Universidade Federal de Pernambuco

Centro de Informática

Text-Independent Speaker Recognition Using 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

The increasing popularity and the intensive use 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 Recognition (ASR) system 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 use 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. Nowadays, fingerprint biometrics is present 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's 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 any other technique.

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 these goals, numerous voice processing techniques have become known in academy and industry, such as Natural Language Processing (NLP), Hidden Markov Models (HMMs) and Gaussian Mixture Models (GMMs). Although all of these are interesting state-of-the-art techniques, this paper covers a subarea of speaker recognition and only a small subset 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.

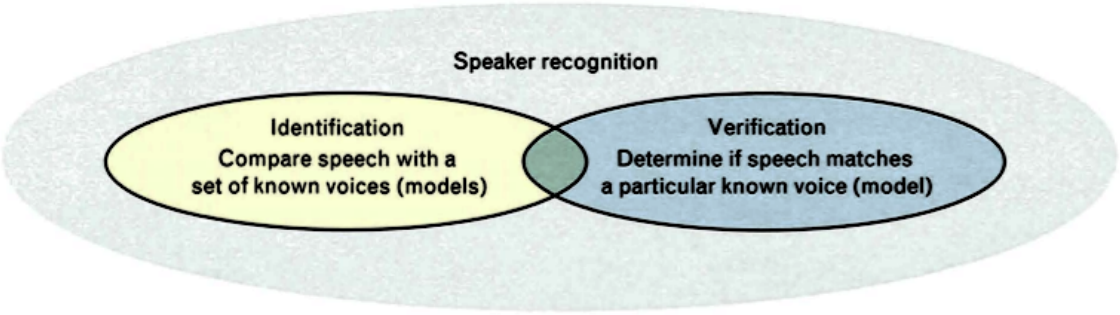

Figure 1.1: Relation between identification and verification of speakers, *Reynolds* [5].

The text inputted 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, *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 present in different sentences, use of foreign languages and even gibberish. Between the extremes in constraints falls the **vocabulary-dependent system**, which restricts the speech to come from a limited vocabulary (e.g., digits such as "two" or "one-two-three"), *Reynolds* [5].

1.2 Gaussian Mixture Models

The focus of this paper is on **text-independent speaker recognition** (both identification and 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 an ASR system.

Recognition systems are constructed using several techniques based on GMM. For the identification process a GMM is trained for each enrolled speaker, referred to as Gaussian Mixture Speaker Model (GMSM), with the identity given by the model with higher likelihood. Verification systems are designed using an Universal Background Model-Gaussian Mixture Model (UBM-GMM) trained to represent all speakers as a single background and a GMSM or a bayesian adaptation of the UBM-GMM, *Brown, Lee and Spohrer* [8], named Adapted Gaussian Mixture Speaker Model (AGMSM). A likelihood ratio test is used to evaluate a speech signal and to decide if it belongs or not to the claimed speaker. All techniques are detailed in Chapter 4.

1.3 Objectives

This study is aimed to implement ASR systems (for both identification and verification processes) and analyze the following:

- Correct identification rates for different sizes of mixture and features.
- False detection and false rejection rates for speaker verification using a DET curve, *Martin et al.* [9].
- Accuracy of a speaker verification system for different sizes of mixture and features.
- Comparison between GMSM and AGMSM verifications.

1.4 Document Structure

Chapter 2 contains basic information about ASR systems, as well as its basic architectures. 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 theories of GMM, UBM-GMM and AGMSM are 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 GMSM verification and Appendix B for AGMSM verification.

2. Speaker Recognition Systems

The speaker recognition process lies on the field of pattern classification, with the speaker's speech signal $\boldsymbol{Y}$ as input for a classifier. For an identification system, the classification is 1 to N (one speaker signal to be identified as belonging to one of the N enrolled speakers), while for a verification system the classification is 1 to 1 (a claimed speaker identity is *enrolled* or *imposter*).

ASR systems are bayesian classifiers, using the following equation to calculate the probabilities of recognition:

$$P(\mathcal{S}|\boldsymbol{Y}) = \frac{p(\boldsymbol{Y}|\mathcal{S})P(\mathcal{S})}{p(\boldsymbol{Y})}, \tag{2.1}$$

where $\mathcal{S}$ is the speaker who produced $\boldsymbol{Y}$. As all speakers are considered equally probable, the *a priori* probability $P(\mathcal{S})$ and the *evidence* $p(\boldsymbol{Y})$ (just used for normalization) can be removed with no loss to the analysis. Eq. 2.1 is then replaced by $p(\boldsymbol{Y}|\mathcal{S})$.

2.1 Basic Concepts

Before start the discussion about the types of ASR systems, some basic concepts must be elucidated. Those are **utterance** and **features**.

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.

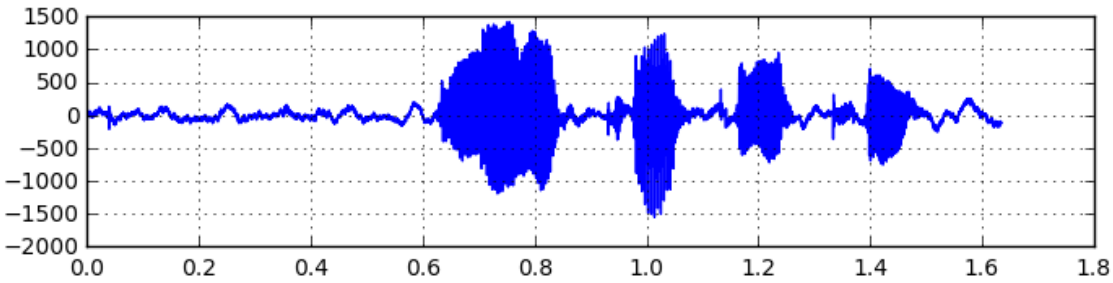

Figure 2.1: Speech signal from utterance "karen livescu", *Woo, Park & Hazen* [10].

