Universidade de Aveiro Departamento de Física,
2015

Luís Manuel Monteiro de Sousa Cruz

Aplicação de Técnicas de Classificação em Séries Temporais

Luís Manuel Monteiro de Sousa Cruz

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Física, realizada sob a orientação científica de Fernão Rodrigues Vístulo de Abreu, Professor Auxiliar do Departamento de Física da Universidade de Aveiro

o júri / the jury

presidente

Prof. Dr. João Filipe Calapez de Albuquerque Veloso
Professor Auxiliar do Departamento de Física da Universidade de Aveiro

vogais

Prof. Dr. Fernão Rodrigues Vístulo de Abreu
Professor Auxiliar do Departamento de Física da Universidade de Aveiro

Prof. Dr. Iveta Pimentel
Professora Associada do Departamento de Física da Faculdade de Ciências da Universidade de Lisboa

agradecimentos / acknowledgements

Eu gostaria de agradecer a todas as pessoas que contribuíram para a realização desta dissertação. Em particular, gostaria de agradecer ao meu orientador, o professor Fernão Abreu, por todo o apoio oferecido. Gostaria de agradecer também à professora Filipa Lã pela sua disponibilidade para me dar dicas importantes para a compreensão da componente fonética. Por fim, agradeço a todas as pessoas que se disponibilizaram para fornecer dados de áudio.

palavras-chave processamento de sinal, reconhecimento da fala, codificação preditiva linear, cepstrum, distorção temporal dinâmica, modelos de Markov ocultos, máquinas de vetores de suporte, florestas aleatórias

Resumo Hoje, há um interesse crescente pela aprendizagem de uma segunda língua, quer seja por razões profissionais ou pessoais. Esta é uma tendência que se vai afirmando num mundo cada vez mais interconectado. Por outro lado, a democratização das tecnologias computacionais torna possível pensar em desenvolver novas técnicas de ensino de línguas mais automatizadas e personalizadas. Esta dissertação teve como objetivo estudar e implementar um conjunto de técnicas de processamento de sinal e de classificação de séries temporais úteis para o desenvolvimento de metodologias do ensino oral com feedback automático. São apresentados resultados preliminares sobre a prestação destas técnicas, e avaliada a viabilidade deste tipo de abordagem.

keywords signal processing, speech recognition, linear predictive coding, cepstrum, dynamic time warping, hidden Markov models, support vector machines, random forests

Abstract Today, there is a growing interest in learning a second language , either for professional or personal reasons. This is a trend that tends to hold in a increasingly interconnected world. On the other hand , the democratization of computer technologys makes it possible to think about developing new more automated and personalized language teaching techniques. This work aimed to study and implement a set of useful signal processing and time series classification techniques to develop methodologies of oral teaching with automatic feedback. Preliminary results on the provision of these techniques are shown, and assessed the feasibility of this approach.

Índice

Índice i

Lista de Figuras iii

Lista de Tabelas vii

1 Introdução 1

2 Conceitos Teóricos da Análise da Fala 3
2.1 Amostragem e Efeito de Aliasing . . . 4
2.2 Análise de Fourier de Tempo Curto . . . 5
2.3 Filtro Pré-Ênfase . . . 6
2.4 Funções de Janelas . . . 6
2.5 Extração de Características . . . 8
2.5.1 Codificação Preditiva Linear . . . 8
2.5.2 Mel-Cepstrum . . . 11
2.5.3 Coeficientes Cepstral derivados da CPL . . . 15
2.5.4 Lifter . . . 15
2.5.5 Coeficientes Dinâmicos: Delta e Aceleração . . . 16
2.6 Fonética . . . 16

3 Algoritmos de Classificação Dinâmicos 19
3.1 Distorção Temporal Dinâmica . . . 19
3.1.1 Programação Dinâmica Para Cálculo da Distorção Total . . . 21
3.2 Modelos de Markov . . . 22
3.2.1 Modelos de Markov Ocultos . . . 22
Escalamento . . . 29

