Universidade Federal de Pernambuco

Centro de Informática

Graduação em Engenharia da Computação

Text-Independent Speaker Recognition Using Gaussian Mixture Models

A Dissertation in Computer Engineering

Eduardo Martins Barros de Albuquerque Tenório

Recife, June 18, 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 and was revised by Prof. Dr. George Darmiton da Cunha Cavalcanti, 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

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

Prof. Dr. George Darmiton da Cunha Cavalcanti

Recife, June 18, 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 intensive use of computational systems in the everyday of modern life creates the need for easier and less invasive forms of user recognition. While enter a password in a terminal and place an expert to identify a person are the status quo for respectively verification 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 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 is common, with people speaking with computers and cash withdrawals allowed through voice authentication.

Current commercial products based on voice technology (e.g., Dragon Naturally Speaking, KIVOX and VeriSpeak) usually intend 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, while computer login and telephone fraud prevention supplement a memorized personal identification code with speaker verification, *Reynolds* [5]. 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 is divided in two subareas. The first, **speaker identification**, is 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 that may be considered is **open set identification**, when a verification is used to guarantee the attributed identity. This type of recognition is not discussed in this paper and is presented here only illustratively. The term *speaker identification* refers only to the closed set modality.

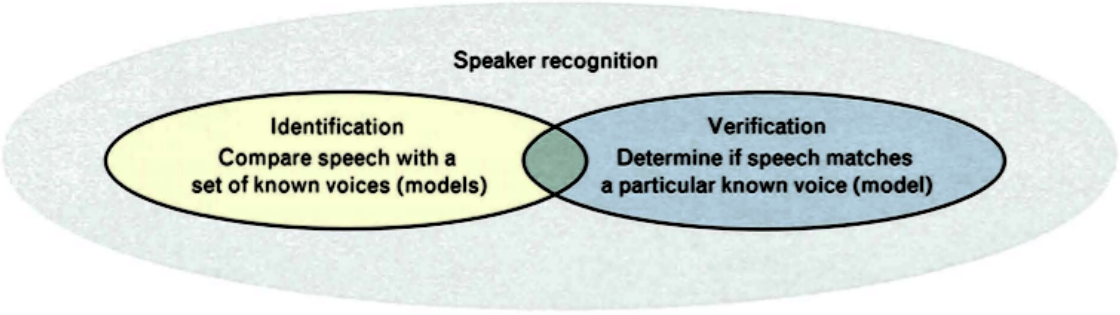

Figure 1.1: Relation between identification and verification of speakers, *Reynolds* [5].

The text inputted may have constraints, such as type (e.g., digits and letters) and number of words used (e.g., one word or sentences). In **text-dependent** systems the speech content 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 completely new training session. **Text-independent** systems have no input restrictions in both sets, with the non-textual characteristics of the speaker'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

This paper is focused on **text-independent speaker recognition**, and as independence on the message spoken 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 Gaussian Mixture Models (GMMs) to model the ASR systems.

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

Besides the previously cited, a new model using the theory of Fractional Covariance Matrix (FCM), *Gao, Zhou & Pu* [9], named Fractional Gaussian Mixture Model (FGMM), is examined and compared with the traditional speaker identification, aimed to verify a possible improvement in the outcomes. All GMM designs are explained in details in Chap. 4, as well as their implementations and experiments in Chap. 5.

1.3 Objectives

This study aims to implement ASR systems for identification and verification processes and analyze the following:

- Success rates for speaker identification with GMM and FGMM, using different sizes of mixture and features.
- Comparison between GMM and FGMM for speaker identification.
- False detection and false rejection rates for speaker verification with GMM and Adapted GMM (AGMM), using different sizes of mixture and features.
- Comparison between GMM and AGMM for speaker verification.

1.4 Document Structure

Chap. 2 contains basic information about ASR systems, as well as their basic architectures. The feature extraction process is explained in Chap. 3, from the reasons for its use to the chosen technique (Mel-Frequency Cepstral Coefficient, MFCC). In Chap. 4 the theories of GMM, UBM, AGMM and FGMM are presented. Experiments are described in Chap. 5, as well as their results. Finally, Chap. 6 concludes the study and proposes future works. Furthermore, Apx. A presents tables and figures for speaker identification with FGMM, while Apx. B presents tables and figures for speaker verifications with GMM and AGMM.

2. Speaker Recognition Systems

Speaker recognition 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 speaker with a claimed identity is either **enrolled** or **imposter**).

Automatic Speaker Recognition (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 $P(\mathcal{S})$ can be removed with no loss to the analysis, along with the *evidence* $p(\boldsymbol{Y})$ (just used for normalization). Eq. 2.1 is then replaced by

$$P(\mathcal{S}|\boldsymbol{Y}) = p(\boldsymbol{Y}|\mathcal{S}). \tag{2.2}$$

2.1 Basic Concepts

Before start the discussion about the types of ASR systems, two basic concepts (**utterance** and **features**) must be elucidated.

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 is defined as an utterance recorded and digitalized. The speech signal is the input for an ASR system.

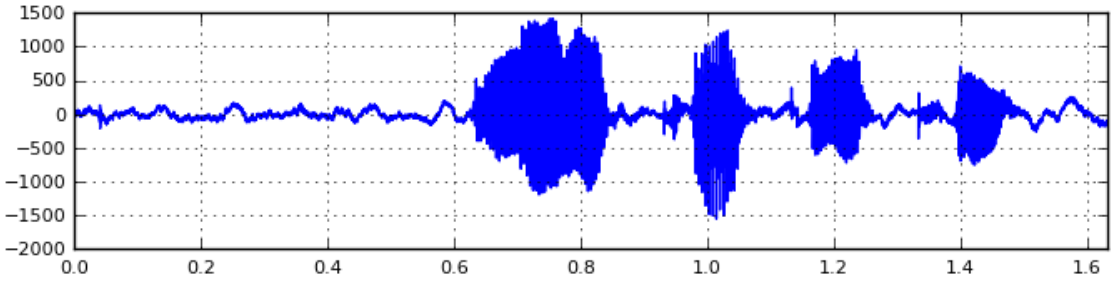

Figure 2.1: Speech signal from utterance "karen livescu", from the corpus defined in *Woo, Park & Hazen* [10].

2.1.2 Features

The raw speech signal is unfit for use by the classifier in an ASR system. For a correct processing, the representative characteristics (i.e., 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 Chap. 3.

2.2 Speaker Identification

Given a sequence of 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:

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

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. 2.3, a proper way to represent the signal's characteristics.

2.2.1 Training

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

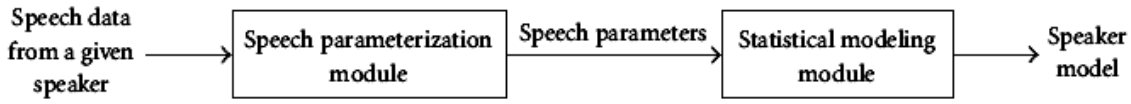

Figure 2.2: The statistical model of $\mathcal{S}$ is created from the speech signal $\boldsymbol{Y}$, *Bimbot et. al.* [1].

The idea behind the training stage is to make λ_j "memorize" the distinct characteristics present in $\mathcal{S}_j$'s vocal tract that provide the best representation. The SSGMM, initially referenced in Sec. 1.2 and described in details in Chap. 4, is a perfect choice to model the speakers.

2.2.2 Test

The system test is performed replacing the speakers $\mathcal{S}_j$'s in Eq. 2.3 by their models λ_j, leading to

$$\mathcal{S} \text{ is } \mathcal{S}_i \text{ if } i = \arg_j \max p(\boldsymbol{X}|\lambda_j), \tag{2.4}$$

where the λ_j with the highest likelihood has its identity assigned to $\mathcal{S}$. The main disadvantage this system presents is that every $\boldsymbol{X}$ must be tested against every $\mathcal{S}_j$ from $\boldsymbol{\mathcal{S}}$, what demands a high amount of time.

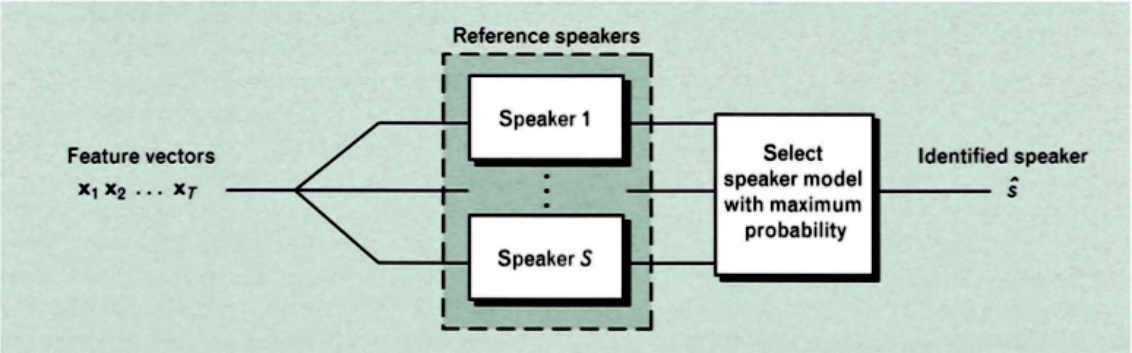

Figure 2.3: Speaker identification test, *Reynolds* [5].

