymbol{X}$ are to the features from $\mathcal{S}_i$ "memorized" by the system. Then a simple equation

$$p(\boldsymbol{Y}|\mathcal{S}_i) \begin{cases} \geq \alpha, & \text{accept } \mathcal{S}, \\ < \alpha, & \text{reject } \mathcal{S}, \end{cases} \tag{2.4}$$

should be enough (again considering all speakers equally probable). However a subset of enrolled speakers may have vocal similarities, leading to a misclassification of one enrolled speaker as another (a false positive). To reduce the error rate, the system must decide not only if a speech signal came from the claimed speaker, but also if it came from a set composed of all other enrolled speakers.

2.3.1 Likelihood Ratio Test

Given the vector of features $\boldsymbol{X}$, and assuming it was produced by only one speaker, the detection task can be restated as a basic test between two hypoteses, *Reynolds* [13]:

H_0: $\boldsymbol{X}$ is from the claimed speaker $\mathcal{S}_i$;

H_1: $\boldsymbol{X}$ is not from the claimed speaker $\mathcal{S}_i$.

The optimum test to decide which hypotesis is valid is the **likelihood ratio test** between both likelihoods $p(\boldsymbol{X}|H_0)$ and $p(\boldsymbol{X}|H_1)$

$$\frac{p(\boldsymbol{X}|H_0)}{p(\boldsymbol{X}|H_1)} \begin{cases} \geq \theta, & \text{accept } H_0, \\ < \theta, & \text{reject } H_0, \end{cases} \tag{2.5}$$

where the decision threshold for accepting or rejecting H_0 is θ. Applying the logarithm, the behavior of the likelihood ratio is maintained and Eq. 2.5 is replaced by the log-likelihood ratio

$$\Lambda(\boldsymbol{X}) = \log p(\boldsymbol{X}|H_0) - \log p(\boldsymbol{X}|H_1). \tag{2.6}$$

2.3.2 Training

Once the features are extracted from the speech signal, they are used to train the models λ_{hyp} and $\lambda_{\overline{hyp}}$ for H_0 and H_1, respectively. A high-level demonstration of the training of λ_{hyp} (mathematical representation of $\mathcal{S}_i$) is shown in Fig. 2.3.

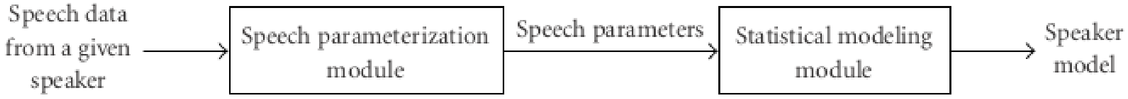
Speech data from a given speaker
Speech parameterization module
Speech parameters
Statistical modeling module
Speaker model

Figure 2.3: The statistical model of $\mathcal{S}$ is created from the speech signal $\boldsymbol{Y}$, *Bimbot et. al.* [1].

Due to λ_{hyp} be a model of $\mathcal{S}_i$, the features used for training (i.e., estimate $p(\boldsymbol{X}|\lambda_{hyp})$) are extracted from speech signals produced by $\mathcal{S}_i$. The model $\lambda_{\overline{hyp}}$, however, is not well-defined. It should be composed of the features extracted from speech signals from all other speakers except $\mathcal{S}_i$, but creating a single $\lambda_{\overline{hyp}}$ for each speaker is complicated and with no expressive gain. Instead, what is normally done is use all speakers to generate a background model λ_{bkg}, *Reynolds* [14], in which the weight of each $\mathcal{S}_i$ is minimized.

2.3.3 Test

As seen in Eq. 2.5, the decision process is based on a function *Score*. Replacing each H_j for its corresponding model, the likelihood of a λ_j given $\boldsymbol{X}$ can be written as

$$p(\boldsymbol{X}|\lambda_j) = \prod_{t=1}^{T} p(\boldsymbol{x}_t|\lambda_j). \tag{2.7}$$

Using the logarithm function, Eq. 2.7 becomes

$$\log p(\boldsymbol{X}|\lambda_j) = \frac{1}{T}\sum_{t=1}^{T} \log p(\boldsymbol{x}_t|\lambda_j), \tag{2.8}$$

where the term $\frac{1}{T}$ is used to normalize the log-likelihood to the duration of the speech signal. That said, the likelihood ratio given by Eq. 2.6 becomes

$$\Lambda(\boldsymbol{X}) = \log p(\boldsymbol{X}|\lambda_{hyp}) - \log p(\boldsymbol{X}|\lambda_{bkg}), \tag{2.9}$$

and the speaker is accepted if $\Lambda(\boldsymbol{X}) \geq \log\theta$, for an arbitrary value of θ.

3. Feature Extraction

As an acoustic wave propagated through space over time, the speech signal is not appropriate to be evaluated by an ASR. In order to deliver decent outcomes, a good parametric representation must be provided. This task is performed by the feature extraction process, which transforms a speech signal into a sequence of characterized measurements, i.e. features. The selected representation compress the speech data by eliminating information not pertinent to the phonetic analysis, and enhancing those aspects of the signal that contribute significantly to the detection of phonetic differences, *Davis & Mermelstein* [11]. According to *Wolf* [15] the ideal features should:

- occur naturally and frequently in normal speech;
- be easily measurable;
- vary highly among speakers and be very consistent for each speaker;
- not change over time nor be affected by the speaker's health;
- be robust to reasonable background noise and to transmission characteristics;
- be difficult to be artificially produced;
- not be easily modifiable by the speaker.

Features may be categorized based on vocal tract or behavioral aspects, divided in (1) short-time spectral, (2) spectro-temporal, (3) prosodic and (4) high level, *Pinheiro* [16]. Short-time spectral features are usually calculated using millisecond length windows and describe the voice spectral envelope, composed of supralaryngeal properties of the vocal tract, e.g. timbre. Spectro-temporal and prosodic occur over time, e.g. rhythm and intonation, and high level features occur during the conversation, e.g. accents.