4 Algoritmos de Classificação Estáticos 31
4.1 Máquinas de Vetores de Suporte . . . 31
4.1.1 MVS de Classificação Binária . . . 31
MVSs de Margens Suaves . . . 34
MVSs Não Lineares . . . 34
4.1.2 MVS Multi-classe . . . 36
4.1.3 MVS de Uma Classe . . . 36
4.2 Florestas Aleatórias . . . 37

4.2.1 Construção de Árvores de Decisão 38
4.2.2 Construção da Floresta Aleatória 39

5 Simulações: Avaliação do Desempenho do Utilizador 41
5.1 Descrição dos testes realizados 41
5.2 Processamento de Sinal 42
5.3 Aplicação das Técnicas de Classificação 42
5.3.1 Aplicação da Distorção Temporal Dinâmica 42
5.3.2 Aplicação das Máquinas de Vetores de Suporte 43
5.3.3 Aplicação das Florestas Aleatórias 45
5.3.4 Aplicação dos Modelos de Markov Ocultos 46
5.4 Discussão de Resultados 47

6 Conclusões e Trabalho Futuro 49

Referências 51

Lista de Figuras

1.1 Diagrama das etapas que compõem um algoritmo de auxílio à aprendizagem de línguas. Inicialmente é necessária a construção de uma base de dados constituída por exercícios padrão. De seguida existe o processamento de sinal que permite traduzir os sinais da fala, num conjunto reduzido de parâmetros. Na imagem os parâmetros são indicados por meio de sequências de retângulos coloridos, que vão indicar de uma forma mais clara os padrões fonéticos dos sinais da fala. Por fim, utiliza-se algoritmos que permitam realizar uma comparação dos padrões fonéticos, de forma a avaliar o desempenho fonético do utilizador. 2

2.1 Na imagem a) mostra-se uma representação simplificada do trato vocal. O trato vocal funciona como uma cavidade ressonante. As regiões que se destacam no espetro de frequência dos sinais correspondem às frequências ressonantes, designadas de formantes. Na imagem b) está representado um espetro típico de uma vogal em que são visíveis três formantes. 3

2.2 Representação do processo de amostragem. 4

2.3 Representação do processo de amostragem de um sinal contínuo com duas frequências de amostragem diferentes. Na parte superior da imagem está ilustrado o sinal contínuo composto por duas frequências (250 e 500 Hz). Na parte inferior esquerda, o sinal foi amostrado com uma $F_s = 750$ Hz. Neste caso as duas frequências são sobrepostas devido ao efeito de *aliasing*. Na imagem inferior direita, o sinal foi amostrado com uma $F_s = 1500$ Hz, ou seja, as frequências são conservadas. 4

2.4 Efeito de *aliasing* no domínio da frequência. 5

2.5 Ilustração de um espetrograma da palavra "Aveiro". 6

2.6 Resposta do filtro pré-ênfase no domínio da frequência, com $\alpha = 0.97$ e frequência de amostragem $F_s = 44100Hz$. 6

2.7 Ilustração do vazamento espetral. A imagem a) é o gráfico de um sinal com uma frequência de 50 Hz, cuja transforma de fourier (normalizada) está ilustrada na imagem b). A imagem c) representa o sinal resultante do corte aplicado ao sinal exposto em a). Esse corte está representado com cor laranja. Através da imagem c) pode-se visualizar que o corte vai gerar descontinuidades, que por sua vez vão contribuir para o vazamento espetral (V.E), como se pode observar na imagem d). 7

2.8 Ilustração da janela Hamming. 8

2.9 Representação de um modelo fonte-filtro simples para a geração de sinais da fala. 8

2.10 Separação filtro-fonte via *cepstrum*. Na imagem a) está apresentado um *frame* de sinal com uma janela *hamming* aplicada e na imagem b) está o seu espetro. Na imagem c) foi aplicado um *lifter* (filtro), representado a laranja, para extrair a componente do filtro $h(n)$. De seguida é aplicada a transformação inversa do *cepstrum* e obtém-se a resposta do filtro $h(n)$ na frequência, representado a laranja na imagem d). 12