2.3 Speaker Verification

If a speaker $\mathcal{S}$ claims to be a particular user $\mathcal{S}_C$ from $\boldsymbol{\mathcal{S}}$, the strength of this claim resides on how similar the features $\boldsymbol{X}$ are to the features from $\mathcal{S}_C$ used to model the system. Then, a simple equation

$$p(\boldsymbol{X}|\mathcal{S}_C) \begin{cases} \geq \alpha, & \text{accept } \mathcal{S}, \\ < \alpha, & \text{reject } \mathcal{S}, \end{cases} \tag{2.5}$$

where α is an arbitrary coefficient, should be enough. 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 determine not only if $\boldsymbol{X}$ is similar to the claimed speaker's features, but also its similarities to a set composed of all other enrolled speakers' features 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 $\mathcal{S}_C$;

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

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],

$$\frac{p(\boldsymbol{X}|H_0)}{p(\boldsymbol{X}|H_1)} \begin{cases} \geq \Theta, & \text{accept } H_0, \\ < \Theta, & \text{reject } H_0, \end{cases} \tag{2.6}$$

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. 2.6 is replaced by the **log-likelihood ratio**

$$\Lambda(\boldsymbol{X}) = \log p(\boldsymbol{X}|H_0) - \log p(\boldsymbol{X}|H_1). \tag{2.7}$$

[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_C and $\lambda_{\overline{C}}$ for H_0 and H_1, respectively. A high-level demonstration of the training of λ_C is shown in Fig. 2.2.

Due to λ_C be a model of $\mathcal{S}_C$, the features used for training (i.e., estimate $p(\boldsymbol{X}|\lambda_C)$) are extracted from speech signals produced by $\mathcal{S}_C$. The model $\lambda_{\overline{C}}$, however, is not well-defined. It should be composed of the features extracted from speech signals from all other speakers except $\mathcal{S}_C$, but creating a single $\lambda_{\overline{C}}$ 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 presence of each $\mathcal{S}_C$ weights approximately the same.

2.3.3 Test

As seen in Eq. 2.7, the decision process is based on a function *Score*. Replacing each H_i, for $i \in \{C, bkg\}$, by its corresponding model, the likelihood of a λ_i given $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$ can be written as

$$p(\boldsymbol{X}|\lambda_i) = \prod_{t=1}^{T} p(\boldsymbol{x}_t|\lambda_i). \tag{2.8}$$

Using the logarithm function, Eq. 2.8 is replaced by

$$\log p(\boldsymbol{X}|\lambda_i) = \frac{1}{T} \sum_{t=1}^{T} \log p(\boldsymbol{x}_t|\lambda_i), \tag{2.9}$$

where the term $\frac{1}{T}$ is inserted[2] to normalize the log-likelihood to the duration of the speech signal. That said, the likelihood ratio given by Eq. 2.7 becomes

$$\Lambda(\boldsymbol{X}) = \log p(\boldsymbol{X}|\lambda_C) - \log p(\boldsymbol{X}|\lambda_{bkg}), \tag{2.10}$$

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

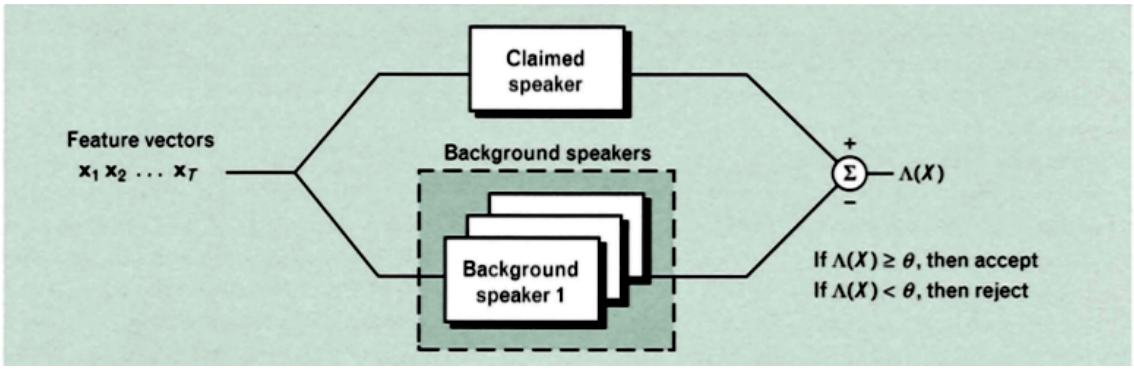

Figure 2.4: Likelihood-ratio-based speaker verification test, *Bimbot et. al.* [1].

[2]Eq. 2.9 is not an accurate application of the log function to Eq. 2.8, but an engineering solution.

3. Feature Extraction

As an acoustic wave propagated through space over time, the speech signal is inappropriate 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] are divided in those based on the Fourier spectrum, such as Mel-Frequency Cepstrum Coefficients (MFCC) and Linear Frequency Cepstrum Coefficients (LFCC), and in 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.

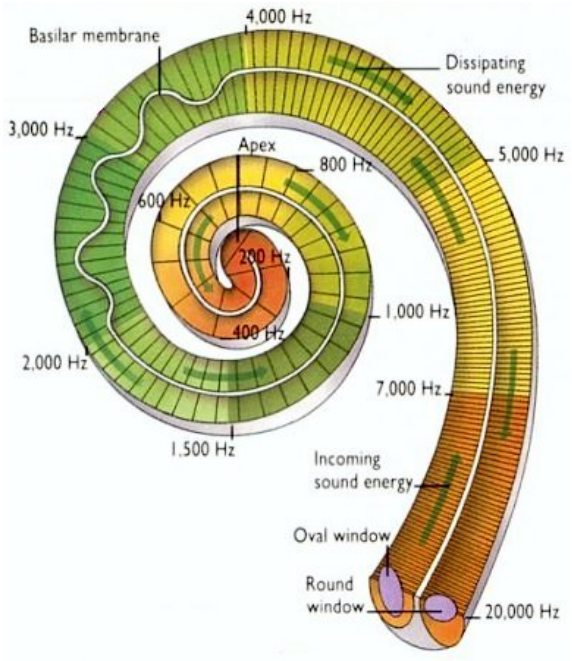

Figure 3.1: Cochlea divided by frequency regions, *ScienceBlogs* [18].

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).

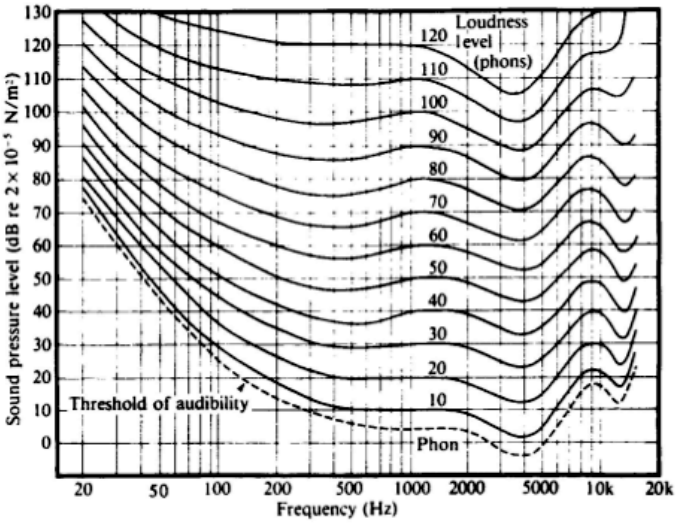

Figure 3.2: Loudness level for human hearing, *Fletcher & Munson* [19].

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 use Eq. 3.1, as shown by Fig. 3.3.

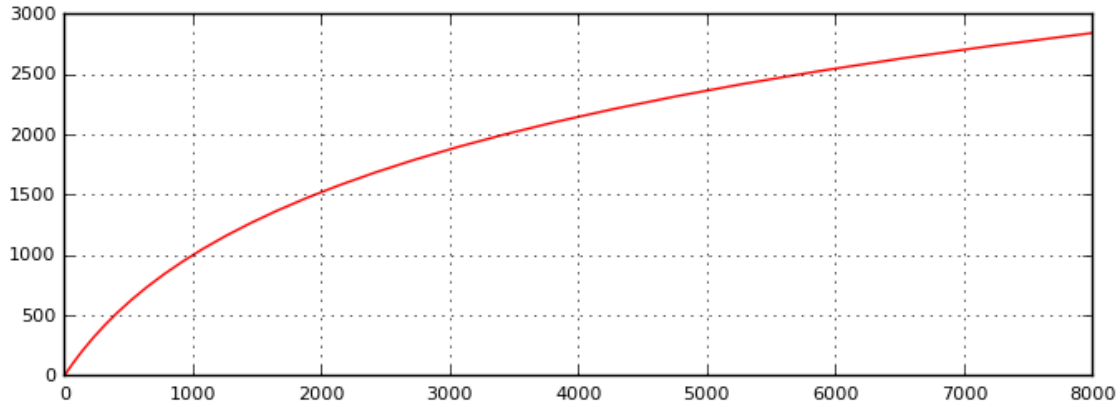

Figure 3.3: The logarithmic curve of the mel scale.

3.1.2 Extraction Process

The feature extraction module in the ASR system receives a raw speech signal and returns a vector of cepstral features in mel scale. 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 present in *Reynolds & Campbell* [6] and *Lyons* [22].

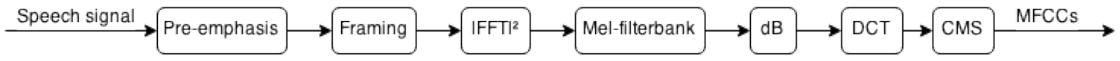

Figure 3.4: Modular representation of the MFCC extraction.

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, given by

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

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.