2.1.2 Features

The raw speech signal is unfit for use by an ASR system. For a correct processing, the representative 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 significance", *Davis & Mermelstein* [11], *Rabiner & Schafer* [12]. This extraction is executed by the MFCC algorithm, explained in details in Chapter 3.

2.2 Speaker Identification

Given the features $\boldsymbol{X}$ extracted from a speech signal $\boldsymbol{Y}$ spoken by an arbitrary speaker $\mathcal{S}$, the task of identify $\mathcal{S}$ as a particular $\mathcal{S}_i$ from $\boldsymbol{\mathcal{S}}$ (set of enrolled users) is given by the following equation:

$$\mathcal{S} \text{ is } \mathcal{S}_i \text{ if } i = \arg_j \max p(\boldsymbol{X}|\mathcal{S}_j), \tag{2.2}$$

for $j = 1, ..., S$ (where S is the size of $\boldsymbol{\mathcal{S}}$). The high level speech $\boldsymbol{Y}$ in $p(\boldsymbol{Y}|\mathcal{S})$ is replaced by $\boldsymbol{X}$ in Eq. 2.2, a proper way to represent the signal's characteristics.

2.2.1 Training

The features extracted from speech signals are used to train models for the speakers. Each speaker $\mathcal{S}_j$ is represented by a model λ_j, generated using only features from the speaker.

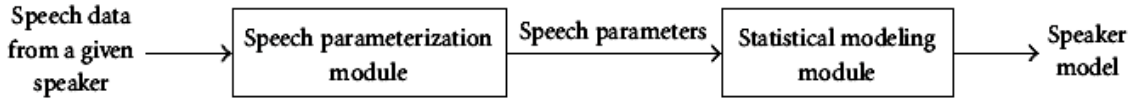

Figure 2.2: The statistical model of $\mathcal{S}$ is created from the speech signal $\boldsymbol{Y}$, *Bimbot et. al.* [1].

The idea behind the training stage is to make λ_j "memorize" the distinct characteristics present in the speaker's vocal tract that perform the identification. The GMSM, initially referenced in Section 1.2 and described in details in Chapter 4, is a perfect choice to model the λ_j's.

2.2.2 Test

The system test is performed replacing $\mathcal{S}_j$ in Eq. 2.2 by the model λ_j, leading to

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

where the λ_j with higher probability has its identity assigned to $\mathcal{S}$. The main difficult this system presents is the fact that every $\boldsymbol{X}$ must be tested with every $\mathcal{S}_j$ from $\boldsymbol{\mathcal{S}}$, as seen in Fig. 2.3, what demands a high amount of time.

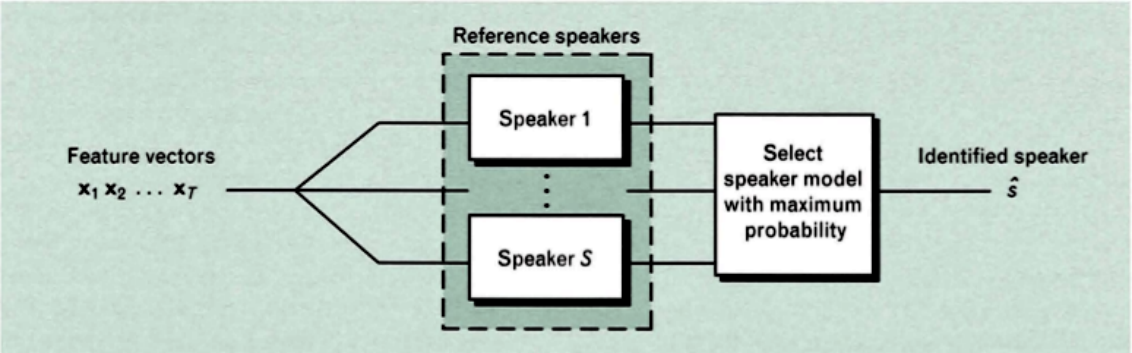

Figure 2.3: Speaker identification test, *Reynolds* [5].

2.3 Speaker Verification

If a speaker $\mathcal{S}$ claims to be a particular user $\mathcal{S}_i$ 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

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

where α is an arbitrary coefficient, should be enough (considering all speakers equally probable). However, a subset of enrolled speakers may have vocal similarities or the features $\boldsymbol{X}$ may be common to a large number of users, leading to a misclassification of an imposter as a registered speaker (a false detection). 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 and compare the likelihoods.

2.3.1 Likelihood Ratio Test

Given the vector of features $\boldsymbol{X}$, and assuming it was produced by only one speaker, the detection[1] 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 <u>not</u> 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 $p(\boldsymbol{X}|H_1)$, *Reynolds, Quatieri & Dunn* [14],

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

where the decision threshold for accepting or rejecting H_0 is θ (a low θ generates a more permissive system, while a high θ, a more restrictive). Applying the logarithm, the behavior of the likelihood ratio is maintained and Eq. 2.5 is replaced by the **log-likelihood ratio**

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

[1] the terms verification and detection are used interchangeably

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. 2.2:

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* [15], 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 $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$ 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 $\Lambda(\boldsymbol{X}) \geq \theta$, for an arbitrary value[2] of θ (see Fig. 2.4).

