# Interactive Game for the Training of Portuguese Vowels

Mara Inês Pires Carvalho

Dissertation developed within the framework of the Mestrado Integrado em Engenharia Electrotécnica e de Computadores under the supervision of Prof. Dr. Aníbal Ferreira

(President of Jury, Prof. Dr. Francisco Oliveira Restivo)

Faculdade de Engenharia da Universidade do Porto
Departamento de Engenharia Electrotécnica e de Computadores
Rua Roberto Frias, s/n, 4200-465 Porto, Portugal

April 7, 2008

# Abstract

In this work an interactive application completely controlled by the utterance of 5 Portuguese vowels was developed. This application (a vowel display) is an important complement to speech therapy or language learning areas, as it allows to overcome many of the major drawbacks of the traditional approaches, namely the lack of motivation, the impossibility of continuing training after the sessions, and the fact that the feedback, that is provided by the speech/therapist in the traditional methods, is delayed, and hence not as effective to self-monitoring as a real-time feedback of the utterances is.

The traditional techniques for achieving a vowel display (formant plots) have many fragilities, related with the formants estimation, that is not reliable for high-pitched vowels (this is the case of the vowels uttered by children).

A basic pattern recognition process flow was followed, but a number of variants, including the use of different features, different mapping techniques and different classifiers was tested. Several hierarchical approaches were also tested, and yield good indications.

The final classifier scheme comprises the use of 16 MFCCs as a parametrization of the speech signal. Then, these parameters are mapped to a 4-dimensional space, using a linear technique (LDA or PCA). The pitch can additionally be added as a feature to this 4-dimensional set. The resulting feature set is then mapped using bayesian classifiers. Both linear and quadratic classifiers were used.

The classifier developed is hence adapted to use in real-time operations. Although lacking further verifications, the classifier performs relatively well for high-pitched vowels, according to tests made regarding a female speaker.

The output of the classifier is used as the controller for a simple car race game, that is intended to be the auxiliary visual display to language learning/speech therapy sessions.

# Resumo

Neste trabalho foi desenvolvida uma aplicação completamente controlada por 5 vogais do português. Este *display* de vogais é um complemento importante para áreas como terapia da fala e aprendizagem de linguagem, pois permite ultrapassar muitos dos principais problemas das abordagens tradicionais, nomeadamente a escassez de motivação, a impossibilidade de continuar o treino após as sessões, e o facto de o *feedback* dado pelo terapeuta ou professor nas abordagens tradicionais aparecer atrasado em relação à produção do som, pelo que não é tão eficaz como o *feedback* em tempo-real seria para aumentar a capacidade de auto-monitorização.

As técnicas tradicionais de obtenção de *displays* de vogais recorrem a gráficos de formantes. Estes apresentam muitas fragilidades, visto os processos de estimação de formantes não serem robustos, e apresentarem falhas para vogais produzidas com um elevado pitch (como é o caso das vogais produzidas por crianças).

Neste trabalho, utilizou-se a sequência tradicional de operações de processamento de padrões. No entanto, várias variantes desta abordagem geral são testadas, incluindo o uso de diferentes características, diferentes técnicas de mapeamento, e diferentes classificadores. Foram também testadas várias abordagens hierárquicas, que apresentaram boas indicações.

O classificador final compreende o uso de 16 MFCCs para caracterizar o sinal de fala. Seguidamente, estes parametros são mapeados para um espaço de 4 dimensões, usando uma técnica linear (LDA ou PCA). O pitch pode ser adicionado nesta fase como uma característica alternativa. O conjunto de características resultante é finalmente classificado por métodos *bayesianos* (lineares ou quadráticos).

De seguida, este classificador foi adaptado para permitir operação em tempo-real. As primeiras indicações mostram que este classificador funciona bem para vogais produzidas com pitch elevado.

A saída dos classificadores é utilizada para controlar um jogo de corridas de carros simples, que foi criado para ser um *display* visual auxiliar para as áreas de terapia da fala ou aprendizagem de linguagem.

# Preface

Language learning and Speech Therapy are areas that can obtain much benefit from the use of new technologies. The use of interactive displays, that provide real-time feedback regarding the utterance of vowels, is a valid alternative to reinforce the lacking acoustic feedback.

Although the benefits of using such applications are enormous, they have to fulfill many requirements in order to be a valid substitute for the human auditory system: real-time performance, together with the robustness of the classifier are mandatory. However, the existing solutions are not very robust, and tend to fail when used with high-pitched vowels. Therefore, with this work, the development of an interactive application, capable of correctly identify high-pitched vowels is intended.

The development of this work would not be possible without the enormous support provided by Prof. Aníbal Ferreira, together with the aid given by the Seegnal Research Team: Joaquim Matos, José Lopes and Filipe Abreu. At last, I thank the patience and encouragement provided to me by my parents and my sister Alexandra.

To all, I express my enormous gratitude.

# List of Acronyms

**ANN** Artificial Neural Network.

**CCA** Conformal Eigenmaps.

**CFA** Coordinated Factor Analysis.

**GDA** Generalized Discriminant Analysis.

**LDA** Linear Discriminant Analysis.

**LDC** Linear Discriminant Classifier.

**LLC** Locally Linear Coordination.

**LLE** Local Linear Embedding.

**LPC** Linear Predictive Coding.

**LPP** Linearity Preserving Projection.

**LTSA** Local Tangent Space Analysis.

**MDA** Multiple Discriminant Analysis.

**MDS** Multidimensional Scaling.

**MFCC** Mel-Frequency Cepstral Coefficient.

**MLP** Multi-layer Perceptron.

**MVU** Maximum Variance Unfolding.

**NNC** Nearest Neighbor Classifier.

**NPE** Neighborhood Preserving Embedding.

**PCA** Principal Component Analysis.

**PSC** Perceptual Spectral Cluster.

**QDC** Quadratic Discriminant Classifier.

**SNE** Stochastic Neighbor Embedding.

**SPE** Stochastic Proximity Embedding.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Overview

In this work, the problem of isolated vowel recognition is addressed. In the last few decades, major improvements were made in the speech recognition area. Although isolated vowel recognition appears to be a minor problem in this context, it is also true that it represents a big challenge: most of today's speech recognition systems rely on the use the contextual information to improve the system's performance. However, in the isolated vowel recognition environment, no contextual information exists. Therefore, the development of a system capable of identifying the distinctive features of the vowels is essential.

A vowel is a speech sound produced with an open configuration of the vocal tract, therefore offering no obstructions to the flow of air from the lungs. This definition arises from the main difference to the consonants, the alternative class of speech sounds. Vowels are one of the most interesting classes of sounds - in most languages speech is not possible without vowels. Nevertheless, their importance to classification and representation of written text is very low: there are only 5 vowels (in fact, if a written word is lacking the vowels, one can easily derive the word from the consonants - the same thing does not happen if there is a lack of consonants). For spoken speech, the opposite happens - most speech recognition systems rely on vowels to achieve high performance [4].

However, isolated vowel recognition systems remain inaccurate and are not robust. An explanation for this may be found in the fact that the identification of the distinctive features of vowels has not yet been done. In fact, one must not forget that the amount of variability that can be associated with the pronunciation of a "normal" vowel is enormous: different genders, possibly with different regional accents exist, so defining a standard vowel sound is impossible. The main challenge of this problem still lies in the achievement of the correct distinctive features for the vowel class. Standard classifiers for vowels sounds rely on the estimation of the first three formants using the short-time Fourier spectrum derived from the speech signals. Also, the traditional vowel displays are simple scatter plots, which use the first two formants as axes. However, these plots show great dispersion and overlap between the several classes [4]: although formants contain information regarding the distinction between classes, this information is not complete, neither the methodologies to estimate formants are reliable to use in a real-time framework. This is obviously incompatible with the idea of a robust automatic vowel recognizer. Latter, the use of spectral-shape features was proposed [5] as a valid alternative to the estimation of formants. However, none of these techniques approaches the performance of the human auditory system ("human performance in recognizing isolated vowels (...) is not even approximated by current state-of-the-art technology" [3]) - in fact, one is able to recognize without great effort an uttered vowel, regardless of the gender of the speaker, or its regional accent. Therefore, the final goal of a "perfect" classifier can be defined as the achievement of a human-like performance.

The use of visual displays to present the uttered vowel is a technique very useful for speech training purposes. Visual displays find their target application areas such as language learning and speech therapy - both groups of "patients" need aid to correct the deviation from the correct utterance, as it is extremely unlikely that any would be able to correct the respective problem simply by self-monitoring the utterances [6]. Usually, some kind of fragility on the usual feedback system, the auditory system, exists, thus an alternative feedback has to be provided. As stated in [3], computer-based visual feedback of speech-related features, namely vowels has the purpose of reinforce or replace the natural acoustic feedback pathway. This visual display must actively change with the different utterances: the necessity of a real-time display, that reflects even the small changes in pronunciation is immediately defined as a goal for language learning and speech therapy computer-based applications [7]. Together with the reinforcement of acoustic feedback, visual displays facilitate and fasten the convergence to the desired pronunciation - extra motivation is provided to the patients, by offering opportunities of self-training in complement of the training performed in speech therapy sessions.

The use of new technologies in the speech training areas represents a abrupt change relatively to the traditional techniques, in which all lacking feedback is provided by the teacher/therapist. These changes must be seen as an aid to the teacher's or therapist's work, and not as a replacement to them.

The aid provided by a computer-based visual feedback is indeed powerful, and may help to overcome some important drawbacks of the traditional training techniques, that are presented in [6]. In the following, the main drawbacks identified are presented:

1. **Insufficient motivational impact and reinforcement**

   The traditional teaching strategies are hardly dissociated from the idea that a treatment is being held. However, and specially when the patient is a child, it is extremely important to maintain a certain degree of motivation. When that does not happen, and the child is being "forced" to do the treatment, the level of concentration is limited, and the probability of success is certainly diminished. Therefore, it is difficult to achieve the degree of reinforcement desired.

2. **No immediate linking and response between the acoustics and the articulation**

   In the usual training methods, the natural feedback is replaced by the one given by the teacher. This introduces necessarily some delay between the production of the utterance and the self-perception of the degree of deviance from the desired production. A real-time feedback provides a patient with the ability to become self-aware of the quality of his/her utterances, and, consequently, to anticipate and correct a bad utterance.

3. **Practice does not continue after the speech training session**

   The traditional techniques usually require the presence of a teacher/therapist. Even when there are specific exercises that can be taught to a patient and performed on his own, they usually lack the capability of motivating the patient to continue training after a session.

4. **Too much time and patience are required from both the teacher and the students**

   The training does not bring immediate results. First it will gradually approximate the patient to the desired production, and latter will help the patient to maintain the desired production. That is specially true if no training is held at home between the sessions. Additionally, the sessions are usually expensive.

The use of a computer-based visual display gives an augmented form of feedback for the patient, as it provides a real-time feedback that can replace or increase the natural acoustic feedback, but also for the teacher/therapist, as it gains improved insight regarding the utterances held by the patient.

Regarding the visual displays, some essential requirements have to be met in order to the feedback provided by such elements can be helpful for both patient and teacher/therapist, as mentioned earlier. Many of these requirements naturally arise from the compensation of the drawbacks of the traditional methodologies, described above. As stated in [6], the main requirements of such a display are:

**Real-time operation** The visual feedback provided must be immediate, i.e., the patient must see the effect of small changes in an utterance in the visual display, so that he can be aware of the effect of changing the articulation. Also, if the feedback is immediate, it is more easily associated with the production made, and hence the patient becomes aware of how to correct deficient productions. Although utility can be drawn from delayed operation, for some specific purposes, the real-time requirement is essential in training application, such as the one that is intended to be developed in this work.

**The level and amount of detail extracted** There are displays that focus only in one feature, and displays that show a big amount of information, such as spectrograms. These high-detailed displays are very useful for professional analysis, however they can be overwhelming, confusing and incomprehensible for patients. Ideally, the display should provide the amount of detail necessary to reinforce the learning.

**Characteristics of visual representation** The chosen visual representation must be related with the speech attribute to be taught - the display should give intuitive representations. Child patients have limited concentration capabilities, so the display must be simple.

**Visual instructions** The game must be intuitive, but there also should exist visual indications of how to reach the intended goal, so that the patient can correct his errors and achieve a correct vowel utterance.

**Meaningful contrasting models** The feedback provided by different utterances must be different, so that the effect of each utterance is perfectly clear, and that no confusions are made.

**A metric to measure speech quality** The visual feedback must provide some kind of information relative to the quality of the current production (similar to the evaluative feedback a teacher/therapist can give, i.e. good, excellent, etc.). Therefore, some metrics to compute the quality of the utterance must be derived.

**Motivational impact and reinforcement of visual feedback** The visual display must be attractive enough to provide an interesting training framework to the patient, so that he/she is motivated to continue the training at home. Therefore, as more training is done, the results are also better.

**Flexibility and suitability** The visual display must be adaptable to the current needs of the patient, i.e., the training process should be increasingly difficult, as the patient improves its condition.

**Accurate and reliable** The visual display must give accurate feedback regarding the current utterance. Errors can result in a disincentive for the patient, hence resulting in the opposite of the effect intended. In worst case scenario, wrong feedback can even reinforce bad productions.

Ideally, these requirements must be met while developing a visual display for speech therapy or language learning purposes.

Different approaches to speech-recognition related problems, such as the isolated vowel recognition problem, can be considered. In [4], three approaches to the speech recognition problem are presented:

**Acoustic-phonetic approach** The basic idea behind this approach is that distinctive phonetic units exist in spoken language, that are characterized by properties existing in the speech signal or in its spectrum. It is assumed that the variability existing regarding both the amount of speakers and the effect of co-articulation of sounds depends on straightforward rules, and thus can be easily learned.

**Pattern recognition approach** In this approach, speech patterns are used to train a classifier. these patterns can be used directly without the explicit feature determination required by the previous technique. If the training set is representative of the general population, the classifier will be able to identify correctly new patterns.

**Artificial intelligence approach** This methods can be seen as a hybrid of the previous two approaches, as ideas and methods from both are exploited.

In this work the pattern recognition approach was followed, as it has many advantages, such as the simplicity of use, the robustness and the proven high performance.

## 1.2 Motivation

Children are naturally motivated to use interactive displays. Furthermore, the use of games controlled by the utterance of vowels provides means to facilitate the motivation needed to induce a child to follow a regular training program.

This motivation mainly comes from the fact that, in an interactive game framework, the child can actually see his/her progress - they attempt to change the articulatory patterns until the produced utterance allows the accomplishment of the game goals. However, traditional vowel classification techniques tend to fail for high pitched vowels - as is often the case of child speakers.

The occurrence of errors in classification can have a very negative impact in training, as child may be led to think that an incorrect utterance is in fact correct, and vice-versa. Therefore, the development of a new classifier, capable of correctly identify high-pitched vowels is necessary.

## 1.3 Objectives

The purpose of this work is thus to build an automatic vowel classifier, suited for be used by children, regarding 5 of the 8 oral European Portuguese vowels. The classifier should be able to work in real-time, and should further be integrated in a simple computer game. Signal processing techniques are used to obtain distinctive features from the speech utterances, and pattern recognition techniques will then be used to achieve the classification in vowels. The final application must give real-time visual feedback regarding the vowels uttered by the patient (that will be preferentially a child). Also, this visual feedback must be given in the form of an attractive and intuitive game, as the target audience for such a system are children. Throughout this thesis the main steps followed to accomplish these goals are presented.

## 1.4 Organization of the Thesis

In Chapter 2, the main signal processing concepts and techniques are presented. An overview of the basic properties of the speech signal is presented, with special care regarding voiced signals, the class to which vowels belong. Then, a summarized presentation of the concept of formants is made, as these are the traditional features used to characterize vowels. The concept of cepstrum is defined, with the basic description of how the concept arised, and the presentation of the formulas regarding both the real and the complex cepstrum. The clarification of this concept is necessary to complement the definition of the Mel-Frequency Cepstrum Coefficients, that are subsequently defined, and that constitute the main features used to characterize the speech signals in the present work. Another important concept is the pitch, as the perceptual equivalent of the fundamental frequency, presented in this chapter. Also, an overview of the basic concepts related with the Linear Prediction Coding are presented.

Chapter 3 presents a review of the main pattern classification concepts. The dimensionality reduction, feature extraction and pattern classification concepts are defined. A large amount of techniques regarding the feature extraction step are presented. For the pattern classification step, two main methods were used and are defined: bayesian classifiers and nearest-neighbor classifiers. The concept of Artificial Neural Networks is approached on a different section, because this is a very important and complex method, used in many of the applications regarding the use of vowel displays.

A summarized state-of-the-art review is presented in Chapter 4, with references to the main applications in this area.

In Chapter 5, the results of preliminary experiments regarding the vowel's spectral shape are presented. These experiments intend to give some insight regarding the connection between the spectral structure of these signals and the human recognition ability.

Chapter 6 provides an overview of the several simulations regarding the development of the most suited classifier for high-pitched vowel identification purpose. These simulations were done in Matlab®.

Details regarding the implementation in C++ of the selected classifier, and the development of the final application (a car race game) are presented in Chapter 7.

In Chapter 8, the main conclusions drawn from this work are presented together with suggestions regarding further work to enhance the application.

## 1.5 Original Contributions

This work addresses the problem of isolated vowel recognition regarding child speakers. Although some solutions for vowel training using speech displays exist, the development of the classifying and mapping techniques does not explicitly regard the problem of high pitched vowels (this problem is usually addressed by providing different classifiers for each gender). In the present work, an attempt to create a robust classifier is addressed, by using a database mainly composed of high-pitched vowels.

Furthermore, the relevance of the use of several classical pattern recognition techniques, including feature extraction, bayesian classifiers and nearest-neighbor classifiers is addressed, by applying these techniques to the vowels database. The resulting classifier was then adapted to real-time operation. Although some experiments regarding the use of these techniques to vowel classification have already been made, no concern with the possibility of adapting to real-time operation was taken. In this work, the use of these techniques is approached regarding this requirement, together with the necessity of finding a robust classifier.

The use of hierarchical approaches to vowel classification, both by separating the vowels accordingly to their spectral characteristics in different stages, and by using different types of features is tested, and has provided good indications.

# Chapter 2

# Fundamentals of Digital Speech Processing

## 2.1 Introduction

The first step in any speech recognition system is the signal processing. This step comprises the parametrization of the input speech signal. In this chapter, the main signal processing techniques for speech signal analysis are presented.

For vowel classification purposes, this parametrization of the input speech signal comprises the computation of speech-related features that are able to characterize the signal, regarding the distinction between vowels. There are many features that can be computed from an input speech signal: from the raw features like pitch or amplitude, to more complex features like MFCCs or LPCs. Therefore, in the remainder of this chapter, the techniques regarding the computation of parameters that are helpful for vowel identification are characterized.

Primarily, however, a brief overview of the speech signal is provided.

## 2.2 Speech

In human beings, the speech production/perception process involves several steps. First, the message is formulated, then it is converted into a language code. Finally, neuromuscular actions have to be taken in order to allow the production of the correct sounds, by making the vocal cords vibrate and adjusting the shape of the vocal tract. The acoustic wave is then formed, and travels around the acoustic channel. Hopefully, it will reach the listener: thus, the speech recognition process begins.

Regarding the speech recognition process, the acoustic signal is first processed along the basilar membrane in the inner-ear. This provides a spectrum analysis of the incoming signal. Next, a neural transduction occurs, converting this spectral signal “into activity signals on the auditory nerve, corresponding roughly to a feature extraction process” [4]. Next, a conversion to a language code is made, and the message comprehension is finally achieved. It is a very complex process, and not fully understandable by the present knowledge. Therefore, machines are still unable to replicate this behavior, despite the enormous efforts made in the last decades.

In the process of human speech production, the source signal is the glottal pulse. An air flow then arises, and accordingly to the state of the vocal cords, different classes of sounds are produced: if they are tense, this air flow makes them vibrate, thus producing the so-called voiced speech sounds. On the other hand, if the vocal cords are relaxed, and the air passes by a constriction in the vocal tract, it becomes turbulent, and unvoiced sounds are produced, or, alternatively, it builds up pressure behind a point of total closure, and when it is opened, this pressure is suddenly released, causing a plosive. The voiced sounds are particulary interesting, as they resemble periodic signals. Examples of voiced sounds are the vowels.

Although speech is a time-varying signal, when considering short amounts of time (between 5 and 100 ms), one can consider it as a stationary signal: that assumption is often made for simplicity reasons on the calculations.

## 2.3 Formants

The formants correspond to the resonance frequencies of the vocal tract, determined by how its cross-sectional area varies. These frequencies are related to those that allow the passage of most of the energy from the source to the output, and thus match peaks in the spectrum. Changing the position of the tongue, lips, jaw and velum influence the sound produced. Therefore, the formants can be seen as a reflex of the position of these articulators - i.e., the first two formants are primarily determined by the position of the tongue. Accordingly, vowels are mainly affected by the position of the tongue - therefore, it is usual to use the first two (or three) formants to identify a vowel.

## 2.4 Cepstrum

The cepstrum has its origin in the problem of deconvolving two or more signals - in fact, the cepstrum was first defined as a technique for finding the echo delay. This definition was motivated by the fact that the logarithm of the Fourier spectrum of a signal containing an echo has an additive periodic component that is dependent only on the echo size and delay [8].

The original definition regards the cepstrum, that will be further referred as the real cepstrum, for disambiguation purposes. Later, a generalization of the concept was achieved with the notion of the complex cepstrum. The two concepts will be described in the following subsections.

### 2.4.1 Real Cepstrum

The cepstrum can be defined as the inverse Fourier transform of the log magnitude spectrum of a signal (Equation 2.1).

$$c[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log |X(\omega)| e^{j\omega n} d\omega \tag{2.1}$$

As stated previously, the root of the definition of cepstrum lies in the attempt of deconvolving different components of a signal. Voiced speech ($s(n)$) is composed of the convolution of the excitation sequence $e(n)$ and the vocal system response $\theta(n)$, $s(n) = e(n) * \theta(n)$. Therefore, in the Fourier domain, $S(\omega) = E(\omega)\Theta(\omega)$.

If instead of a convolution, our voiced signal was obtained by simply adding the components, i.e. $s(n) = x_1(n)+x_2(n)$, a simple Fourier transform would allow a separation in frequency of both signals: $S(\omega) = X_1(\omega)+X_2(\omega)$. Therefore, a the separation of both components would be easily achieved simply by applying a filter, and then transform the resulting signal back to the time domain. That is not the case: the convolution does not allow such simplicity. However, if we define $C_s(\omega)$ as $C_s(\omega) = \log |S(\omega)|$, using the properties of the logarithm, it is obvious that $C_s(\omega) = \log |E(\omega)| + \log |\Theta(\omega)| = C_E(\omega) + C_\theta(\omega)$. We are now in the presence of a linear combination of both signals. Thus, the same techniques of filtering can be applied to separate the signals.

The creators of this concept also defined a new terminology to avoid confusions: quefrency, cepstrum and liftering, are the terms that prevailed.

### 2.4.2 Complex Cepstrum

The real cepstrum has the major flaw of discarding the phase information (it only uses the magnitude of the spectrum). Therefore, it is not allowed the inverse transformation to the original nonlinear domain. With a simple generalization of this concept, this problem is resolved, and the complex cepstrum of a signal is defined (Equation 2.2):

$$\hat{x}[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} \log\{S(\omega),\}e^{j\omega n}d\omega \tag{2.2}$$

where

$$\log\{X(e^{j\omega})\} = \log|X(e^{j\omega})| + jarg[X(e^{j\omega})]. \tag{2.3}$$

The cepstrum is the even part of the complex cepstrum.

Although the complex cepstrum has a series of interesting properties when compared to the real cepstrum, the latter is still the one widely used.

## 2.5 Pitch

Psychoacoustics is a branch of psychophysics that studies the human auditory perception. The studies made in this field have shown that there are differences between the physical frequency content of sound, and the human perception of these frequencies. Although the term pitch is usually used interchangeably with the term fundamental frequency, this term actually represents the perceived fundamental frequency of a sound, that may or not correspond to the actual fundamental frequency.

The fundamental frequency is related with the vibration of the vocal cords: when voiced sounds are produced, the glottis is opened and closed. This movement makes the vocal cords vibrate. The rate of this vibration is the fundamental frequency.

The pitch is one of the basic parameters that characterize a sound. There are “normal” ranges of pitches for the speech of each gender (man, female and child). Also, vowels have intrinsic pitches [9], and can be further classified according to it.

There are several proposed methods for detecting pitch, namely methods regarding time-domain waveform similarity and methods regarding frequency domain spectral similarity. Only a brief overview of two methods regarding each of these groups will be presented.