2.11 Relação entra a escala de Mel e a escala de frequências linear. 13

2.12 Representação de 12 filtros triangulares. Cada filtro está representado com uma cor diferente. 14

3.1 Exemplo de uma distorção dinâmica. Na imagem a) estão representadas duas séries temporais S_x e S_y, constituídas por símbolos discretos. Para além disso, estão representadas as funções de mapeamento ϕ_x e ϕ_y que foram utilizadas para a normalização temporal ótima, segunda a métrica representada na imagem b). Na imagem c), o caminho ótimo (CO) está representado numa grelha por uma linha azul. 20

3.2 Exemplo de um modelo de Markov com três estados, com as respetivas probabilidades de transição. 23

3.3 Descrição de sinais audio em termos de modelos de Markvov ocultos. Nestes modelos, é considerada uma direção pois os fonemas estão encadeados temporalmente. Cada estado representa um fonema (indicado por /$\mathbf{f_n}$/) dando origem a diferentes vetores de características, que caracterizam o som pronunciado. . 24

4.1 Representação de um espaço de decisão a partir de vetores de entrada em $\Re^1$, com duas classes diferentes (azul representa a classe -1 e vermelho representa a classe 1). Como os vetores de entrada pertencem a $\Re^1$, o hiperplano é reduzido a um ponto (representado a verde). A reta preta representa a função de decisão. 32

4.2 Nesta figura está representado o processo de separação dos dados de forma linear, após um mapeamento dos mesmos num espaço de maior dimensão. Na imagem a) estão representados os vetores de entrada $\Re^1$, pertencentes a duas classes diferentes (azul representa a classe 1 e vermelho representa a classe -1). Estes vetores são mapeados num espaço $\Re^2$, através da função $\Phi(x) = [x, (x-2.5)^2]$, como se pode observar pela imagem b). Após este mapeamento, já é possível separar os dados linearmente como se pode observar na imagem c). Através da imagem c), observa-se que o hiperplano está a reduzido a uma reta (representada a verde). A função de decisão está definida em $\Re^3$ e é representada pelo plano de cor preta. 35

4.3 Representação de uma árvore de decisão arbitrária. Neste exemplo, o espaço de entrada X é constituído por duas variáveis, X_1 e X_2. A classificação é realizada a partir de dois pontos de divisão arbitrários, P_{d1} e P_{d2}. Neste caso, as folhas (classes) estão representadas por cores: azul, verde e vermelho. . . . 38

5.1 Representação da análise espetral realizada para a extração das características das séries temporais. 42

5.2 Ilustração do processo de classificação utilizando a DTD como métrica discriminante. É calculada a distância média $\langle D_c \rangle$ entre um exercício de teste e os exercícios de treino das várias classes e atribuída a classe à menor distância média. 43
5.3 Ilustração do processo da extração de vetores de entrada. Neste exemplo, a série está dividida em cinco partes iguais. Em cada uma das partes foi extraído um vetor de características para a construção de cada vetor de entrada. Note-se que o número de partes em que se divide a série temporal não é sempre igual ao número de vetores de entrada. 44
5.4 Ilustração da estrutura utilizada na construção dos modelos de Markvov ocultos. Nestes modelos, é considerada uma direção pois os fonemas estão encadeados temporalmente. Cada estado representa um fonema (indicado por /$\mathbf{f_n}$/). As observações (vetores de características) associadas a cada fonema estão ilustradas por cores diferentes. 46

Lista de Tabelas

2.1 Representação dos fonemas das vogais e dos ditongos portugueses. 17
2.2 Representação dos fonemas das consoantes portuguesas. 17