The parametric representations evaluated in *Davis & Mermelstein* [11] may be divided into those based on the Fourier spectrum, Mel-Frequency Cepstrum Coefficients (MFCC) and Linear Frequency Cepstrum Coefficients (LFCC), and those based on the Linear Prediction Spectrum, Linear Prediction Coefficients (LPC), Reflection Coefficients (RC) and Linear Prediction Cepstrum Coefficients (LPCC). The better evaluated representation was the MFCC, with minimum and maximum accuracy of 90.2% and 99.4%, respectively, leading to its choice as the parametric representation in this work.

3.1 Mel-Frequency Cepstral Coefficient

MFCC is a highly used parametric representation in the area of voice processing, due to its similarity with the mode the human ear operates. Despite the fact the ear is divided in three sections, i.e. outer, middle and inner ears, only the last is mimicked. The mechanical pressure waves produced by the triad hammer-anvil-stirrup are received by the cochlea

(Fig. 3.1), a spiral-shaped cavity with a set of inner hair cells attached to a membrane (the basilar membrane) and filled with a liquid. This structure converts motion to neural activity through a non-uniform spectral analysis, *Rabiner & Schafer* [12] and passes it to the pattern recognizer in the brain.

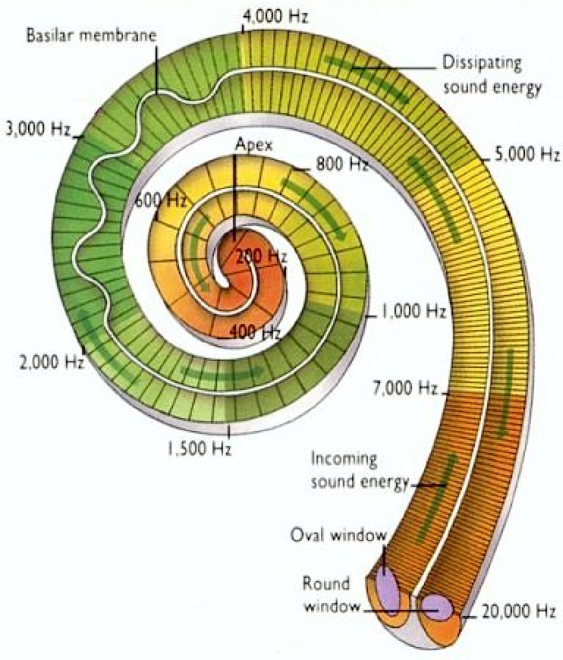
4,000 Hz
Basilar membrane
Dissipating sound energy
3,000 Hz
Apex
800 Hz
5,000 Hz
600 Hz
200 Hz
1,000 Hz
400 Hz
2,000 Hz
7,000 Hz
1,500 Hz
Incoming sound energy
Oval window
Round window
20,000 Hz

Figure 3.1: Cochlea divided by frequency regions, *ScienceBlogs* [17].

A key factor in the perception of speech and other sounds is **loudness**, a quality related to the physical property of sound pressure level. Loudness is quantified by relating the actual sound pressure level of a pure tone (in dB relative to a standard reference level) to the perceived loudness of the same tone (in a unit called phons) over the range of human hearing (20 Hz–20 kHz), *Rabiner & Schafer* [12]. As shown in Fig. 3.2, a 100 Hz tone at 60 dB is equal in loudness to a 1000 Hz tone at 50 dB, both having the **loudness level** of 50 phons (by convention).

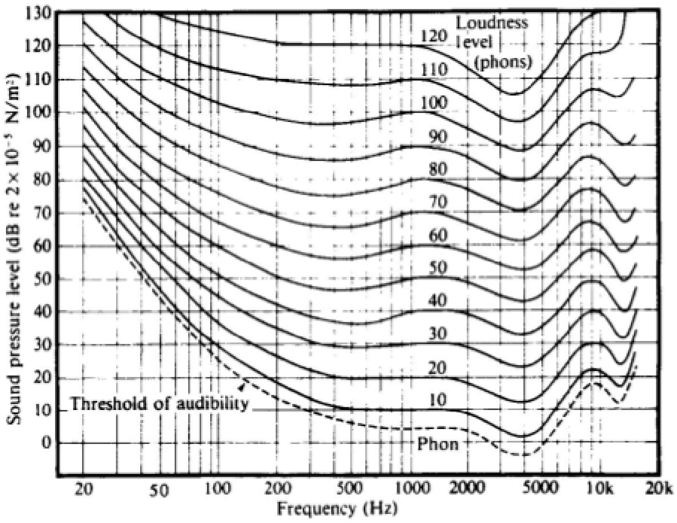

Figure 3.2: Loudness level for human hearing, *Fletcher & Munson* [18].

3.1.1 The Mel Scale

The mel scale is the result of an experiment conducted by *Stevens, Volkmann and Newman* [19] intended to measure the perception of a pitch and construct a scale based on it. Each observer was asked to listen to two tones, one in the fixed frequencies 125, 200, 300, 400, 700, 1000, 2000, 5000, 8000 and 12000 Hz, and the other free to have its frequency varied by the observer for each fixed frequency of the first tone. An interval of 2 seconds separated both tones. The observers were instructed to say in which frequency the second tone was "half the loudness" of the first. A geometric mean was taken from the observers' answers and a measure of 1000 mels was assigned to the frequency of 1000 Hz, 500 mels to the frequency sounding half as high (as determined by Fig. 1 in *Stevens et. al.* [19]) and so on.

Decades after the scale be defined, *O'Shaughnessy* [20] presented an equation to convert frequencies in Hertz to frequencies in mels.

$$f_{mel} = 2595 log_{10}(1 + \frac{f}{700}) \tag{3.1}$$

Being logarithmic, the growth of a mel-frequency curve is slow when Eq. 3.1 is applied to a linear growth of the frequency in Hertz. Sometimes the mel conversion is used only for frequencies higher than 1000 Hz, while in lower, f_{mel} and f_{Hz} share the same value. In this work all conversions will use Eq. 3.1, as shown by Fig. 3.3.