### 2.5.1 Autocorrelation Method

The autocorrelation of a signal $s(n)$ assumed stationary can be defined by the expression in Equation 2.4.

$$R(\tau) = \sum_{n=0}^{N-1} s(n)s(n+\tau) \tag{2.4}$$

It is straightforward to understand that, if the signal is periodic, then $s(n) = s(n+T)$, where $T$ is the period. Therefore, the autocorrelation of a periodic signal will have a maximum for $\tau = T$. As was already mentioned, voiced signals can be seen as quasi-periodic signals with period equal to the pitch. Therefore, the autocorrelation can be used to achieve estimations of the pitch period, by determining its maxima.

The main problem with the use of this technique is that the first formant frequency often is near or bellow the pitch frequency, and thus can interfere with this detection. Also, as the signal is simply quasi-periodic, the peaks can be less prominent, and its identification can be difficult.

### 2.5.2 Harmonic-peak detection

Frequency analysis methods can also be employed to determine the pitch frequency by noting that the distance between harmonics is the reciprocal of the pitch period. Hence, an obvious form of determining pitch is to detect the harmonic peaks and measure the pitch as the spacing between harmonics.

## 2.6 Linear Predictive Coding

Linear Predictive Coding (LPC) is one of the most important speech analysis methods. It is used for estimating basic speech parameters: pitch, formants, spectra, vocal tract area functions, and for representing speech for low bit rate transmission or storage. This method models short-term correlation between speech samples (the long term correlation are modeled by pitch prediction models).

The LPC analysis basically states that "a speech sample can be approximated as a linear combination of past speech samples" ([10]). In order to understand LPC analysis, one should be aware of the source filter model of speech production (see Figure 2.1). In this model, the time varying filter incorporates the influences of the vocal tract, radiation, and excitation. Its steady-state system function is presented in Equation 2.5, where $G$ represents the gain.

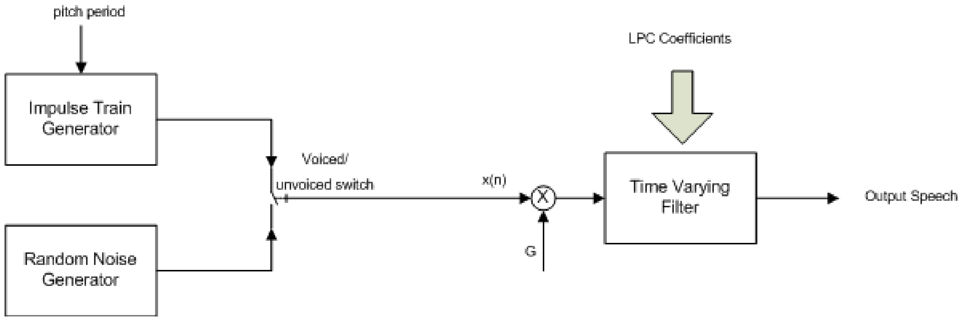

Figure 2.1: Source Filter model of Speech Production [1]

$$H(z) = \frac{S(z)}{X(z)} = \frac{G(1 - \sum_{j=1}^{M} b_j z^{-j})}{(1 - \sum_{i=1}^{N} a_i z^{-i})} \tag{2.5}$$

This equation can be simplified if we consider an all-pole system. In fact, this is only true for non-nasal voiced sounds. However, if the order of the denominator considered is high enough, the resulting model provides a fairly good representation for almost all sounds. In Equation 2.6, the simplified all-pole model system equation is presented:

$$H(z) = \frac{G}{(1 - \sum_{j=1}^{p} a_j z^{-j})} = \frac{G}{A(z)} \tag{2.6}$$

where

$$A(z) = (1 - \sum_{j=1}^{p} a_j z^{-j}). \tag{2.7}$$

The corresponding sampled-time domain formula, commonly referred as LPC difference equation (Equation 2.8) shows what was previously mentioned: the value of the present output, $s(n)$ may be determined by the weighted sum of the present input, $x(n)$ and the past output samples. The problem of the LPC analysis consists in determining the parameters $a_j, j = 1, \ldots p$, given the input signal $s(n)$.

$$s(n) = Gx(n) + \sum_{j=1}^{p} a_j s(n - j) \tag{2.8}$$

The derivation of the LPC Analysis Equation, together with the presentation of the most used methods to derive the solutions for the problem, are presented in Appendix A.

## 2.7 Mel-Frequency Cepstrum

As stated in the previous section, there are differences between the physical parameters (such as the Fundamental Frequency), and the perceived ones (such as the pitch). This perception does not follow a linear scale. Therefore, the "mel" scale was created, a perceptual scale of pitches, that was derived from measures obtained from listeners, who were asked to change the physical frequency, until the perceived pitch was doubled. The reference point is the pitch of a 1 kHz tone, defined arbitrarily as 1000 mels.

With the definition of a perceptual pitch scale, variants of the standard cesptrum appeared: mel-frequency cepstral coefficients, proposed by Davis and Mermelstein [11] are one of the most popular approaches. The design of this representation was motivated by perceptual factors: the desired characteristic of such a representation is the ability to capture the perceptually relevant information. The computation of the MFCCs uses the discrete cosine transform (DCT). In Equation 2.9, the form of computation of these coefficients as described in [11] is presented.

$$MFCC_i = \sum_{j=1}^{N} X_j \cos(i(j - \frac{1}{2})\frac{\pi}{N}), \qquad i = 1, 2, ..., M \tag{2.9}$$

In 2.9, $M$ is the number of cepstrum coefficients, $X_k$, $k = 1, \ldots, N$ represents the log-energy output of the $k$th filter (Fig. 2.2).

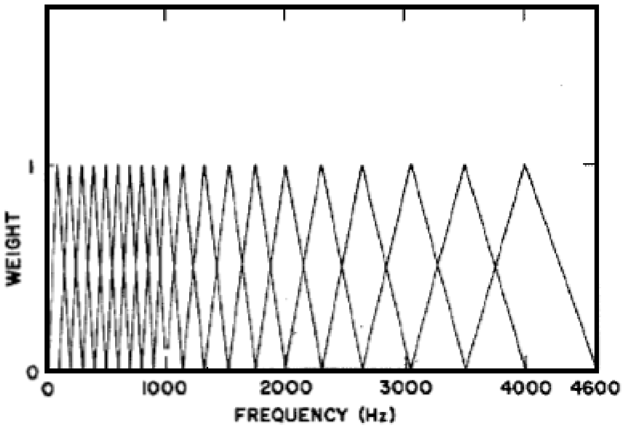

Figure 2.2: Filters for generating MFCCs [1]

The MFCCs are hence a parametric representation of acoustic data based on the Fourier spectrum, that preserve information that LP (Linear Prediction) features (see 2.6) omit. Furthermore, the MFCC has proven advantage over the linear frequency cepstral coefficients [11], and allow a better suppression of perceptually less important spectral variation on higher frequency bands.

## 2.8 Conclusions

In the present chapter, a general overview of the signal processing techniques regarding speech analysis was presented.

The human auditory system performs an analysis of the speech signal regarding recognition that is still a mystery. Therefore, the replication of this process by machines is not yet achievable: no complete knowledge regarding which speech features are used by the human auditory system to perform vowel recognition exists.

In this chapter, the main techniques regarding the computation of the principal parameters for speech recognition are presented. Linear prediction coefficients appear as the most traditional technique to parameterize a speech signal. MFCCs arise as the most used state-of-the-art speech features. Pitch is a raw feature of speech but with proven impact on vowel recognition. These parameters will be used further in this work, and the corresponding ability to characterize vowels will be studied.

# Chapter 3

# Pattern Recognition Techniques

## 3.1 Introduction

Pattern Recognition can be understood as “the act of taking in raw data and making an action based in the “category” of the pattern” [12]. Such techniques are employed by humans, with no apparent effort. Transposing such capability to machines is of extreme importance - areas such as isolated vowel recognition (the aim of the present work) are good candidates to be solved with these techniques. What is being said is that the isolated vowel recognition problem can be seen as a problem of simply classifying a certain pattern, derived from a speech frame. This pattern is obtained using the signal processing techniques referred in the previous chapter.

A pattern is, in this context, a quantitative description of an object (in this work, the object is a vowel utterance). Hence, the pattern is a vector, containing several components, that are features extracted from the input speech signal, somehow chosen to reflect the distinctive characteristics of the several sounds.

Each object is associated to a class (i.e., vowel), and objects belonging to the same class should have similar patterns. The classifier goal is to assign a class to a given object, accordingly to the pattern associated to this object. This is done by first creating a series of discriminant functions, that will give a numeric result for each pattern fed to them. The classifier assigns an object to a class if the value obtained by the respective discriminant function is the highest of the values obtained by all the discriminant functions (see Equation 3.1).

$$\mathbf{x} \in Class_i \qquad \Rightarrow g_i(\mathbf{x}) = \max_{forallj \in K} g_j(\mathbf{x}) \tag{3.1}$$

In 3.1, $x$ is the pattern (i.e. a vector of features) one wishes to classify, $g_i$ is a given discriminant function, and $K$ is the set of classes.

The derivation of the discrimination functions is usually made using training samples, i.e., labeled patterns, that are supposed to provide a good characterization of the generality of the data. However, care has to be taken to keep the classifier from overfitting to the training samples, i.e., the discrimination functions must also work well for similar patterns that are not in the training set - the classifier must have a good generalization capability.

Automatic pattern recognition systems comprises several steps. The first step usually regards the simple extraction of important parameters from the input. In our system, signal processing techniques, described in the previous chapter, were used to obtain parameters that provide good representations of the important and distinctive features of the speech signal. However, in this step the parameters obtained often comprise more information than the distinctive one: redundant or not distinctive information, that serves mainly to difficult the classifier task. Therefore, an intermediate step between parameter extraction and classification exists, often referred as feature extraction or dimensionality reduction[1]. With this step, one intends to achieve representations of the input that keep only the distinctive features to the classification task. The basic flowchart of such a process is presented in Fig. 3.1.

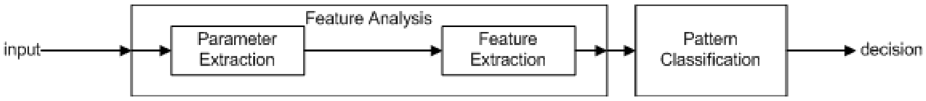

Figure 3.1: A simple pattern recognition flowchart

The idea of seeking among the several parameters extracted from the input the distinctive features for a certain classification task has the purpose of easing the following classification task. As defended in [12], there is no rigid boundary between feature extraction and classification, as the ideal feature extraction should be able to construct features from the input parameters

[1] The notion of dimensionality reduction is more wide than the feature extraction, as it comprises also the feature selection techniques. However, the first group (feature extraction techniques) has the advantage of allowing a combination of features in order to obtain new and more distinctive features, while the feature selection techniques simply choose among the several parameters the ones that provide better discrimination. Therefore, the first have obvious advantages over the second, and in this work only feature extraction techniques will be used for dimensionality reduction. Therefore, the two terms will be used interchangeably.

that would make the classification task extremely easy. Ideally, a perfect classifier would also be able to distinguish between the several features fed to him and perform a good classification despite of the quality of the feature extractor. However, as is often the case, the reality differs from the ideal concept. Therefore, several approaches to these problems, each one having advantages and flaws, arise. In the following sections, the main techniques to feature extraction and pattern classification will be described.

## 3.2 Feature Extraction and Dimensionality Reduction

Dimensionality Reduction is composed of both the feature extraction techniques, and the feature selection techniques. As mentioned early, only the feature extraction techniques will be addressed in the present work. The purpose of the feature extraction step is to derive, from the several parameters extracted from the input, a set of features, such that these features have similar values when the corresponding objects belong to the same class, and distinct values when that is not the case. Several approaches to this problem exist, including linear and nonlinear methods. Linear methods, including Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), assume that the data lie in a near linear subspace of a high-dimensional space, and have the advantage of allowing a simpler implementation. As for the nonlinear techniques, they do not rely on the linearity assumption, and can be grouped in three sets [2]: global nonlinear methods, local nonlinear methods and global alignment of linear models.

### 3.2.1 Linear Techniques

Linear Feature Extraction Techniques provide new features by making linear combinations of the several parameters. These techniques are very easy to implement, and thus are very attractive. The traditional linear feature extraction techniques are the Principal Component Analysis (PCA) and the Linear Discriminant Analysis (LDA) or Multiple Discriminant Analysis (MDA). Both will be presented following.

#### Principal Component Analysis

The Principal Component Analysis technique finds a lower dimensional representation of the data by maximizing the resulting variance of the data.

The main goal of the technique is hence to find representations in a lower-dimensional space of the data. Considering a multi-dimensional set of samples, a good zero-dimensional representation of this set is its mean value. However, the mean value gives no information about the dispersion of the samples. If instead of a point the data is mapped to a line, the representation can be more interesting. By minimizing the squared-error function between the samples and their projection along the line, a direction will be found as the one where the data dispersion is higher. These directions are hence the ones were most of the information is represented - the PCA technique uses these directions (the Principal Components) to achieve the new representation of the data. A graphical representation of what was said is presented in Fig. 3.2[2], where the original 3-dimensional data can be seen together with the lower dimensional representation.

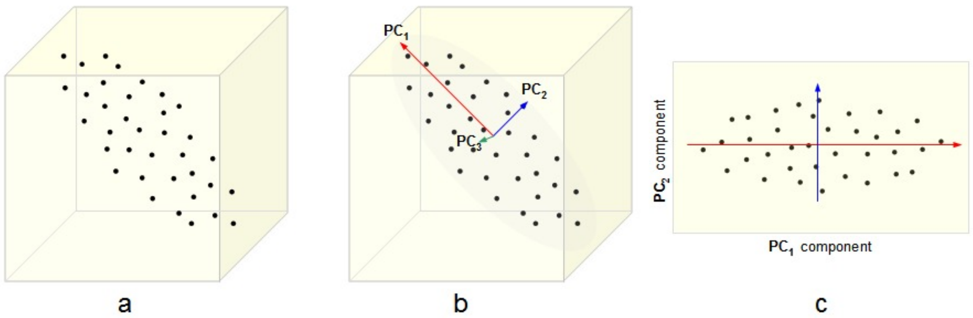

Figure 3.2: The PCA technique: A.The 3-dimensional original representation B.The representation of the directions of the 3 principal components C.The 2-dimensional representation, using the 2 principal components

Mathematically, the problem can be addressed by considering a set of $n$ d-dimensional samples, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, and a one-dimensional representation of the data as a line running through the sample mean $m = \frac{1}{n}\sum_{k=1}^{n} \mathbf{x}_k$, $\mathbf{x} = \mathbf{m} + a\mathbf{e}$. The goal is hence to find the direction $e$ that allows a minimization of the squared-error between the original samples, and the data projected onto this line. The solution for this problem involves the scatter matrix $S$, defined in Equation 3.2. This matrix is simply $n-1$ times the sample covariance matrix.

$$S = \sum_{k=1}^{n} (\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^t \tag{3.2}$$

[2] This image was obtained from http://cnx.org/content/m11461/latest/

In Equation 3.2, $\mathbf{x}_k$ represents each of the samples, and $\mathbf{m}$ the mean of all the samples.

It can be proven [12] that the direction where the dispersion is higher (and hence the squared-error is minimized) is the one given by the eigenvector corresponding to the higher eigenvalue of the scatter matrix. Generalizing this result, a mapping in a subspace with $k$ dimensions is achieved by making a linear transformation using the $k$ eigenvectors matching the $k$ higher eigenvalues of the scatter matrix, which are hence called principal components.

### Linear Discriminant Analysis

LDA, or Fisher's Linear Discriminant, was primarily derived for problems with two classes. When the number of classes is more than two, some authors also address to this natural generalization of Fisher's linear discriminant by Multiple Discriminant Analysis (MDA). In this work, for simplicity reasons, the term Linear Discriminant Analysis will be used to refer to the general method.

In contrast with the previously described technique, LDA is a supervised technique (the mapping is done considering the class labels of the training data), that has the goal of maximizing the linear separability between classes, maximizing the Fisher criteria: the ratio between the variance between classes to the variance within the classes.

The former mapping (PCA) gives rise to components that are useful for compact data representation, but has no concerns in whether these components are suitable for discriminating the different classes. This problem is addressed in Fig. 3.3[3], where it can be seen that the mapping achieved by the PCA techniques does not provide the intended separation between the classes: in fact, the technique induces the mixture of the samples.

[3] This image was obtained from http://www.cc.gatech.edu/ dminn/mini-proj.html

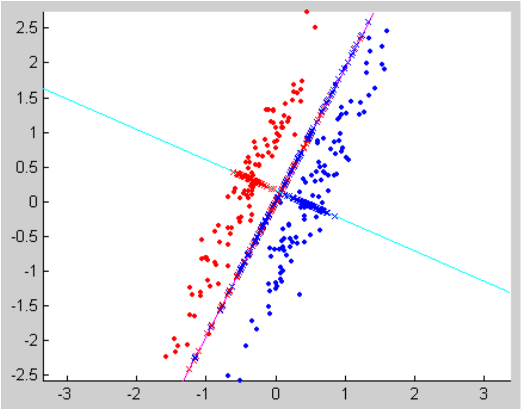

Figure 3.3: The LDA technique: The purple line is the principal component derived by the PCA technique, the cyan line is the vector found by LDA

The LDA approach tries to project the data in directions such that the different classes are well-separated. We will first address this problem considering only two classes.

Consider a set of $n$ d-dimensional samples, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, where $n_1$ samples belong to class $\omega_1$ and $n_2$ samples belong to class $\omega_2$. A linear combination of the components of a sample $\mathbf{x}$ is obtained by the scalar dot product $y = \mathbf{w}^t\mathbf{x}$. Therefore, each of the new samples $y_1, \ldots, y_n$ is also divided in two subsets, accordingly to the class they belong. Geometrically, these new samples, $y_i$, are simply the projection of the initial samples $\mathbf{x}_i$ onto a line in the direction of $\mathbf{w}$. Therefore, the problem can be defined as finding the best direction $\mathbf{w}$ that allows a bigger separation between classes, hence simplifying the classification process.

A measure of separation between classes can be given by the distance between the projected means of each class. In the original high-dimensional space, the mean is given by Equation 3.3, where $\omega_i$ represents the current class, $n_i$ the corresponding number of samples, and $\mathbf{x}$ the samples. In the projected space, the mean is given by Equation 3.4, and is simply the projection of the mean regarding the high-dimensional samples. In 3.4, $\omega_i$ represents the current class, $n_i$ the corresponding number of samples, $y$ the samples in the low-dimensional space and $\mathbf{w}$ the transformation matrix.

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in \omega_i} \mathbf{x} \tag{3.3}$$

$$\boldsymbol{m}_i = \frac{1}{n_i} \sum_{y \in \omega_i} y = \frac{1}{n_i} \sum_{\mathbf{x} \in \omega_i} \mathbf{w}^t\mathbf{x} = \mathbf{w}^t\mathbf{m}_i \tag{3.4}$$

Therefore, one can obtain a measure of separation between classes in the projected space by calculating $|\boldsymbol{m}_1 - \boldsymbol{m}_2| = |\mathbf{w}^t(\mathbf{m}_1 - \mathbf{m}_2)|$. Hence, if we define the scatter for the projected samples as presented in Equation 3.5, the criterion function presented in 3.6 can be defined. This function is maximized when the projection $\mathbf{w}$ that provides a better separation between classes is found.

$$s_i^2 = \sum_{y \in \omega_i} (y - m_i)^2 \tag{3.5}$$

$$J_{\mathbf{W}} = \frac{|\boldsymbol{m}_1 - \boldsymbol{m}_2|}{s_1^2 + s_2^2} \tag{3.6}$$

The Equation 3.6 can be rewritten as an explicit function of $\mathbf{w}$, by defining the scatter matrices $\mathbf{S}_i$, the between-class scatter $\mathbf{S}_B$ and the within-class scatter $(\mathbf{S}_W)$.

The scatter matrices represent the scatter of the data in the original high-dimensional space (Equation 3.7). The within-class scatter is defined in 3.8.

$$\mathbf{S}_i = \sum_{\mathbf{X} \in \omega_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t \tag{3.7}$$

$$\mathbf{S}_w = \mathbf{S}_1 + \mathbf{S}_2 \tag{3.8}$$

The projected-space scatter matrix can hence be rewritten as shown in Equation 3.9, and the separation between means as Equation 3.10:

$$s_i^2 = \sum_{\mathbf{X} \in \omega_i} (\mathbf{w}^t\mathbf{x} - \mathbf{w}^t\mathbf{m}_i)^2 = \sum_{\mathbf{X} \in \omega_i} \mathbf{w}^t(\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t\mathbf{w} = \mathbf{w}^t\mathbf{S}_i\mathbf{w} \tag{3.9}$$

$$(\boldsymbol{m}_1 - \boldsymbol{m}_2)^2 = (\mathbf{w}^t\mathbf{m}_1 - \mathbf{w}^t\mathbf{m}_2)^2 = \mathbf{w}^t(\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^t\mathbf{w} = \mathbf{w}^t\mathbf{S}_B\mathbf{w} \tag{3.10}$$

where

$$\mathbf{S}_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^t. \tag{3.11}$$

Hence, the criterium function can be rewritten as in Equation 3.12. This criterium function is referred to as Fisher's criterion. This criterium basically states that the between-class scatter $(\mathbf{S}_B)$ should be maximized as the within-class scatter $(\mathbf{S}_w)$ is minimized. In other words, one should increase the difference between the projected means of each class, and minimize the data separability within each class.

$$\mathbf{J}_w = \frac{\mathbf{T}^t\mathbf{S}_B\mathbf{T}}{\mathbf{T}^t\mathbf{S}_w\mathbf{T}} \tag{3.12}$$

In this equation, $\mathbf{T}$ is the transformation matrix.

This criterium function is a generalized Rayleigh quotient. The vector $\mathbf{w}$ that maximizes this quotient must satisfy $\mathbf{S}_B\mathbf{w} = \lambda\mathbf{S}_w\mathbf{w}$, that is easily identified as a generalized eigenvalue problem. Therefore, one should solve the eigenvalue problem $\mathbf{S}_w^{-1}\mathbf{S}_B\mathbf{w} = \lambda\mathbf{w}$ by finding the eigenvalues and eigenvectors of $\mathbf{S}_w^{-1}\mathbf{S}_B$. That is not necessary, because a solution can be easily found by noting that $\mathbf{S}_B\mathbf{w}$ is always in the direction of $(\mathbf{m}_1 - \mathbf{m}_2)$, and that the scale factor for $\mathbf{w}$ is not important [12]. Therefore, a solution that optimizes the criterium function is $\mathbf{w} = \mathbf{S}_w^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$.

When there are $c$ classes, the problem can be easily generalized by defining the scatter matrices for each of the c classes. Therefore, the within-class scatter is given by Equation 3.13.

$$\mathbf{S}_w = \sum_{i=1}^{c} \mathbf{S}_i \tag{3.13}$$

As for the between-class scatter, one must find the generalization by noting that a total scatter matrix is such that $\mathbf{S}_T = \mathbf{S}_w + \mathbf{S}_B$

$$\mathbf{S}_T = \sum_{\mathbf{X}} (\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^t \tag{3.14}$$

where $\mathbf{m}$ is the total mean. This equation can be rearranged as

$$\mathbf{S}_T = \sum_{i=1}^{c} \sum_{\mathbf{X}\in\omega_i} (\mathbf{x} - \mathbf{m}_i + \mathbf{m}_i - -\mathbf{m})(\mathbf{x} - \mathbf{m}_i + \mathbf{m}_i - -\mathbf{m})^t \tag{3.15}$$

$$\mathbf{S}_T = \sum_{i=1}^{c} \sum_{\mathbf{X}\in\omega_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t + \sum_{i=1}^{c} \sum_{\mathbf{X}\in\omega_i} (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^t \tag{3.16}$$

$$\mathbf{S}_T = \mathbf{S}_w + \sum_{i=1}^{c} n_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^t. \tag{3.17}$$

Therefore, the generalized between scatter matrix is obtained in Equation 3.18;

$$\mathbf{S}_B = \sum_{i=1}^{c} n_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^t. \tag{3.18}$$

### 3.2.2 Global nonlinear techniques

The Global nonlinear techniques, similarly to the linear techniques, attempt to preserve the global properties of the data, while constructing nonlinear transformations between the high dimensional data $X$ and the low dimensional data $Y$. Therefore, global approaches attempt to preserve the geometry at all scales, mapping nearby points on the high-dimensional space to nearby points in low-dimensional space, and faraway points to faraway points [13]. Therefore, these techniques tend to provide a faithful representation of the data's global structure.