5.1 Resultados obtidos utilizando a DTD no reconhecimento de vogais, dígitos e as palavras que compõem os conjuntos Engenharia Física (EF) e Universidade de Aveiro (UA). 43
5.2 Resultados das MVSs na classificação de vogais orais. O *kernel* utilizado foi o gaussiano, sendo que o parâmetro γ utilizado para cada teste está indicado na tabela. 45
5.3 Resultados obtidos com a aplicação das MVSs na classificação de dígitos e das palavras que compõem os conjuntos EF e UA. O *kernel* utilizado foi o gaussiano, sendo que foi utilizado o parâmetro $\gamma = 0.001$ em todos os testes. . 45
5.4 Resultados obtidos com a aplicação das FAs na classificação de dígitos e das palavras que compõem os conjuntos EF e UA. 46
5.5 Resultados obtidos com a aplicação dos MMOs na classificação de vogais, de dígitos e das palavras que compõem os conjuntos EF e UA. 47

Capítulo 1

Introdução

Hoje há um interesse crescente na aprendizagem de uma segunda língua. Podem ser diversas as razões. Há pessoas que encontram prazer nesse desafio. Para outras a aprendizagem de uma segunda língua é determinante para a sua integração social. Há ainda outras que precisam de uma segunda língua por questões profissionais. De qualquer forma, a aprendizagem de uma segunda língua é hoje muito mais comum num mundo mais interconectado.

Nos dias de hoje já existe uma grande quantidade de aplicações com vista a auxiliar a aprendizagem de novas línguas. Dois exemplos de aplicações para o auxílio na aprendizagem de inglês, são o EyeSpeak English e o English Central. Estas aplicações são bastante completas, proporcionando treino com feedback personalizado. Oferecem gravações de uma conversa que o utilizador deve repetir. O software analisa então o desempenho fonético do utilizador, indicando aspetos a melhorar. Estas aplicações só existem para a língua inglesa.

O grande desafio destes softwares é o de conseguirem apontar de forma precisa os aspetos a melhorar. Efetivamente, um algoritmo deste tipo deverá contemplar uma enorme diversidade de dicções, tons de voz, entoações e velocidades diferentes, para além de ter de reconhecer se realmente o que foi pronunciado está correto.

Um algoritmo deste tipo terá que contemplar várias etapas, como se pode observar pela figura 1.1. Inicialmente é necessário construir uma base de dados com os exercícios padrão, gravados por um conjunto de pessoas. Esta base de dados será utilizada posteriormente para comparação com o exercício de um utilizador, com vista a avaliar o seu desempenho fonético. No entanto, os sinais da fala têm que ser processados de forma a serem traduzidos num conjunto reduzido de parâmetros, mas que ao mesmo tempo contenham informação suficiente à cerca da estrutura fonética. Esses parâmetros vão indicar de uma forma mais clara os padrões fonéticos dos sinais da fala. Por simplicidade esses parâmetros foram indicados na figura 1.1 por meio de sequências de retângulos coloridos. É possível observar pela imagem, que o padrão de cores é semelhante em todas as sequências. Isto é, todas iniciam com tons de verde, mudam para tons de vermelho e terminam com tons de azul. A tarefa final é utilizar algoritmos que permitam realizar uma comparação entre os sinais parametrizados, dos exercícios padrão e do exercício do utilizador, de forma a avaliar o desempenho fonético do utilizador. A título de exemplo, podemos indicar que se os padrões fonéticos dos exercícios padrão iniciarem todos com tons de verde e o padrão fonético do exercício do utilizador iniciar com tom de azul, então, à partida o desempenho fonético do utilizador está a ser mau.

O objetivo principal deste trabalho foi estudar e implementar um conjunto de técnicas de processamento de sinal e de classificação de séries temporais úteis para o desenvolvimento de