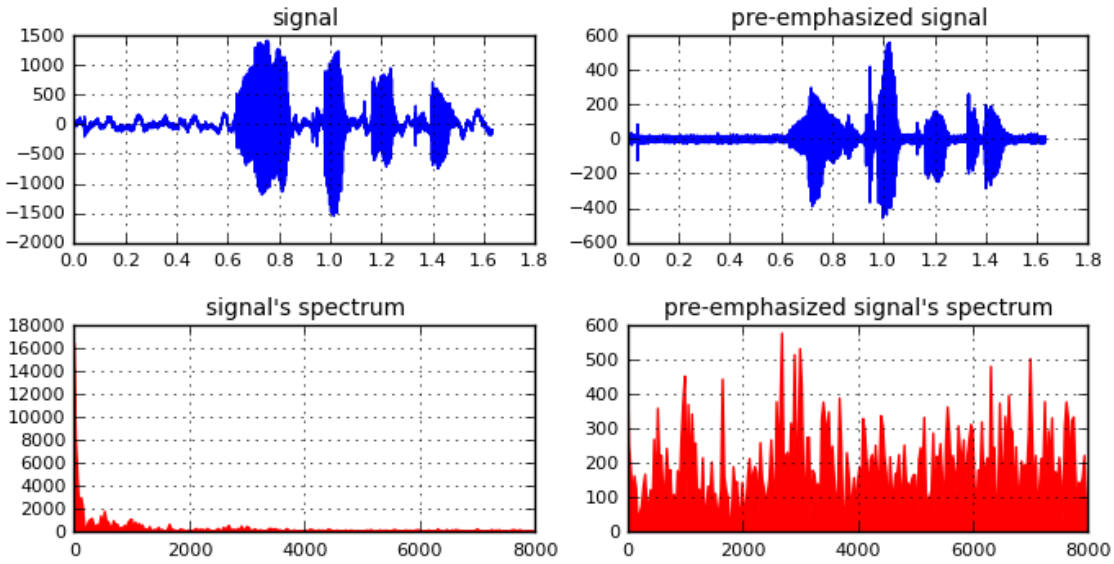

Figure 3.5: Raw and pre-emphasized ($\alpha = 0.97$) speech signals, with respective spectral magnitudes.

The first mandatory stage of the feature extraction 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.

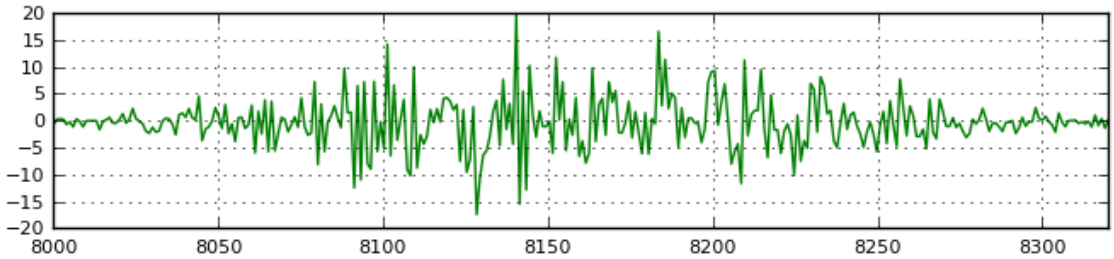

Figure 3.6: 51st frame. The samples at the ends are thinner than the ones at the middle.

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.

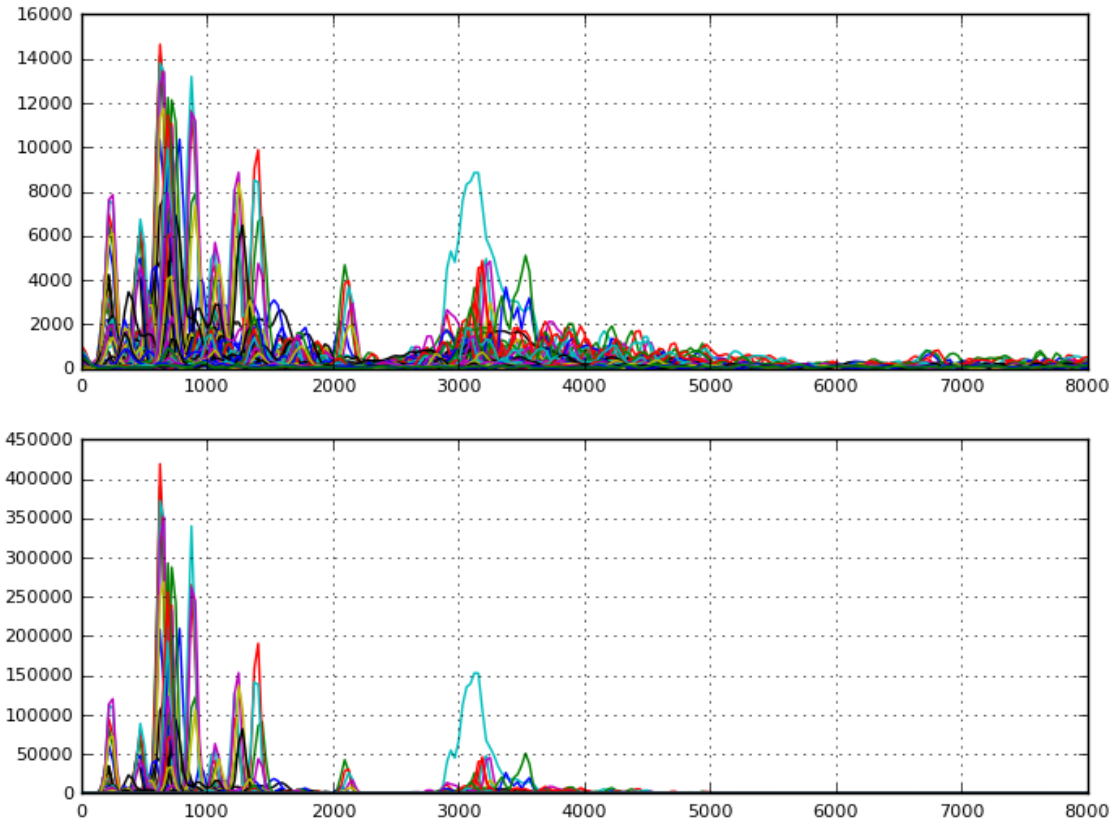

Figure 3.7: $|FFT|$ (top) and $|FFT|^2$ (bottom)

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, 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, as shown by Fig. 3.9.

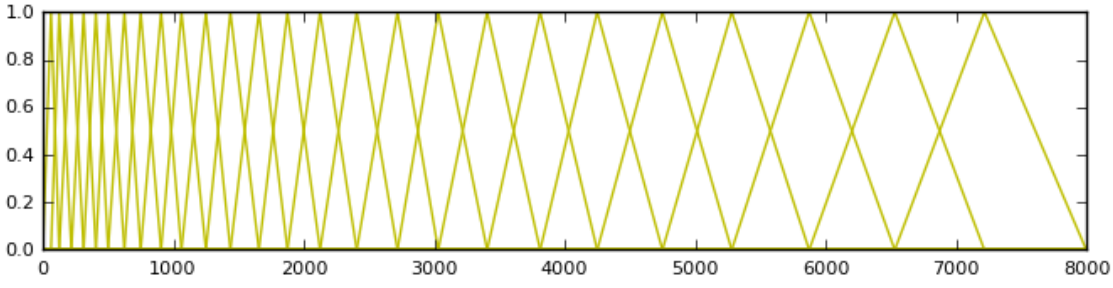

Figure 3.8: Filter bank with 26 filters.

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**:

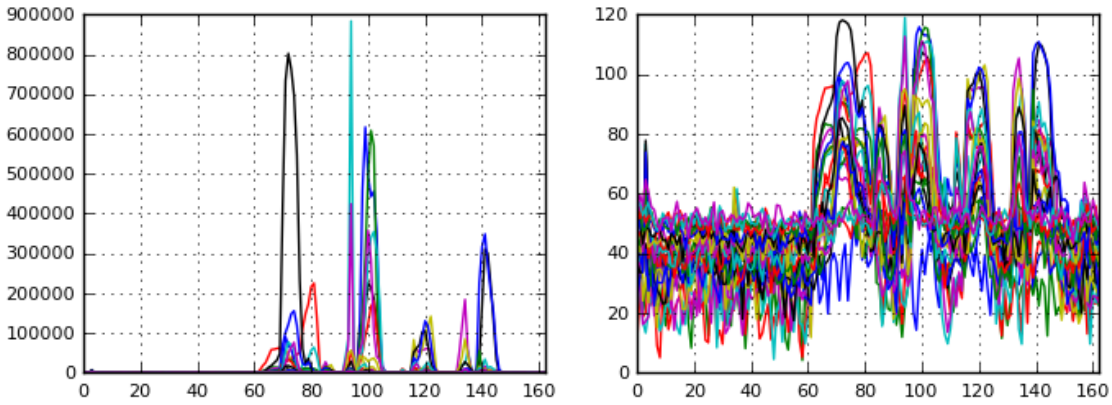

Figure 3.9: Spectral coefficients after the filterbank (left) and after the log conversion (right).

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

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.

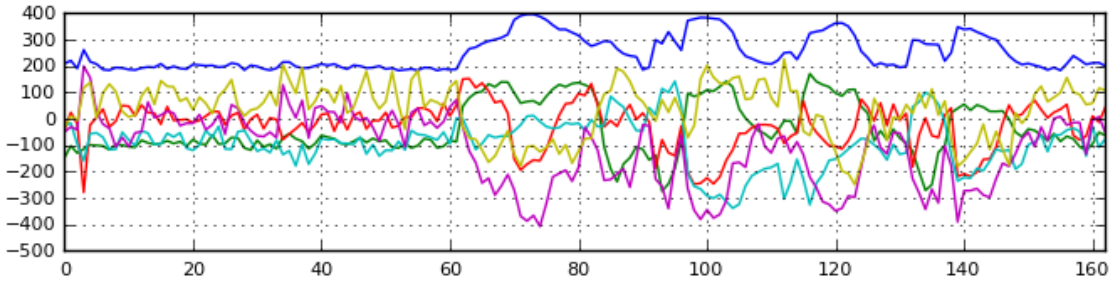

Figure 3.10: 6 MFCCs for each frame over time.

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.