### Multidimensional Scaling (MDS)

This method comprises a set of techniques that attempt to determine a mapping between a high dimensional space and a low dimensional space, trying to preserve the pairwise distance between points. The quality of the mapping is expressed using a stress function, that is simply a quantification of the error between the pairwise distance in the high dimensional space and in the low dimensional space. Examples of stress functions are the raw stress function (Equation 3.19) and the Sammon cost function (Equation 3.20).

$$\phi(Y) = \sum_{i,j}(\|x_i - x_j\| - \|y_i - y_j\|)^2 \tag{3.19}$$

$$\phi(Y) = \frac{1}{\sum_{i,j}(\|x_i - x_j\|)} \sum_{i,j} \frac{(\|x_i - x_j\| - \|y_i - y_j\|)^2}{\|x_i - x_j\|} \tag{3.20}$$

In Equations 3.19 and 3.20, $\|x_i - x_j\|$ is the euclidian distance between two points of the high dimensional space, and $\|y_i - y_j\|$ is the euclidian distance between two points in the low dimensional space. The Sammon cost function represents an attempt to maintain the distances that originally were smaller.

Minimizing the stress functions can be made using various methods, like the conjugate gradient method.

### Stochastic Proximity Embedding (SPE)

This is an iterative algorithm that attempts to minimize the raw stress function previously presented. The initial positions in the low-dimensional space $y_i$ are randomly selected in $[0, 1]$. In each iteration, these coordinates are updated, by randomly selecting $s$ pairs of points $(y_i, y_j))$ and computing the low-dimensional Euclidean distance for the selected pairs of points. Then, the coordinates of $y_i$ and $y_j$ are updated in order to decrease the difference between the distance in the low dimensional space $d_{ij}$ and the corresponding distance in the high-dimensional space $r_{ij}$:

$$y_i = y_i + \lambda \frac{r_{ij} - d_{ij}}{2d_{ij} + \epsilon}(y_i - y_j), \tag{3.21}$$

$$y_j = y_j + \lambda \frac{r_{ij} - d_{ij}}{2d_{ij} + \epsilon}(y_j - y_i). \tag{3.22}$$

where $\lambda$ is a learning parameter that decreases with the number of iterations, and $\epsilon$ is a regularization parameter to prevent divisions by zero. This procedure is repeated for a large number of iterations (i.e. $10^5$ iterations). The enormous amount of iterations is not is not a major disadvantage as the computational cost of the update procedure is very small.

### Isomap

The MDS set of techniques has the fragility of using euclidian measures. That means that, if the data points are disposed along a curve surface, these techniques could consider two points to be close, although the distance along the curve surface could be much bigger. The Isomap technique attempts to maintain the pairwise curvilinear or geodesic distances. These distances are computed by first constructing a neighborhood graph $G$ where each datapoint $x_i$ is connected to its $k$ nearest neighbors $x_{i_j}$. Then, the shortest path between two points in the graph can be easily computed using Dijkstra's shortest-path algorithm, and provides a good overestimate of the geodesic distance. When the geodesic distances between all datapoints are computed, a pairwise distance matrix is obtained. The low dimensional representations $y_i$ is finally computed by applying MDS algorithm to this matrix.

### Fast Maximum Variance Unfolding (FastMVU)

This technique is similar to the Isomap technique: it defines a neighborhood of each point and stores the point-to-point distances. The main difference is that this technique explicitly tries to unfold the data surfaces, by maximizing the euclidian distances between points, with the restriction of keeping the distances in the neighborhood (not distorting the geometry of the data surface). The first step of this technique is the definition of a neighborhood graph $G$, where each point $x_i$ is connected to its $k$ closest neighbors. Next, we try to maximize the euclidian distances between every point, keeping the distances in the neighborhood graph $G$:

$$max \sum_{ij} \|y_i - yj\|^2 \qquad \text{restricted to } \|y_i - yj\|^2 = \|x_i - xj\|^2 \qquad \forall (i,j) \in G. \tag{3.23}$$

### Kernel PCA

This technique is a reformulation of the linear PCA technique on a higher dimensional space constructed using a kernel function. The first step is hence to transpose the data into a higher dimensional space (using, i.e., Support Vector Machines). The entries of the kernel matrix $K$ are defined by $k_{ij} = \kappa(x_i, x_j)$, where $\kappa$ is a kernel function. Next, the matrix is centered, using the modified entries presented in Equation 3.24.

$$k_{ij} = k_{ij} - \frac{1}{n}\sum_{l} k_{il} - \frac{1}{n}\sum_{l} k_{jl} + \frac{1}{n^2}\sum_{lm} k_{lm} \tag{3.24}$$

Finally, the $d$ principal components (eigenvectors) $v_i$ of the kernel matrix are calculated.

The performance of this method is highly dependent on the choice of the kernel function.

### Generalized Discriminant Analysis (GDA) or Kernel LDA

Like the previous technique, this is a reformulation of the LDA technique in a high dimensional space using a kernel function. This technique tries to maximize the Fisher's criterium in the high dimensional space, constructed using a kernel function. The main purpose of this technique is to maximize the separability of the data in the high dimensional space by using nonlinear mappings.

### Diffusion Maps

These maps are based in the definition of a random Markov walk in the data graph. By performing the random walk for a certain number of timesteps, a measure of proximity between datapoints is obtained. Using this measure one can obtain the diffusion distance. In the low dimensional space, the pairwise diffusion distances are retained as well as possible.

In this method, the first step is to construct a graph of the data, which edges have weights computed using the Gaussian kernel function. A matrix $W$ is computed, with entries:

$$w_{ij} = \exp^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}, \tag{3.25}$$

where $\sigma$ is the variance of the Gaussian. Next, a normalization of the matrix $W$ is performed, so that its rows add up to 1. Therefore, a matrix $P^{(1)}$ is formed, with entries:

$$p_{ij}^{(1)} = \frac{w_{ij}}{sum_k w_{ik}}. \tag{3.26}$$

This is considered a Markov matrix that defines the forward probability of a transition between datapoints in a single timestep. The forward probability matrix regarding $t$ timesteps $P^{(t)}$ is given by $P^{(t)} = P^{(1^t)}$. The diffusion distance matrix can hence be defined as

$$D^{(t)}(x_i, x_j) = \sum_k \frac{(p_{ik}^{(t)} - p_{jk}^{(t)})^2}{\psi^{(0)}(x_k)} \tag{3.27}$$

where $\psi^{(0)}(x_i) = \frac{m_i}{sum_j m_j}$, $m_i = \sum_j p_{ji}$ is a term whose purpose is to give more weight to parts of the graph with bigger density. Pairs of datapoints with high forward transition probability have small diffusion distance. This measure of distance is more robust to noise, as it is based on many paths through the graph. In the low-dimensional representation $Y$, one tries to maintain these diffusion distances. This representation is formed by the $d$ principal eigenvectors of $P^{(t)}Y = \lambda Y$.

### Stochastic Neighbor Embedding (SNE)

This is an iterative technique that tries to maintain the pairwise distances in the low dimensional space. The main differences between this technique and MDS are the distance measure used and the cost function it minimizes.

The probability $p_{ij}$ that datapoints $x_i$ and $x_j$ are generated by the same gaussian is computed for all combinations of datapoints possible, and stored in a matrix $P$. These probabilities are calculated using the Gaussian kernel function. Next, the low-dimensional coordinates $y_i$ are set to random values close to zero, and the probabilities $q_{ij}$ that the datapoints $y_i$ and $y_j$ are generated by the same Gaussian are computed for all possible combinations and stored in matrix $Q$. A perfect low-dimensional representation would be achieved if P=Q. SNE then attempts to minimize the sum of Kullback-Leibler divergences (a natural distance measure between two probability distributions):

$$\phi(Y) = \sum_{ij} p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{3.28}$$

This minimization can be achieved by using, i.e., the gradient descent method. To avoid problems with local minima, a decreasing amount of Gaussian jitter is added in every iteration.

### Multilayer Autoencoders

These are feedforward neural networks, with an odd number of hidden layers. The mid hidden layer has $d$ nodes, where $d$ is the target dimensionality. Both the input and the output layer have $D$ nodes, where $D$ is the original dimensionality. The network is trained to minimize the mean squared error between the input and the output (ideally, they would be equal). Hence, the d-dimensional representation is obtained by extracting the node values in the middle layer. This gives a representation that preserves as much information of the original representation as possible. The schematic structure of an autoencoder is presented in Fig. 3.4.

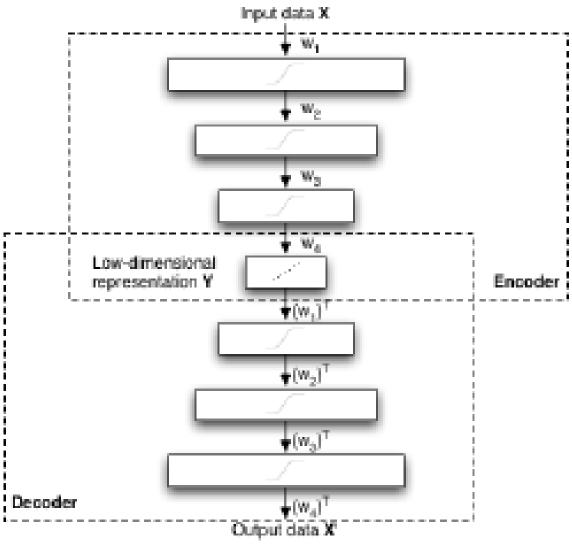

Figure 3.4: Schematic structure of an autoencoder [2]

This approach has the main drawback of tending to be stuck in local minima. There are some techniques to overcome this difficulty, namely the use of Restricted Boltzmann Machines (RBMs) for pre-training the network.

## 3.2.3 Local nonlinear techniques

These techniques, contrarily to the previously presented ones, do not attempt to maintain the global properties of the data. They actually attempt to preserve the properties of small neighborhoods of each point - the main idea is the preservation of the local properties of the data, seeking to map nearby points on the high-dimensional space to nearby points in the low-dimensional representation. Although the main idea behind these approaches may appear strange, as no care is being taken to preserve the global structure, they have important advantages, such as the computational efficiency and the representational capacity [13].

### Local Linear Embedding (LLE)

This technique is somehow similar to the Isomap technique, as the construction of a neighborhood graph is also the first step of the LLE technique. However, being a local technique, here the attempt is to preserve the local properties. This is done by writing the datapoints $x_i$ as a linear combination $W_i$ of its $k$ nearest neighbors $x_{i_j}$. When computing the low-dimensional representation, this technique attempts to retain $W_i$ - the local linearity assumption implies that these values can also be obtained when combining the low-dimensional datapoints $y_i$. Mathematically, this technique corresponds to minimizing the cost function presented in Equation 3.29.

$$\phi(Y) = \sum_i (y_i - \sum_{j=1}^{k} w_{ij} y_{i_j})^2 \tag{3.29}$$

### Laplacian Eigenmaps

In this technique, the local properties are based on the pairwise distances between datapoints. The low-dimensional representation is computed by minimizing the distances between a datapoint and its $k$ nearest neighbors, in a weighted manner (the distance between the datapoint and its nearest neighbor contributes more to the cost function that the distance between the datapoint and its second nearest neighbor.

The first step of this technique is also the construction of a neighborhood graph $G$, where every datapoint $x_i$ is connected to its $k$ nearest neighbors. For each set of points $x_i$ and $x_j$ connected in $G$, a weight is associated to its edge, using the Gaussian kernel function $w_{ij} = \exp^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$. These weights are the entries of a matrix $W$. The computation of the low-dimensional representations $y_i$ are obtained by minimizing the cost function presented in Equation 3.30.

$$\phi(Y) = \sum_{ij} (y_i - y_j)^2 w_{ij} \tag{3.30}$$

Therefore, as large weights $w_{ij}$ correspond to small distance in the high-dimensional space, the distances between the corresponding datapoints in the low-dimensional representation contribute highly to the cost function. Therefore, nearby points in the high dimensional space are brought closer in the low dimensional representation.

### Hessian LLE

This technique is a variant of the LLE, that tries to minimize the "curviness" of the high-dimensional representation when constructing the low-dimensional representation. The first step is identifying, for each datapoint $x_i$, the $k$ nearest neighbors $x_{i_j}$, using Euclidean distance. In this neighborhood, local linearity is assumed. Therefore, a basis for this tangent space of a datapoint can be found applying the PCA technique on the nearest neighbors. Thus, one can define the matrix $M$ that contains the $d$ principal eigenvectors computed with PCA technique for the tangent space. The estimator for the Hessian of the high dimensional representation at the point $x_i$ is computed by first calculating a matrix $Z_i$ that contains in its columns all the cross products of $M$ up to the $d$th order. Next, $Z_i$ is normalized applying Gram-Schmidt orthonormalization. The tangent Hessian $H_i$ is obtained transposing the last $\frac{d(d+1)}{2}$ columns of $Z_i$. Hence, having the Hessian estimators, one can construct matrix $H$ with entries:

$$H_{lm} = \sum_i \sum j (H_i)_{jl} \times H_i)_{jm}). \tag{3.31}$$

By performing a eigenanalysis of matrix $H$, the low-dimensional data representation is found.

### Local Tangent Space Analysis

This technique has some similarities with the Hessian LLE, has it also describes the local properties of the high-dimensional representation by using the tangent space of each datapoint. As local linearity is assumed, a linear mapping between the high-dimensional representation to its local tangent space must exist, together with a linear mapping from the low dimensional datapoint to the same tangent space. These local tangent spaces at each datapoint $x_i$ are obtained by applying PCA on the $k$ nearest neighbors. Hence, a mapping $M_i$ is constructed, in order to map data between the high-dimensional representation and the tangent space $\Theta_i$. The linearity assumption allows to consider a linear mapping $L_i$ between the local tangent space coordinates $\theta_{i_j}$ and the low-dimensional representations $y_{i_j}$. Therefore, this technique preforms the following minimization:

$$\min_{Y_i, L_i} \sum_i \|Y_i J - k - Li\Theta_i\|^2, \tag{3.32}$$

where $J_k$ is the centering matrix of size $k$. This minimization finds its solution in the eigenvectors of the alignment matrix $B$ that correspond to its $d$ smallest nonzero eigenvalues. The entries of this matrix are obtained for iterative summation, for all matrices $V_i$, and starting from $b_{ij} = 0 \quad \forall ij$:

$$B_{N_i N_i} = B_{N_i N_i} + J_k(I - V_i V_i^T) J_k, \tag{3.33}$$

where $N_i$ is a selection matrix that contains the indices of the nearest neighbors od datapoint $x_i$.

The low dimensional representation is obtained by computing the $d$ smallest nonzero eigenvectors of the symmetric matrix $\frac{1}{2}(B + B^T)$.

### 3.2.4 Extensions and variants of local nonlinear techniques

In [2], a third group of nonlinear techniques is presented, containing techniques that are derived from the local nonlinear ones.

#### Conformal Eigenmaps (CCA)

This technique tries to overcome a restriction of the local nonlinear techniques, that do not employ the information regarding geometry of the high-dimensional data representation, that is contained in the eigenvectors corresponding to small eigenvalues, since these eigenvectors are discarded. The CCA technique first employs LLE or other local nonlinear technique, reducing the original dimensionality $D$ to $d_t$, where $d < d_t < D$. Then, this intermediate solution is used to construct a d-dimensional solution that is conformal.

One can define conformality by considering a triangle $(x_h, x_i, x_j)$ and its low dimensional counterpart $(y_h, y_i, y_j)$. Conformality occurs if

$$\frac{\|y_h - y_i\|^2}{\|x_h - x_i\|^2} = \frac{\|y_i - y_j\|^2}{\|x_i - x_j\|^2} = \frac{\|y_h - y_j\|^2}{\|x_h - x_j\|^2}. \tag{3.34}$$

Hence, a measure $C_h$ is defined to measure conformality of the triangles present in the neighborhood graph of datapoint $x_h$:

$$C_h = \sum_{ij} \eta_{hi}\eta_{ij}(\|y_i - y_j\|^2 - s_h\|x_i - x_j\|^2)^2, \tag{3.35}$$

where $\eta_{ij}$ is a variable that is 1 if $x_i$ and $x_j$ are connected in the neighborhood graph, and $s_h$ is a variable for scaling corrections. Summing over $C_h$ yields a measure $C(Y)$, that this technique tries to minimize.

#### Maximum Variance Unfolding (MVU)

MVU is a technique very similar to the FastMVU technique previously mentioned, as both share the same optimization problem. However, this technique starts from an intermediate solution obtained with LLE with dimensionality $d_t$, with $d < d_t < D$. Then, the FastMVU procedure is applied to this intermediate solution. The main advantage of applying MVU instead of FastMVU, is that in MVU the FastMVU procedures are applied to a much smaller problem, minimizing the computational effort.

#### Linearity Preserving Projection (LPP)

This and the following two techniques represent an attempt to combine the advantages of both linear and local nonlinear techniques. This is done by finding a linear mapping that minimizes the cost function of Laplacian Eigenmaps.

#### Neighborhood Preserving Embedding (NPE)

Like LPP, this technique minimizes the cost function of a local nonlinear technique, namely LLE, by applying a linear mapping.

#### Linear Local Tangent Space Analysis (LLTSA)

This technique uses a linear mapping to minimize the cost function of LTSA.

### 3.2.5 Global alignment of linear models

This set of models represent a group of techniques that attempt to combine the global nonlinear models and the linear models, by computing a collection of linear models and perform an alignment on these models.

#### Locally Linear Coordination (LLC)

The LLC technique can be seen as a two-step process. In the first step, a mixture of local linear models, namely factor analyzers or probabilistic PCA, is computed by means of an Expectation-Maximization(EM) algorithm. Secondly, these models are aligned to obtain the low dimensional data representation. This is done by minimizing the LLE cost function.

### Manifold Charting

As the LLC technique, Manifold Charting constructs a low-dimensional representation of the data by aligning a mixture of factor analyzers (MFA). This alignment is done, not by the minimization of a cost function of another dimensionality reduction technique, but by minimizing a convex cost function that measures the amount of disagreement between linear models on the global coordinates of datapoints.

### Coordinated Factor Analysis (CFA)

CFA is somehow similar to the previous techniques, as it also constructs the low dimensional representation of the data by performing a global alignment of a mixture of factor analyzers. However, a main difference from the previous techniques arises from the fact that, in CFA, the mixture and the alignment are made in a single stage, using an EM algorithm that maximizes the normal MFA log-likelihood function minus a term that measures the resemblance between the mixture and the Gaussian.

## 3.3 Artificial Neural Networks

Artificial Neural Networks are state-of-the-art classifying tools [4]. They can be seen as a adaptative system that changes its structure according to internal or external information. They are very useful for modeling complex relations between inputs and outputs - that is done by connecting simple elements, that together can model very complex relations. The name artificial neural network arises from the main inspiration for such system: the central nervous system.

Through training, the heights associated to each connection is changed until the desired behavior is reached. The output, $f(x)$ is obtained by combining functions $g_i(x)$, that can also be defined by combining other function. Mathematically, this notion is represented in Equation 3.36.

$$f(x) = k(\sum_i \omega_i g_i(x)) \tag{3.36}$$

In 3.36, $k()$ represents a pre-defined function.

The most interesting characteristic of a neural network is its learning ability: using a set of observations, the weights to achieve the "perfect" mapping between input and output are derived.

### 3.3.1 Neurons

Neurons are the basic processing unit of an Artificial Neural Network. They have an input from where signals are received from other units, a body were the inputs are somehow integrated, passed through a threshold function and directed to the output, that is connected to other units. This thresholding function can be one of several alternatives, i.e. linear, ramp, step or sigmoid. The sigmoid functions follow the relation $S(x) = (1{+}\exp(-x))^{-1}$. The relation between the input and output nodes of a neuron is given by the expression in Equation 3.37, where $S(.)$ is the thresholding function, and $y'_i$ represents each of the inputs, and $\omega_{ki}$ the weight of the connections between node $k$ and node $i$. Sometimes an additional thresholding bias has to be considered, therefore adding an extra input to the neuron, $\theta_k$.

$$y_k = S(\sum_{i=1}^{N}(\omega_{ki}y'_i) - \theta_k) \tag{3.37}$$

ANNs consist of connections between these simple units. These interactions between many simple nonlinear elements result in a very powerful tool. There are two basic concepts regarding the type of connections allowed between units: the feedforward concept regards the fact that no connections between a cell output and its inputs are allowed. The recurrent concept regards networks were feedback is allowed.

### 3.3.2 Multi-Layer Neural Network

There are an unlimited number of possible connections between neurons, hence the amount of architectures regarding ANNs is enormous. However, the architecture most used in Speech Recognition applications is the Multilayer Neural Network [4]. In this architecture, the neurons are organized in layers. A layer can be defined as a set of neutrons for which the inputs are the weighted outputs of the previous layer (and possibly a thresholding bias). Hence, the first layer receives the external inputs, and the final layer provides the external outputs. The hidden layers are those which the outputs cannot be directly accessed by the outside world.

These organization in layers usually requires sequential connections, therefore the recurrent concept is excluded (there are some deviations from this concept, but will not be addressed). The term Multilayer Perceptron (MLP) or Feedforward ANN regards a nonrecurrent layered network.

### 3.3.3 Backpropagation Algorithm

The learning ability is one of the most precious characteristics of ANNs. The backpropagation algorithm is a learning algorithm commonly used. This algorithm has the goal of finding the weights that minimize the total squared output error. That derivation is based on a set of training parameters. Although the main idea is very simple, one must not forget that the MLP implements nonlinear mappings, thus there will be generally multiple minima in the error surface. Our point is therefore to find the global minimum. In practice, the weights are adjusted to a local minimum, and the procedure is repeated until a good local minimum is achieved.

The backpropagation algorithm is a recursive algorithm, that converges after an unknown number of iterations. The behavior is highly dependent on the initial choice of weights.

## 3.4 Classification

### 3.4.1 Bayesian Decision Theory

Bayesian Classifiers are one of the most simple kind of classifiers. The Bayesian Decision Theory is a statistical approach to the pattern recognition problem [12]. These classifiers are based on the computation of the posterior probabilities, for each class. According to the Bayes formula (Equation 3.38), the posterior probabilities can be calculated from the prior probabilities and the class conditional probabilities [12].

$$P(w_j|\mathbf{x}) = \frac{p(\mathbf{x}|w_j)P(w_j)}{p(\mathbf{x})} \tag{3.38}$$

In this equation, $w_j$ represents class $j$ and $\mathbf{x}$ represents the current sample. In order to achieve a decision, one has to decide to which class a certain sample belongs. Defining discriminant functions ($g_i(\mathbf{x})$) as the posterior probabilities $P(w_i|\mathbf{x})$, it is straightforward that for minimizing the average probability of error we should choose the class based on the following rule ($w_i$ denotes class $i$):

$$\mathbf{x} \in w_i \Rightarrow g_i(\mathbf{x}) > g_j(\mathbf{x}) \qquad \forall j \neq i. \tag{3.39}$$

Developing a classifier basically consists in defining the set of discriminant functions. According to the assumptions made, different discriminant functions may be devised. In this work, two functions were considered. Thus, two different bayesian classifiers were used.

**Linear discriminant classifier** Assumes that the covariance of each class is equal. Equation 3.40 presents the corresponding discriminating function.

$$g_i(\mathbf{x}) = -\frac{1}{2\sigma^2}[\mathbf{x}^t\mathbf{x} - 2\mathbf{m}_i^t\mathbf{x} + \mathbf{m}_i^t\mathbf{m}_i] + \ln P(w_i) \tag{3.40}$$

where $\sigma$ is the standard deviation and $\mathbf{m}_i$ is the mean of class $i$.

**Quadratic discriminant classifier** Assumes that the covariance matrices are different for each category. The surfaces delimiting each class space are hyperquadrics.

$$g_i(\mathbf{x}) = \mathbf{x}^t\mathbf{W}_i\mathbf{x} + \mathbf{w}_i^t\mathbf{x} + w_{i0} \tag{3.41}$$

where

$$\mathbf{W}_i = -\frac{1}{2}\Sigma_i^{-1} \tag{3.42}$$

$$\mathbf{w}_i = \Sigma_i^{-1}\mathbf{m}_i \tag{3.43}$$

$$w_{i0} = -\frac{1}{2}\mathbf{m}_i^t\Sigma_i^{-1}\mathbf{m}_i - \frac{1}{2}\ln|\Sigma_i| + \ln P(w_i) \tag{3.44}$$

$$\Sigma = \mathbf{E}[(\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^t] \tag{3.45}$$

In these equations, $\Sigma_i$ represents the covariance of class $i$, defined in Equation 3.45, $\mathbf{E}[]$ denotes the expected value, and $\mathbf{m}_i$ is the mean of class $i$.

### 3.4.2 Nearest Neighbor Classifiers

The nearest neighbor rule is of simple understanding: consider a set of $n$ labeled prototypes, $D^n = \{\mathbf{x}_1 \dots \mathbf{x}_n\}$. Given a test point $\mathbf{x}$, it is assigned to the class to each the prototype $\mathbf{x}'$ nearest to it belongs.