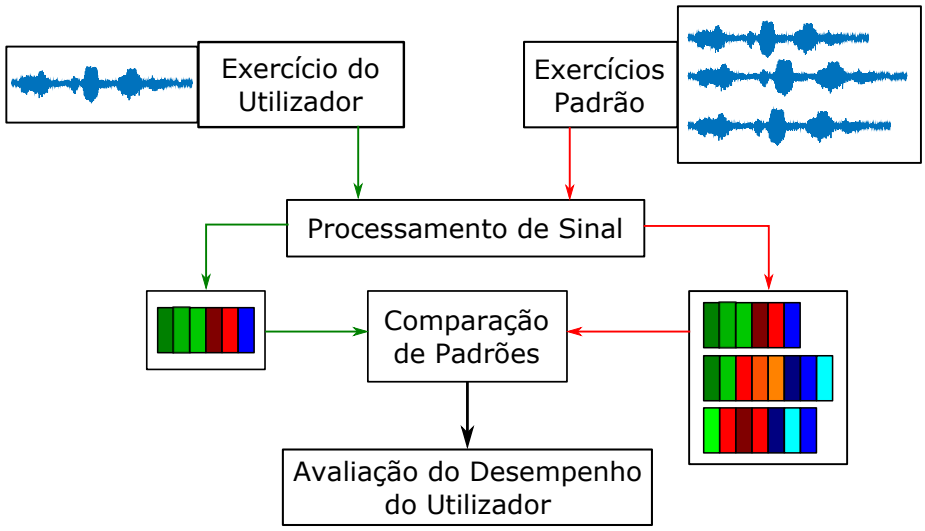


Figura 1.1: Diagrama das etapas que compõem um algoritmo de auxílio à aprendizagem de línguas. Inicialmente é necessária a construção de uma base de dados constituída por exercícios padrão. De seguida existe o processamento de sinal que permite traduzir os sinais da fala, num conjunto reduzido de parâmetros. Na imagem os parâmetros são indicados por meio de sequências de retângulos coloridos, que vão indicar de uma forma mais clara os padrões fonéticos dos sinais da fala. Por fim, utiliza-se algoritmos que permitam realizar uma comparação dos padrões fonéticos, de forma a avaliar o desempenho fonético do utilizador.

---

metodologias do ensino oral com feedback automático. Com vista a atingir os objetivos, foram estudadas técnicas de parametrização dos sinais baseadas na codificação linear preditiva e no *cepstrum*. Estas técnicas serão apresentadas no capítulo 2. Foram estudadas quatro técnicas para a comparação dos padrões, a distorção temporal dinâmica (DTD), os modelos de Markov ocultos (MMO), as máquinas de vetores de suporte (MVS) e as florestas aleatórias (FA). Estes algoritmos de classificação serão abordados nos capítulos 3 e 4. A aplicação de todas as técnicas de processamento de sinal e de classificação de séries temporais será apresentada no capítulo 5.

A abordagem mais utilizada para o reconhecimento da fala de uma forma geral nos dias de hoje, é a aplicação de modelos de Markov ocultos. Porém, este é um método com algumas desvantagens. Pois, faz assunções incorretas em relação aos sinais da fala e necessita de grandes quantidades de dados para atingirem um desempenho razoável.

Na aplicação de algoritmos de classificação como MVSs e as FAs, enfrentam-se duas grandes dificuldades. Em primeiro lugar, os sinais da fala apresentam tipicamente tamanhos varáveis, ao contrário das MVSs e FAs que são classificadores estáticos. Isto é, assume-se que as observações são vetores de dimensão fixa. Em segundo lugar, o sinal da fala corresponde a uma sequência de palavras, e cada palavra corresponde a uma porção de sinal desconhecido. Assim, o reconhecimento da fala é um problema de classificação sequencial, enquanto que os algoritmos MVS e FA foram projetados para classificação singular. Neste trabalho, serão apresentadas estratégias para contornar estas dificuldades.

Capítulo 2

Conceitos Teóricos da Análise da Fala

A fala consiste em ondas de pressão que são geradas pelo aparelho fonador humano. Estas ondas são geralmente classificadas em fala vocalizada ou não vocalizada. Os sons vocalizados produzem pulsos quase-periódicos de ar devido à vibração das cordas vocais, sendo o caso das vogais. Enquanto que os sons não vocalizados geram uma onda não periódica e aleatória. Estes sons são modificados à medida que atravessam o trato vocal, gerando o sinal da fala.