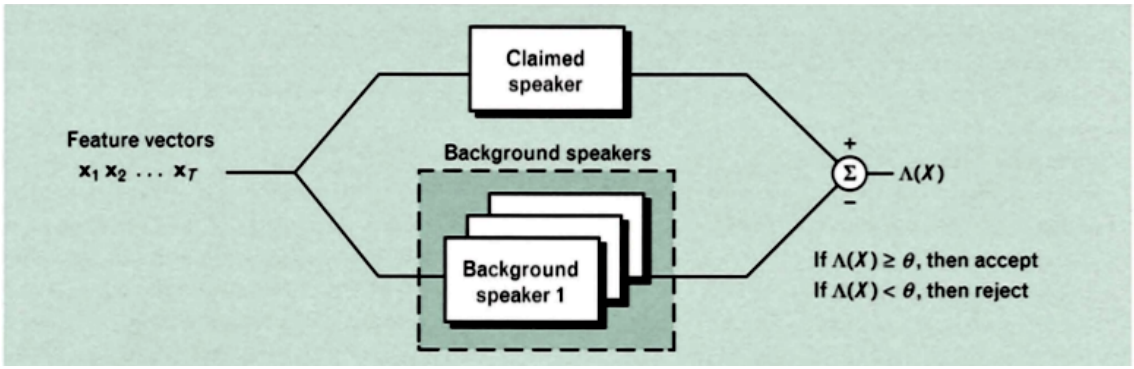

Figure 2.4: Likelihood-ratio-based speaker verification test, *Bimbot et. al.* [1].

[2] θ is loosely used here. The proper equation would be $\Lambda(\boldsymbol{X}) \geq \log\theta$.

3. Feature Extraction

As an acoustic wave propagated through space over time, the speech signal is not appropriate to be evaluated by an ASR system. 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* [16], 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* [17]. 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 conversation (e.g., accent).

The parametric representations evaluated in *Davis & Mermelstein* [11] may be divided into those based on the Fourier spectrum, such as Mel-Frequency Cepstrum Coefficients (MFCC) and Linear Frequency Cepstrum Coefficients (LFCC), and those based on the Linear Prediction Spectrum, such as 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 way the human ear operates. Despite the fact the ear is divided in three sections (i.e., outer, middle and inner ears), only the innermost 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.

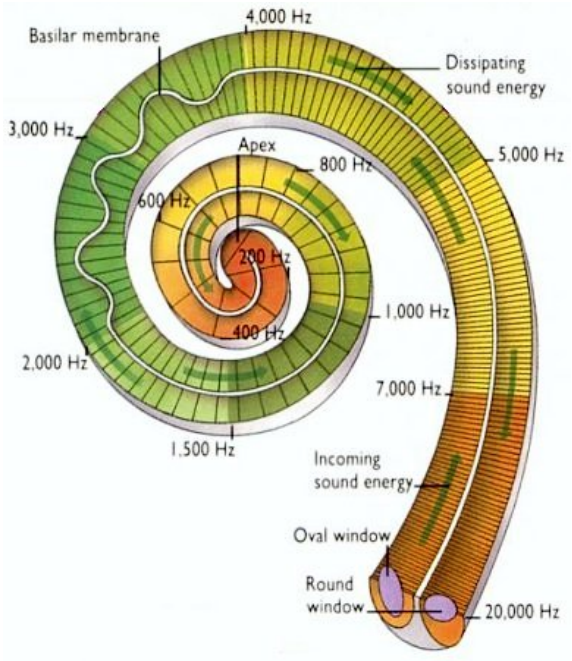

Figure 3.1: Cochlea divided by frequency regions, *ScienceBlogs* [18].

A key factor in the perception of speech and other sounds is **loudness**, a quality related to the physical property of the 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).

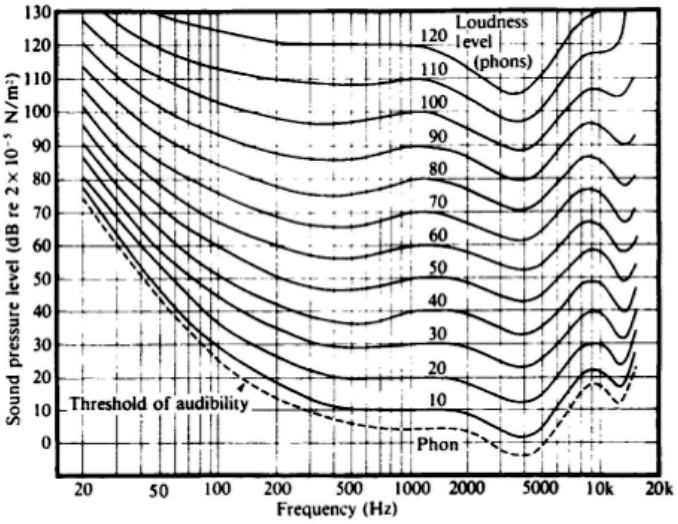

Figure 3.2: Loudness level for human hearing, *Fletcher & Munson* [19].

3.1.1 The Mel Scale

The **mel scale** is the result of an experiment conducted by *Stevens, Volkmann and Newman* [20] 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.* [20]) and so on.

Decades after the scale definition, *O'Shaughnessy* [21] 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 used Eq. 3.1, as shown by Fig. 3.3.

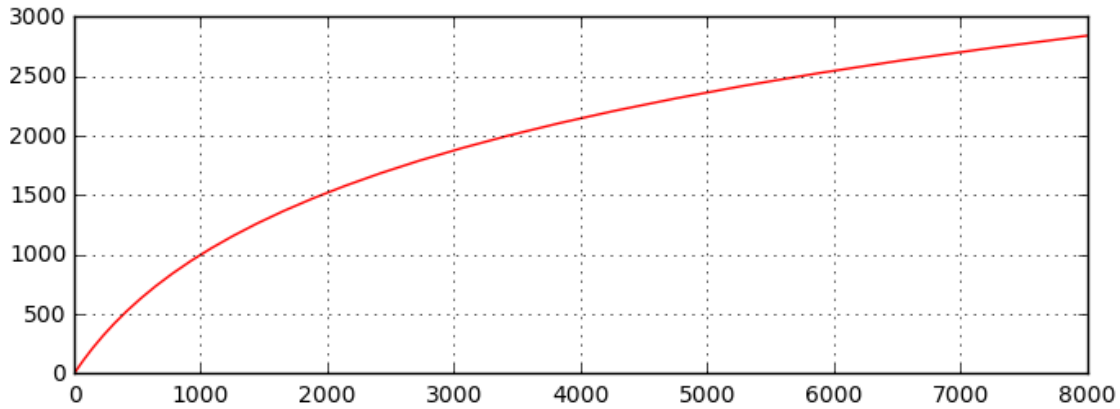

Figure 3.3: The logarithmic curve of the mel scale.