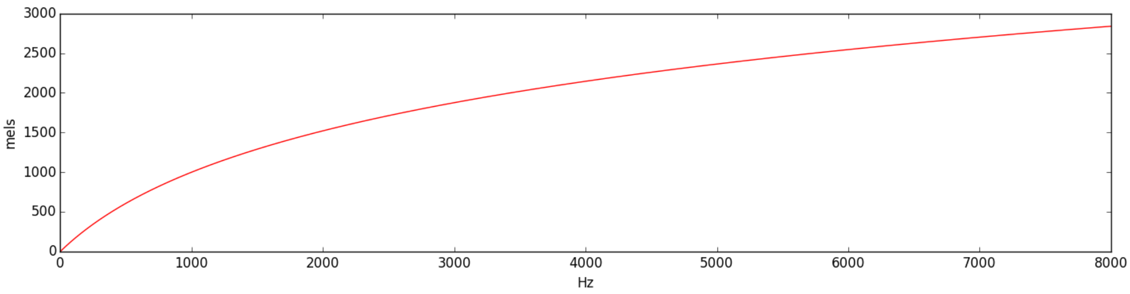

Figure 3.3: The logarithm curve of the mel scale.

3.1.2 Extraction Process

In an ASR the feature extraction module receives a raw speech signal and returns a vector of cepstral features in mel scale (see Fig. 3.4). The number of features in each frame is defined at the moment of extraction (e.g., 6, 13 or 19), but the user has the option to append time variations of the MFCCs (i.e., delta coefficients) in order to improve the representation.

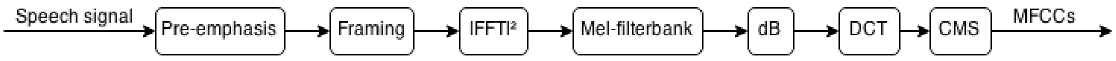

Figure 3.4: Modular representation of the MFCC extraction.

As the human voice is concentrated in the lower frequencies (see Fig. 3.5), the higher ones are enhanced to improve the classification. A first order FIR filter is used

$$s_{emph}[n] = s[n] - a \cdot s[n-1], \tag{3.2}$$

with values of a usually in the interval $[0.95, 0.98]$, *Bimbot et. al.* [1]. This stage is not necessary to extract the the MFCCs.

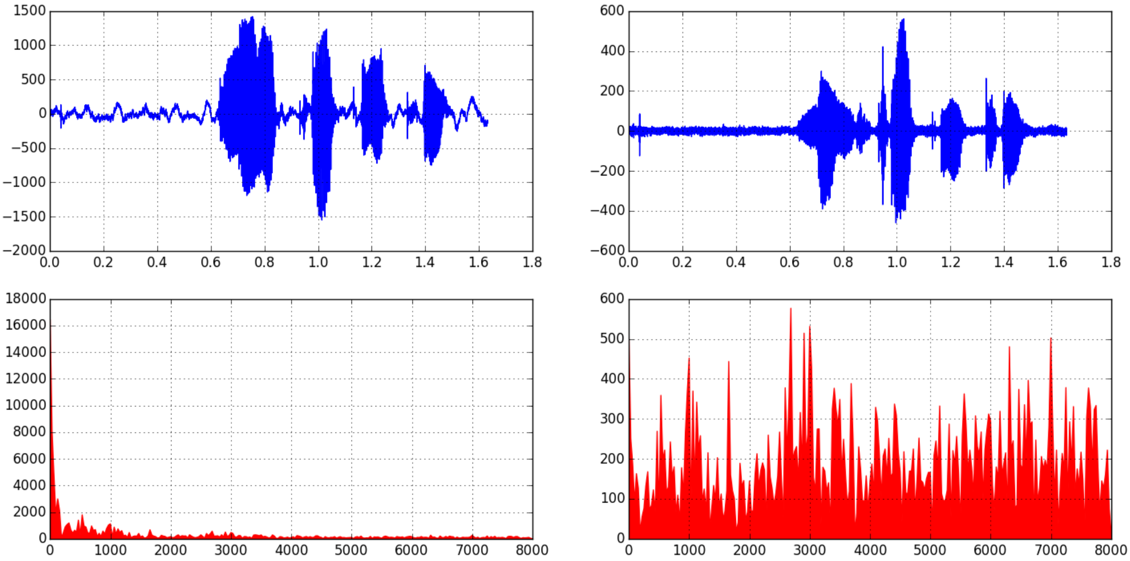

Figure 3.5: Raw (left) and pre-emphasized (right) speech signal, with respective spectral magnitudes (bottom).

The first mandatory stage of the feature process is the division of the input signal in overlapping frames, by the application of a sliding window (commonly Hamming, to taper the signal on the ends and reduce the side effects, *Bimbot et. al.* [1]). The window has a width in the order of milliseconds (to perform a short-time analysis) and a shift that must be shorter than the width, or the frames will not be overlapped. Usual values for the window's width are between 20 and 40, and for the shift, 10.