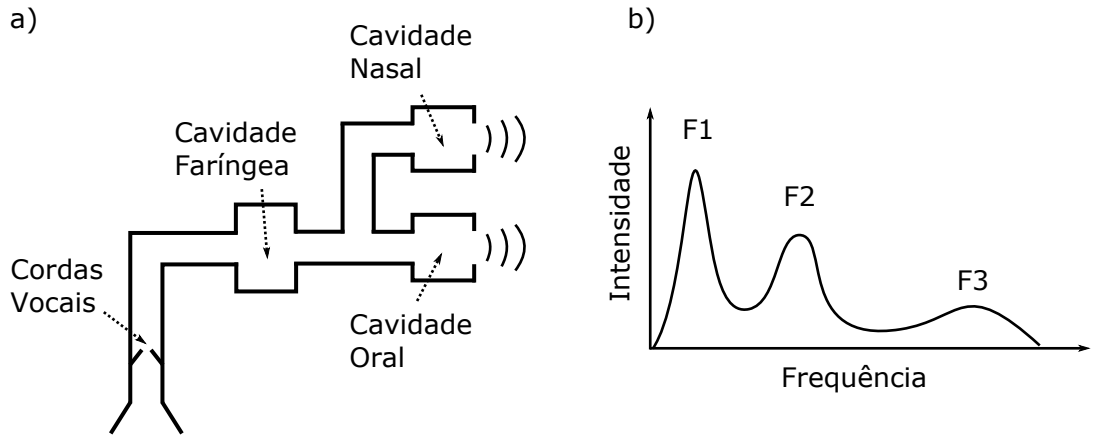


Figura 2.1: Na imagem a) mostra-se uma representação simplificada do trato vocal. O trato vocal funciona como uma cavidade ressonante. As regiões que se destacam no espetro de frequência dos sinais correspondem às frequências ressonantes, designadas de formantes. Na imagem b) está representado um espetro típico de uma vogal em que são visíveis três formantes.

Neste capítulo serão abordadas ferramentas que são importantes no processamento de sinais da fala. Inicialmente serão apresentados alguns procedimentos para o processamento do sinal, como a amostragem, a filtragem de pré-ênfase, a transformada de fourier de tempo curto e funções de janelas. Por fim, serão abordadas formas de extração de características a partir dos sinais da fala. Essas técnicas de extração de características são baseadas na codificação linear preditiva e no *cepstrum*.

2.1 Amostragem e Efeito de Aliasing

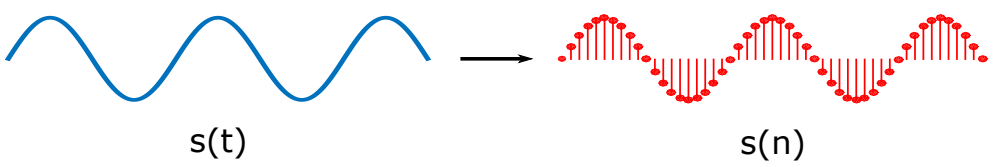


Figura 2.2: Representação do processo de amostragem.

No processamento de sinal, a amostragem é o processo de conversão de um sinal contínuo $s(t)$ para um sinal discreto $s(n)$, como se pode ver na figura 2.2. Neste processo, define-se a frequência de amostragem (F_s) como o número de pontos utilizado para descrever um segundo de sinal. O teorema de amostragem de Nyquist–Shannon afirma que a frequência de amostragem (F_s) deve ser no mínimo o dobro da frequência mais alta contida num sinal (F_N), isto é, $F_s \geq 2F_N$ [1][2]. Usualmente F_N é definida como a frequência de Nyquist.

Assim, uma baixa frequência de amostragem pode causar o efeito de *aliasing*, distorcendo o sinal original.