3.1.2 Extraction Process

In an ASR system 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. The process described ahead is mostly based on the ones in *Reynolds & Campbell* [6] and *Lyons* [22].

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 Finite Impulse Response (FIR) filter is used,

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

with values of α usually in the interval $[0.95, 0.98]$, *Bimbot et. al.* [1]. This is an optional stage of the MFCC extraction process.

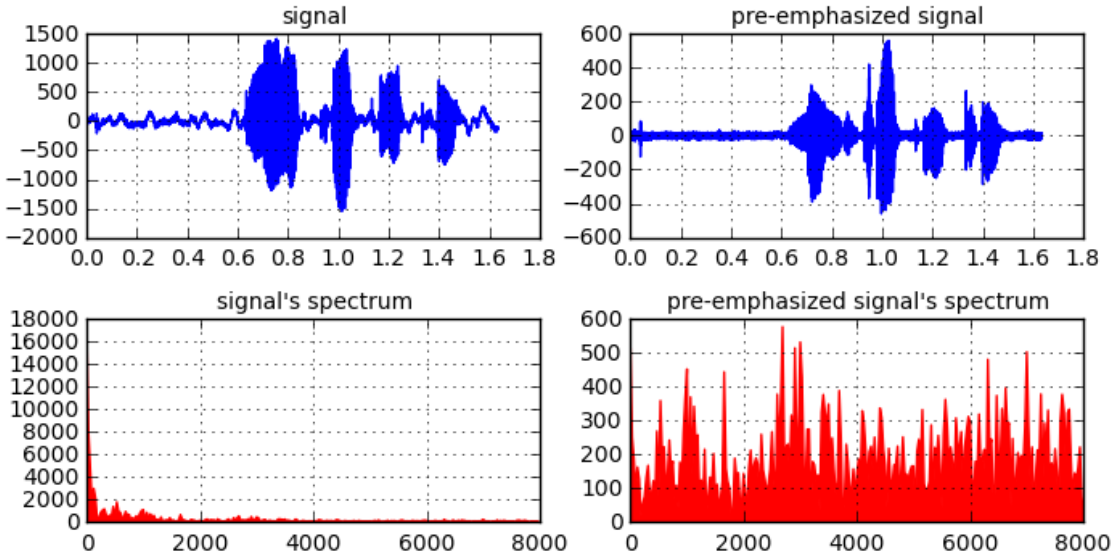

Figure 3.5: Raw and pre-emphasized ($\alpha = 0.97$) speech signals, with respective spectral magnitudes.

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 usually between 20 and 40 milliseconds (to perform a short-time analysis) and a shift that must be shorter than the width (commonly 10 milliseconds), or the frames will not overlap.

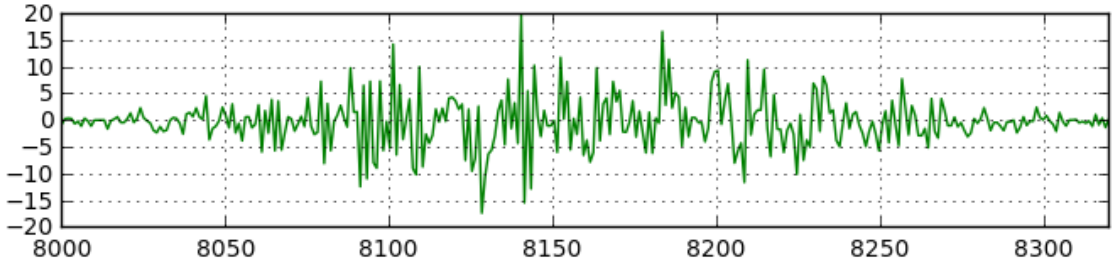

Figure 3.6: 51st frame. The samples at the ends are thinner than the ones at the middle.

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 taken and the power spectrum is obtained. Due to its symmetry, only the non-negative half is kept.