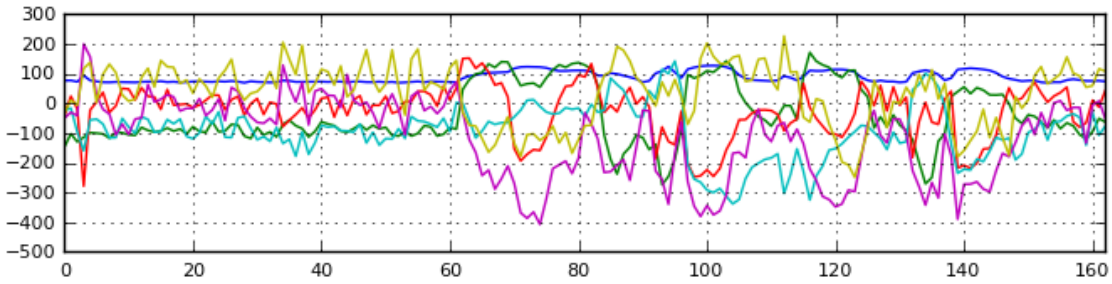

Figure 3.11: First feature changed by the summed energy of each frame.

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].

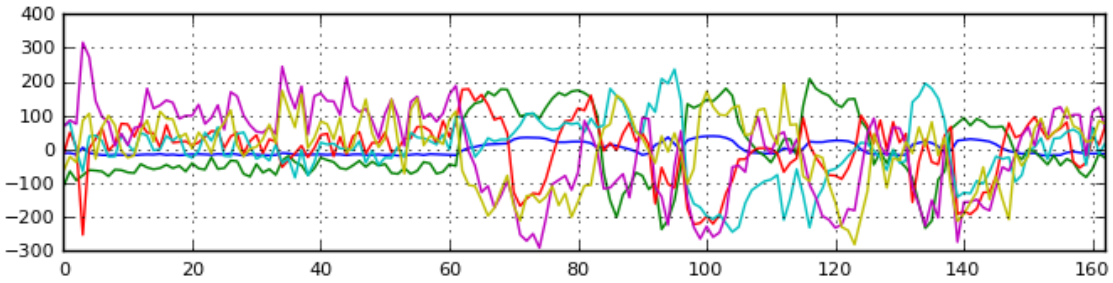

Figure 3.12: CMS applied to the MFCCs in Fig. 3.11.

In order to improve the speech parameters, the differences in time for each coefficient may be added as new features. In a vector with 6 features per frame, the velocity and acceleration of each coefficient provide 12 more features to the parametrization (totaling 18 features), all of them related to the ones previously extracted. These new features are the $\mathbf{\Delta}$s (deltas) of the MFCCs, given by

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

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 $\mathbf{\Delta}$s of first and second orders.

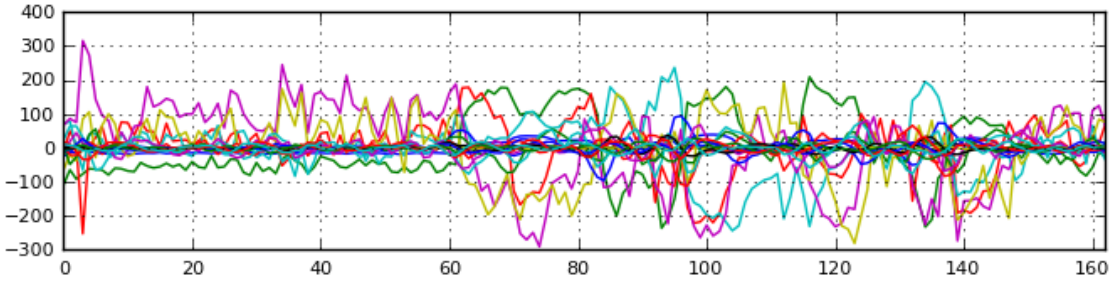

Figure 3.13: MFCCs from Fig. 3.12 with $\mathbf{\Delta}$s of 1st and 2nd orders added.

Eq. 3.5 may be used to calculate $\mathbf{\Delta}$s of any order, just as acceleration (second order) is derived from velocity (first order). However, as seen in Fig. 3.13, each order of $\mathbf{\Delta}$ delivers lower coefficients, providing a marginal gain for high orders.

4. Gaussian Mixture Models

Chap. 2 briefly discussed the use of models λ_i to perform an identification process and models λ_C 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 have unknown values until the moment of extraction, it is reasonable to model the ASR system to work with 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, $\boldsymbol{\mu}$ is a D-dimensional vector of means, $\boldsymbol{\Sigma}$ is a $D \times D$ matrix of covariances, $|\boldsymbol{\Sigma}|$ is the determinant of $\boldsymbol{\Sigma}$, and $(\boldsymbol{x} - \boldsymbol{\mu})'$ is the transposed of the colum-matrix $(\boldsymbol{x} - \boldsymbol{\mu})$.

4.1 Definition

For the general case, a single Gaussian distribution does not provide the most accurate representation. This issue is reduced using a linear combination of $p(\boldsymbol{x})$'s to model the ASR system, estimating the one that best represents the training data. This combination is named Gaussian Mixture Model (GMM), first used for speaker recognition in *Reynolds* [24], and given by

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

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 $i = 1, ..., M$. Each Gaussian in each model has its own covariance matrix (nodal covariance). 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 reason to model a speaker $\mathcal{S}$ using a GMM 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 $\boldsymbol{X} = \{\boldsymbol{x}_t\}$, the GMM's 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 parameter λ and direct maximization is not possible, *Reynolds* [25], leading to estimate $p(\boldsymbol{x}|\lambda)$ iteratively using the Expectation-Maximization (EM) algorithm.

In this paper, the GMM that models a single speaker will be reffered to as Single Speaker Gaussian Mixture Model (SSGMM), as initially cited in Sec. 1.2.

4.2 Expectation-Maximization

The idea of the EM algorithm is to estimate a new model $\lambda^{(k+1)}$ from a previous model $\lambda^{(k)}$, that obeys $p(\boldsymbol{X}|\lambda^{(k+1)}) \geq p(\boldsymbol{X}|\lambda^{(k)})$, better representing 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 $\boldsymbol{\sigma}_i^2$ of variances).

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

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

The λ present in Eq. 4.5 is the previously cited $\lambda^{(k)}$ for the current iteration.

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 $\lambda^{(k+1)}$ represents the training data better than the previous ones. From *Reynolds* [25], the updates of w_i, $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i^2$ 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:

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

[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.

Variances:

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

where $\lambda^{(k+1)} = \{(\overline{w}_i, \overline{\boldsymbol{\mu}}_i, \overline{\boldsymbol{\sigma}}_i^2)\}$, for $i = 1, \ldots, M$, and $\lambda^k = \lambda^{(k+1)}$ in the next iteration. This algorithm trains the GMMs used in the ASR systems shown in sections 2.2 and 2.3 and previously described in Sec. 4.1.

1: **procedure** EXPECTATION–MAXIMIZATION($\lambda, \boldsymbol{X}, threshold$)
2: $\lambda^k = \lambda$
3: $\lambda^{(k+1)} =$ M-Step($\lambda^k, \boldsymbol{X}$)
4: **if** $p(\boldsymbol{X}|\lambda^{(k+1)}) - p(\boldsymbol{X}|\lambda^{(k)}) \leq threshold \implies$ **goto** line 7
5: $\lambda^k = \lambda^{(k+1)}$
6: **goto** line 3
7: **end procedure**

The pseudo code above describes the EM algorithm. The *E-Step* is not shown, but is used inside the *M-Step*. Obviously, in an implementation the *E-Step* is calculated once for each iteration. Fig. 4.1 shows λ's gaussians before and after the EM algorithm.

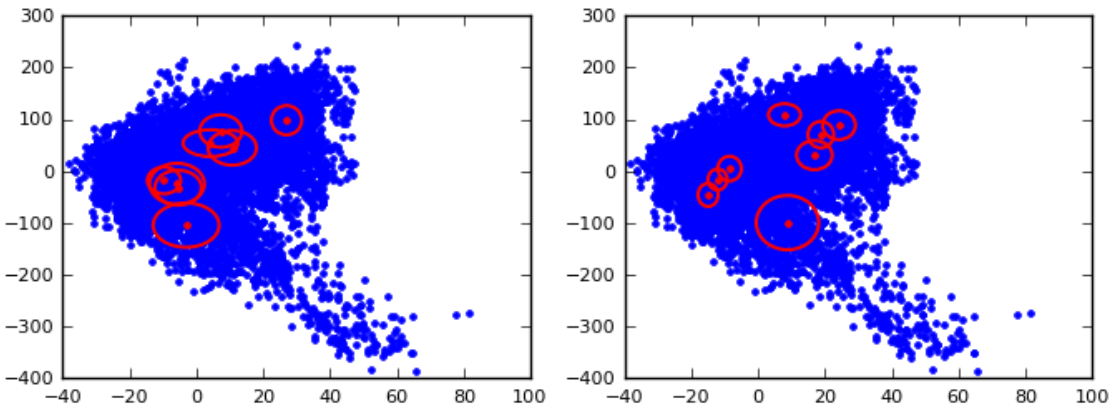

Figure 4.1: Gaussians before (left), partitioned using a single iteration k-means, and after (right) the EM algorithm. Only the first deviation is shown.

4.3 Universal Background Model

An Universal Background Model (UBM) is a GMM composed of features from all enrolled speakers and used in speaker verification. The idea is to generate a model λ_{bkg} where common characteristics present in this group are well represented. Then, a speech mostly composed of these characteristics is more difficult to succeed the likelihood ratio test, due to the low score produced by Eq. 2.10.

There are many configurations for an UBM, however, as seen 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 of order $M/2$ (see Fig. 4.2).

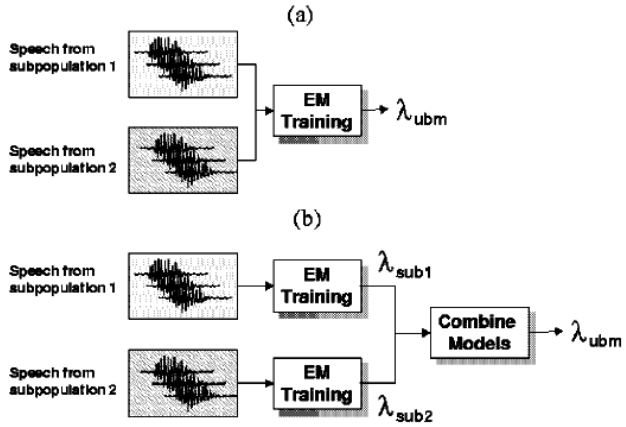

Figure 4.2: UBM with gender trained (a) together and (b) separately and combined, *Reynolds et. al.* [14].