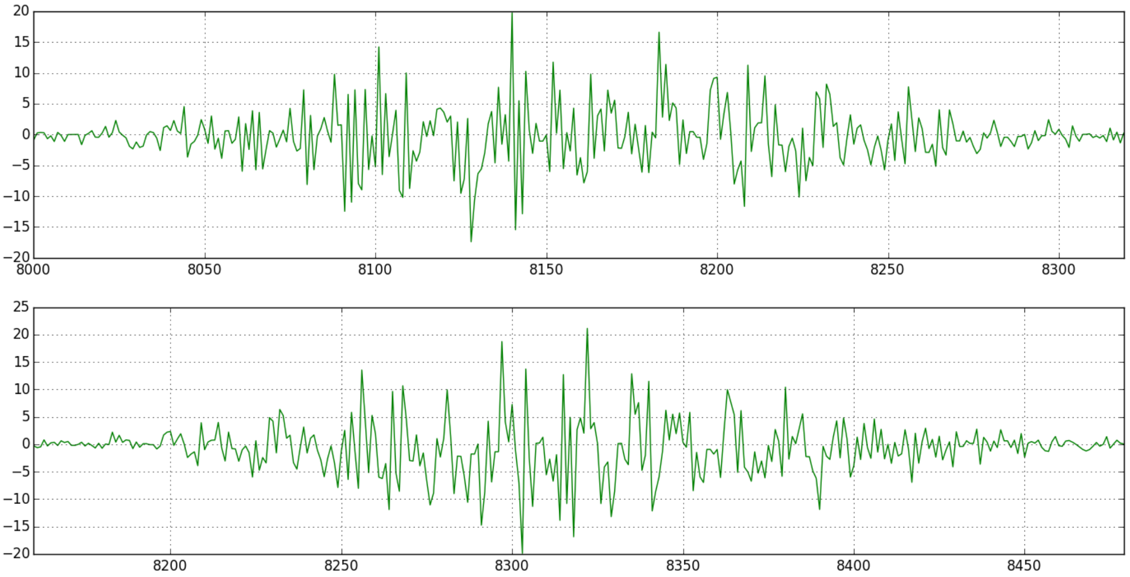

Figure 3.6: 51^{o} and 52^{o} frames. Notice the initial and final samples of each figure.

For each frame the Fast Fourier Transform (FFT) is calculated, with number of points greater than the width of the window (usually 512). Finally, the modulus of the FFT is

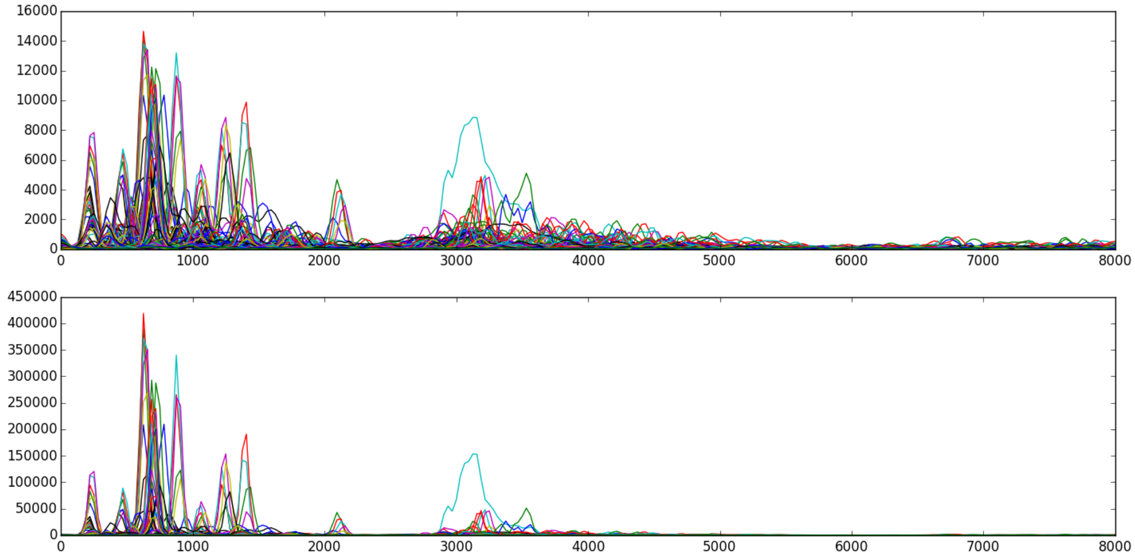

Figure 3.7: $|FFT|$ (top) and $|FFT|^2$ (bottom)

calculated and the power spectrum is obtained. Due to its symmetry, only half of the points are kept.

To get the envelope (and to reduce the size of spectral coefficients), the spectrum is multiplied by a filterbank in the mel scale. As seen in Fig. 3.8, the width of the filters enlarge when the frequency increases (these frequencies bands have the same width in mels). This is an approximation of the filtering process executed by the cochlea (see Fig. 3.1), and is done this way due to the higher accuracy of human hearing in lower frequencies than in higher ones. The result of the filtering is the energy in each sample of the frame (see Fig. 3.9 (left)).

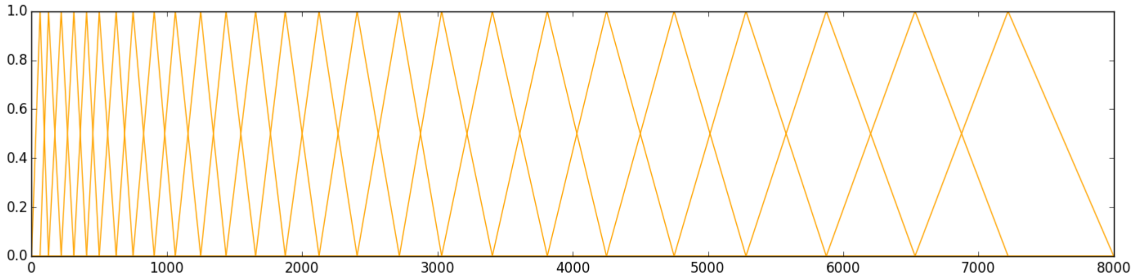

Figure 3.8: Filter bank with 22 filters.

The spectral coefficients are then converted to dB by the application of the function $10\log(\cdot)$ to each sample of each frame, reducing the differences between energy values.

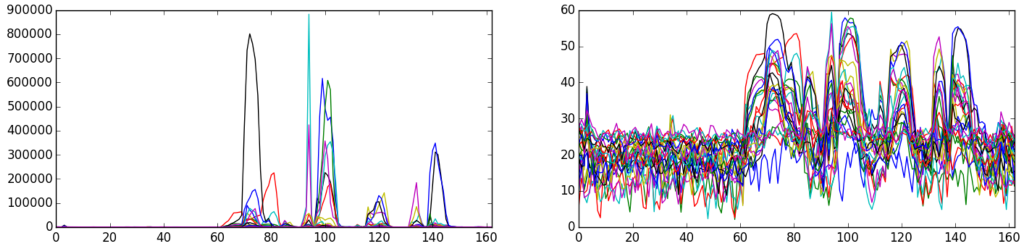

Figure 3.9: Spectral coefficients after the filterbank (left) and after the log conversion (right).