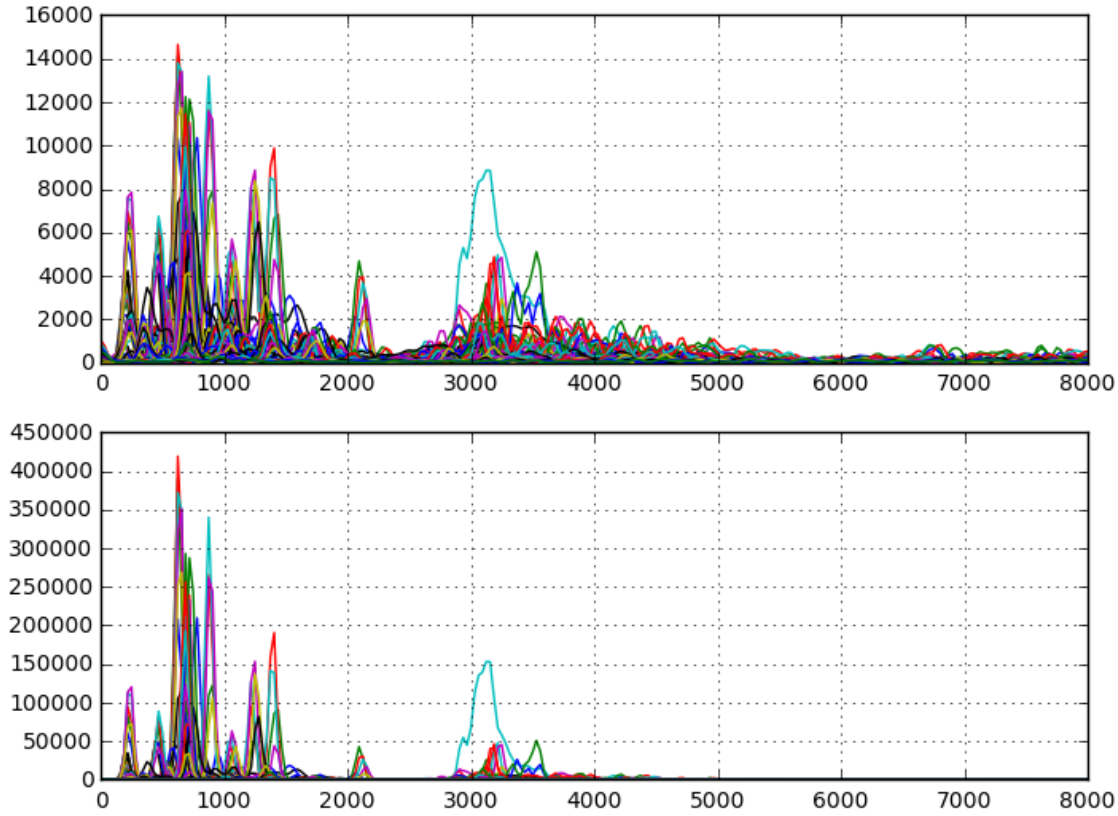

Figure 3.7: $|FFT|$ (top) and $|FFT|^2$ (bottom)

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).

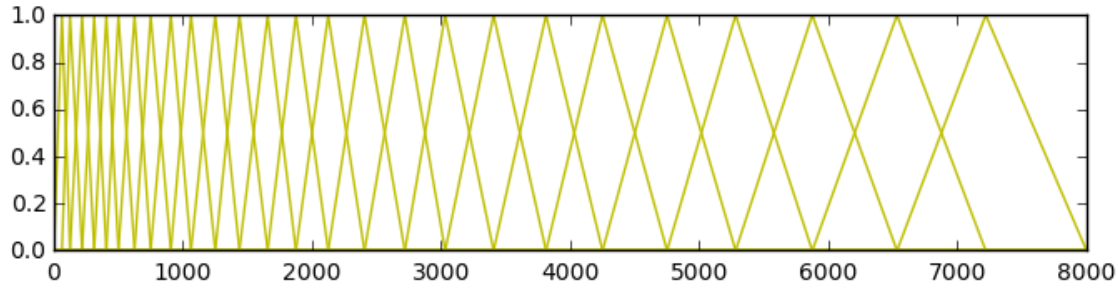

Figure 3.8: Filter bank with 26 filters.

The spectral coefficients are then converted to dB by the application of the function $20\log(\cdot)$ to each sample of each frame, reducing the differences between energy values.

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:

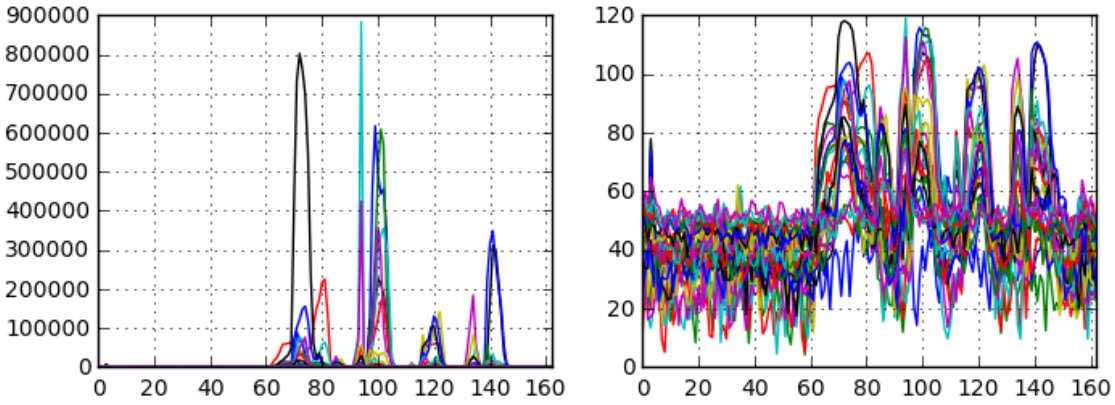

Figure 3.9: Spectral coefficients after the filterbank (left) and after the log conversion (right).

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

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.

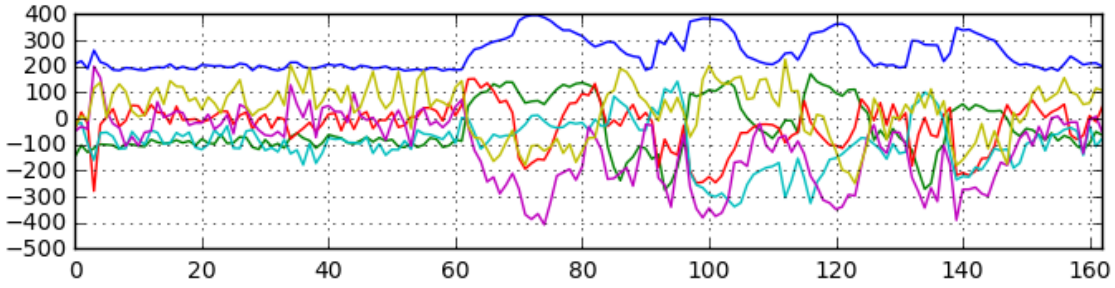

Figure 3.10: 6 MFCCs for each frame over.