As shown in Sec. 2.3, the likelihood ratio test is performed using the models λ_C and λ_{bkg}. The default ASR system is a SSGMM-UBM system, turning Eq. 2.10 in

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

4.4 Adapted Gaussian Mixture Model

As seen in Chap. 2 and in the previous sections, to perform the verification process a GMM for the claimed speaker and an UBM must be trained. Verify the entire corpus demands the training of SSGMMs 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 SSGMMs and the UBM must have the same order to use Eq. 4.9, and adapt its parameters to generate a new SSGMM for a speaker, *Brown et. al.* [8]. This technique provides a faster training than in the SSGMM-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 Single Speaker Adapted Gaussian Mixture Model (SSAGMM). Refactoring Eq. 4.9, the log-likelihood ratio test for Adapted Gaussian Mixture Model (AGMM) is

$$\Lambda(\boldsymbol{X}) = \log p(\boldsymbol{X}|\lambda_{SSAGMM}) - \log p(\boldsymbol{X}|\lambda_{UBM}). \tag{4.10}$$

The Bayesian adaptation[2] recalculates the gaussians from the UBM using features from the desired speaker only. If a gaussian represents the new data better than the old data, the change is relevant.

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

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

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

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

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

$$E_i(\boldsymbol{x}^2) = \frac{1}{n_i} \sum_{t=1}^{T} P(i|\boldsymbol{x}_t)\boldsymbol{x}_t^2 \tag{4.13}$$

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 with the equations taken from *Reynolds et. al.* [14]:

$$\hat{w}_i = [\alpha_i n_i/T + (1-\alpha_i)w_i]\gamma \tag{4.14}$$

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

$$\hat{\boldsymbol{\sigma}}_i{}^2 = \alpha_i E_i(\boldsymbol{x}^2) + (1-\alpha_i)(\boldsymbol{\sigma}_i^2 + \boldsymbol{\mu}_i^2) - \hat{\boldsymbol{\mu}}_i{}^2. \tag{4.16}$$

The scale factor γ normalizes the weights. The adaptation coefficient controlling the balance between old and new estimates is α_i, given by[4]

$$\alpha_i = \frac{n_i}{n_i + r}, \tag{4.17}$$

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 olds.

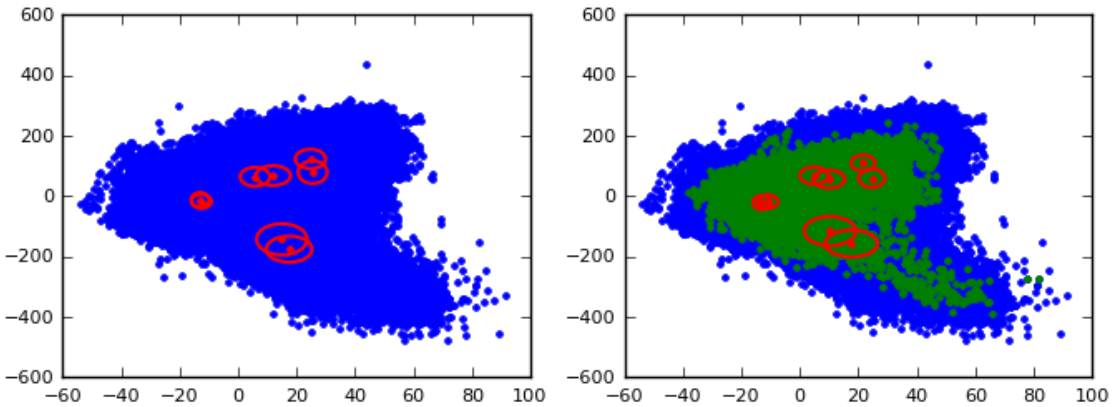

Figure 4.3: UBM trained (left) and weights, means and variances adapted for a female speaker (right) with $r = 16$. The blue dots are the background features and the green the speaker's. Only the first deviation is shown.

[4]The equation for α_i is a simplification used in *Reynolds et. al.* [14] with negligible loss. For the original equations for α_i^ρ, $\rho = \{w, \boldsymbol{\mu}, \boldsymbol{\sigma}^2\}$, visit the referenced paper.

4.5 Fractional Gaussian Mixture Model

Gao et. al. [9] uses the definition and applications of fractional moments to propose a new technique applying Fractional Covariance Matrix (FCM) in Principal Component Analysis (PCA) and Two-Dimensional Principal Component Analysis (2D-PCA), named Fractional Principal Component Analysis (FPCA) and Two-Dimensional Fractional Principal Component Analysis (2D-FPCA), respectively. The experiments are executed on two face image databases (ORL and Yale), using

$$\sigma^2 = E[(X^r - \mu^r)^2], \tag{4.18}$$

where E is the expected value and r is a real number, and show superior performance when choosing different values for r between 0 and 1. A value of 1 for r reduces Eq. 4.18 to the usual variance. As demonstrated in *Gao et. al.* [9], FPCA and 2D-FPCA deliver better projections than the usual PCA and 2D-PCA, respectively, making natural to extrapolate this idea for other types of signals and parametrizations.

The technique described in this section, named Fractional Gaussian Mixture Model (FGMM), also uses the theory of FCM to calculate matrices of covariances. As the matrices are diagonal, Eq. 4.18 is sufficient, changing Eq. 4.8 to

$$\overline{\boldsymbol{\sigma}}_i^2 = \frac{\sum_{t=1}^{T} P(i|\boldsymbol{x}_t, \lambda)(\boldsymbol{x}_t^r - \overline{\boldsymbol{\mu}}_i^r)^2}{\sum_{t=1}^{T} P(i|\boldsymbol{x}_t, \lambda)}. \tag{4.19}$$

One problem the FCM generates when applied to MFCCs is the emergence of complex numbers. A practical solution is to shift the MFCCs (see Fig. 4.4), turning all values in positive numbers and their minimums equal to 1:

$$c_n = c_n + (1 - \min_t c_{n,t}). \tag{4.20}$$

The distances in each dimension between points remain the same, maintaining the usual variances unchanged. Eq. 4.20 works correctly for GMMs with diagonal covariance matrices, due to the independence of each variable to the others. This ensures the correct uses of Eq. 4.18 to calculate the initial fractional variances and of Eq. 4.19 to train the models.

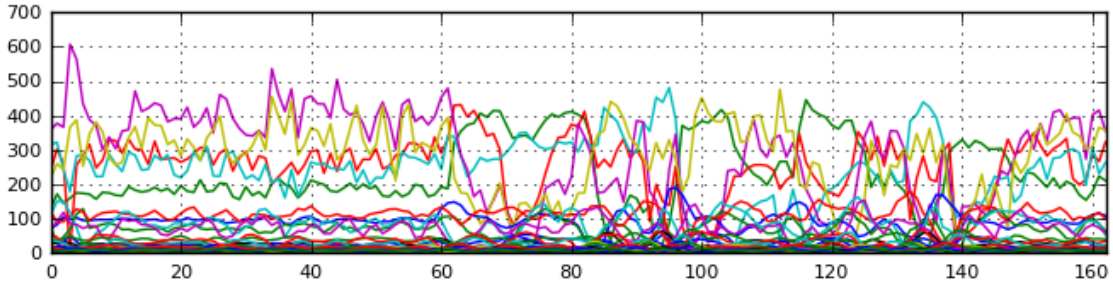

Figure 4.4: MFCCs from Fig. 3.13 shifted up. The minimum value for each feature is 1.

After the variances are calculated, the means are shifted again,

$$\mu_n = \mu_n - (1 - \min_t c_{n,t}), \tag{4.21}$$

returning to their supposed "original" values. This procedure avoids having to shift every vector of features, be it in the training or in the test sections, and is a practical solution to circumvent the problem.

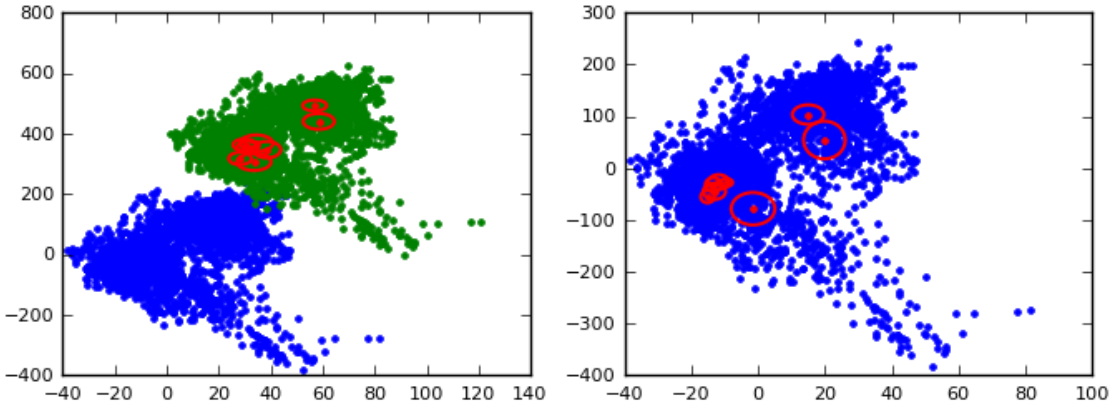

Figure 4.5: Features before (left), partitioned using k-means, and after (right) the EM algorithm. All variances (before and after EM) were calculated using FCM with $r = 0.95$. The blue points are the original features, while the greens are the shifted. Only the first deviation is shown.

The choice of shift all features to 1 is mostly based in common sense. To avoid non-negative values, just shift to 0 would be sufficient. However, when $r \to 0$, the results for X^r amd μ^r could fall in the indefinition 0^0, due to approximations of floating points in numeric computation. Also, for values greater than or equal to 1, the exponentiation remains monotonic.