In this technique, the feature space is partitioned into cells, each corresponding to the points closer to a certain prototype sample - this is called Voronoi tessellation of the space.

The k-nearest neighbor extension of this rule classifies a point $x$ to the class most represented among the $k$ nearest samples.

This method, although simple, requires a considerable amount of calculation, as, for each point, the distance to all the prototypes has to be computed. To overcome this difficulty, there are three methods that reduce the computational complexity of this technique. The first is called partial distance. In this method, the distance is calculated using only a subset of the full $d$ dimensions, and if this distance is too big, no further calculations are made.

The second method is referred to as prestructuring, and consists in the creation of a type of search tree in which the prototypes are selectively linked. For each test point, the distance is calculated to the "root" prototypes, and then the distance to those prototypes linked to the "root" closest to the test point.

The third method is editing. This step basically regards the elimination of the prototypes that are "useless' in the classification process, i.e, one can eliminate prototypes that are surrounded with prototypes belonging to the same class - by doing this, the decision boundaries remain unchanged, and the number of points to be considered in the calculation of distances is reduced.

## 3.5 Conclusions

In this chapter, the basic pattern recognition process is presented. This approach will be followed in the development of the vowel classifier. Therefore, the techniques regarding the two major steps of the pattern classification process that were used in the present work are described is this chapter.

Several feature extraction were presented. These different approaches were tested, and a visual evaluation of the quality of the mapping achieved was then performed - it was noted that for this data, the performance of nonlinear techniques is inferior to the performance of linear techniques . This conclusions will be presented in the subsequent chapters.

Also, two groups of classifiers were presented: the bayesian classifiers, and the nearest-neighbors. These are different approaches to the classification problem, and both were tested further in this work.

ANN are very important classifying tools and will also be used in the subsequent tests.

# Chapter 4

# State of the Art

## 4.1 Introduction

Visual Displays have been used to provide feedback regarding speech articulation for several decades. The traditional users of such systems are hearing impaired, that urge for feedback systems alternative to the auditory system. However, the utility of such systems can be expanded to other areas, namely to patients with different voice pathologies, or to the language learning area.

Several approaches to this problem have been proposed. In this chapter, the main achievements regarding the definition of visual displays for vowel representation are presented.

## 4.2 Spectrograms

In [6], spectrographic analysis is presented as “the first attempt to relate what were already known as articulatory properties to what are now known as acoustic or spectral properties”. However, although the spectrographic analysis is extremely useful to phoneticians, it provides a complex display, that can be only understood after careful study. Therefore, it is not an obvious choice to replace the auditory feedback - simpler displays are needed.

## 4.3 Formant Plots

Concerning the isolated vowel recognition problem, the most traditional visual displays are the formant maps. These displays rely on the estimation of the first three formants using the short-time Fourier spectrum from the speech signal. This technique provides solutions “very reliable and robust for male speakers, but not for female or child speakers” [14]. Also, the estimation of the formant frequencies in real-time is not trivial [15] - when the fundamental frequency is close or higher than the first formant (child and women speech) the LP (Linear Prediction) techniques for estimating the formant frequencies fail, as the magnitude spectrum becomes undersampled, and important spectral peaks are “missed” [3]. However, even in these cases the human auditory system is able to correctly identify the uttered vowel. Therefore, features different from formants should provide a better discrimination ability, and should be used in this identification. Furthermore, this realization is quite obvious when observing the formant plots: this technique do not provide solutions that are capable of completely separate the several vowels [4], as Fig. 4.1 [1] clearly shows.

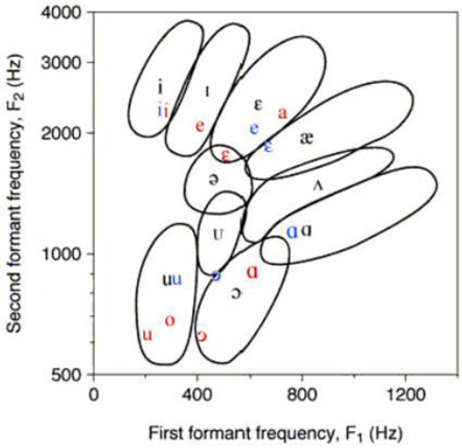

Figure 4.1: Formant Plot for English Vowels

## 4.4 Alternatives to Formant Plots

Given the obvious fragilities of the use of formants regarding vowel classification, alternative approaches have been proposed. In [14], a new concept is presented: Perceptual Spectral Cluster (PSC), an alternative approach to formants. The main purpose of the introduction of these new features is to capture perceptual cues in a more effective way than formants. These concept is defined in the same paper: “PSC denotes a spectral region with a significant local concentration of spectral power, the characteristics of which determine important perceptual cues used in vowel recognition. A PSC may

[1] Figure obtained from http://www-ccrma.stanford.edu/˜jmccarty/formant.htm

be primarily characterized by its center frequency, power, and a measure of shape such as bandwidth or level differences among close harmonic partials" [3]. In order to characterize the performance of these new features, an euclidian-based distance measure between vowels was defined: Vowel Separation Measure (VSM). The tests made with 3 speakers have shown that these new features yield minimum VSM values higher than when using formants: 1.8 versus 0.6. Further research regarding these new features is presented in [3]. In this work, the idea of a human auditory system organized in stages, starting by raw features like intensity, until more elaborated features, like pitch, timbre and finally the speaker recognition, is pursued. Therefore, the several stages must be taken in account, even for apparently simple tasks, namely vowel recognition. Four alternative features sets were used: the first used PSC-related features, namely the pitch, the center of gravity of PSC1, the center of gravity of PSC2, the right border of PSC1 and the decibel difference between the average magnitude of PSC1 and PSC2; the second set uses a LP technique to obtain the values of the first four formants, to which the pitch value is added; the third set uses also LP techniques to estimate formants, however this is a modified algorithm, designed to prevent estimation errors common in the first technique; finally, the fourth technique uses 16 MFCCs. The purposed feature set yields results of about 88.4% of correct identification. This is not, however, better than the results that the MFCC set yield (94.0%), but it should be noted that the dimensionality of this set is significantly higher.

Vowel Articulation Training Aid (VATA), developed by Dr. Stephen A. Zahorian [5], is a speech therapy tool, that intends to replace insufficient or missing acoustic feedback by real-time visual feedback. The first implementations of such tool were hardware-based, using analog filter banks, and the output was simply a change in the color of a television screen. The goal of his work is defined has the development of a technique to map vowels to colors, in a speaker-independent manner. In [7], Zahorian points out the necessity of finding a visual display that could replace the deficient or lacking auditory feedback (thus doing the mapping in real-time): the efforts done until that point had resulted in complex and hard to interpret displays. The proposed method is to use the output of a 16-channel filter, and mapping it to a 3-dimensional space in which each coordinate represents the amounts of red, green and blue. The mapping is achieved by one of two methods: a linear transformation, whose coefficients are computed so that the mean square error between the target coordinates and the actual coordinates of each vowel is minimized; and a nonlinear transform using a 2-layer feedforward perceptron neural network, followed by a linear transformation to convert vowels in colors. The second technique yielded better results in the tests made.

In [15], the natural evolution of the previous work is presented: a new visual display, based on the traditional F1/F2 displays. The features used to characterize the speech remain the same (the output of a 16-channel filter bank) - a form of cepstral coefficients. This representation, based in the overall spectral shape is preferred to the use of formants, as formants are hard to track in real-time. The 16-dimensional data is mapped to the two-dimensional display using an ANN.

In [16], the previous work is adapted to run in a personal computer environment, thus discarding the need of expensive specialized equipment. However, the basic steps of the processing remain the same, with the use of an Artificial Neural Network to achieve the desired mapping.

In [17], although the same basis was still used, some improvements to the application were introduced. The features used are hence cepstral coefficients, referred to as Discrete Cosine Transform Coefficients, that are mapped by means of a neural network. The main improvement was the addition of a Maximum Likelihood Classifier that is used together with the Neural Network to avoid the occurrence of false corrects. This inclusion was justified in [17] as follows: "the NN (Neural Network) must choose from among only the vowel categories for which it was trained; it cannot "choose" another category. This has the unfortunate consequence of occasionally producing feedback corresponding to a "correct" pronunciation when in fact no valid vowel sound is in the audio stream". Therefore, an MLC was added to the system, to measure the proximity between the features of the correct sound, and the features of the "typical" phoneme. If they are not close enough (with this limit being defined by an empirical procedure) they are discarded. Another important improvement was the use of a larger amount of data. As mentioned in the work, "for "good" performance, a neural network classifier requires a large amount of training data". Also, other changes were added to the program: namely, the existence of two displays - the continuous F1/F2-like display, referred to as ellipse display, that reflects the changes in articulation in a continuous manner, and a discrete display, referred to as bargraph, in which the presently identified vowel is highlighted with the activation of the corresponding bar (the activation of two bars could happen when incorrect utterances were produced, but when a bar is significantly higher than the rest, the corresponding vowel has been identified). As for the expected performance of the system, the authors present the expected upper limits for the classification rates, obtained using all available data to train the classifier. The results obtained were 89.99% of recognition rate for male speakers, 87.33% for female speakers, 83.33% for child speakers and 84.96% when combining all the samples. These tests were made considering 10 English vowels.

In [5], the same test results and the same methodology is described. The main difference is the introduction of three game displays, in addition to the two main displays already described. These game displays are controlled by the utterance of vowels, and constitute an alternative type of display, specially suited for children training, as these displays provide higher motivation to this audience.

IBM SpeechViewer, Indiana Speech Training Aid (ISTRA), Video Voice Speech Training System, Speech Training Aid for Hearing Impaired (HARP), Jogos Fonoarticulatórios [18] are examples of applications that comprise a set of games, including vowel-controlled games or vowel displays. However, these applications are now obsolete, and no new releases have been provided.

OLTK is an interactive tool designed to build a 2D visual display of vowels, and thus allowing the visualization of the distance between the current utterance and the intended one. The application developed is based in the “Speak and Look Cycle”, that is presented as an alternative approach to the traditional ”Speak-Hear Cycle”. As the preferential feedback source (the auditory system) is somehow defective, the visual feedback is used as an alternative - basically, the user produces an utterance, and the resulting feedback is given in a visual manner, in real-time, in the visual display. In this approach 9 cepstral coefficients are obtained from the input sounds, and then mapped to a 2D space by means of an ANN. The advantages of using a 2D vowel display are presented: the ability of viewing the several speech events effect and understand what goes wrong is a major quality. Therefore, the output of this system is a simple 2D display, that includes tools designed for more technically oriented persons. This is hence a tool that is meant to be used in speech therapy sessions, and that has proven to contribute to improve the patient’s motivation.

## 4.5 Application of Pattern Classification Techniques

In [19] and [20], pattern recognition techniques are applied to speech recognition problems, namely vowel recognition, but with no real-time application concerns. The work hence provides an overview of the behavior of the several algorithms when applied to speech recognition problems. Five algorithms are studied: the use of LDA (Linear Discriminant Analysis) and PCA (Principal Component Analysis) as feature extractors used prior to the classification step is compared to the use of MCE (Minimum Classification Error) and GMCE (Generalized Minimum Classification Error) as techniques that provide the joint feature extraction and classification tasks. The state-of-the-art classifiers SVMs (Support Vector Machines) are also used. PCA and LDA are independent feature extraction techniques that have the great advantage of being linear techniques, and hence simple to implement. The MCE and GMCE algorithms provide an integration of the feature extraction and classification steps. Like the other methods, the MCE technique extracts features from the parameter vector through a linear transformation matrix $T$, but derives discriminating functions dependent of such matrix. The GMCE method is a generalization of MCE that uses an initialization procedure to avoid local minima determination when applying the normal MCE algorithm. The SVM is a nonlinear classification algorithm that uses a kernel method: first the original parameter values are mapped to a higher dimensional feature space through nonlinear kernel functions, and then, in the higher dimensional space, the SVM tries to represent the class by the samples which are closest to the boundaries. The tests regarding the several approaches were made using vowels extracted from the TIMIT database,with parameter vectors composed of 20 MFCCs and 1 energy coefficient. Speaker-independent tests proved similar performance of both linear techniques (PCA and LDA), but LDA appers to have more stability. Also, SVM did not perform well in speaker-independent tests. The MCE and GMCE algorithms performed better than the linear algorithms

In [21], pattern recognition techniques are applied to European Portuguese vowels. The results of using these techniques are promising. It is also shown that the inclusion of pitch as a preliminary feature (which is done here using a gender classifier based mainly in pitch) has obvious advantages.

## 4.6 Conclusions

In this chapter, the main achievements regarding the development of visual displays for representation of vowels are presented.

The traditional approach, that uses formant plots, has many flaws: first, a complete separation is not achieved; secondly, the techniques for estimation of formants are not robust. Therefore, alternative approaches where proposed. Spectral-shape features have proven to perform well to vowel identification. Also, some indications regarding the use of features corresponding to different stages of perception are provided.

Some studies regarding the application of pattern recognition techniques were also made. However, no concern with real-time application of the techniques was taken.

The use of MFCCs to characterize the speech signal appears as the most effective approach. These will be hence the parameters used primarily in the following experiments.

# Chapter 5

# Preliminary Experiments with Vowel Signals

## 5.1 Introduction

The purpose of this work is the development of an automatic classifier for 5 of the 8 European Portuguese oral vowels. In this chapter, some of the preliminary experiments made with these vowel signals are presented. These experiments regard mostly the spectral behavior of the vowels. As it was already mentioned, overall spectral shape features have proven good performance in vowel recognition operations [5]. Hence, a general insight regarding the relation between the spectral shape and the human ability of recognizing a vowel is presented in this chapter.

## 5.2 Spectral Analysis of Vowel Signals

Preliminary observations consisted in the observation of the spectral characteristics of each vowel, for the several genders (adult male, adult female and child).

The observation of the several spectra clearly shows a similar shape exists for the several vowels. The overall spectral shape of the 5 vowels is presented in Fig. 5.1, and examples of spectrograms regarding different utterances of each vowel are presented in Fig. 5.2.

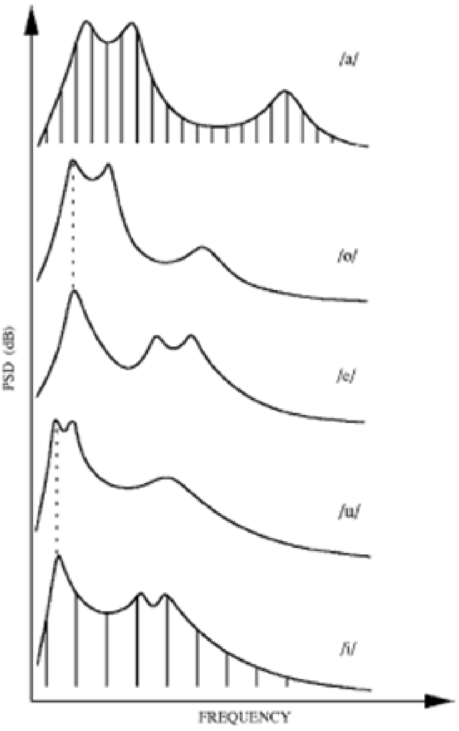

Figure 5.1: Overall spectral shape of the 5 vowels [3]

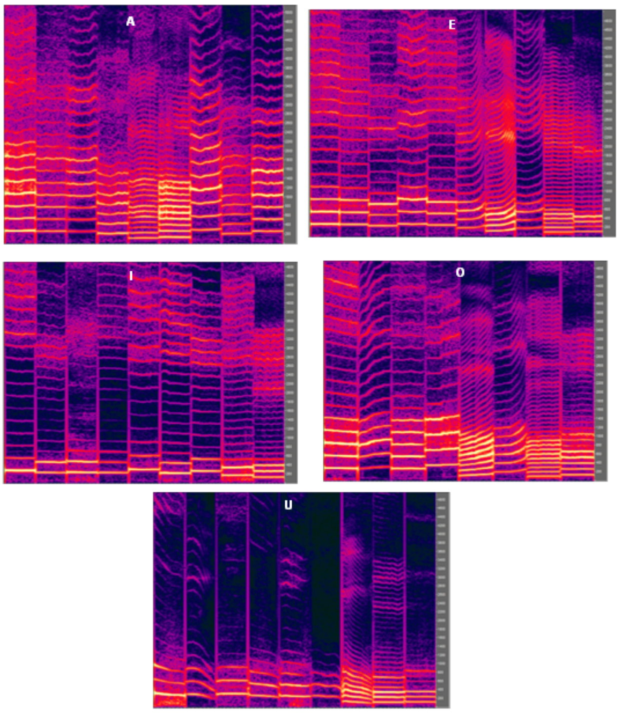

Figure 5.2: Spectrograms of several utterances (from different speakers) of each vowel

The main conclusions regarding the several spectra are:

**A** The spectral envelope maintains high values for a large frequency interval.

**E** The valley in the spectral envelope is the most prominent characteristic

**I** This vowel in mainly characterized by the high amplitude values of the spectrum at low frequencies.

**O** Very similar to the spectrum of vowel /a/, this is a slightly compressed version of the former spectrum

**U** The important information is concentrated at low frequencies

Although the basic shape is essentially the same, which is a good indication regarding the use o spectral-shape features for vowel classification as defended in [5] , there are obvious differences between the several genders (which is consistent to the use of different classifiers for each gender as presented in VATA [5], or with a hierarchic classifier that first comprises a gender classifier [21]). These differences are obviously related with the different pitches, that result in different "compressions" of the partials (see Fig.

5.3). This is coherent with the conclusions presented in [3] regarding the feasibility of LP techniques for predicting the formants when high pitched vowels are considered.

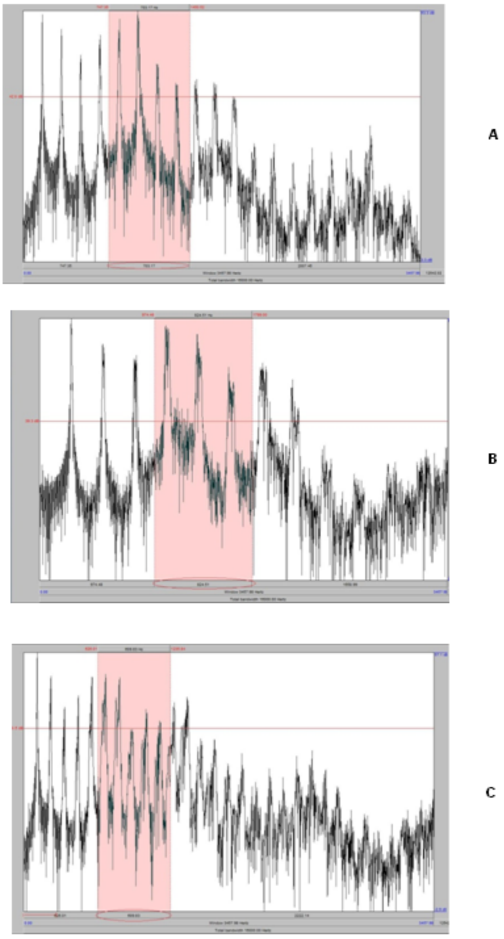

Figure 5.3: Spectral Shape of vowel /a/ for A.Adult Female B.Child C.Adult Male

Another interesting property is that the essential information regarding the perception of a certain vowel if contained in a specific frequency interval: in Fig. 5.3, if one apply a bandpass filter maintaining only the highlighted frequencies, the vowel “a” is still perceived. Furthermore, if the signal is bandpassed for other frequencies, different vowels can be synthesized (i.e., if only the first two or three spectral peaks are maintained, an “u” is easily produced).

## 5.3 Partial Equalization

Further insight regarding the importance of the spectral shape was obtained by performing several experiments regarding partials equalization. The partial equalization step consist in an identification of the several partials of the acoustic signal, and a subsequent attenuation of its amplitude value until the amplitude of all is the same. Therefore, the shape of the spectrum will be degraded at each equalization step. The purpose of such a proceeding is to understand to each extent one is able to degrade the spectral shape of a vowel and maintain its intelligibility.

The experiments consisted in the following: a partial tracking algorithm was used to detect its frequency position. Subsequently, the amplitude corresponding to the partial was scaled using different factors, until the amplitude of all the partials is equalized. For the several scale factors, the signals were resynthesized, and listened, in a random order, and the corresponding vowel was identified. Fig. 5.4 shows the spectra regarding several steps of the equalization.

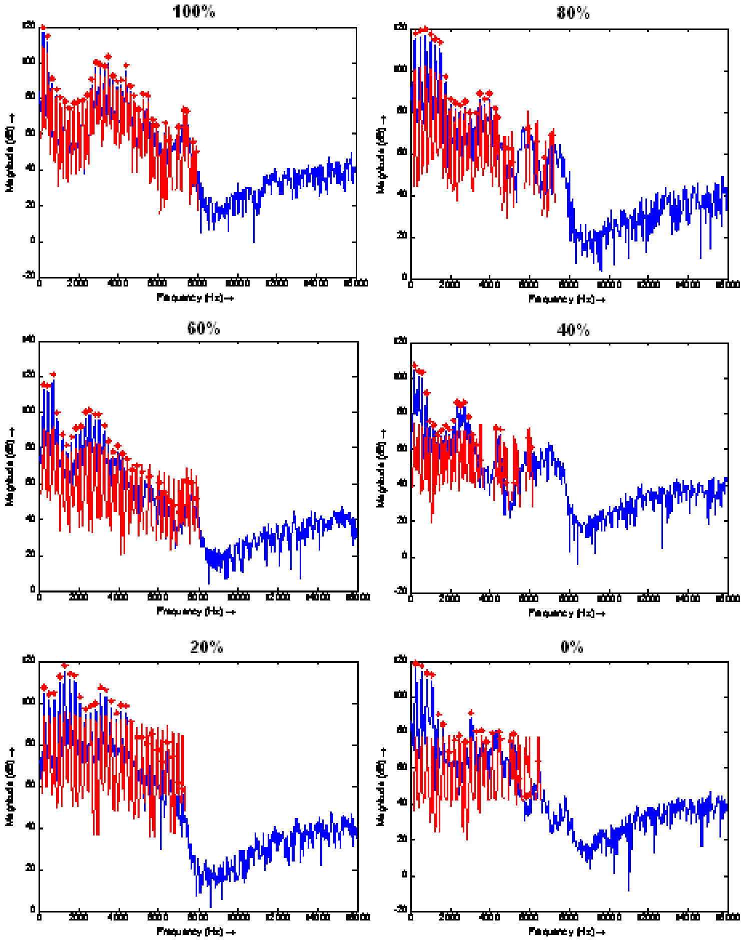

Figure 5.4: Partials Equalization

From the experiments it was noted that, although an obvious gradual decrease in the vowel intelligibility occurs with the equalization, while the overall spectral shape is maintained, the vowels are fairly well recognized. This reinforces the idea of the importance of the spectral shape to the classification of vowels.

## 5.4 Conclusions

In this chapter, a preliminary study of the spectral characteristics of each vowel, and its importance to human comprehension was made. These experiments allowed to draw the following conclusions:

- The spectral shape is obviously a distinctive feature for vowels. Furthermore, while the basic shape is maintained, humans can still recognize the uttered vowel.

- There are remarkable differences between genders, that are responsible for a difficult formant estimation in high-pitched vowels.

- Different vowels have distinctive characteristics in different frequency bands of the spectrum. Furthermore, if the analysis is very limited in frequency, mistakes arise because it has been observed that different vowels can be synthesized from a single original vowel, by bandpassing the signal in different frequency bands.

- The spectra of vowels /a/ and /o/ are very similar.

# Chapter 6

# Algorithmic Development and Simulation

## 6.1 Introduction

In the development of the classifier, the first step consisted in the development and simulation of several different approaches to the problem of vowel recognition. Those preliminary approaches will be presented in this chapter.

The first tests regard the use of MFCCs as a parametrization of the speech signal, because the previous works analyzed in Chapter 4 show that these features perform well in vowel classification. The effect of using different numbers of MFCCs is also addressed.

Several feature extraction techniques are applied to this dataset, and the resulting 2-dimensional plots are analyzed. From the several techniques presented in Chapter 3, only 5 are shown here, as the remainder nonlinear techniques did not perform well. Some of these results are presented in Appendix B.

Another issue addressed is the effect of increasing the length of the segments considered.

Features as pitch and LPCs are also used, together with MFCCs, in proposed hierarchical methods, that attempt to achieve better mappings.

Finally, the vowel classifier is developed, and the simulation results regarding its performance are shown.

The tests were made in the Matlab®environment, with the aid of the PRTools package [22] and the Matlab Toolbox for Dimensionality Reduction [2].