In Fig. 3.10, the blue line represents the first feature, and as is clear, 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).

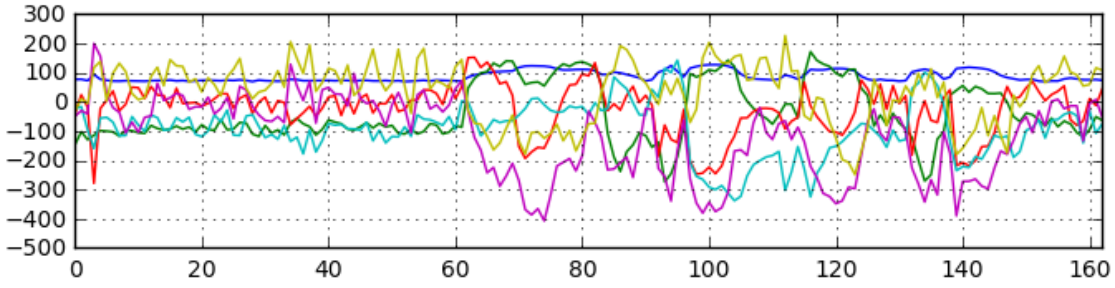

Figure 3.11: First feature changed by the summed energy of each frame.

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 (CMS),

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

reduces the disturbing channel effect before the ASR system be trained, delivering a cleaner signal to the models, *Westphal* [23].

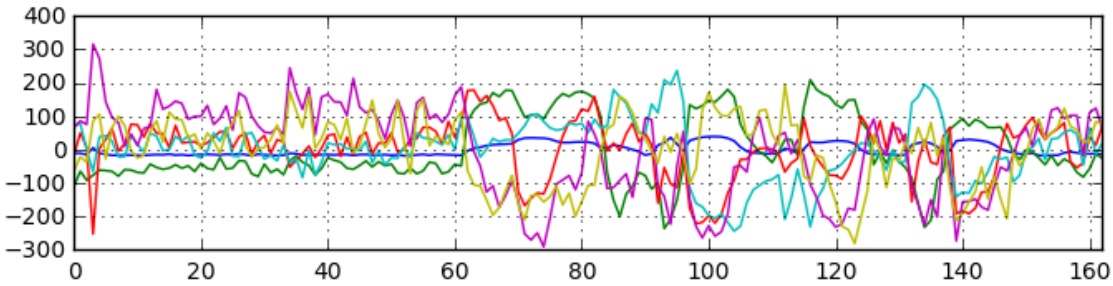

Figure 3.12: CMS applied to the MFCCs from Fig. 3.11.

In order to improve the speech parameters, the differences in time around 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 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.

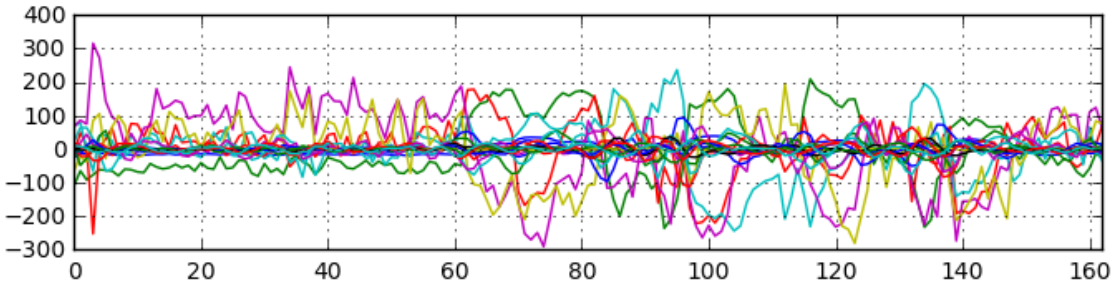

Figure 3.13: MFCCs from Fig. 3.12 with deltas of order 1 and 2 added.

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

4. Gaussian Mixture Models

Chapter 2 briefly discussed the use of models λ_i to perform an identification process and models λ_{hyp} and λ_{bkg} for a claimed speaker and for a background composed of all enrolled speakers, respectively, to a verification process. As the features from the speech signal (see Chapter 3) have unknown values until the moment of extraction, it is reasonable to model the ASR system 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, as demonstrated by 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, $\boldsymbol{\mu}$ the D-dimensional vector of means, $\boldsymbol{\Sigma}$ the $D \times D$ matrix of covariances, $|\boldsymbol{\Sigma}|$ the determinant of $\boldsymbol{\Sigma}$, and $(\boldsymbol{x} - \boldsymbol{\mu})'$ the transposed of the colum-matrix $(\boldsymbol{x} - \boldsymbol{\mu})$.

4.1 Definition

A weighted sum of $p(\boldsymbol{x})$'s is used to model the ASR system, estimating the composition that best represents the training data. This weighted sum is named Gaussian Mixture Model (GMM), first used for speaker recognition in *Reynolds* [24], and is given by

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

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

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

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* [25], leading to estimate $p(\boldsymbol{x}|\lambda)$ iteratively using the Expectation-Maximization (EM) algorithm.

4.2 Expectation-Maximization

The idea of the EM algorithm is to estimate a new model $\lambda^{(j+1)}$ from a previous model $\lambda^{(j)}$, that obeys $p(\boldsymbol{x}|\lambda^{(j+1)}) \geq p(\boldsymbol{x}|\lambda^{(j)})$, 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 following description of the steps uses a λ with **diagonal**[1] $\boldsymbol{\Sigma}_i$ (i.e., change the $D \times D$ matrix $\boldsymbol{\Sigma}_i$ for a D-dimensional vector $\boldsymbol{\sigma}_i$ of variances).