Until now the features are in the mel scale, but are not yet "cepstral". The last necessary stage is to apply a Discrete Cosine Transform (DCT) to the spectral coefficients in order to yield the cepstral coefficients:

$$c_n = \sum_{k=1}^{K} S_k \cdot \cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{K}\right], \qquad n = 1, 2, ..., L, \tag{3.3}$$

where K is the number of spectral coefficients, S_k is a spectral coefficient, and L is the number of cepstral coefficients to calculate ($L \leq K$). The application of a lifter (a cepstral filter) is usual after the computation of the DCT, to smooth the coefficients. After this stage, the MFCCs are extracted.

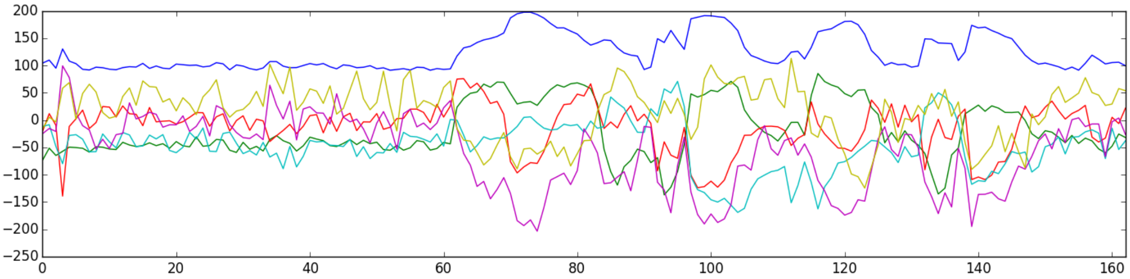

Figure 3.10: 6 MFCCs for each frame.

In Fig. 3.10, the blue line represents the first feature, and as it is obvious, its values over time are much higher than the values of the others. To correct this discrepancy, the feature is changed by the summed energy of each frame, bringing it closer to the others (see Fig. 3.11).

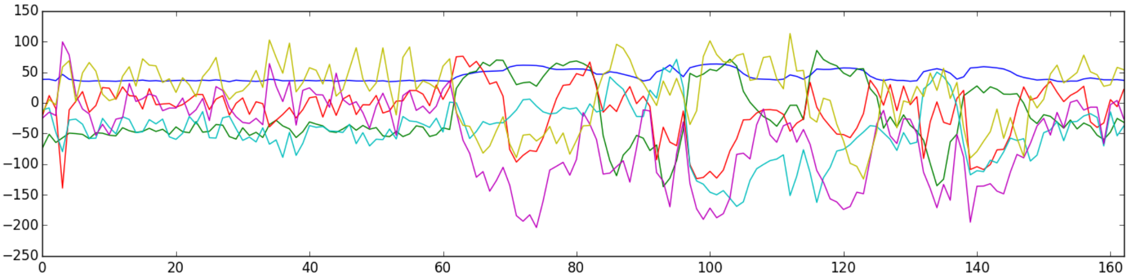

Figure 3.11: First feature changed by the summed energy of each frame.

Even an utterance recorded in a quiet environment still suffers with the side effects of any noise captured during the recording, what may degrade the performance. For speeches recorded in regular places (e.g., a living room or a park), the environment robustness is a need. Cepstral Means Subtraction,

$$c_n = c_n - \frac{1}{T}\sum_{t=1}^{T} c_{i,t}, \tag{3.4}$$

tries to eliminate the disturbing channel effect before the ASR be trained, delivering a cleaner signal to the models, *Westphal* [21].

In order to improve the speech parameters, the differences in time of each coefficient may be added as new features. In a vector with 6 features per frame, the velocity and acceleration of each coefficient may be added, providing 12 more features to the parametrization, all of them related to the ones previously extracted. These new features are the **deltas** of the MFCCs, given by

$$d_t = \frac{\sum_{n=1}^{N} n(c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2}. \tag{3.5}$$

where N (usually 2) determines how far from the frame t the calculation is taken. Fig. 3.13 shows the MFCCs from Fig. 3.12 improved by the addition of deltas of first and second orders.

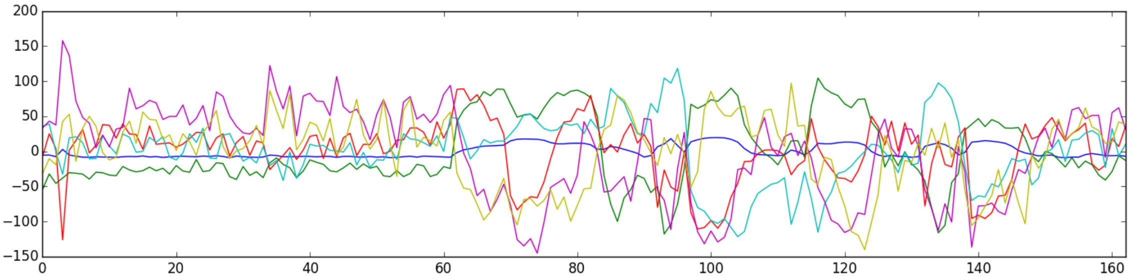

Figure 3.12: CMS applied to the MFCCs from Fig. 3.11.

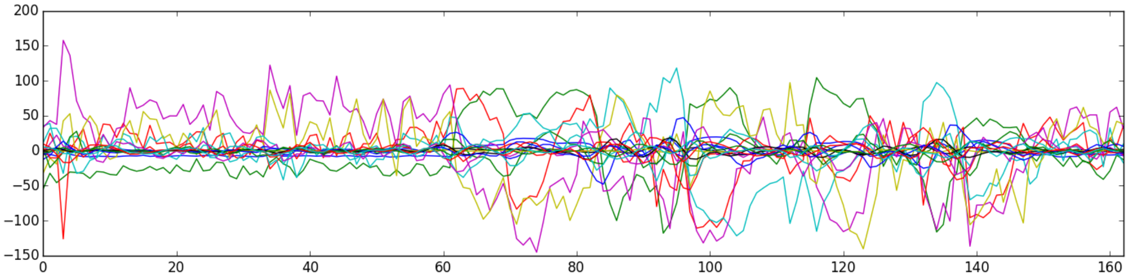

Figure 3.13: MFCCs from Fig. 3.12 with deltas or order 1 and 2 added.