## 6.2 Database Characterization

The database used in this work was created by Aníbal J. S. Ferreira, and is presented in [3]. It is particulary suited for the present work, as it is mainly composed by vowels uttered at high pitch, produced by children and women (the group for which traditional approaches fail): the target group of the present work. A total of 27 child speakers, 11 adult female speakers, and 6 adult male speakers contributed for this database. Each speaker uttered in sequence the 5 vowels /a/, /e/, /i/, /o/, /u/, thus, a total of $(27+11+6) = 44$ files was produced. The duration of each vowel was between 1 and 2 seconds. The samples were recorded in a quiet environment, using a laptop, an electret microphone, and a audio editor using a sampling frequency of 32kHz.

Latter, the files were manually processed, and two variants for the database were created: the first composed by $44 \times 5 = 220$ files of 100 ms, corresponding to the most stable region of the spectrogram. An identical approach allowed the creation of a variant of this database, composed by files with the duration of 400ms. In both variants, the data was carefully labeled.

## 6.3 Classification without applying feature extraction techniques

In [3] it is shown that a simple classifier using 16 MFCC has performed better than those that used LP based features and PSC features (see Chapter 4), for identification of the 5 European Portuguese vowels considered in the present work. Therefore, the starting point of our research regards the use of simple classifiers (described in Section3.4) and MFCCs as features.

MFCCs are features that have proven effectiveness in speech recognition problems. The use of a mel scale, that approximates the human auditory system response is undoubtedly an advantage to be considered. Furthermore these are indeed the most used state-of-the-art features in the area of speech recognition. However, it is also true that some important information is discarded by these features, namely pitch. As stated in [3], the idea that our auditory system approaches even simple recognition problems using information regarding different stages of perception would require the additional inclusion of raw features. At this point however, our main concern is to understand to which extent are the MFCCs capable of characterizing the different vowels.

These first tests had the purpose of giving a general insight regarding the general behavior of the classifiers when using this data. In fact, the dimensionality reduction step was skipped in this phase (Fig. 6.1).

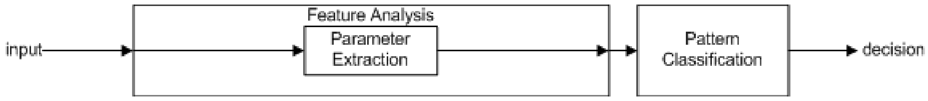

Figure 6.1: The simplified approach (skipping feature extraction step)

In the following subsections, several tests regarding analysis on the performance of these features for vowel classification are presented, specifically analysis regarding the effect of increasing the number of MFCCs.

### 6.3.1 Derivation of the dataset

The dataset used in this step of the work was obtained by calculating, for each file belonging to the 100ms database described in 6.2, the corresponding MFCCs.

The proceeding concerning the derivation of these coefficients is as follows: for each file, a frame-by-frame analysis was made. For each frame, the MFCCs were calculated. Subsequently, for each file, the mean vector of the MFCCs was calculated (see Fig. 6.2). Therefore, the dataset obtained is composed by 220 samples, each of each containing as parameters the MFCCs and the corresponding label (/a/, /e/, /i/, /o/, /u/).

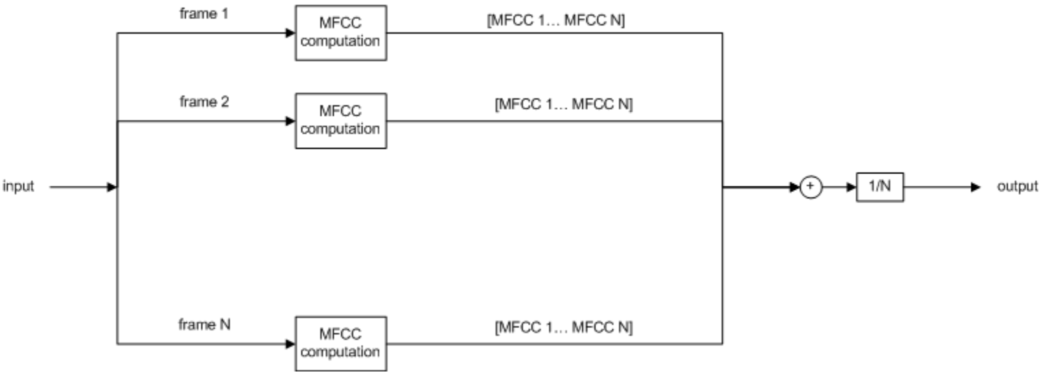

Figure 6.2: The schematic representation of the MFCC computation process

### 6.3.2 Tests with 16 MFCC

The results presented in this subsection regard the use of 16 MFCCs to characterize each vowel. The use of 16 coefficients appears as the logic sequence of the work presented in [3].

The test procedure was made as follows: the classifiers were trained using 70% of the total number os samples, randomly selected from the dataset, the remaining samples were used for testing, and the error rates were stored. The procedure was repeated 100 times, and finally the mean regarding the error rate associated with each classifier was taken. The results obtained are presented in Table 6.1.

Table 6.1: Tests using 16 MFCCs as features

| Classifier | Error Rate |
|---|---|
| LDC | 5.94% |
| QDC | 11.94% |
| 1-NNC | 4.36% |
| 4-NNC | 4.88% |

The first indications, although lacking further validation, gave good indications concerning the pattern recognition approach selected to treat this problem. However, these are obviously positively biased results: first, segments of 100ms are being used in this classification, which are undoubtedly segments longer than the real-time requirement will allow us to use, secondly, the speaker-independent requirement is not being tested, as at this point the existence of samples (corresponding to different vowels) regarding the same speaker in both training and testing sets was allowed. These considerations are obviously important, and will be discussed in subsequent steps of this analysis. However, at this point, only a general analysis was intended.

### 6.3.3 Effect of varying the number of MFCC

The next logical step is to analyze the general effect of varying the number of MFCCs considered as features. The same approach previously described was repeated using different datasets which have as features vectors with different numbers of MFCCs. The error rates obtained are presented in Table 6.2.

Table 6.2: Effect of varying the number of MFCCs

| Classifier | 4 MFCCs | 8 MFCCs | 12 MFCCs | 16 MFCC | 20 MFCCs | 24 MFCCs |
|---|---|---|---|---|---|---|
| LDC | 8.39% | 9.48% | 8.50% | 5.94% | 3.97% | 4.56% |
| QDC | 5.77% | 6.36% | 8.70% | 9.64% | 16.97% | 31.14% |
| 1-NNC | 9.85% | 6.67% | 6.45% | 4.36% | 2.36% | 3.27% |
| 4-NNC | 6.86% | 5.77% | 5.50% | 4.88% | 4.41% | 4.03% |

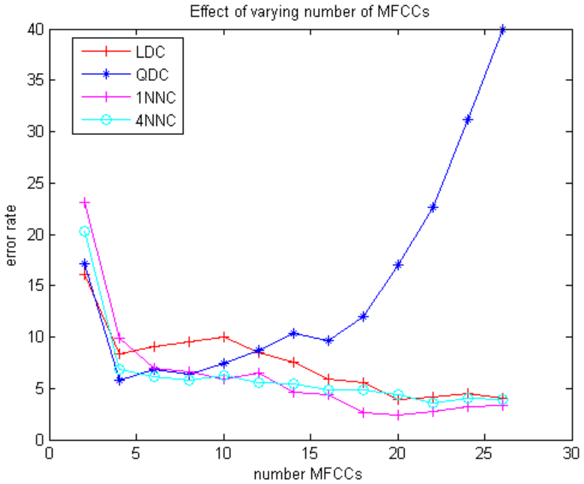

Figure 6.3: Effect of varying the number of MFCCs

Regarding the results presented in Table 6.2 and in Fig. 6.3, some considerations can be held. For both the nearest neighbor classifiers and the linear bayesian classifier, the expected behavior is observed - the error rate decreases as the number of MFCCs considered is augmented. This decreasing in the error rate is more expressive until 16 features are considered. Further increasing in the number of MFCCs corresponds to a smaller impact in the corresponding decreasing in error rate, that eventually tends to stabilize. As for the bayesian quadratic classifier, it is observed that the error rate tends to increase when more than 5 MFCCs are considered. An explanation for this phenomena may be the overfitting of this classifier to the training samples, thus providing worse test results.

### 6.3.4 Spacial distribution of samples using MFCC

The lacking step in the pattern recognition approach is the dimensionality reduction, that often tends to simplify the classifier task, that can hence produce more favorable results. Although it has already been mentioned the advantages of using feature extraction techniques opposingly to feature selection techniques, one must not forget that feature selection techniques represent a simplification of the computation process. Therefore, an observation of the spacial distribution of the samples regarding small sets of MFCCs was made, using scatter plots. The spacial distribution regarding the use of the first 2 (Fig. 6.4) and 3 (Fig. 6.5) is presented. This visual representation is also a good complement to the classifier results presented previously, as it shows the spacial distribution of the considered samples.

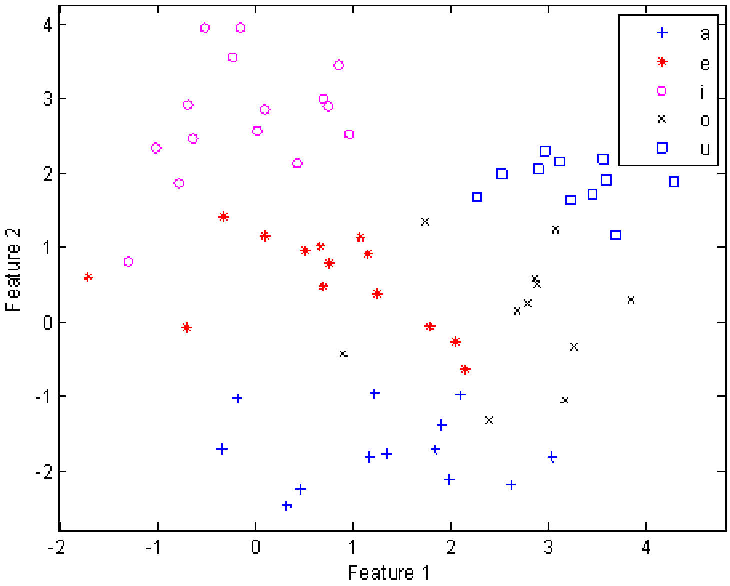

Figure 6.4: Spacial Distribution using 2 MFCC

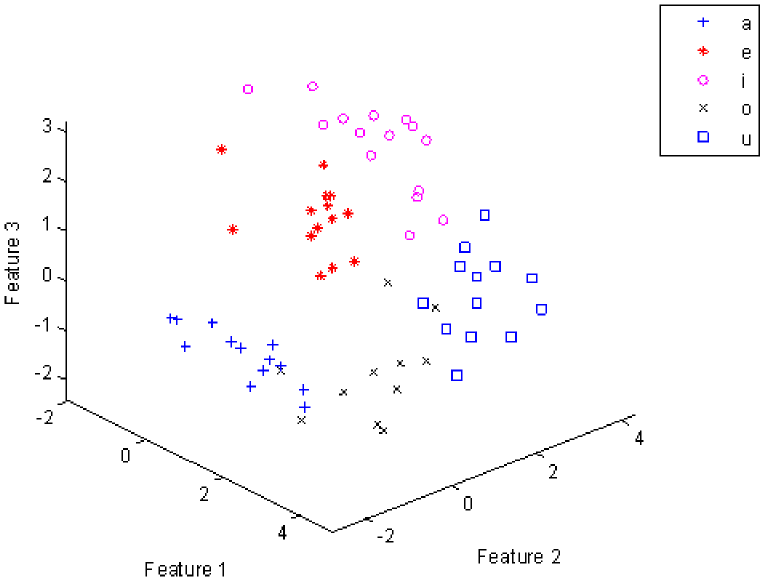

Figure 6.5: Spacial Distribution using 3 MFCC

Both figures show that the use of a small set of MFCCs do not allow the achievement of a good separation between vowels: therefore, the use of feature extraction techniques is necessary.

Furthermore, it can be seen that vowels /i/ and /u/ appear to be more grouped and isolated from the rest. Therefore, the classification task regarding these vowels will be easier. This is not unexpected, as vowels /a/ and /o/ have very similar spectral shapes.

## 6.4 2-dimensional spacial distribution of samples applying feature extraction techniques

After the first tests regarding a simplified pattern classification approach, which gave some insight regarding each of the considered classifiers behavior, and noting the inefficiency of the feature selection techniques, the introduction of the feature extraction step was made. The enormous amount of techniques capable of doing such task (see 3.2) requires a previous study of the performance of each technique. For that purpose, two steps were taken: first, a visual observation of the performance of the several techniques, by mapping a considered dataset to a 2-dimensional space. Then, a subsequent selection of the techniques worth further analysis was done. For these techniques, the computation of the error rate obtained when using the different classifiers can be used as a performance indicator.

### 6.4.1 Dimensionality reduction using 100ms segments

After the first approach regarding classification based on the parameters obtained from the speech samples, the complete pattern recognition flow, described in 3 was resumed. Therefore, the dimensionality reduction techniques were applied.

The several techniques referred in 3.2 were applied to a dataset consisting of 12 MFCCs. Only those who yield better results are shown, the remaining scatters are presented in appendix B.

From the several nonlinear techniques, 3 techniques were selected: the use of ANN, recurrent in these type of applications, MDS and SPE. Both MDS and SPE were presented as global nonlinear techniques, that have proved to perform better with this data. Actually, these are very similar techniques: the main difference is that SPE is an iterative technique. Both techniques attempt to preserve global properties of the data. The nonlinear techniques are suited for highly nonlinear data, that is not the case: hence, one can infer that the data is organized in a linear manner in the high-dimensional data.

### ANN

The use of an ANN to directly provide a 2D-mapping was achieved by defining the targets associated to each label as the 2D coordinates of display. The target coordinates were chosen to resemble the distribution on the classical vowel plot shown in Fig. 6.6. In this figure, the several vowels, represented in its IPA (International Phonetic Alphabet) notation, are represented, accordingly to the corresponding position of the articulators. Therefore, a MLP with one hidden layer was used. The input layer has 12 entries, and the output layer provides 2 values, that correspond to the two coordinates. For the hidden layer, 50 neurons were considered. It must be noted that the number of neurons in the hidden layer has great influence on the performance of the ANN. Different numbers of neurons in the hidden layer have been tested, and it was concluded that the use of 50 neurons is suited for the current problem.

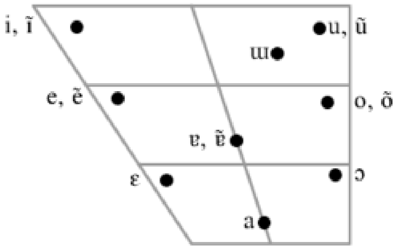

Figure 6.6: Vowel Plot for Portuguese Vowels

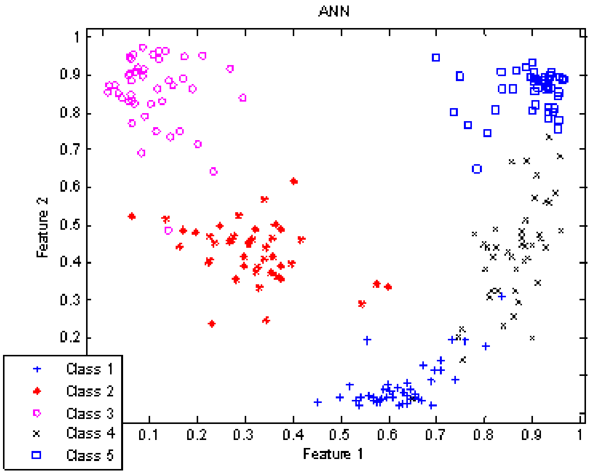

Figure 6.7: Dimensionality Reduction using ANN

The mapping achieved using this technique is very positive, as the several classes are fairly separated. It must be noted that the poorest separation occurs for the vowels /a/ and /o/, which is coherent with the spectral shape similarities already referred.

## PCA

The PCA technique was applied to the dataset, and a mapping to a 2-dimensional subset was achieved. The scatter plot regarding the resulting low-dimensional data is shown in Fig. 6.8.

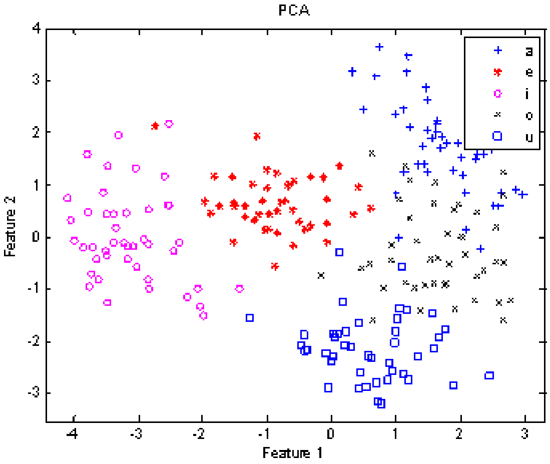

Figure 6.8: Dimensionality Reduction using PCA

Although the separation is inferior to the one obtained using ANNs, it can be observed the clustering of the samples of the several classes, which is a good indication.

## LDA

As the PCA technique, LDA was applied to obtain a mapping from the 12-dimensional space to a 2-dimensional space. The result is presented in Fig. 6.9.

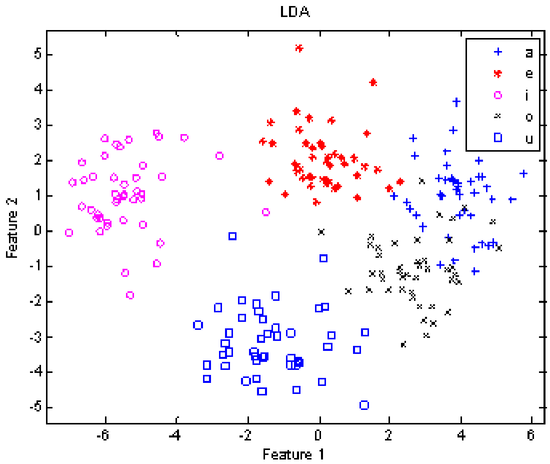

Figure 6.9: Dimensionality Reduction using LDA

The separation obtained is apparently better than the one obtained using the PCA technique, as the separation between classes is higher.

### MDS

In the MDS technique, the minimization of a raw stress function (a measure of the maintenance of the pairwise distances in both high and low-dimensional spaces) is pursued. This is a global nonlinear technique, that is somehow similar to the linear techniques, as it tries to preserve the global geometry of the data. The mapping is made using nonlinear transformations.

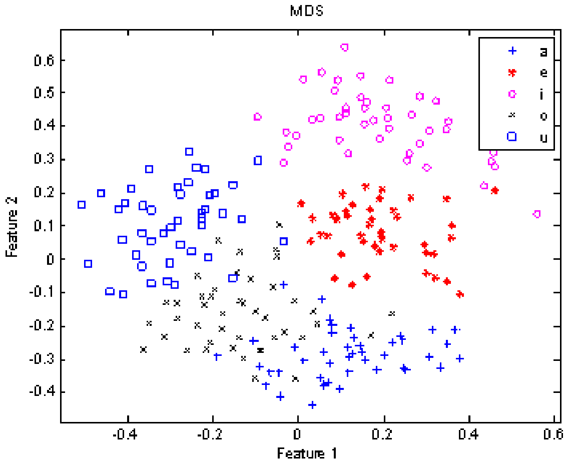

Figure 6.10: Dimensionality Reduction using MDS

The resulting mapping, shown in Fig. 6.10 is undoubtedly poorer than the previous mappings.

### SPE

SPE is a technique very similar to the previous one, but the problem of minimization of the raw stress function is addressed with an iterative approach.

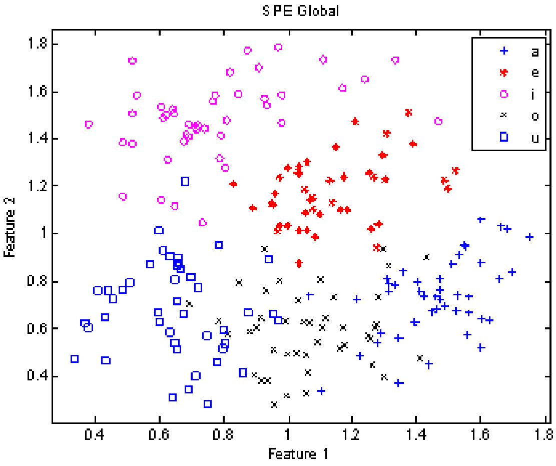

Figure 6.11: Dimensionality Reduction using GlobalSPE

The result obtained is worst than all the previously presented mappings. Although a obvious clustering of the samples of each class occurs, the overlap of the several classes is notorious.

### Conclusions

The linear techniques together with the ANN have proved to perform better than the rest of the nonlinear techniques. The latter perform better with highly nonlinear data, that is obviously not the case.

As for the linear techniques, the performance of the LDA is quite remarkable, providing a good separation between classes. However, none of the techniques achieved a good separation between the vowels with similar spectral shapes, namely /a/ and /o/ (and to a smaller extent, /e/).

### 6.4.2 Dimensionality reduction using 400ms segments

The several mapping techniques used in the previous subsections are here applied to data regarding longer segments. Hence, the alternative database, with 400ms samples, is used, and a new dataset is constructed, with features obtained using the same strategy mentioned earlier: for each file, a frame analysis is performed, and the MFCCs are computed. Later, for each file, the mean of the parameters obtained for the several frames is computed. The use of longer segments will allow an improved normalization of the result.

#### ANN

The main difference regarding the application of the same technique to the previous dataset is that, when using the results drawn from the longer segments, the data concerning each class appears to be more concentrated (the dispersion is lower).

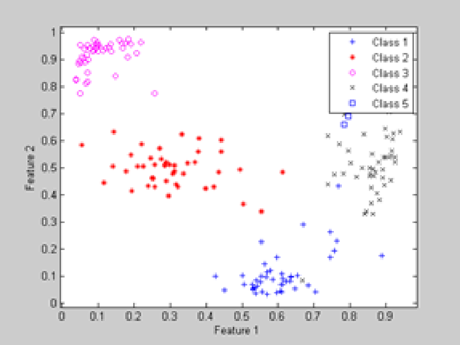

Figure 6.12: Dimensionality Reduction using ANN with 400ms segments

#### PCA

When using PCA with this new dataset, no relevant differences arise from the observation of both low-dimensional representations.

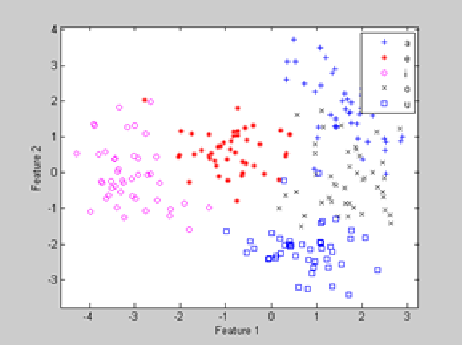

Figure 6.13: Dimensionality Reduction using PCA with 400ms segments

### LDA

As for PCA, the LDA technique is not affected by the increase in the length of the considered segments.

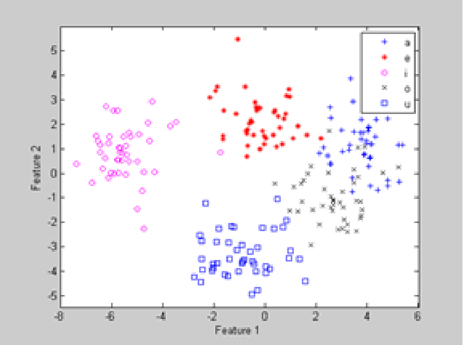

Figure 6.14: Dimensionality Reduction using LDA with 400ms segments

### MDS

The MDS technique has the same behavior of the linear techniques: the increase on the segments length is not reflected in the quality of the mapping.

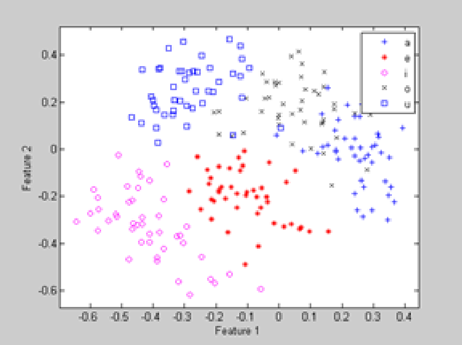

Figure 6.15: Dimensionality Reduction using MDS with 400ms segments

### SPE

In the SPE technique, like in the ANN, the increase in the segment length provided a better clustering of the samples of each class.