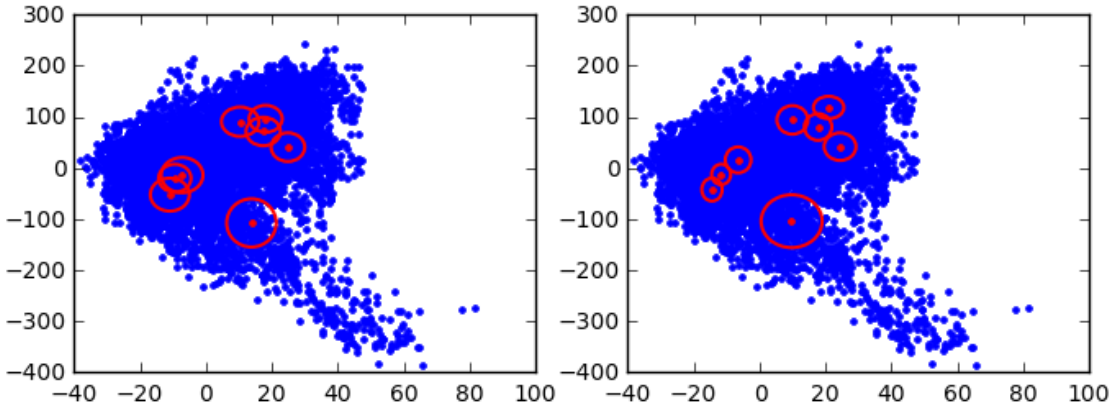

Figure 4.1: Features before (left), with random means and equal variances, and after (right) the EM algorithm. Only the first deviation is shown.

E-Step

The **expectation step** consists of estimating the *a posteriori* probabilities $P(i|\boldsymbol{x}_t, \lambda)$ for each distribution i and each feature vector $\boldsymbol{x}_t$, defined as

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

M-Step

In the **maximization step** the model is updated by recalculation of the parameters w_i, $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$, and the algorithm guarantees that each new $\lambda^{(j+1)}$ represents the training data better than the previous ones. From *Reynolds* [25], the updates of w_i, $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ are given by the equations below.

[1] As stated in *Reynolds et. al.* [14], diagonal covariance matrix GMMs outperform and are more computationally efficient than full covariance matrix GMMs. Also, the density modeling of an M-th order full covariance matrix GMM can equally well be achieved using a larger order diagonal covariance.

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 trains the GMMs used in the ASR system shown in Sections 2.2 and 2.3 and previously described in Section 4.1. Fig. 4.1 shows the mixture before and after the training.

4.3 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. The idea is to generate a model where common characteristics present in the corpus are well represented. This done, a speech mostly composed of common characteristics from enrolled speakers is more difficult to pass the likelihood ratio test (see Eq. 2.5).

There are many configurations for an UBM, but, as it is possible to see in *Reynolds et. al.* [14], male and female speakers present distinct vocal traits and are better represented when trained separately. Also, female voices have more intrasimilarities than males, leading to more distinct male configurations. The M-th order UBM in this study is created merging trained male and female models, both of order $M/2$ (see Fig. 4.2).

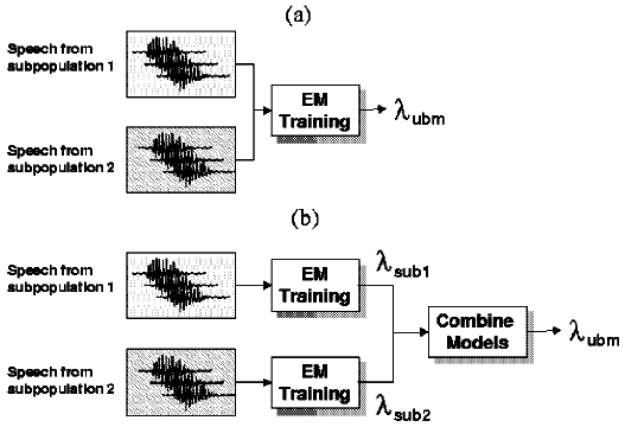

Figure 4.2: UBM with gender trained (a) together and (b) separately and combined, *Reynolds et. al.* [14].

As shown in Section 2.3, the likelihood ratio test is performed using the models λ_{hyp} and λ_{bkg}. The default ASR system is a GMM-UBM system, turning Eq. 2.9 in

$$\Lambda(\boldsymbol{X}) = \log p(\boldsymbol{X}|\lambda_{GMM}) - \log p(\boldsymbol{X}|\lambda_{UBM}). \tag{4.9}$$

4.4 Adapted Gaussian Mixture Model

As seen in Chapter 2 and in the previous sections, to perform the verification process a GMM for the claimed speaker and an UBM must be trained. Verify all speakers demands the training of Speaker Gaussian Mixture Models (SGMMs) for all enrolled speakers, a highly costly action (in time). An effective alternative is to take advantage of the well-trained M-th order UBM, since the GMMs and the UBM must have the same order to use Eq. 4.9, and adapt its parameters to generate a GMM for a speaker, *Brown et. al.* [8]. This technique provides a training faster than the GMM-UBM system (there is no loop such as in the EM algorithm) and tighter coupling between the speaker's model and the UBM, *Reynolds et. al.* [14]. The resultant GMM is named Adapted Speaker Gaussian Mixture Model (ASGMM).

The idea behind the Bayesian Adaptation[2] is to recalculate the gaussians from the UBM using only the desired speaker's features. If a gaussian does not well represent a portion of the data, the change is relevant, as seen in Fig. 4.3.