Eq. 3.5 may be used to calculate deltas of any order, just as the acceleration (second order) are derived from the velocity. However, as seen in Fig. 3.13, each order of delta delivers lower coefficients, providing a marginal gain.

4. Gaussian Mixture Models

In Chapter 2 was discussed the use of a model λ_j for each enrolled speaker to be identified, and models λ_{hyp} and λ_{bkg} for a claimed speaker and for the background composed of all enrolled speakers, respectively, to perform a verification process. As the features from the speech signal (discussed in Chapter 3) have unknown values until the moment of extraction, it is reasonable to model the ASR to accept random values.

For all sorts of probability distributions, the Gaussian (or normal) is the one that best describes the behavior of a random variable of unknown distribution, due to the central limit theorem. Its equation for a D-dimensional space is

$$p(\boldsymbol{x}) = p(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}, \tag{4.1}$$

where $\boldsymbol{x}$ is a D-dimensional input vector, $\boldsymbol{\mu}$ is the D-dimensional vector of means and $\boldsymbol{\Sigma}$ is the $D \times D$ matrix of covariances. The vector $(\boldsymbol{x} - \boldsymbol{\mu})'$ is the transposed of the colum-matrix $(\boldsymbol{x} - \boldsymbol{\mu})$.

For the speaker recognition, a weighted sum of $p_i(\boldsymbol{x})$ is used to model the system, trying to estimate the composition that best represents the training data. This weighted sum is named Gaussian Mixture Model (GMM), *Reynolds* [22], and is given by

$$p(\boldsymbol{x}|\lambda) = \sum_{i=1}^{M} w_i p(\boldsymbol{x}), \tag{4.2}$$

where M is the number of distributions used, $\sum_{i=1}^{M} w_i = 1$, and $\lambda = \{w_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}$, for $i = 1, ..., M$. Applying Eq. 4.1 to Eq. 4.2, the likelihood for the GMM is

$$p(\boldsymbol{x}|\lambda) = \sum_{i=1}^{M} \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}_i|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)'\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)}. \tag{4.3}$$

The idea behind use a GMM as a model for a speaker $\mathcal{S}$ is to achieve a λ that maximizes the likelihood when applied to features $\boldsymbol{X}$ extracted from a speech signal produced by $\mathcal{S}$. This value is found by a Maximum Likelihood Estimation (MLE) algorithm. For a sequence of T training vectors $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_T\}$, the GMM likelihood can be written as

$$p(\boldsymbol{x}|\lambda) = \prod_{t=1}^{T} P(\boldsymbol{x}_t|\lambda). \tag{4.4}$$

Unfortunately, this expression is a nonlinear function of the parameters λ and direct maximization is not possible, *Reynolds* [23], leading to estimate $p(\boldsymbol{x}|\lambda)$ iteratively using the Expectation-Maximization (EM) algorithm.

4.1 Expectation-Maximization

The idea of the EM algorithm is to estimate a new model $\lambda^{(j+1)}$ from a previous model $\lambda^{(j)}$, such that $p(\boldsymbol{x}|\lambda^{(j+1)}) \geq p(\boldsymbol{x}|\lambda^{(j)})$, approximating the GMM to the training data at each iteration, until some convergence threshold is reached. The algorithm is composed of 2 steps, an expectation of the *a posteriori* probabilities for each distribution i, and a maximization step, when the parameters w_i, $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ are updated. The description ahead for the steps is for a λ with all $\boldsymbol{\Sigma}_i$ diagonal, i.e., change the $D \times D$ matrix $\boldsymbol{\Sigma}_i$ for a D-dimensional vector $\boldsymbol{\sigma}_i$ of variances.

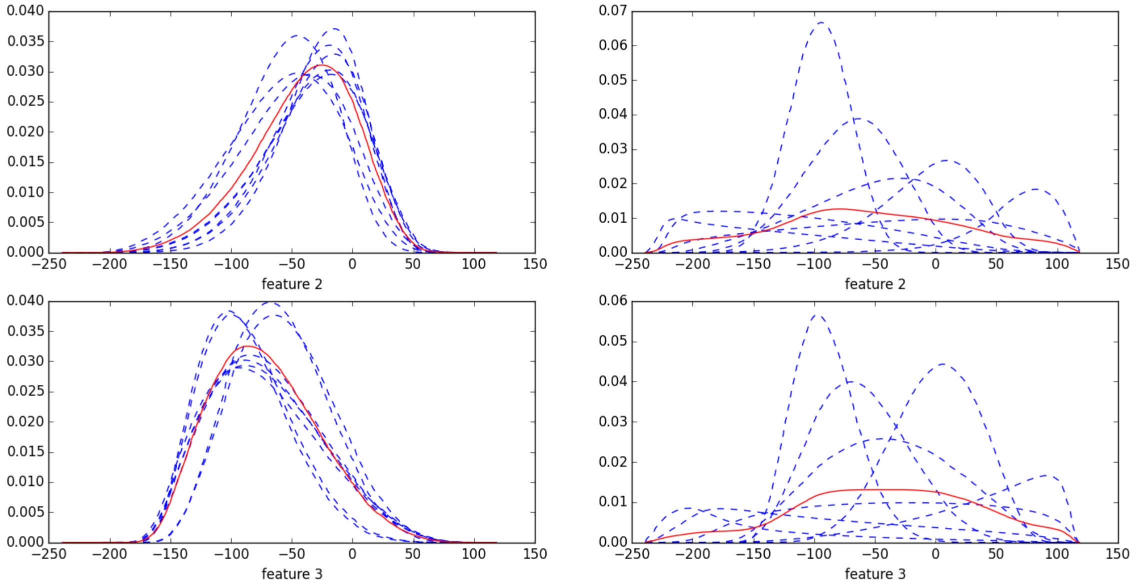

Figure 4.1: Features 2 and 3 before (left) and after (right) the EM algorithm. The dashed blue lines are the 8 gaussians and the solid red lines the mixture.