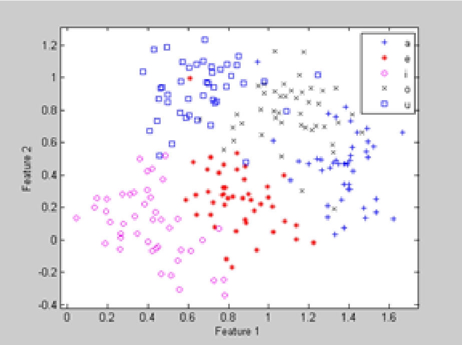

Figure 6.16: Dimensionality Reduction using GlobalSPE with 400ms segments

### Conclusions

In this section we have tested the effect of using longer segments in the quality of the mappings. The scatter plots regarding the several techniques show that only the ANN and SPE techniques (that are iterative techniques) were positively affected by this measure. The regarding techniques showed no change in the performance.

## 6.4.3 Comparison of Results

A numeric comparison of the results can be held by applying simple classifiers to the several 2-dimensional datasets derived by each of the considered techniques.

The results present in Table 6.3 are hence the average classification errors, obtained for each combination mapping/classifier/duration, after repeating each test 100 times, with different random separation of test and training results (this repetition is made to avoid biased results by a suitable distribution of the data in training and test sets if only one random separation was done).

Table 6.3: Comparison of the several feature extraction techniques

| | **ANN** | | **PCA** | | **LDA** | | **MDS** | | **Global SPE** | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 100ms | 400ms | 100ms | 400ms | 100ms | 400ms | 100ms | 400ms | 100ms | 400ms |
| LDC | 9.83% | 9.17% | 9.21% | 9.91% | 10.44% | 10.94% | 10.68% | 12.55% | 13.72% | 14.14% |
| QDC | 9.47% | 9.32% | 10.50% | 10.74% | 10.33% | 10.41% | 9.45% | 10.98% | 13.15% | 13.85% |
| 1NNC | 10.21% | 9.82% | 11.89% | 11.48% | 14.45% | 12.35% | 12.83% | 14.43% | 18.69% | 18.78% |
| 4NNC | 9.48% | 9.28% | 10.74% | 10.05% | 13.21% | 11.73% | 12.62% | 12.42% | 15.23% | 15.06% |

These results give us insight regarding two aspects:

**The effect of using longer segments** There are no significant changes in the behavior of the mapping techniques by increasing the size of the segments. Although some improvement was seen in the visual evaluation of the performance of the iterative techniques, this improvement is not significant. This provides indications that the essential information regarding vowel classification is achieved in the shortest segments, and no further information arises from considering longer segments. Hence, the same information might be found in shorter segments: a good indication to real-time performance.

**The performances of each mapping technique** The nonliner techniques do not bring significant improvements to the quality of the mappings. In fact, only the ANN technique has performed better than the linear techniques, Therefore, the increase in complexity the implementation of nonlinear techniques such as MDS or SPE brings is not compensated by an improvement in the performance.

## 6.5 Effect of increasing the number of MFCC to the performance of linear mapping techniques

Although the effect of increasing the number of MFCCs for classification purposes was already tested, in this section the mappings achieved by using two sets of initial parameters: one with 16 MFCCs and other with 18 MFCCs. The mapping technique used was LDA.

### 16 MFCCs

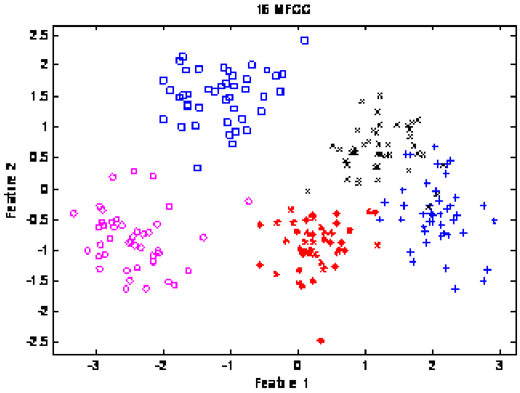

Figure 6.17: Dimensionality Reduction with LDA using 16 MFCCs

### 18 MFCCs

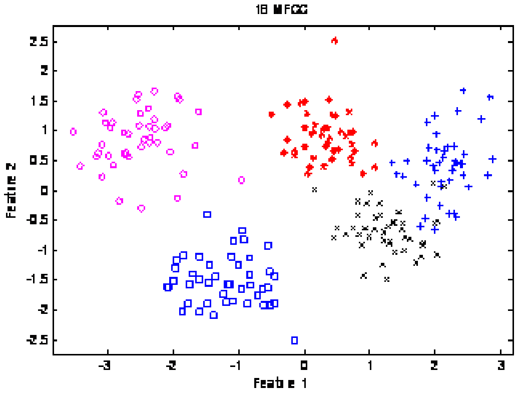

Figure 6.18: Dimensionality Reduction with LDA using 18 MFCCs

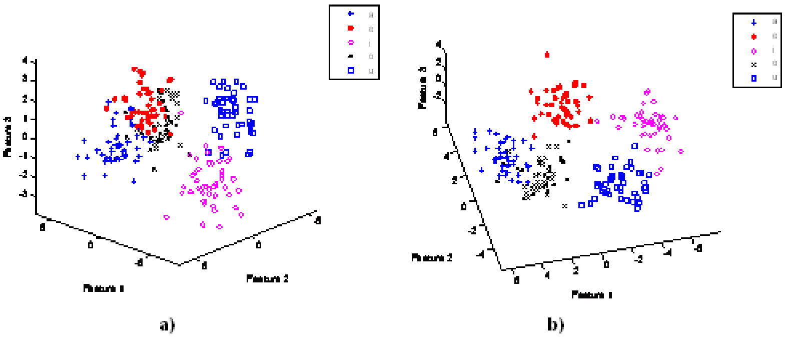

Figure 6.19: Dimensionality Reduction to 3 dimensions with LDA using 18 MFCCs

### Conclusions

The increase of the dimension of the high-dimensional representation appears to bring no significant changes in the quality of the mapping.

## 6.6 Effect of using different parameters

The previous results show a fairly good separation of classes when using MFCCs as initial parameters. However, some confusion between vowels /a/, /o/ and /e/ remains. This occurs because the MFCCs are not able to provide the desired distinctive classification of the vowels: the human auditory system must rely on different or additional features to perform its notable classification. Hence, in this section, results regarding the use of different sets of parameters, mapped using LDA, are shown.

### 6.6.1 16MFCC+5LPC

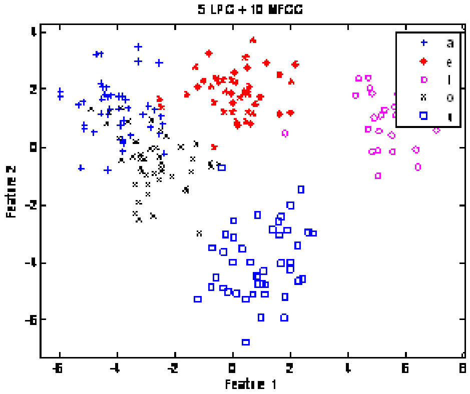

Figure 6.20: Dimensionality Reduction with LDA using 16 MFCCs and 5 LPCs

The use of LP features in addition to the MFCCs, namely 5 LPCs does not provide a increase in the performance of the mapping technique.

### 6.6.2 16MFCC+14LPC

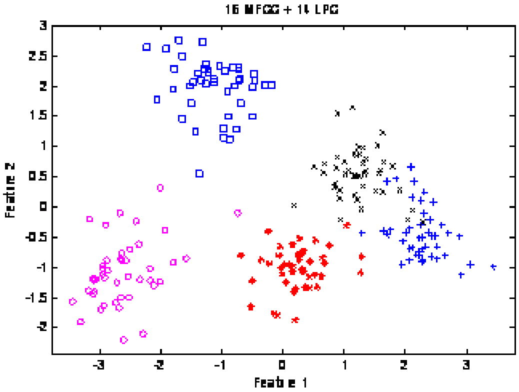

Figure 6.21: Dimensionality Reduction with LDA using 16 MFCCs and 14 LPCs

Increasing the number of LPCs considered does not seem to provide a better discrimination between classes.

### 6.6.3 16MFCC+10DCTC

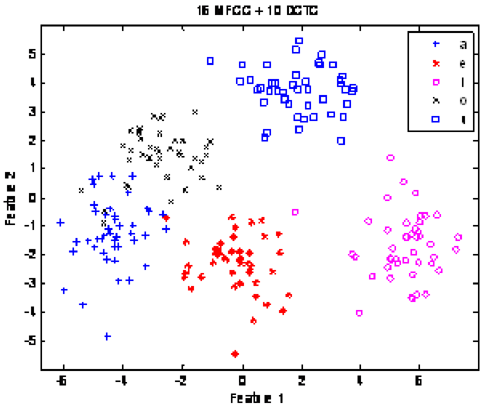

Figure 6.22: Dimensionality Reduction with LDA using 16 MFCCs and 10 DCTCs

The use of 10 Discrete Cosine Transform Coefficients, in addition to the 16 MFCCs does not bring improvements to the mapping technique.

### 6.6.4 Summary of Results

In this section, the results of using a mapping technique after adding different features to the initial set of parameters were shown. This approach did not show improvements when compared to the previous results (using simply the 16 MFCCs). Reasons for this behavior may rely in the fact that the additional parameters might not bring additional information to the MFCCs, concerning vowel separation. Alternatively, the additional information brought is somehow overlapped by the information contained in the MFCCs.

## 6.7 Hierarchical approaches based on vowel similarity

The use of hierarchical approaches arise naturally from the conclusions held in the previous chapter: the similarity observed between spectral shapes of some vowels suggests the use of these techniques.

It has been seen that vowels /a/ and /o/ have very similar spectral shapes. Additionally, vowel /e/ also resembles these spectra. Therefore, the approach described in Fig. 6.23 was followed. The main idea is to train linear mappings and bayesian classifiers for a three different stages of classification: the first regards separation between vowels /i/, /u/, and the remaining vowels. The second stage is applied to the vowels classified as belonging to the last group, and is trained to separate vowel /e/ from the remaining. The objects classified as belonging to this latter group are hence fed to a new classifier, that makes the final separation. As the figure shows, the separation obtained is fairly good, although a big dispersion between the samples of each class exists.

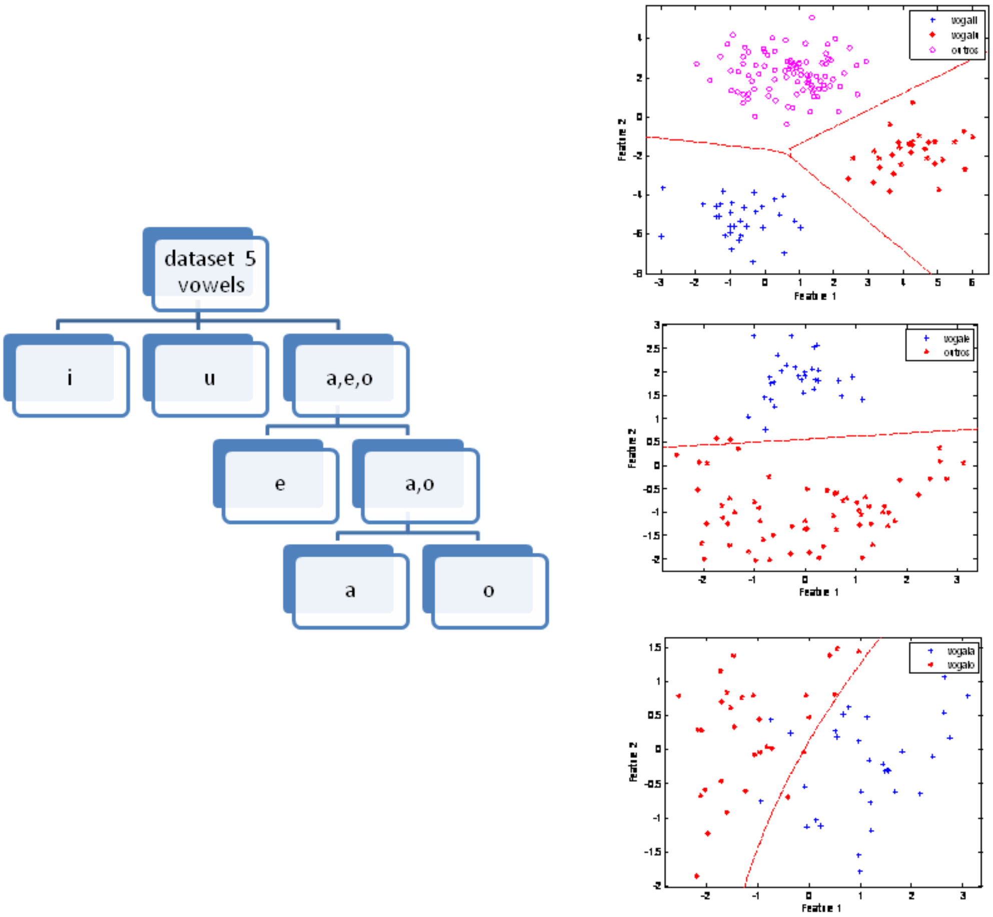

Figure 6.23: Hierarchical Approach 1

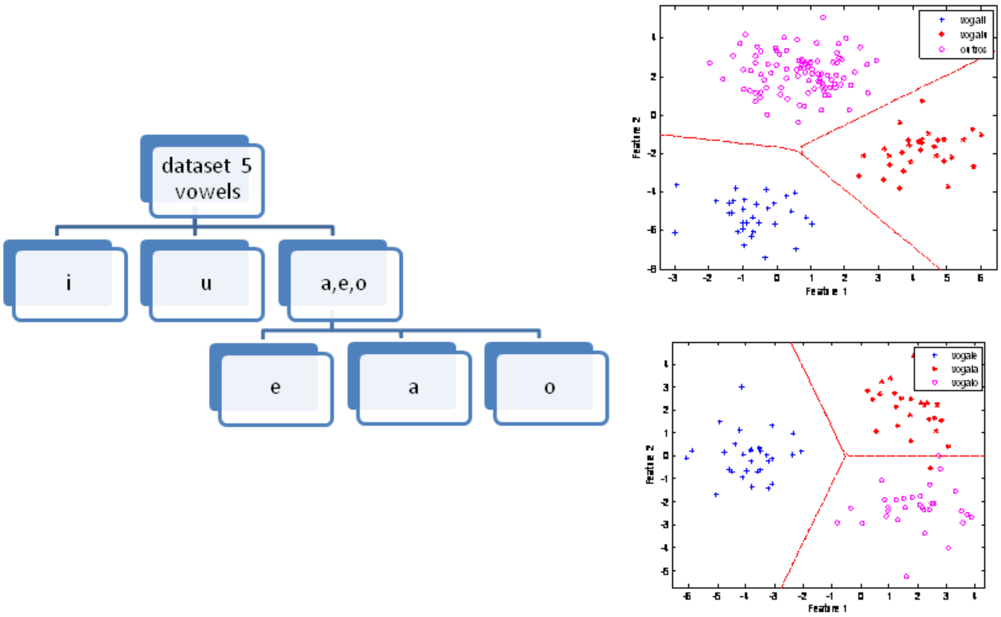

Figure 6.24: Hierarchical Approach 2

Fig. 6.24 shows a simplified approximation of the previous hierarchic approach, using only two stages. The dispersion of samples of each class is apparently smaller.

Although these techniques seem very logical approaches, and the simulation results yield good indications, a major problem concerning the application of such techniques arise: the propagation of errors, that can bring a very negative impact to the performance of such a classifier.

## 6.8 Hierarchical approaches based on linear / nonlinear mappings combination

The idea presented on the previous section, of using hierarchical approaches, has a number of possible variants that are worth exploring. In this section, these approaches are used by combining linear and nonlinear mapping techniques.

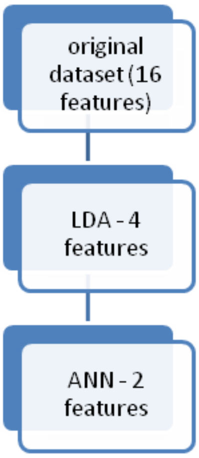

Figure 6.25: Hierarchical Approach 3 Scheme

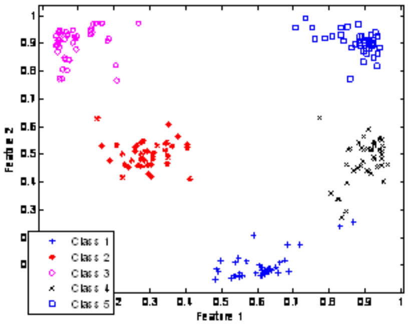

Figure 6.26: Hierarchical Approach 3

In Fig. 6.25 and 6.26 the idea of using LDA to achieve an initial mapping, of 4 dimensions, followed by the use of an ANN to achieve the final 2D mapping is presented. It can be seen that the dispersion between samples belonging to the same class is diminished, when comparing to the use of ANN by itself.

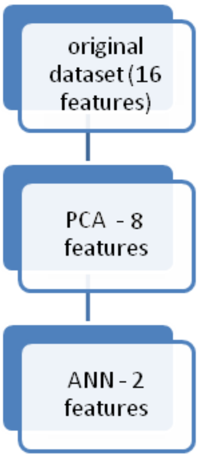

Figure 6.27: Hierarchical Approach 4 Scheme

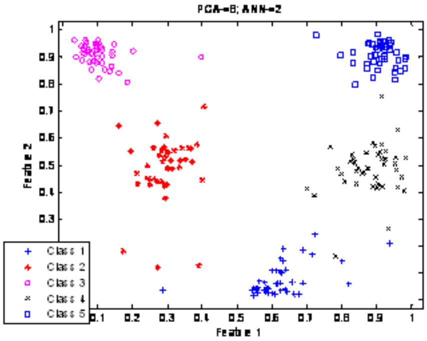

Figure 6.28: Hierarchical Approach 4

In Fig. 6.27 and 6.28 a similar technique is used, but now the linear mapping technique used is PCA. The result is slightly worst.

The results show a clear improvement when comparing to the use of an ANN with no previous mapping. The improvement of the clustering between samples of each class appears as a result of a simplification of the ANN task. In fact, by applying a linear technique preceding the ANN operation, the features fed to it are already the most discriminating ones. Hence, the remarkable ANN classification capabilities are merely concerned with finding a nonlinear mapping regarding this intermediate representation and a 2D final mapping, obtaining better results.

## 6.9 Hierarchical Approaches based on combination of different features

The previously referred results are very promising. However, we are considering a small set of samples. Although no more files are available, one can consider that, in the 400ms database, each frame has unique properties, that are being filtered as the average is computed - we are indeed making a pre-filtering step. Furthermore, in the real-time operation, the mapping will be done regarding frame. Therefore, at this point a new dataset was considered, by assuming that each frame is a new sample.

This new dataset was primarily used with the first technique presented in the previous section, so that the effect of using the frame dataset is analyzed.

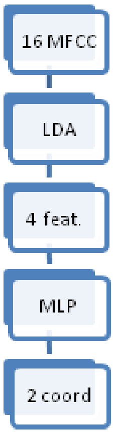

Figure 6.29: Hierarchical Approach 5 Scheme

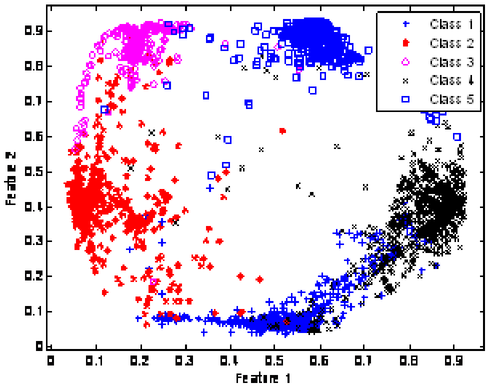

Figure 6.30: Hierarchical Approach 5

As expected, a much bigger dispersion is observed when considering a frame basis, as a pre-filtering step is lacking. However, this behavior resembles more the real-time expected behavior.

Subsequently, this new dataset was used to test new hierarchical approaches, that use linear techniques to obtain the more significant features derived by different sets of parameters. Subsequently, a mapping using linear/ANN, exploited in the previous section, is used to achieve the final mapping.

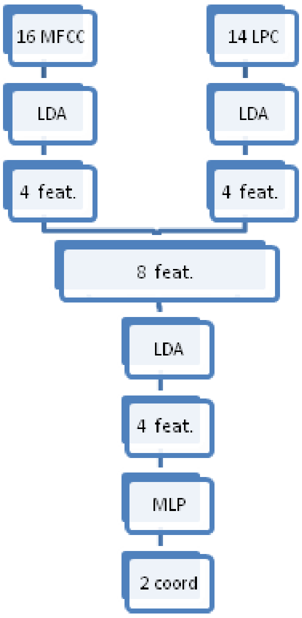

Figure 6.31: Hierarchical Approach 6 Scheme

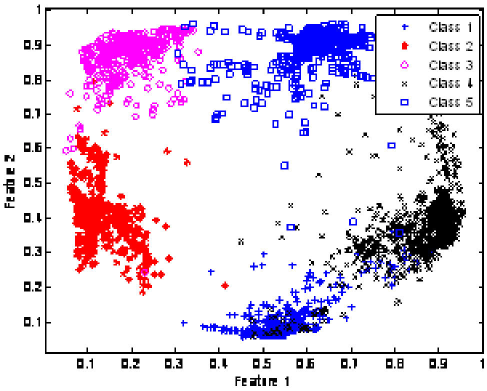

Figure 6.32: Hierarchical Approach 6

Fig. 6.31 and 6.32 regard the use of LPC features in addition to the MFCCs. Each of the two types of features are first mapped to a 4-dimensional space. Then, the 8 features are presented to a linear/nonlinear mapping technique. the resulting mapping shows, in general, an improved concentration of each class features. However, for vowel /u/ the dispersion has increased.

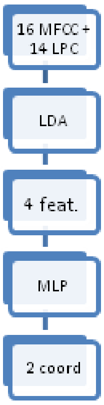

Figure 6.33: Hierarchical Approach 7 Scheme

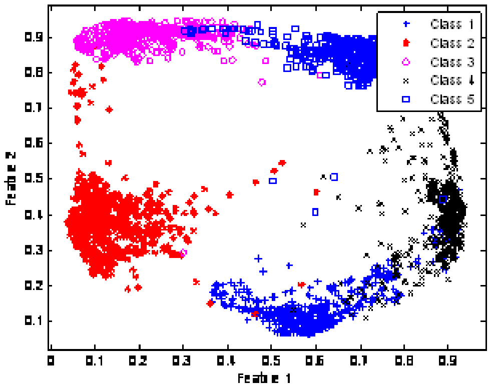

Figure 6.34: Hierarchical Approach 7

Fig. 6.33 and 6.34 regard the jointly map of LPCs and MFCCs. The dispersion is diminuished.

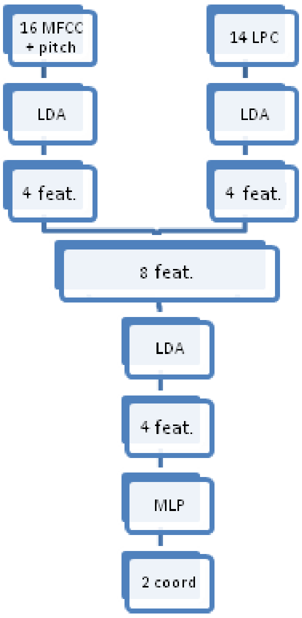

Figure 6.35: Hierarchical Approach 8 Scheme

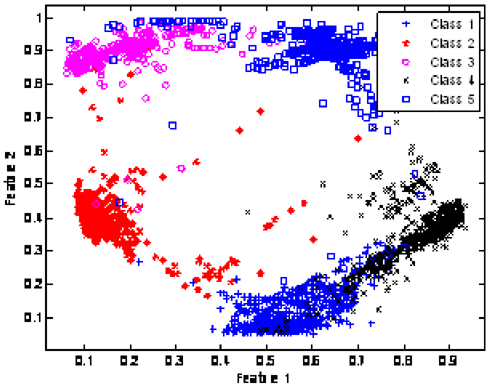

Figure 6.36: Hierarchical Approach 8

Fig. 6.35 and 6.36 regard the inclusion of pitch as a feature. As referred previously, the use of raw features may approximate these classifiers to the human auditory system behavior, considering the theory that his behavior is characterized by an analysis regarding different stages of perception. However, no major improvements were observed.