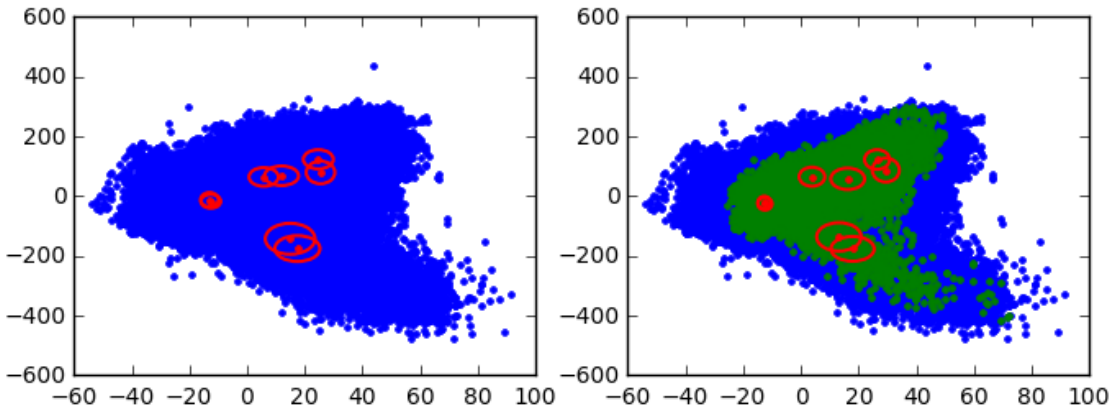

Figure 4.3: UBM trained (left) and weights, means and variances adapted for a male speaker (right). The blue dots are the background features and the green the speaker's.

The adaptation process is composed of two steps. The first is an expectation step, similar to the EM algorithm. Using $P(i|\boldsymbol{x}_t)$ (see Eq. 4.5) is possible to compute the sufficient statistics for the weight, mean, and variance parameters:[3]

$$n_i = \sum_{t=1}^{T} P(i|\boldsymbol{x}_t) \tag{4.10}$$

$$E_i(\boldsymbol{x}) = \frac{1}{n_i} \sum_{t=1}^{T} P(i|\boldsymbol{x}_t)\boldsymbol{x}_t \tag{4.11}$$

[2] Also known as **maximum a posteriori** (MAP) estimation.

[3] $\boldsymbol{x}^2$ is shorthand for diag($\boldsymbol{x}\boldsymbol{x}'$).

$$E_i(\boldsymbol{x}^2) = \frac{1}{n_i} \sum_{t=1}^{T} P(i|\boldsymbol{x}_t)\boldsymbol{x}_t^2 \tag{4.12}$$

Finally, these new sufficient statistics from the training data are used to update the old UBM sufficient statistics and adapt the parameters for mixture i (see Fig. 4.3) with the equations taken from *Reynolds et. al.* [14]:

$$\hat{w}_i = [\alpha_i n_i/T + (1 - \alpha_i)w_i]\gamma \tag{4.13}$$

$$\hat{\boldsymbol{\mu}}_i = \alpha_i E_i(\boldsymbol{x}) + (1 - \alpha_i)\boldsymbol{\mu}_i \tag{4.14}$$

$$\hat{\boldsymbol{\sigma}}_i{}^2 = \alpha_i E_i(\boldsymbol{x}^2) + (1 - \alpha_i)(\boldsymbol{\sigma}_i^2 + \boldsymbol{\mu}_i^2) - \hat{\boldsymbol{\mu}}_i{}^2. \tag{4.15}$$

The scale factor γ normalizes the weights. The adaptation coefficient controlling the balance between old and new estimates is α_i, given by

$$\alpha_i = \frac{n_i}{n_i + r}, \tag{4.16}$$

where r is a fixed relevance factor. If a mixture component has a low probabilistic count n_i, then $\alpha_i \to 0$ causing the deemphasis of the new (potentially undertrained) parameters and the emphasis of the old (better trained) parameters. For mixture components with high probabilistic counts, $\alpha_i \to 1$, causing the use of the new speaker-dependent parameters. The relevance factor r controls the strength of the new data in the adaptation process. Higher values of r demand that more data be observed in a mixture before new parameters begin replacing old ones (i.e., more relevant).

5. Experiments

This chapter details the experiments performed on the system described in the previous chapters, contemplating from the front-end processes until the speaker modeling and the log-likelihood ratio test (see Eq. 2.9). First, a description of the corpus is made. Later, the results are exhibited using the feature extraction process and the GMM-UBM techniques.

5.1 Corpus

The database used in the experiments of this work is *The MIT Mobile Device Speaker Verification Corpus* (MIT-MDSCV), *Woo et. al.* [10], a **corpus** designed to evaluate voice biometric systems of high mobility. All utterances were recorded using mobile devices of different models and manufacturers.

This corpus is composed of three sections. The first, named "Enroll 1", contains 48 speakers (22 females and 26 males), each with 54 utterances (names of ice cream flavors) of 1.8 seconds average duration, and is used to train the ASR system. The utterances were recorded in three different locations (a quiet office, a mildly noisy hallway, and a busy street intersection) as well as two different microphones (the built-in internal microphone of the handheld device and an external earpiece headset) leading to 6 distinct test conditions. The second section, named "Enroll 2", is similar to the first with a difference in the order of the spoken utterances, and is used to test the enrolled speakers. The third section, named "Imposters" is similar to the first two, but with 40 non-enrolled speakers (17 females and 23 males), and is used to test the robustness of the ASR system.

Section	Training	Test
Enroll 1	**X**	
Enroll 2		**X**
Imposters		**X**

Table 5.1: Corpus divided in training and test sets.

5.2 Coding and Data Preparation

5.3 Experiments and Results

5.3.1 Speaker Identification

5.3.2 Speaker Verification using SGMM

5.3.3 Speaker Verification using ASGMM

6. Conclusion

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

A. Results for Verification using Gaussian Mixture Speaker Model

B. Results for Verification using Adapted Gaussian Mixture Speaker Model

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] 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.

[15] 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.

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

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

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

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

[20] 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.

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

[22] James Lyons. *Mel Frequency Cepstral Coefficient (MFCC) tutorial*. 2012. URL: `http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/`.

[23] 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.

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

[25] 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.