E-Step

The expectation step consists of estimate the *a posteriori* probabilities for each distribution i and each feature vector $\boldsymbol{x}_t$,

$$P(i|\boldsymbol{x}_t, \lambda) = \frac{w_i p_i(\boldsymbol{x}_t)}{\sum_{k=1}^{M} w_k p_k(\boldsymbol{x}_t)}. \tag{4.5}$$

M-Step

In the maximization step, the parameters are updated, and the algorithm guarantees that each new $\lambda^{(}j+1)$ represents the training data better than the previous ones. From *Reynolds* [23], the updates of w_i, $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ are given by the equations below.

Weights:

$$\overline{w}_i = \frac{1}{T}\sum_{t=1}^{T} P(i|\boldsymbol{x}_t, \lambda), \tag{4.6}$$

Means:

$$\overline{\boldsymbol{\mu}}_i = \frac{1}{T}\frac{\sum_{t=1}^{T} P(i|\boldsymbol{x}_t, \lambda)\boldsymbol{x}_t}{\sum_{t=1}^{T} P(i|\boldsymbol{x}_t, \lambda)}, \tag{4.7}$$

Variances:

$$\overline{\boldsymbol{\sigma}}_i = \frac{1}{T}\frac{\sum_{t=1}^{T} P(i|\boldsymbol{x}_t, \lambda)\boldsymbol{x}_t^2}{\sum_{t=1}^{T} P(i|\boldsymbol{x}_t, \lambda)} - \overline{\boldsymbol{\mu}}_i^2. \tag{4.8}$$

This algorithm is used to train the GMMs described in sections Section 2.2 and Section 2.3 of Chapter 2. Fig. 4.1 shows the mixture before and after the training.

4.2 Universal Background Model

An Universal Background Model-Gaussian Mixture Model (UBM-GMM), shortened to UBM, is a GMM composed of features from all enrolled speakers. Its idea is to model a system as if speech signals were recorded by the same microphone with all speakers talking at once, i.e., a background of voices, a situation where it is impossible to understand the content of what each speaker is saying.

The training and evaluation of an UBM are the same for single speaker GMMs, with difference in the time taken (due to the number of speakers composing it). In this paper, there is two types of UBMs. In the first, the subpopulations of male and female speeches are combined and used to train a single, unisex UBM. In the second, each subpopulation is used to train its own model, and then both models are combined in one with twice the number of distributions.

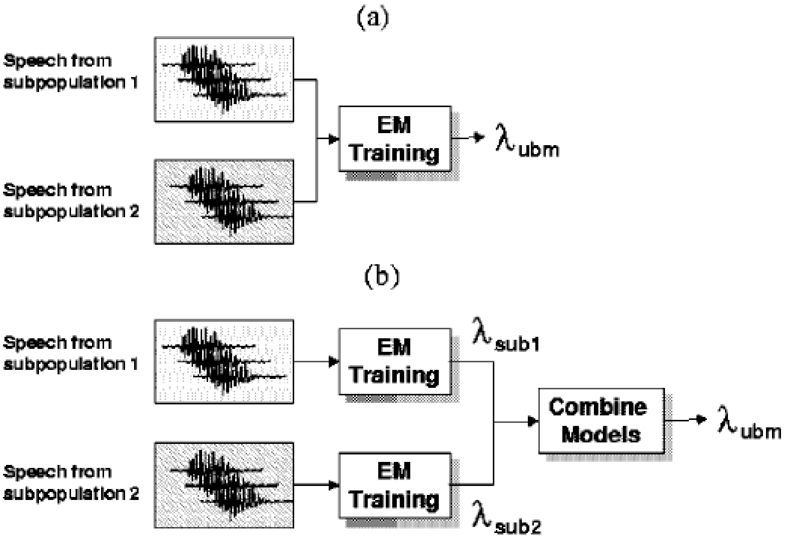
(a)
Speech from subpopulation 1
Speech from subpopulation 2
EM Training
λ ubm
(b)
Speech from subpopulation 1
Speech from subpopulation 2
EM Training
EM Training
λ sub1
λ sub2
Combine Models
λ ubm

Figure 4.2: Gender (a) independent and (b) dependent UBMs. *Reynolds, Quatieri & Dunn* [24]

5. Experiments

The ASR was implemented in its entirety using the programming language Python (version 3.4) and the frameworks NumPy (version 1.8.1) and SciPy (version 0.14.0), meaning the feature extraction, the GMM and UBM models, and the identification and verification classifiers. All codes and data are stored in the public repository `https://github.com/embatbr/tg`, as well as this document and secondary resources.

The database used for this work is the **MIT Mobile Device Speaker Verification Corpus (MIT-MDSCV)**, *Woo et. al.* [10]. Two different sets of models were trained. The first was a single speaker GMM for each enrolled speaker, using the utterances relative to the speaker presented in the dataset *enroll_1*. The trainings took 2-12 seconds each, by the EM algorithm. The second set was of UBMs, also using the utterances from the dataset *enroll_1*. Two types of UBMs were created: an unisex and a divided by gender, both described in Section 4.2 from chapter Chapter 4. The unisex UBM took 3-6 minutes of training, while the divided by gender took 7-10 minutes. Both sets of models were created and trained for 32, 64 and 128 distributions, 6, 13 and 19 cepstral coefficients and 0, 1 and 2 orders of delta.

Three experiments were conducted: identification, verification using unisex trained UBMs and verification using gender trained UBMs.

5.1 Identification

The identification is performed by the application of Eq. 2.3 to vectors of features $\boldsymbol{X}$ extracted from speech signals in *enroll_2* and models λ_j stored in the directory *bases/gmms*. The set of enrolled speakers is composed of all 26 males and 22 females. Each speaker in *enroll_2* has 54 recorded utterances, with all tested and the percentage of correct identification calculated. TABELA ERRADA!