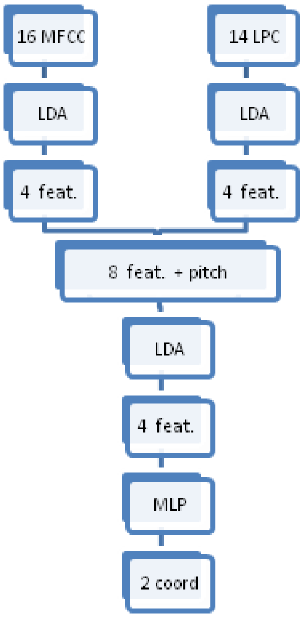

Figure 6.37: Hierarchical Approach 9 Scheme

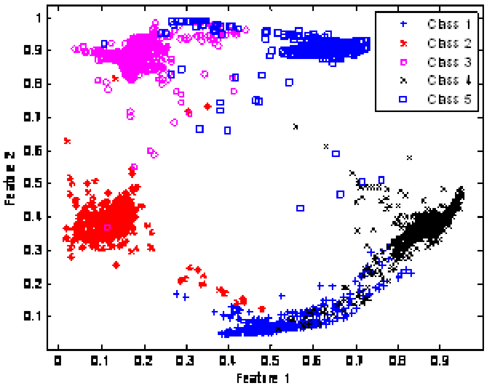

Figure 6.38: Hierarchical Approach 9

Fig. 6.37 and 6.38 regard the inclusion of pitch in a higher stage of the mapping. With this the inclusion of pitch as a discriminative feature is being forced. The performance has improved, as the dispersion within classes has diminished.

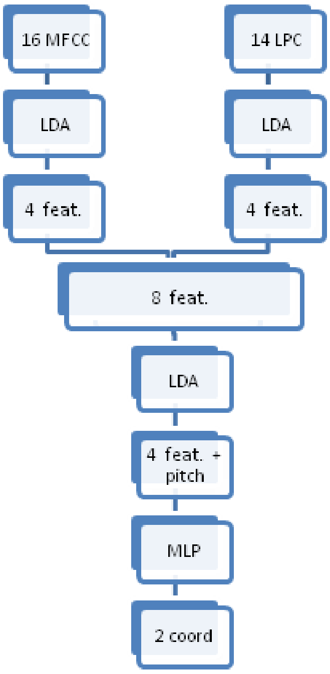

Figure 6.39: Hierarchical Approach 10 Scheme

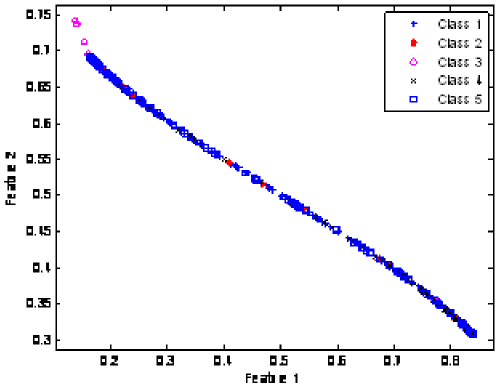

Figure 6.40: Hierarchical Approach 10

Fig. 6.39 and 6.40 regard an approach were pitch was introduced in a even more advanced stage of the mapping process: before the MLP operation. It is visible that the results are awful: the inclusion of this feature in such an advanced stage “confuses” the operation of the ANN.

## 6.10 Derivation of the final classifier scheme

### 6.10.1 Summary of the several tests

From the presented results, a set of conclusions important to the definition of the final classifier scheme have to be considered, together with the requirements of this specific work, namely the real-time operation mode. The main conclusions drawn are presented:

- Among the several vowels, it was proved that vowels /u/ and /i/ are more clearly separated from the rest, and hence allow a simpler classification task. Opposingly, vowels /a/ and /o/ have very similar spectral shape, and hence are very difficult to separate.

- It was shown that using more than 16 MFCCs does not provide a improvement on the performance of the classifier (in terms of the diminution of the error rate) comparable with the improvement of the computational complexity associated with the use of more features.

- It has been seen that, as expected, feature selection techniques do not yield good results, as one cannot consider a small group of signal-related features that completely capture the essential distinctive features of the several vowels. Therefore, there are clear advantages in using feature extraction techniques, as these techniques provide combinations of all the initial parameters to achieve the best distinctive features.

- Among the several nonlinear mapping techniques tested, ANN was the one that yield better results. The remaining nonlinear mapping techniques do not significantly improve the performance of the linear mapping techniques. Also, the local nonlinear techniques perform worse. This is an indication that the data is not arranged in a nonlinear way in the high-dimensional space.

- As for the hierarchical approaches, they have given good indications. Those based on vowel similarity seem logical approaches, however the occurrence of propagation errors has invalidated the pursue of such techniques. From the other hierarchical techniques analyzed, the use of ANN has proven its advantages, as did the inclusion of pitch as a feature in intermediate steps of mapping.

- Although the use of ANN yield good results, generalization issues [17] require the use of large training sets, which is not the case. In addition, these techniques are more complex to use in real-time operations. Therefore, for the final classifier specifications, this technique was dropped, and the dimensionality reduction techniques used are the linear ones: LDA and PCA. However, the results previously provided can be seen as indication regarding the advantages of using such techniques, if an appropriated training set is available.

- Regarding the features used, the behavior of MFCCs has proven to be good. No significant advantages were observed when LP features were added, although several techniques regarding the combination of the two groups of features were tested. Hence, the decision was to maintain 16 MFCCs. However, the results of hierarchical approaches using pitch in intermediate steps of the mappings, together with the knowledge that the separation in genders (that are usually classified according to its pitches) yields usually better results [5], the pitch was chosen as a feature. Also, it has been defended that the use of raw features together with more elaborated ones resembles the human auditory system behavior. Also, it is known that different vowels have different intrinsic pitches [9].

- Regarding the classifier, both bayesian and nearest neighbor classifiers were tested. Although the behavior of the nearest neighbor classifiers was acceptable, it is an algorithm that is highly dependent on the training data - in this case however, the training set is small. In addition, it has been seen that the use of these classifiers can bring difficulties to the real-time operation, as the computational complexity of such techniques tends to be big. Therefore the bayesian classifiers will be used. As for the initial problems regarding the quadratic classifier, it has been proved that the use of dimensionality reduction techniques prevents the occurrence of overfitting, so the use of QDC, together with LDC, will be pursued.

With these conclusions in mind, the final classifier will follow the basic pattern recognition flow, with the use of linear feature extraction techniques and bayesian classifiers. 4 scenarios were defined:

**Scenario 1** From the 16 MFCC coefficients existing in the parameter vector, we used LDA to achieve a mapping to a 4-dimensional subspace. To these 4 features we added the pitch. These 5 features were used by the bayesian classifiers to identify the current vowel.

**Scenario 2** From the 16 MFCC coefficients existing in the parameter vector, we used LDA to achieve a mapping to a 4-dimensional subspace. These 4 features were used by the bayesian classifiers to identify the current vowel.

**Scenario 3** From the 16 MFCC coefficients existing in the parameter vector, we used PCA to achieve a mapping to a 4-dimensional subspace. To these 4 features we added the pitch. These 5 features were used by the bayesian classifiers to identify the current vowel.

**Scenario 4** From the 16 MFCC coefficients existing in the parameter vector, we used PCA to achieve a mapping to a 4-dimensional subspace. These 4 features were used by the bayesian classifiers to identify the current vowel.

### 6.10.2 Number of mapped dimensions

The scenarios previously described regard the use of 4-mapped features to feed the classifiers. The choice was not random: for LDA, this is in fact the maximum number of dimensions allowed (corresponding to $c-1$, where $c$ is the number of classes - 5). Experiments regarding further reducing of the number of dimensions tend to yield worse results. For the PCA technique, experiments using different numbers of mapping dimensions have shown that there are no effective benefits in keeping more than 4 dimensions (see Fig. 6.41). In fact, although keeping more than 4 dimensions might result in a slightly smaller error rate, the corresponding increase in complexity of the problem is not justifiable.

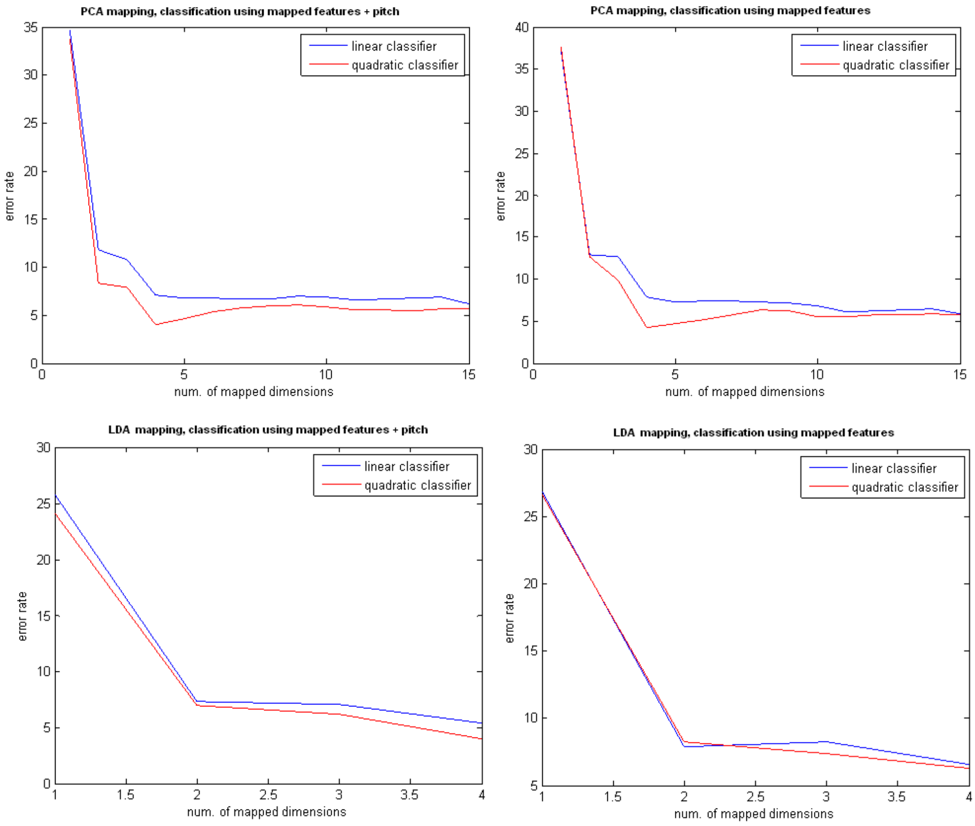

Figure 6.41: Effect of changing number of mapped dimensions

### 6.10.3 Simulation of the final classifier

The four approaches described previously were primarily evaluated using Matlab. In this environment, we created the parameter vectors corresponding to the several samples (a total of 5060 samples). These samples were separated in training and testing sets, by keeping 70% of the samples in the training set, and the remaining samples in the test set. The separation was made randomly, but careful to avoid that samples belonging to the same speaker appeared in both training and testing sets was taken. Also, the separation was made by using 70%-30% of each gender's samples in each of the sets. The final sets had 3645 samples for training, and 1495 for test. Next, both the mappings and the classifiers were trained and tested. This process was repeated 50 times. Table 6.4 presents the recognition rates obtained for the several approaches (corresponding to the average value of the 50 trials), for the linear and quadratic classifiers, respectively. It is visible the advantage of including pitch as a feature in the dataset, particularly for the quadratic classifier: the scenarios using this feature in addition to the MFCC-derived features yield better results. Also, and as expected, the results obtained with the quadratic classifier were superior, as this allows the definition of more complex discriminating functions.

Table 6.4: Comparison of recognition rates obtained

| **Method** | **Scenario 1** | **Scenario 2** | **Scenario 3** | **Scenario 4** |
|---|---|---|---|---|
| LDC | 94.59% | 93.40% | 92.92% | 92.11% |
| QDC | 96.01% | 93.75% | 95.97% | 95.67% |

The results are apparently very good, and are higher to those obtained with other applications, namely VATA. However, one must not forget that only 5 vowels are being used here, instead of the 10 regarding the results presented in [5].

## 6.11 Conclusions

In this chapter, the several experiments regarding the final definition of a classifier scheme are presented.

The final classifier follows the basic pattern classification scheme, and uses linear feature extraction methods. Although several nonlinear methods were tested, these did not outperform the linear ones, and were excluded. The only nonlinear technique that provided good results was the ANN. Furthermore, the use of such a technique in the proposed hierarchical schemes provides good indications regarding classification. However, the use of ANN was eventually excluded, as the number of training samples is not big enough to allow good generalization capabilities to this technique. Hence, linear techniques (PCA and LDA) are used.

Regarding the classifiers, the use of bayesian classifiers was pursued, as these are more suited to convert to real-time operation.

Four different variations of the final classifier were tested. The results of the simulation, show that the use of LDA is more advantageous than the use of PCA, which is somehow expected, because LDA is a supervised technique, and hence uses information regarding the labeling of the training set. Furthermore, the addition of pitch has also proven enhance the classification: the idea of using raw features in addition to higher-stage features as MFCCs seems to provide good results.

# Chapter 7

# Interactive Application Developed

## 7.1 Introduction

The advantages associated with the use of visual displays have been widely referred in the present report. The use of simple 2D vowel displays is an important aid to speech therapy sessions, improving the motivation of child patients. The use of interactive games provides extra motivation for children, by making the necessary training a pleasure instead of an obligation. Furthermore, interactive computer games are tools that allow the continuation of the training at home, after the speech therapy sessions. Therefore, in this work, a simple interactive game was developed, using OpenGL: a car race game.

In this chapter, the implementation of the classifier (derived in the previous chapter) in C++ will be addressed, together with some implementation details regarding the development of the final application. Also, the results of a preliminary experiment, regarding the efficiency of the classifier, using only one speaker, are presented.

## 7.2 Adaptation of the Classifier to Real-Time Performance

The first step in developing the interactive application is the adaptation of the classifier defined in the previous chapter to real-time operation. In this section, this adaptation process is briefly described.

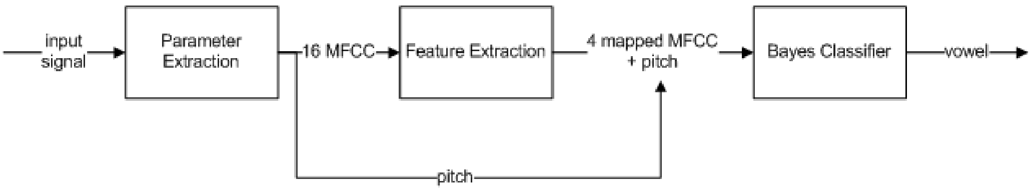

Figure 7.1: Diagram of the developed classifier

In Fig. 7.1, a diagram of the final classifier structure is presented. Hence, there are three major steps in such approach: parameter extraction, feature extraction, and classification.

The parameter extraction comprises the computation of both the MFCCs and the pitch from the input speech signal. Therefore, the first step is the development of an input interface, capable of capturing the speech signal obtained from the computer's audio input, and present it to the parameter extraction block as a vector of time samples. This interface was created using the RTAudio[1] API, that provides a set of classes capable simplifying the process of dealing with the computer's audio hardware to obtain realtime audio input. Using this set of functions, one can easily obtain in a buffer the several samples corresponding to the sound captured by the computer's audio input.

Finetuning the RTaudio functions to capture each of the input audio frames involves also the definition of a silence threshold, to avoid that the classifier attempts to classify noise. The next step comprises the parameter extraction. MFCCs are computed accordingly with the definition previously presented, using the outputs of a bank of mel-frequency spaced filters. As for the pitch, its computation is made using a cepstral peak analysis technique.

The following step comprises the dimensionality reduction. This step was highly simplified by the use of linear techniques. Therefore, for both LDA and PCA techniques, mean vectors $X_{mean}$ regarding the MFCCs values of the training data and Linear Transformation matrices $T$ can be computed in the training step, accordingly to the previously described techniques. Thus, at this phase, to obtain a low-dimensional representation $Y$ from the high-dimensional representation $X$, one has to simply compute the mapping, that is basically a simple matrix multiplication operation: $Y = (X - X_{mean}) \times T$.

Once the dimensionality reduction step is completed, one has to classify the resulting data. Once again, in the training step the necessary data for obtaining the discriminant functions is calculated. Therefore, in real-time operation, one has simply to calculate the values of the 5 discriminant func-

[1] http://www.music.mcgill.ca/~gary/rtaudio/

tions, and assign the input data to the class which discriminant function yield the bigger result.

A final post-processing step was added, in order to avoid that sounds captured that did not correspond to any of the considered vowels were classified. This post-processing step basically regards verification if the biggest value obtained by the discriminant functions is higher then a defined threshold.

## 7.3 Game Development

The final purpose of the work was the development of an interactive game controlled by vowels. The game selected was a car race game. The selection of such a game was made concerning both the simplicity of use (every child easily realizes the purpose and how to play the game), and the capability of integrating the 5 vowels as commands.

Hence, the game has, in this initial phase, a single circuit, as well as a single car (latter, more cars and circuits should be added to improve the ability of motivating in a prolonged way the child). Also, the surrounding environment developed is quite simple: although some elements for enrichment of the surrounding environment may be added, excessive ornaments may contribute to diminish the child concentration.

The basic framework of the OpenGL application is straightforward and will not be described. Suffice to say that this basic framework comprises functions responsible for generating the window, closing the window, resizing the window, etc. This framework was was developed in a previous work (regarding the real-time identification of pitch as the controller of a simple game).

The race track is defined by a vector with a set of coordinates corresponding to the limitations of the curve sectors of a track. Therefore, it is supposed that the track is constituted by an alternated sequence of straight segments and curves. The construction of the straight parts of the circuit is simple: one must design a set of polygons from the initial to the final coordinates. The design of the curve parts is not as straightforward. For this purpose, Bézier curves (a kind of parametric curves, recurrent in graphical computation) were used. These curves use some control points to define the several intermediate points of the curve. Cubic Bézier curves were used, as the definition of the 4 control points from the coordinates is very simple: the projection of the previous and following coordinates suffices (see Fig. 7.2).

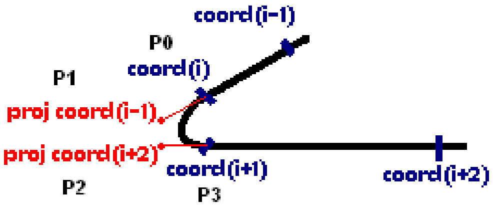

Figure 7.2: Cubic Bézier Curve

Each of the intermediate point is defined Equation 7.1.

$$B(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3, \qquad t \in [0,1] \quad (7.1)$$

At this point, one of the barriers of the circuit was defined for the entire circuit. For completing the track, a vector perpendicular to each point was calculated, using the properties of the scalar product, and the fact that the norm of the width of the track is constant. This vector is added to each point, to obtain the corresponding opposite side of the track.

As for the player, a 3DS (3D Studio Max) model was used. The use of such a model is related to the fact that this object requires the use of complex geometry to achieve a satisfactory appearance. Additional algorithms for collision detection were included to prevent the car from passing through the circuit boundaries.

The surrounding environment was created using a set of textures, in the form of a skybox.

As for visualization purposes, the camera was set slightly behind the player, to allow a “1st person shooter” type of visualization.

## 7.4 Characterization of the Game

In Figure 7.3, a screenshot of the developed application is shown.

The purpose of the game is to complete the circuit in the shortest amount of time.

The continuous utterance of each of the 5 vowels match to one of controls:

**Vowel /a/** allows a progressive increase of the car speed until a maximum speed is achieved.

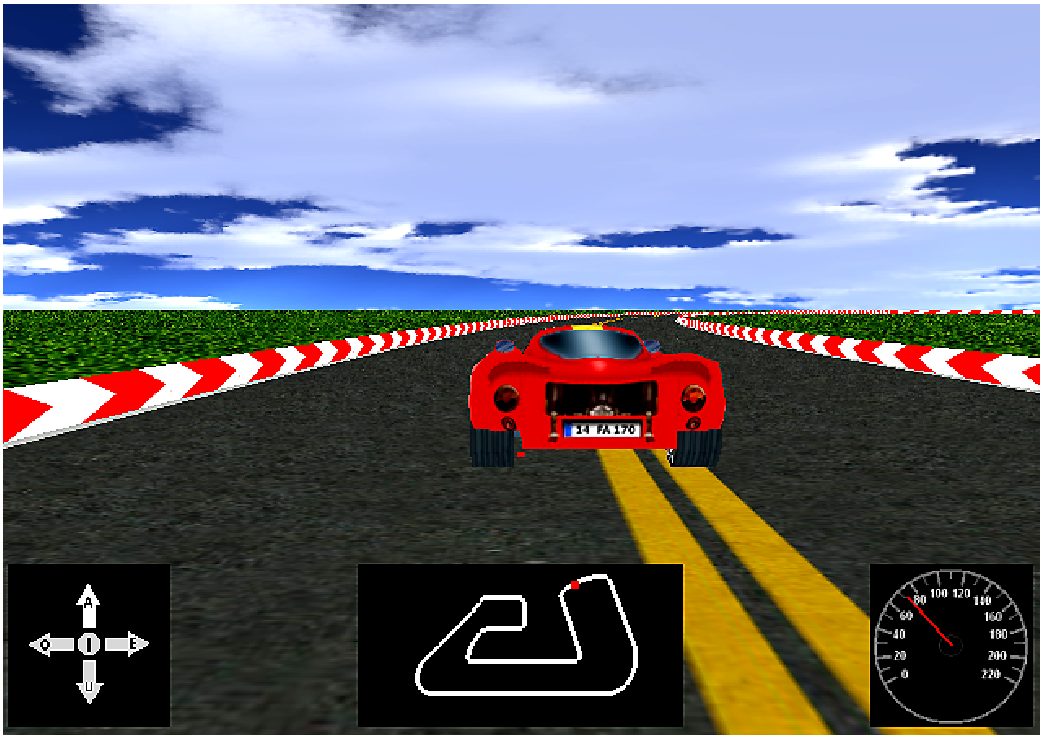

Figure 7.3: Screenshot of the developed Application

**Vowel /u/** allows driving the car in reverse gear, decreasing speed until the minimum value allowed is reached.

**Vowel /i/** allows slowing down the car till full stop.

**Vowels /o/** allows changing the direction of the car movement by turning left, without changing the speed

**Vowel /e/** allows changing the direction of the car movement by turning right, without changing the speed

To help the user understand the basic mechanics of the control of the car, three auxiliary elements were added to the visual display: a graphical representation of each command (showing the effect of each utterance, as well as the currently identified vowel) - Fig. 7.4, a small representation of the entire circuit where the current position of the car is emphasized - Fig. 7.5, and also a representation of the current car speed - Fig. 7.6.

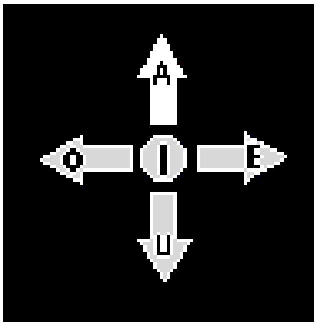

Figure 7.4: Representation of the Command auxiliary element

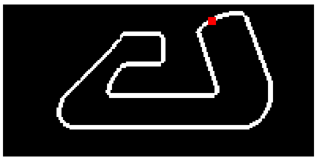

Figure 7.5: Representation of the Circuit auxiliary element

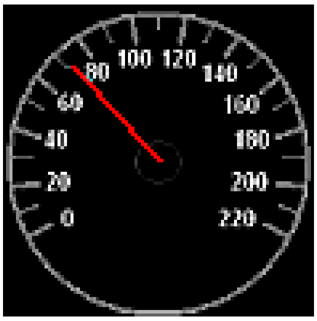

Figure 7.6: Representation of the Velocity auxiliary element

The game is intuitive, although in the first plays some confusion regarding the commands may exist. However, the corresponding auxiliary element is always represented to overcome this difficulty.

## 7.5 Real-Time Performance

The test results have shown that, after a few trials by a user, the game is easily controlled using only vowels, although some problems were identified in distinguish vowels /a/ and /o/. This problem is not unexpected, as these two vowels have very similar spectral envelopes. Also, some tests were made with speakers from different genres, and some fragilities were reveled when male speakers used the program. However, that is mainly due to the lack of sufficient representatives of this genre in the training set: as the purpose of this game is to serve children, we used mainly child speakers in the training database, as supported in [3].

For observing the real-time behavior of the developed classifier, series of 6 tests for each approach (from the 4 approaches mentioned in the previous chapter) were held, with one female speaker. In each test, the 5 vowels were uttered in sequence. The results were saved in a .txt file, and latter the middle samples (corresponding to a sustained vowel utterance) were selected to calculate the statistics. The recognition rates obtained are presented in Table 7.1. Nearly 3000 samples were obtained from each test: a total of 12000 samples were considered for calculating the statistics for the LDC classifier, and 11890 for the QDC classifier.

### 7.5.1 Comparison with Simulation Results

In Table 7.1, the results obtained in the several tests in real-time and the simulation results referred in the previous chapter are presented.