5. Experiments

This chapter details the experiments performed on the systems described previously, contemplating from the front-end processes until the speaker modeling and the log-likelihood ratio test (see Eq. 2.10). First, a description of the corpus is made. Then, explanations about the implementation are given. At last, the results are exhibited using the feature extraction process and the GMM techniques.

5.1 Corpus

The database used 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, to avoid issues related to a specific device. Also, the recordings were performed in actual noisy environments, presenting the Lombard effect (i.e., speakers alter their style of speech in noisier conditions in an attempt to improve intelligibility).

The corpus is composed of three sessions. 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 an 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 training conditions equally dividing the session. The second, 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 session, 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 (imposters should be rejected). The table below summarizes the division of the corpus.

Session	Training	Test
Enroll 1	**X**	
Enroll 2		**X**
Imposter		**X**

Table 5.1: Corpus divided in training and test sets.

All utterances are recorded in uncompressed WAV files using a single channel. For each utterance record there is a correspondent text file containing pertinent information, such as speaker, microphone, location, message content and etc. (see Tab. 5.2).

Despite being a base for speaker verification systems, in this paper MIT-MDSCV is also used for speaker identification experiments. The difference is that only "Enroll 1" and "Enroll 2" are used, for training and test respectively. In an ideal identification system

all utterances from "Enroll 2" are correctly identified and in an ideal verification system the false detection and false rejection rates are zero.

Speaker	f00
Session	1
List	female_list_3a
Gender	female
Location	Office
Microphone	Headset
Phrase	karen livescu

Table 5.2: Information from first utterance of first speaker in session "Enroll 1".

5.2 Coding and Data Preparation

The systems described throughout the paper were implemented in the Python programming language, version 3.4.3, and the frameworks NumPy 1.8.1 and SciPy 0.14.0 were used to perform most of the calculations as matrices. Also, Matplotlib 1.4 was used to plot the results as figures. All codes can be found in `github.com/embatbr/tg` and are free to be used, since properly referenced.

The implementation is divided in 6 modules. Module **features** contains codes to execute the feature extraction through the MFCC algorithm. Each stage is executed by a different function. The main function joins all stages, receiving a speech signal in waveform and delivering a matrix of features over frames. Most of this module was written using codes from *Lyons* [22]. The GMM is implemented in module **mixtures** as a class containing methods for training usign EM algorithm and for bayesian adaptation. The module also contains functions to execute the *k-means* clustering, used during the GMM creation. Functions to extract the features from MIT-MDSCV and to read groups of feature data (e.g., single speaker and background) are present in module **bases**. The module **main**, as the name denounces, is filled with functions to execute every functionality needed. Through command lines it is possible to extract MFCCs from the base, train and adapt models, identify or verify speakers, calculate rates and draw curves and etc. Module **show** is aimed to generate figures to fill the previous chapters (mostly chapters 3 and 4). At last, module **common** contains useful constants and functions shared by the other modules.

5.2.1 Parameters

The MFCCs extracted used a filterbank of size 26, maintaining only the first 19 features, and deltas of order 0, 1 and 2, leading to MFCCs with final feature number of 19, 38 and 57, respectively. The deltas were calculated using $K = 2$. Before any delta calculation, the energy appending and CMS steps were performed. The relevance factor r in AGMM had a default value of 16 for all combinations of adaptations. The implemented EM algorithm stops the training when $\log p(\boldsymbol{X}|\lambda^{(k+1)}) - \log p(\boldsymbol{X}|\lambda^{(k)}) \leq 10^{-3}$.

All speakers were modeled using SSGMMs with sizes (Ms) equal to 8, 16, 32, 64 and 128, each divided in 4 types of environment configuration: quiet office, mildly noisy hallway, busy street intersection and all three together. This leads to 48 speakers $\times$ 5 Ms $\times$ 4 environments $\times$ 3 delta orders $=$ 2880 trained SSGMMs. The UBM is a combination of

a trained male UBM and a trained female UBM (see Fig. 4.2b), for all sizes of mixtures, types of environments and orders of delta, totaling 60 models. Each one of the 4 combinations of SSAGMM also contains 2880 models. Adding the Single Speaker Fractional Gaussian Mixture Models (SSFGMMs) for all 5 values of r (0.95, 0.99, 1, 1.01 and 1.05), the total number of trained models is 28860.

5.2.2 Algorithmic Issues

Initialization:

According to *Reynolds* [25], different methods of initialization lead to different local maxima reached by the EM algorithm. Following the approach used in the referenced paper, in this work the models are generated randomly selecting the means and executing a single iteration *k-means* to initialize the means, nodal variances and weights. A complete k-means (iterate until all clusters remain unchanged) during the initialization, followed by EM training, leads to a similar result, making the choice of single iteration k-means in the initialization the logical decision.

Variances:

During the M-step of the EM algorithm, some variances may have their values decreased significantly. This represents a degeneration of the model, occuring when the size of mixture increases more than needed to represent the data or when the data is insufficient to train the model, *Reynolds* [25]. This issue occurred in SSGMMs and SSFGMMs with 64 or 128 gaussians and UBMs with 128 gaussians. To prevent the degeneration, the EM algorithm receives a constraint: when the variance σ^2 is lower than a minimum variance $\sigma^2_{min} = 0.01$, σ^2_{min} is assigned to σ^2. This test is made in every iteration and provides more robustness to the ASR systems (correct the degradation when it occurs, avoiding error propagation).

Monotonicity:

The Expectation-Maximization is a monotonic algorithm. For every iteration k, the rule $\log p(\boldsymbol{X}|\lambda^{(k+1)}) \geq \log p(\boldsymbol{X}|\lambda^k)$ is always true. The algorithm stops only when a local maximum is reached (the likelihood remains stable) or when the increase is smaller than a defined threshold and there is no practical gain in continue. This rule is not true for an EM performed by a FGMM. The 4 non-unitary values of r presented this discrepancy, more frequent with the increase in distance to 1. When $\log p(\boldsymbol{X}|\lambda^{(k+1)}) < \log p(\boldsymbol{X}|\lambda^k)$ occurs, the stop condition is satisfied and the EM returns the last calculated likelihood.

5.3 Experiments and Results

Using the trained models, two types of experiments were performed. The first, speaker identification, was executed using SSGMMs and SSFGMMs. Identification through SSGMM is an experiment known since *Reynolds* [24] was published in 1992, with countless revisions, improvements and derivations. Conversely, identification through SSFGMM is a technique never tried before and for that reason FGMM was used in speaker identification to validate the idea before further experiments. At the end SSGMM and SSFGMM were compared. The second type, speaker verification, was executed using SSGMMs and

SSAGMMs, providing false detection and false rejection rates. The verification through SSAGMM was executed using only combinations of adaptations that produced curves with a behavior similar to the verification through SSGMM (see the previous section). Finally, a comparison between SSGMM and SSAGMM was performed.

The metrics used for evaluation differ according to the type of ASR system studied. For identification the interest is in know how well the system correctly identify an enrolled user. The logical way is by analyzing the success rate (the closer to 100% the better). For verification, the major concern for the system designer is to avoid misclassification. An imposter can be considered an enrolled speaker (false detection), and an enrolled speaker can be considered an imposter (false rejection). To evaluate the correctness of a system, a Detection Error Tradeoff (DET) curve is used. The perfect point of operation is when false detection and false rejection rates are equal, a measure named Equal Error Rate (EER). EER is used as a starting point to create more restrictive or permissive systems.

In the experiments performed, all 54 utterances from each speaker were tested against any one of the models created and trained for each environment configuration. The first, **office**, is practically a noise free space. The second, **hallway**, presents a moderate level of noise. The third, **intersection**, is a very lound environment. The last is a combination of all three previous environments, providing a model trained with different levels of noise, and thus, more robust. This approach allows a more refined analysis of the system's response to background noise.

5.3.1 Speaker Identification using SSGMM

As shown in Eq. 2.4, given a set $\boldsymbol{\lambda} = \{\lambda_1, ..., \lambda_S\}$ of SSGMMs, a sequence of features $\boldsymbol{X}$ is assigned to a speaker $\mathcal{S}_i$, represented by a model λ_i of $\boldsymbol{\lambda}$, if $p(\boldsymbol{X}|\lambda_i)$ is greater than all $p(\boldsymbol{X}|\lambda_j)$, for $\lambda_j \in \boldsymbol{\lambda}$ and $j \neq i$.

In this study Eq. 2.4 is used for all 54 speech signals of each enrolled speaker from session "Enroll 2". The number of tests is 54 utterances $\times$ 48 enrolled speakers $\times$ 4 environment configurations $\times$ 5 sizes of mixture, totaling 51840. The following table shows the speaker identification success rates (in percent).

Δ	M	Office	Hallway	Intersection	All
0	**8**	41.55	52.66	42.48	64.66
	16	46.76	55.79	45.64	72.65
	32	50.08	58.68	47.53	77.93
	64	50.08	57.52	47.22	80.52
	128	47.84	52.93	44.48	81.21
1	**8**	44.41	53.28	43.98	66.20
	16	50.58	61.30	50.81	78.12
	32	53.78	65.20	53.09	85.03
	64	54.21	64.43	52.43	87.85
	128	52.82	59.53	49.42	88.46
2	**8**	45.02	56.06	46.60	68.56
	16	50.62	62.81	50.89	79.32
	32	54.44	65.39	54.98	85.69
	64	56.33	63.93	54.67	89.54
	128	52.47	62.00	51.08	89.97

Table 5.3: Identification rates for enrolled speakers.