Model Order	**#Coeffs.**	**#Deltas**		
		0	1	2
$M = 32$	6	29.55	40.28	51.74
	13	57.68	69.60	74.77
	19	71.76	81.52	84.53
$M = 64$	6	29.55	40.28	51.74
	13	57.68	69.60	74.77
	19	71.76	81.52	84.53
$M = 128$	6	29.55	40.28	51.74
	13	57.68	69.60	74.77
	19	71.76	81.52	84.53

Table 5.1: Average correct identification of speakers.

The values shown in Table 5.1 are the mean of the success rates for every speaker in the given configuration.

5.2 Verification

5.2.1 Unisex

5.2.2 Gender

6. Conclusion

TODO escrever a conclusão após terminar tudo (antes do abstract)

A. Results for Adapt-GMM-UBM

Appendix with results for the technique Adapt-GMM-UBM

B. Results for C-Adapt-GMM-UBM

Appendix with results for the technique C-Adapt-GMM-UBM

C. Codes

Codes here

References

[1] Frédéric Bimbot et al. “A Tutorial on text-independent speaker verification”. In: *EURASIP Journal on Applied Signal Processing* 4 (Apr. 2004), pp. 430–451.

[2] P.T. Wang and S.M. Wu. “Personal fingerprint authentication method of bank card and credit card”. Pat. US Patent App. 09/849,279. Nov. 2002. URL: `https://www.google.com/patents/US20020163421`.

[3] M. Angela Sasse. “Red-Eye Blink, Bendy Shuffle, and the Yuck Factor: A User Experience of Biometric Airport Systems”. In: *Security & Privacy, IEEE* 5.3 (June 2007), pp. 78–81.

[4] Ahmad N. Al-Raisi and Ali M. Al-Khouri. “Iris recognition and the challenge of homeland and border control security in UAE”. In: *Telematics and Informatics* 25.2 (2008), pp. 117–132.

[5] Douglas A. Reynolds. “Automatic Speaker Recognition Using Gaussian Mixture Speaker Models”. In: *The Lincoln Laboratory Journal* 8.2 (1995), pp. 173–192.

[6] Douglas A. Reynolds and William M. Campbell. “Springer Handbook of Speech Processing”. In: ed. by Jacob Benesty, M. Mohan Sondhi, and Yiteng Huang. Berlin: Springer, 2008. Chap. Text-Independent Speaker Recognition, pp. 763–780.

[7] Martial Hébert. “Springer Handbook of Speech Processing”. In: ed. by Jacob Benesty, M. Mohan Sondhi, and Yiteng Huang. Berlin: Springer, 2008. Chap. Text-Dependent Speaker Recognition, pp. 743–762.

[8] Peter F. Brown, Chin-Hui Lee, and James C. Spohrer. “Bayesian Adaptation in Speech Recognition”. In: *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP ’83*. Vol. 8. IEEE, Apr. 1983, pp. 761–764.

[9] A. Martin et al. “The DET curve in assessment of detection task performance”. In: *Proceedings of the European Conference on Speech Communication and Technology*. 1997, pp. 1895–1898.

[10] Ram H. Woo, Alex Park, and Timothy J. Hazen. “The MIT Mobile Device Speaker Verification Corpus: Data Collection and Preliminary Experiments”. In: *Odyssey 2006: The Speaker and Language Recognition Workshop, San Juan, Puerto Rico, June 28-30, 2006*. IEEE, 2006, pp. 1–6.

[11] Steven B. Davis and Paul Mermelstein. “Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* ASSP-28.4 (Aug. 1980), pp. 357–366.

[12] Lawrence R. Rabiner and Ronald W. Schafer. “Introduction to Digital Speech Processing”. In: *Foundations and Trends in Signal Processing* 1.1-2 (Dec. 2007), pp. 1–194.

[13] Douglas A. Reynolds. “Speaker identification and verification using Gaussian mixture speaker models”. In: *Speech Communication* 17.1 (1995), pp. 91–108.

[14] D. A. Reynolds. “Comparison of background normalization methods for text-independent speaker verification”. In: *Proceedings of the European Conference on Speech Communication and Technology*. Sept. 1997, 963–966.

[15] Jared J. Wolf. “Efficient acoustic parameters for speaker recognition”. In: *Journal of the Acoustical Society of America* 51 (1972), pp. 2044–2056.

[16] Hector N. B. Pinheiro. *Sistemas de Reconhecimento de Locutor Independente de Texto*. Trabalho de Graduação. Universidade Federal de Pernambuco, Jan. 2013.

[17] Ethan. *Don’t you hear that?* May 10, 2010. URL: `http://scienceblogs.com/startswithabang/2010/05/10/dont-you-hear-that/`.

[18] Harvey Fletcher and Wilden A. Munson. “Loudness, Its Definition, Measurement and Calculation”. In: *Bell Telephone Laboratories* 12.4 (Oct. 1933), pp. 82–108.

[19] Stanley S. Stevens, John Volkmann, and Edwin B. Newman. “A Scale for the Measurement of the Psychological Magnitude Pitch”. In: *The Journal of Acoustical Society of America* 8.3 (Jan. 1937), pp. 185–190.

[20] Douglas O’Shaughnessy. *Speech Communications: Human and Machine*. Addison-Wesley, 1987.

[21] Martin Westphal. “The Use Of Cepstral Means In Conversational Speech Recognition”. In: *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech)*. 1997, pp. 1143–1146.

[22] Douglas A. Reynolds. “A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification”. PhD thesis. Georgia Institute of Technology, Aug. 1992.

[23] Douglas A. Reynolds. “Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models”. In: *IEEE Transactions on Speech and Audio Processing* 3.1 (Jan. 1995), pp. 72–83.

[24] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn. “Speaker Verification Using Adapted Gaussian Mixture Models”. In: *Digital Signal Processing* 10.1 (Jan. 2000), pp. 19–41.