Table 7.1: Comparison of recognition rates obtained

| **Method** | **Scenario 1** | **Scenario 2** | **Scenario 3** | **Scenario 4** |
|---|---|---|---|---|
| LDC Real-Time | 93.93 % | 96.43 % | 89.37 % | 92.30 % |
| LDC Simulation | 94.59 % | 93.40 % | 92.92 % | 92.11 % |
| QDC Real-Time | 91.33 % | 97.08 % | 92.87 % | 91.04 % |
| QDC Simulation | 96.01 % | 93.75 % | 95.97 % | 95.67 % |

The results show that, contrarily to the simulation results, the behavior when using pitch as a feature is worse. Scenario 2 (using 16 MFCCs and LDA mapping) is the approach showing better results in real-time. Possible reasons for this fact may be the limitations in the real-time algorithm for detecting pitch.

Although an obvious degradation between the real-time results and the simulation results, the recognition rates are very promising. This is in fact a fairly good indication concerning the reliability of the linear mapping methods. However, it should be noted that these results were obtained only for one speaker, and thus they are merely indications of the performance of our classifier.

## 7.6 Conclusions

In this chapter, details regarding the implementation of the classifier in C++, together with the development of the final application, are presented.

A simple and attractive application, suited for children training, was developed, as intended. The application is fully controlled by the utterance of vowels.

Although only one speaker has been used for real-time testing, the results provide good indications regarding the performance of the final classifier, as they are not significantly different from the simulation results.

# Chapter 8

# Conclusion and Further Work

## 8.1 Conclusions

In this report, an approach to developing a real-time vowel classifier, suited for classifying 5 European Portuguese vowels is presented.

Initially, a spectral analysis of the 5 considered vowels was performed. The importance of the spectral shape in the definition of each vowel, together with the observation of the difficulties associated with identification of high-pitched vowels was observed. Also, it was noted that vowels /a/ and /o/ have spectral shapes very similar, what can compromise the classification of these two vowels (what was in fact verified in further tests).

Traditional techniques based on formant detection have proven to fail when the vowels are uttered at high pitch [3]. This is usually the case when children or women are the speakers. Therefore, a more robust method is needed. Spectral-shape features have been used with some success, for English Vowels, in applications like VATA[17] or OLTK[6]. In this work, the parametrization of vowels, using MFCCs was tested, and has proven to suit relatively well the vowel classification purpose.

Several approaches regarding pattern recognition techniques were tested, and the advantage of using feature extraction techniques for dimensionality reduction (opposing to feature selection techniques) was shown. Also, several linear and nonlinear feature extraction techniques were applied, and it was observed that only the ANN approach yields results that are comparable to the ones obtained by the linear techniques. Thus, the high-dimensional distribution of the data must be linear, and the use of complicated nonlinear techniques is discarded.

ANN have proven their classification capabilities by allowing the achievement of very interesting mappings. Furthermore, the use of hierarchical approaches combining the use of ANNs and linear techniques shown good indications. However, as the number of training samples is not big enough to allow the use of an ANN with good generalization capabilities, this approach was dropped.

As for the linear techniques, LDA has proven its advantage relatively to the PCA technique for this data. That is not unexpected, as the LDA technique is a supervised technique, hence regards the training sample labels when constructing the mapping.

A classifier suited for real-time operation was hence derived. The simulation tests using the selected technique showed promising results.

The real-time implementation of this technique was done, and the classifier outputs were used as commands for a simple car race game. The use of interactive games is a precious aid to speech therapy and language learning areas. The developed game provides a simple display that gives an insight on the technique.

The test results regarding the real-time operation of the classifier were computed for only a single female speaker (within the target group) and have shown good indications. However, further testings should be done to achieve a more reliable estimation of the performance of the classifier. Also, it has been noted that the classifier works better with female speakers than with male speakers. That is not strange, as the classifier was trained using mostly female and child speakers. Hence, the dependency of vowel classifiers on the gender of the speaker still relies. The use of a wider training set would certainly provide a more robust classifier.

Although the final result is fairly good, a final remark has to be done regarding the developed classifier. The technique used provides indications of its suitability for such problem, but the initial parameters used for classifying the signal are not capable of completely characterize the distinctive features between vowels. Therefore, the use of more significant features for these purpose would yield better results. However, such features have not yet been found.

## 8.2 Further Work

The essential goals of the present work have been met. However, regarding the initial considerations made about the requirements of visual displays, it is noted that important benefits can arise by adding some improvements. Following, some suggestions regarding this enhancement are presented.

- The developed classifier should be tested with more speakers.

- The use of a database comprising a increased amount of speakers can significantly enhance the results, and can further allow the use of ANN, or one of the proposed hierarchical approaches in real time operation.

- The improvement of the game scenario, by adding more elements to it and the improvement of the challenge of the game, by adding other cars to the car race, and increasing the number of tracks can improve the motivation associated with the gameplay.

- The use of different parametrizations of the input speech signals can provide better results. Although MFCCs have proved to perform well, more investigation regarding the identification of the features used by the human auditory system must be done. This identification would allow a major improvement in speech recognition tasks.

# Bibliography

[1] A. M. Kondoz, *Digital Speech Coding for low bit rate Communication Systems*. John Wiley and Sons, 1994.

[2] L. J. P. van der Maaten, “An introduction to dimensionality reduction using matlab,” tech. rep., Universiteit Maastricht, 2007.

[3] A. J. S. Ferreira, “Static features in real-time recognition of isolated vowels at high pitch,” *Journal of Acoustical Society of America*, vol. 122, pp. 2389–2404, 2007.

[4] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[5] S. A. Zahorian, A. M. Zimmer, and F. Meng, “Vowel classification for computer-based visual feedback for speech training for the hearing impaired,” *International Conference on Spoken Language Processing*, vol. 78, pp. 973–976, 2002.

[6] A. Hatzis, *Optical Logo-Therapy (OLT): Computer-Based Audio-Visual Feedback Using Interactive Visual Displays for Speech Training*. PhD thesis, University of Sheffield, 1999.

[7] S. A. Zahorian, “Colour display of vowels as a speech articulation training aid,” in *Proceedings of the Annual International Conference of the IEEE on Engineering in Medicine and Biology Society*, vol. 4, pp. 1539–1540, 1988.

[8] L. R. Rabiner and R. W. Schafer, “Introduction to digital speech processing,” *Foundations and Trends in Signal Processing*, vol. 1, 2007.

[9] I. Guimarães, *A Ciência e a Arte da Voz Humana*. Escola Superior de Saúde de Alcoitão, 2007.

[10] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*. Prentice Hall, 1978.

[11] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition is continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, pp. 357–366, 1980.

[12] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. Wiley Interscience, 2001.

[13] V. de Silva and J. Tenenbaum, "Global versus local methods in nonlinear dimensionality reduction," 2003.

[14] A. J. S. Ferreira, "New signal features for robust identification of isolated vowels," in *InterSpeech 2005*, 2005.

[15] S. A. Zahorian and S. Venkat, "Vowel articulation training aid for the deaf," *Acoustics, Speech and Signal Processing*, vol. 2, pp. 1121–1124, 1990.

[16] S. Auberg, "Speech feature computation for visual speech articulation training," Master's thesis, Old Dominion University, 1996.

[17] A. M. Zimmer, "Vata: An improved personal computer based vowel articulation training aid," Master's thesis, Old Dominion University, 2002.

[18] A. M. de Lima Araújo, *Jogos Computacionais Fonoarticulatórios para Crianças com Deficiência Auditiva*. PhD thesis, Universidade Estadual de Campinas, 2000.

[19] X. Wang, *Feature Extraction and Dimensionality Reduction in Pattern Recognition and their Application in Speech Recognition*. PhD thesis, Griffith University, 2002.

[20] X. Wang and K. K. Paliwal, "Feature extraction and dimensionality reduction algorithms and their applications in vowel recognition," *Pattern Recognition*, vol. 36, pp. 2429–2439, 2003.

[21] M. Carvalho, H. Gonçalves, and A. Campilho, "Vowel recognition in european portuguese," in *RecPad 2007 - 13ª Conferência de Reconhecimento de Padrões*, 2007.

[22] R. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, and D. Tax, "Prtools4, a matlab toolbox for pattern recognition," 2004.

# Appendix A

# Linear Predictive Coding

## A.1 Derivation of LPC Analysis Equation: The Least Mean Square Approach

The derivation of the LPC analysis equation can be made using different approaches. The most common one is the Least Mean Square approach. There are other formulations, such as the Maximum Likelihood method. In the present work, only the first approach will be presented, as “this approach leads to a set of linear equations that can be efficiently solved to obtain the predictor parameters” [10].

It was already mentioned that the purpose of LPC analysis is the determination of the parameters $a_j$. In the following text, the estimates of this parameters will be represented by $\alpha_j$. Therefore, the estimation of the current sample is given by Equation A.1. The residual or error can be easily calculated as the difference between A.1 and 2.8, as shown in Equation A.2.

$$s(n) = \sum_{j=1}^{p} \alpha_j s(n-j) \tag{A.1}$$

$$e(n) = s(n) - \ s(n) = s(n) - \sum_{j=1}^{p} \alpha_j s(n-j) \tag{A.2}$$

The Least Mean Square approaches states that the best estimates $\alpha_j$ are obtained when minimizing the mean squared error, $E$, represented in Equation A.3;

$$E = E\{e^2(n)\} = E\{[s(n) - \sum_{j=1}^{p} \alpha_j s(n-j)]^2\} \tag{A.3}$$

To solve Equation A.3, one must set the partial derivatives of $E$ with respect to $\alpha_j, j = 1, \ldots, p$ to zero (Equation A.4). This result shows that $e(n)$ is orthogonal to $s(n-i)$ for $i = 1, \ldots, p$.

$$\frac{\partial E}{\partial \alpha_i} = 0 \Longrightarrow E\{[s(n) - \sum_{j=1}^{p} \alpha_j s(n-j)]s(n-i)\}, i = 1, \ldots, p \tag{A.4}$$

Defining $\phi_n(i,j) = Es(n-i)s(n-j)$, Equation A.4 can be rearranged (Equation A.5). The derivation of this equation in made with the assumption of stationarity of the signal, which, in a speech signal, can be assumed to be true for short segments. In consequence, the expectations ($E\{\}$) can be replaced by finite summations (Equation A.6).

$$\sum_{j=1}^{p} \alpha_j \phi_n(i,j) = \phi_n(i,0), i = 1, \ldots, p \tag{A.5}$$

$$\phi_n(i,j) = Es(n-i)s(n-j) = \sum_{m} s_n(m-i)s_n(m-j), i = 1, \ldots, p, j = 1, \ldots, p \tag{A.6}$$

## A.2 Solutions for the LPC analysis problem

After formulating the LPC problem, it is important to clarify the methods used to solve the problem, i.e., how does one gets the desired estimations $\alpha_j$? There are several approaches to this problem. In [10], seven essentially equivalent formulations are referred:

1. Autocorrelation Method
2. Covariance Method
3. Lattice Method
4. Inverse Filter Formulation
5. Spectral Estimation Formulation
6. Maximum Likelihood Formulation
7. Inner Product Formulation

The first two approaches will be briefly described in the following subsections.

### Autocorrelation Method

In this method, $s_n(m)$ is assumed to be zero outside the interval considered, $0 \leq m \leq N-1$. Therefore, for $m < 0$ and $m > N-1+p$, as the sample values are assumed to be zero (what is, in fact, not true), and thus there is no prediction error for this areas. However, in the region from $m=0$ to $m=p-1$ the prediction is made using samples that were assumed to be zero (although they in fact aren't zero). This generates potentially large prediction errors in the beginning of each frame. The end of each frame is another place where potentially large prediction errors can occur, as from $m=N-1$ to $m=N-1+p$ we are forcing the prediction of zero-valued samples. These effects are more prominent in voiced speech samples, when the beginning of the pitch period is coincident with the beginning or end of the frame. The use of a window that decreases the signal near the end-points of the frame can minimize these errors.

As the prediction error is being forced to zero outside the interval $0 \leq m \leq N-1+p$, the calculation of the mean-squared error can be reduced to this interval, as stated in A.7. We can also rearrange equation A.6 to equation A.8.

$$E_n = \sum_{m=0}^{N-1+p} e_n^2(m) \tag{A.7}$$

$$\phi_n(i,j) = \sum_{m=0}^{N-1+p} s_n(m-i)s_n(m-j), i=1,\ldots,p, j=1,\ldots,p \tag{A.8}$$

That can be rewritten as Equation A.14, as $s_n(m)$ is zero outside $0 \leq m \leq N-1$.

$$\phi_n(i,j) = \sum_{m=0}^{N-1-(i-j)} s_n(m)s_n(m+i-j), i=1,\ldots,p, j=1,\ldots,p \tag{A.9}$$

This expression is simply the short-time autocorrelation function (Equation A.10).

$$\phi_n(i,j) = R_n(|i-j|), i=1,\ldots,p, j=1,\ldots,p \tag{A.10}$$

Equation A.5 can thus be rewritten as Equation A.11, or, in the matrix form in A.12.

$$\sum_{j=1}^{p} \alpha_j R_n(|i-j|) = R_n(i), 1 \leq i \leq p \tag{A.11}$$

$$\mathbf{X} = \begin{bmatrix} R_n(0) & R_n(1) & \dots & R_n(p-1) \\ R_n(1) & \vdots & \ddots & R_n(p-2) \\ \vdots & \vdots & \ddots & \dots \\ R_n(p-1) & \vdots & \ddots & R_n(0) \end{bmatrix} \times \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_p \end{bmatrix} = \begin{bmatrix} R_n(1) \\ R_n(2) \\ \vdots \\ R_n(p) \end{bmatrix} \tag{A.12}$$

Although this equation can be solved by simply inverting the matrix, this approach usually leads to propagation of errors in computer operations. On the other hand, the matrix can be easily identified as a Toeplitz Matrix(a symmetrical matrix in which all elements along a given diagonal are equal), hence, recursive methods, such as the Durbin Algorithm, can be used to compute the estimates.

### Covariance Method

The Covariance Method is an alternative approach, that is based in the assumption that the interval over which the mean squared error is calculated is fixed: $0 \leq m \leq N-1$, therefore, there is no need for defining windows - the speech signal can be used directly (Equation A.13).

$$E_n = \sum_{m=0}^{N-1} e_n^2(m) \tag{A.13}$$

Hence, we can define $\phi_n(i,j)$ as follows.

$$\phi_n(i,j) = \sum_{m=0}^{N-1} s_n(m-i)s_n(m-j), \qquad i = 1,\dots,p, \qquad j = 1,\dots,p \tag{A.14}$$

Redefining the limits of the summation, yields the following equation.

$$\phi_n(i,j) = \sum_{m=-i}^{N-1-i} s_n(m)s_n(m+i-j), i = 1,\dots,p, j = 1,\dots,p \tag{A.15}$$

That is slightly different form the equation derived in the Autocorrelation Method, in the limits of the summation. The matrix model of LPC analysis equations derived by this method is presented in Equation A.16

$$\mathbf{X} = \begin{bmatrix} \phi_n(1,1) & \phi_n(1) & \dots & \phi_n(1,p) \\ \phi_n(2,1) & \vdots & \ddots & \phi_n(2,p) \\ \vdots & \vdots & \ddots & \dots \\ \phi_n(p,1) & \vdots & \ddots & \phi_n(p,p) \end{bmatrix} \times \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_p \end{bmatrix} = \begin{bmatrix} \phi_n(1,0) \\ \phi_n(2,0) \\ \vdots \\ \phi_n(p,0) \end{bmatrix} \tag{A.16}$$

This matrix is not a Toeplitz matrix, thus matrix inversion solutions have to be used, i.e. Cholesky Decomposition.

# Appendix B

# Dimensionality reduction techniques

The several nonlinear feature extraction techniques were tested, and a visual evaluation of their performance was made. It was concluded that the nonlinear techniques do not perform well in mapping the vowel data, specially the local nonlinear techniques. In this appendix, some of the additional scatterplots obtained are presented.

## B.1 Global Nonlinear Techniques

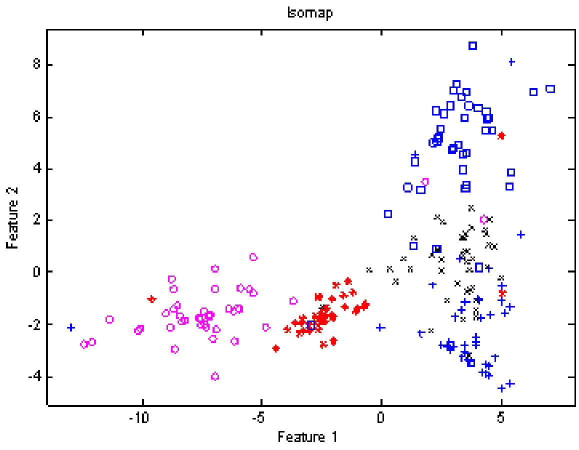

Figure B.1: Isomap

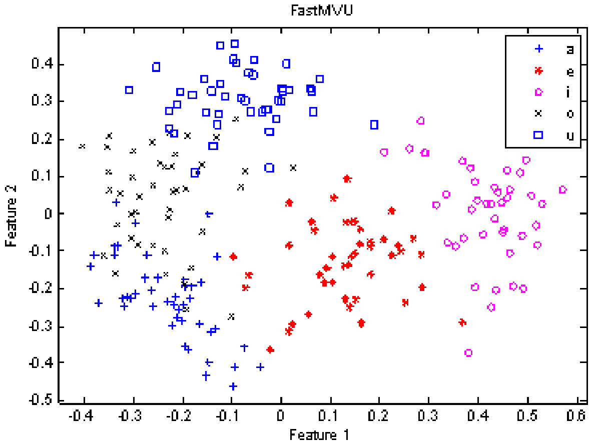

Figure B.2: Fast MVU

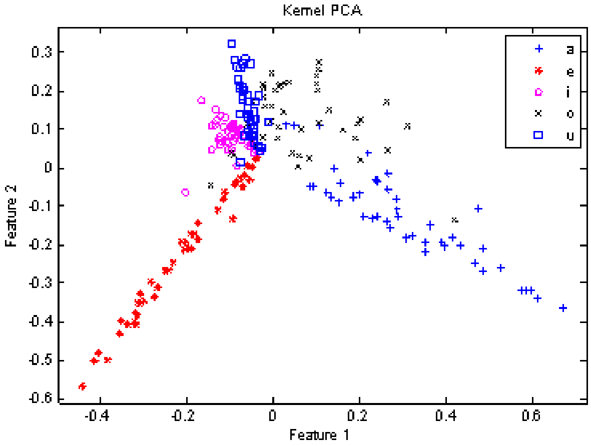

Figure B.3: Kernel PCA using gaussian Kernel functions

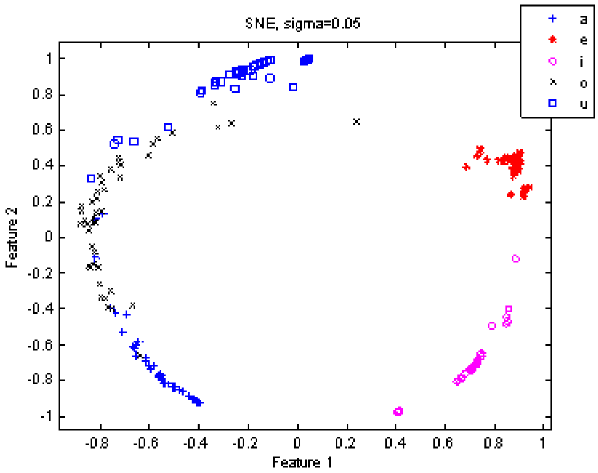

Figure B.4: SNE with small sigma

## B.2 Local nonlinear Techniques

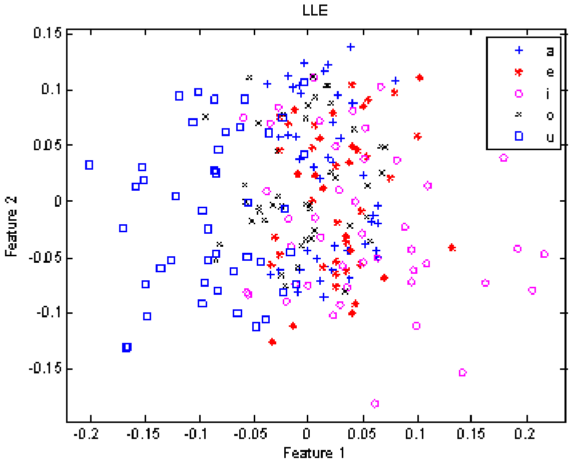

Figure B.5: LLE

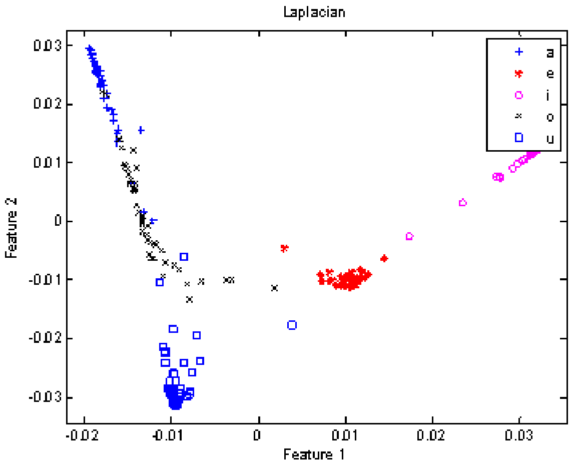

Figure B.6: Laplacian Eigenmaps

## B.3 Extensions and variants of local nonlinear techniques

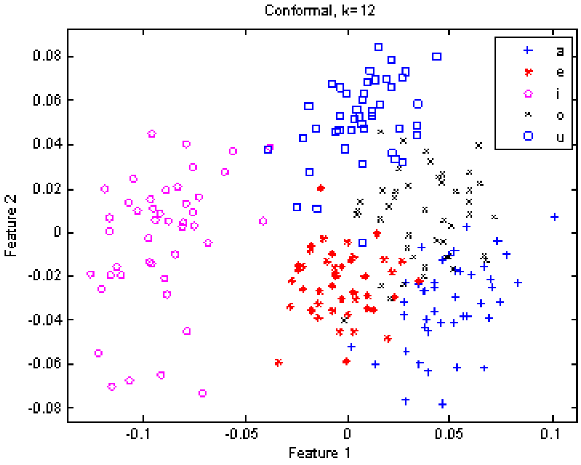

Figure B.7: CCA

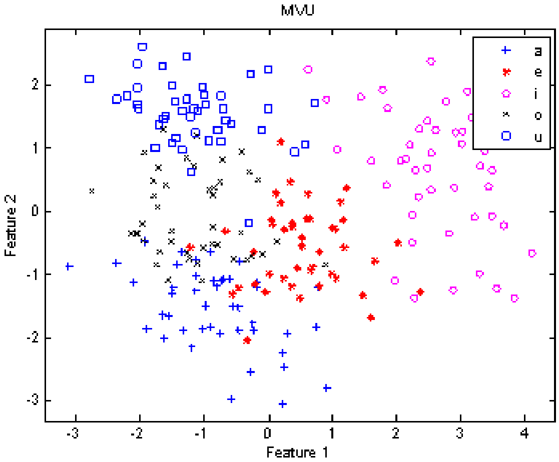

Figure B.8: MVU

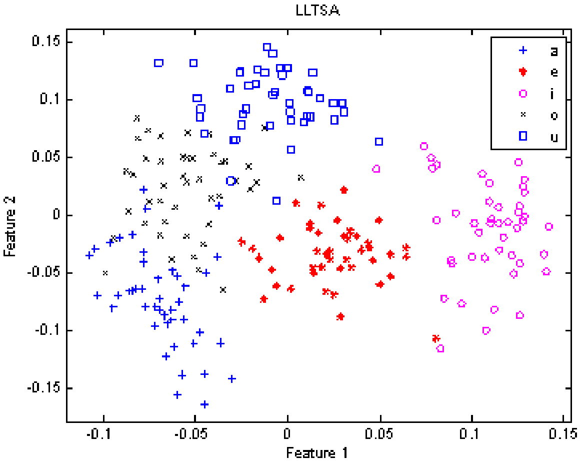

Figure B.9: LLTSA

## B.4 Global Alignment of Linear Models

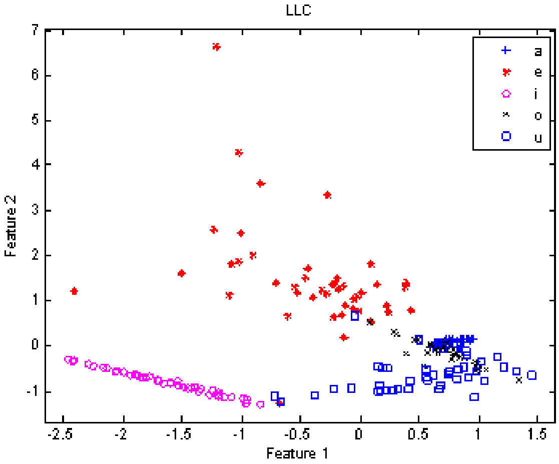

Figure B.10: LLC