As seen in the table above and in Fig. 5.1, for the fourth environment configuration,

named *all*, the success rates increase with M, while for *office* they reach a maximum value at $M = 64$, and for *hallway* and *intersection*, at $M = 32$. As explained in the previous section, SSGMMs suffered degeneration due to a decrease in the values of variances for 64 and 128 gaussians. A higher value of M led to higher misrepresentation of the samples by the SSGMM. The exception to this "rule" occurs when all environments are used in the training, that may be explained due to the higher number of samples (each speaker has 18 utterances recorded in each environment, leading to all 54 speech signals being used to train the model). Also as expected, when all environments are present, the SSGMM represents different levels of background noise, what produces a more robust modeling.

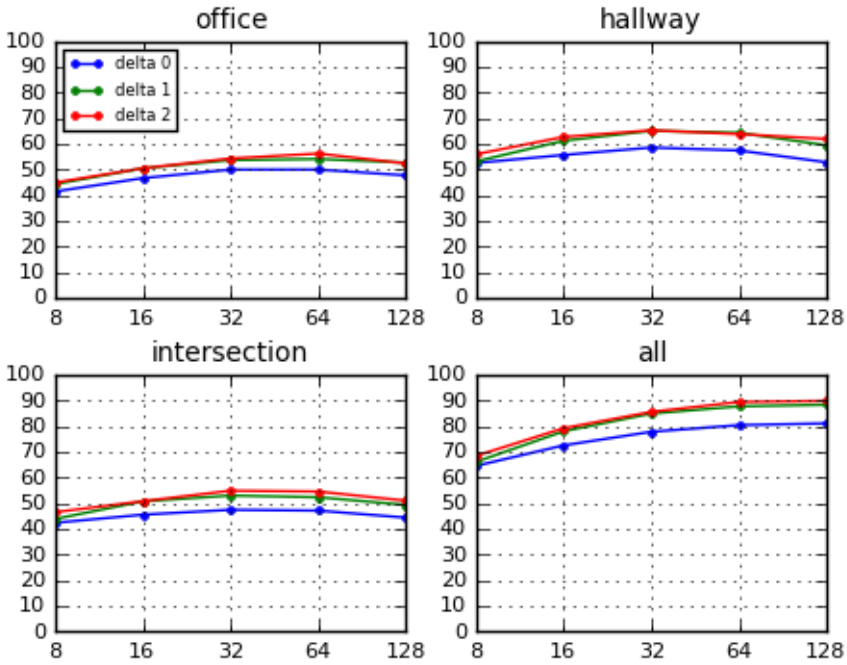

Figure 5.1: Identification rates for enrolled speakers.

5.3.2 Speaker Identification using SSFGMM

The only difference between this experiment and the previous is the type of GMM used. In traditional speaker identification, each speaker $\mathcal{S}_j$ is represented by a λ_j, using the GMM defined in Sec. 4.1 and the EM algorithm described in Sec. 4.2. When FCM is used to model the covariance matrix, the SSGMM is replaced by SSFGMM with values 0.95, 0.99, 1.00, 1.01 and 1.05 for r, maintaining the rest of the experiment unchanged.

The choice of values for r was made using

$$r = r_0 + (-1)^u \delta, \tag{5.1}$$

where $r_0 = 1$, $u \in \{0, 1\}$ and $\delta \in \{0.01, 0.05\}$, providing rs higher and lower than r_0 to conduct a primary study of which values increase the success rates. Apx. A contains all tables and figures showing the results for this experiment.

A comparison of Tab. 5.3 and Fig. 5.1 with the tables and figures from Apx. A indicates that SSFGMMs trained using utterances from all environments and values 0.99, 1 and 1.01 for r present similar speaker identification success rates to the experiment using SSGMM. The differences are around 1 percentage point, what may not be characterized

as an improvement before further investigations (a slightly better success rate may be attributed to the random initialization of the model). For r equals to 0.95 and 1.05, the success rates are near to 20 percentage points lower. A value lower than 1 decreases the variances, approaching zero when $r \to 0$, while a value higher than 1 increases the variance (with the deviation passing the data limits) and tends all means to the same point. Also, the EM algorithm violates the convergence rule faster (1 or 2 iterations).

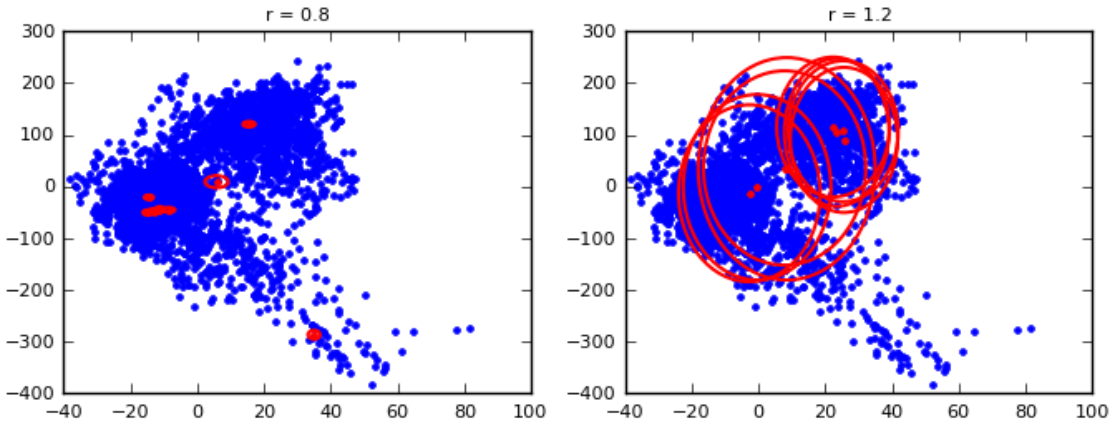

Figure 5.2: GMM "trained" using FCM, with $r = 0.8$ (left) and $r = 1.2$ (right). Only the first deviation is shown.

6. Conclusion and Future Studies

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

A. Identification (SSFGMM)

Δ	M	Office	Hallway	Intersection	All
0	**8**	38.70	44.41	32.37	50.50
	16	41.63	46.37	32.56	62.35
	32	47.72	48.53	37.46	68.06
	64	43.75	50.31	37.27	72.80
	128	38.62	42.75	31.06	72.15
1	**8**	33.37	31.67	26.35	44.87
	16	41.13	42.32	26.62	54.71
	32	44.95	47.92	30.29	64.47
	64	43.13	43.36	31.64	70.95
	128	33.14	37.15	21.10	73.84
2	**8**	32.21	33.49	26.66	43.02
	16	41.09	42.40	31.10	54.67
	32	46.33	44.14	31.75	66.78
	64	40.93	43.60	33.53	72.72
	128	39.16	37.89	23.26	73.53

Table A.1: Identification rates for enrolled speakers with $r = 0.95$.

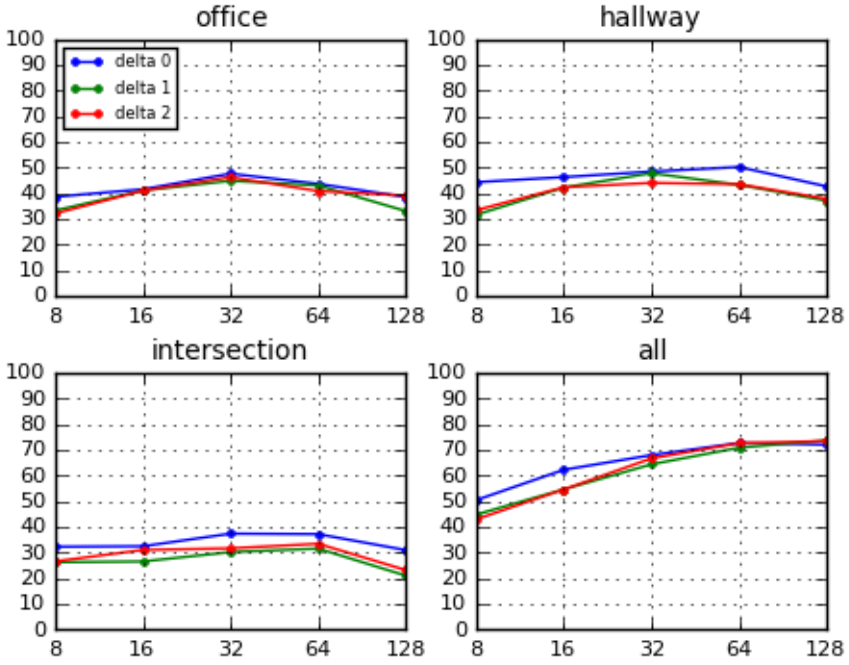

Figure A.1: Identification rates for enrolled speakers with $r = 0.95$.

Δ	M	Office	Hallway	Intersection	All
0	**8**	41.55	51.31	41.13	63.70
	16	47.42	56.13	45.10	71.64
	32	48.73	56.98	43.83	78.32
	64	49.61	55.52	43.21	80.83
	128	47.15	50.69	38.93	81.13
1	**8**	43.90	52.16	43.09	65.90
	16	49.31	58.68	47.22	76.85
	32	52.16	60.42	48.73	83.37
	64	53.94	60.03	48.77	86.03
	128	49.88	54.63	45.83	87.15
2	**8**	43.87	55.25	43.94	66.63
	16	49.65	60.61	48.11	77.97
	32	53.28	62.77	52.20	84.14
	64	53.40	61.11	51.93	88.31
	128	50.23	54.17	46.03	88.43

Table A.2: Identification rates for enrolled speakers with $r = 0.99$.

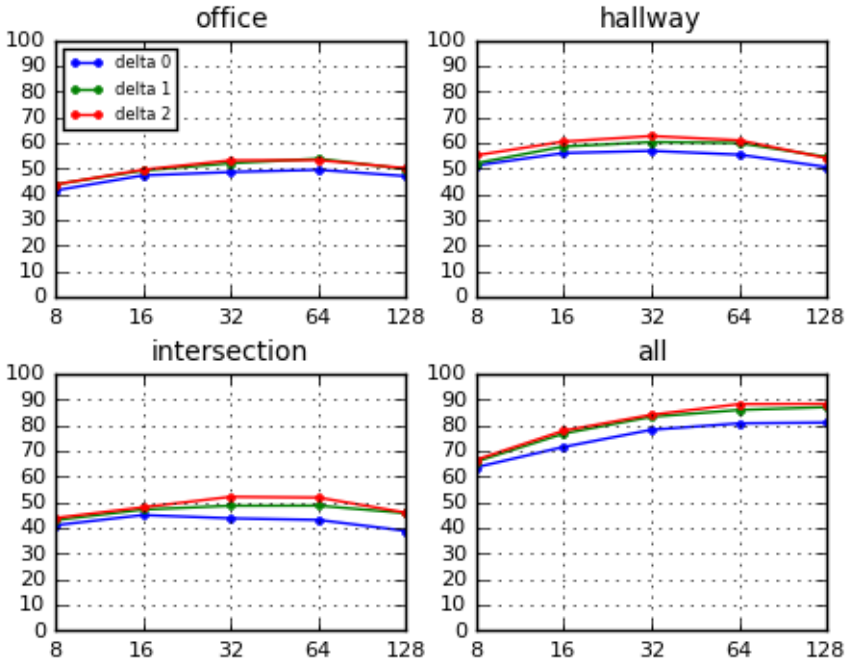

Figure A.2: Identification rates for enrolled speakers with $r = 0.99$.

Δ	M	Office	Hallway	Intersection	All
0	**8**	40.86	52.01	41.32	64.47
	16	47.69	56.52	44.79	72.22
	32	49.50	57.72	47.61	77.74
	64	50.00	57.95	45.68	81.25
	128	48.65	53.43	42.63	81.67
1	**8**	44.25	53.97	45.60	66.94
	16	50.42	62.00	50.54	78.24
	32	54.28	63.54	53.86	84.45
	64	55.09	64.81	52.85	87.31
	128	53.32	59.99	50.46	88.85
2	**8**	44.37	57.06	47.30	69.64
	16	50.89	62.81	52.12	78.78
	32	54.90	65.01	56.29	86.00
	64	56.06	64.70	56.56	89.16
	128	52.55	60.73	49.58	90.66

Table A.3: Identification rates for enrolled speakers with $r = 1.00$.

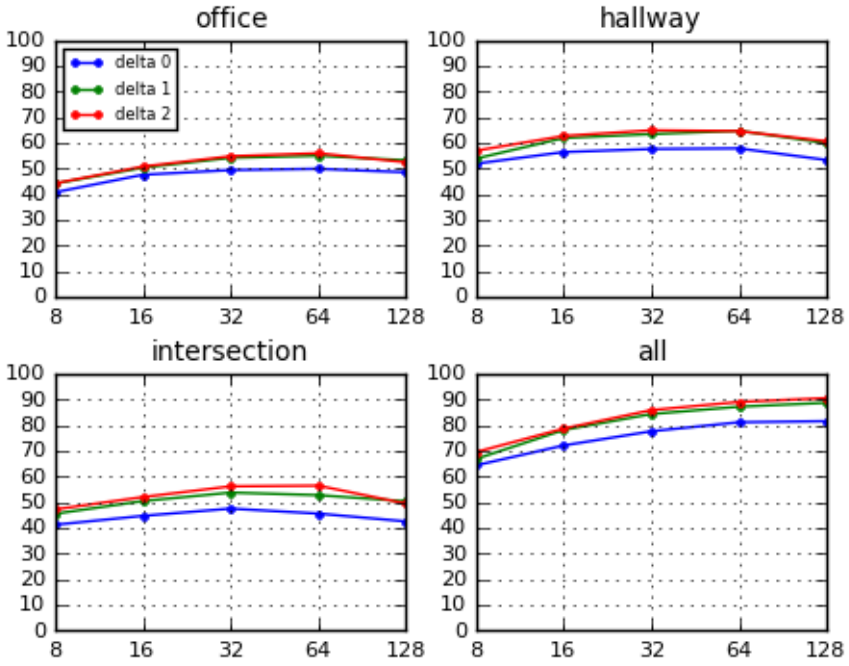

Figure A.3: Identification rates for enrolled speakers with $r = 1.00$.

Δ	M	Office	Hallway	Intersection	All
0	**8**	40.16	52.51	43.02	61.69
	16	46.88	57.10	47.80	71.84
	32	49.92	59.30	49.11	76.66
	64	50.19	58.95	48.92	79.94
	128	48.38	55.56	45.22	81.52
1	**8**	43.36	54.90	45.18	65.28
	16	49.58	61.07	53.74	76.74
	32	55.02	66.44	56.64	83.60
	64	56.02	66.28	56.25	88.00
	128	55.17	62.23	54.32	89.51
2	**8**	45.10	53.74	47.22	66.44
	16	50.81	64.31	53.59	78.05
	32	56.56	67.09	58.49	84.72
	64	56.10	66.90	58.33	89.74
	128	55.02	63.54	56.33	90.55

Table A.4: Identification rates for enrolled speakers with $r = 1.01$.

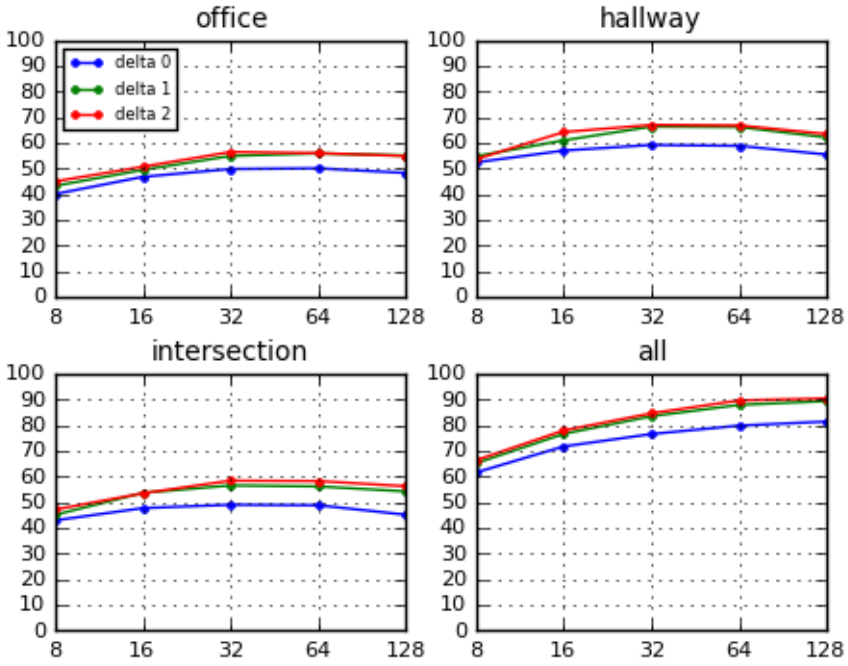

Figure A.4: Identification rates for enrolled speakers with $r = 1.01$.

Δ	M	Office	Hallway	Intersection	All
0	**8**	22.22	33.02	34.80	30.71
	16	32.52	41.32	42.67	42.32
	32	40.78	51.20	48.92	52.70
	64	46.68	56.56	53.51	62.19
	128	49.15	59.57	55.13	69.91
1	**8**	18.56	23.88	26.97	22.15
	16	28.20	39.00	39.78	32.87
	32	38.39	51.58	50.46	50.08
	64	49.11	63.46	58.33	62.96
	128	57.99	66.32	59.41	75.96
2	**8**	17.52	25.15	28.43	19.41
	16	28.94	38.27	42.44	34.30
	32	40.74	50.31	49.38	49.88
	64	50.42	61.23	58.87	63.77
	128	57.68	67.52	62.00	77.55

Table A.5: Identification rates for enrolled speakers with $r = 1.05$.

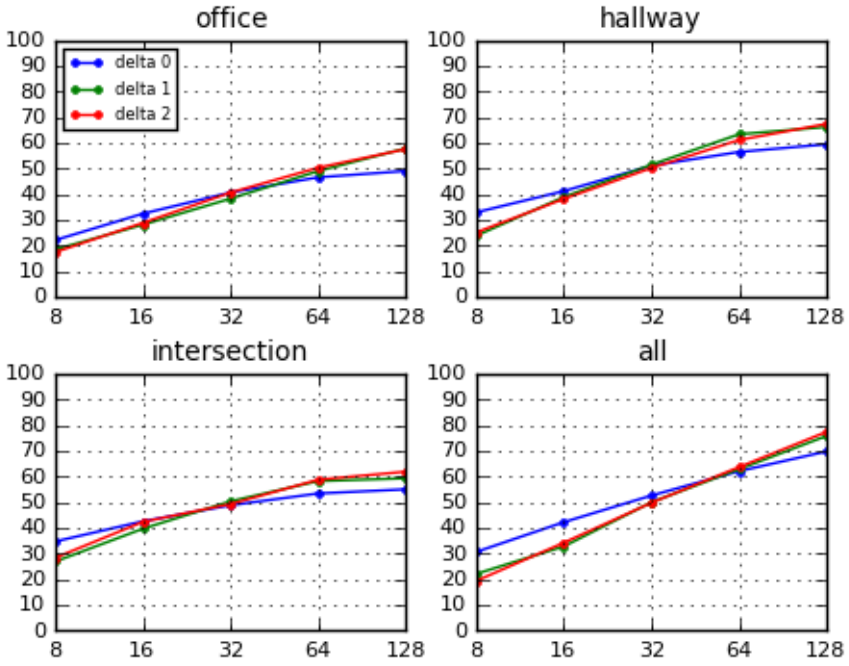

Figure A.5: Identification rates for enrolled speakers with $r = 1.05$.

B. Verification

B.1 Speakers

B.2 Adapted: m

B.3 Adapted: mv

B.4 Adapted: wm

B.5 Adapted: wmv

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] Chaobang Gao, Jiliu Zhou, and Qiang Pu. “Theory of fractional covariance matrix and its applications in PCA and 2D-PCA”. In: *Expert Systems with Applications* 40.13 (Oct. 2013), 5395–5401.

[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.