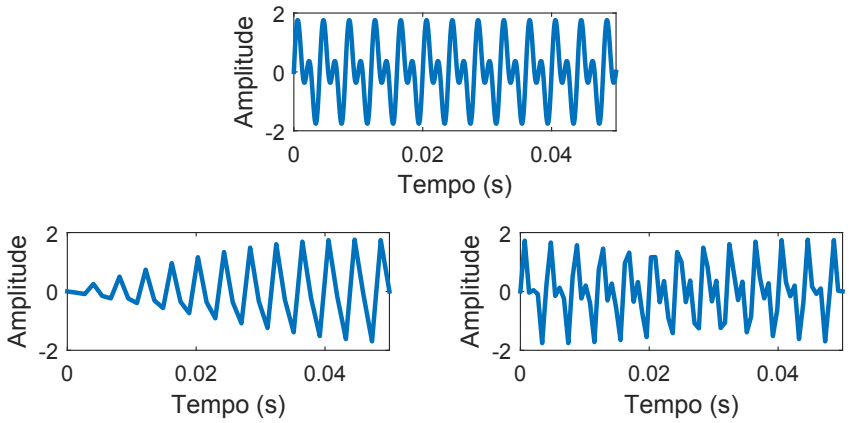


Figura 2.3: Representação do processo de amostragem de um sinal contínuo com duas frequências de amostragem diferentes. Na parte superior da imagem está ilustrado o sinal contínuo composto por duas frequências (250 e 500 Hz). Na parte inferior esquerda, o sinal foi amostrado com uma $F_s = 750$ Hz. Neste caso as duas frequências são sobrepostas devido ao efeito de *aliasing*. Na imagem inferior direita, o sinal foi amostrado com uma $F_s = 1500$ Hz, ou seja, as frequências são conservadas.

Podemos observar pela figura 2.4 que $f' = f$ até à frequência de Nyquist (F_N). Porém, as frequências acima de F_N sofrem o efeito de *aliasing*, em que estas são mapeadas para a banda de frequências abaixo de F_N, distorcendo o sinal original. O mapeamento pode ser definido da forma

$$f' = |mF_s - f|, \quad m \in \mathbb{N} \tag{2.1}$$

em que m é o múltiplo inteiro de F_s mais próximo de f.

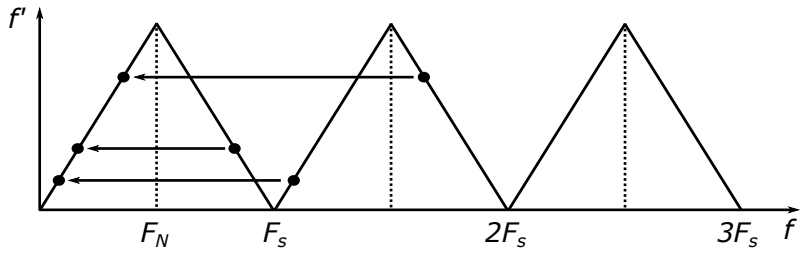


Figura 2.4: Efeito de *aliasing* no domínio da frequência.

2.2 Análise de Fourier de Tempo Curto

Uma das preocupações deste trabalho consiste em aplicar técnicas que consigam extrair informação útil do sistema áudio inicial. As técnicas baseadas na informação espectral de um sinal são muito utilizadas para atingir este fim. A transformada de fourier (TF) converte qualquer sinal no domínio temporal para o domínio da frequência. Esta é uma técnica para analisar séries temporais estacionárias, ou seja, quando não há alteração das frequências no período de observação. No entanto, os sinais que representam a fala não são estacionários, ou seja, as frequências variam no tempo [3]. Assim, uma TF aplicada em todo o sinal não apresenta a evolução temporal do sinal. Por outro lado, os sinais podem ser considerados estacionários numa pequena janela temporal (20 a 30 ms), devido aos tempos de resposta do aparelho fonador humano. Por esta razão, o sinal é decomposto em pequenas sequências designadas de *frames* e é aplicada uma TF em cada uma dessas sequências. Essas transformadas são designadas de transformada de fourier de tempo curto (TFTC). De forma a melhorar a resolução da frequência é usual multiplicar a TFTC por uma janela, que é escolhida dependendo do tipo de resolução que se pretende [4]. No sentido de reduzir descontinuidades da evolução temporal das frequências, geralmente é aplicada uma sobreposição de janelas.

Considere o sinal $s(n)$ com N amostras e $F(\omega_k, Tm)$ a TFTC de $s(n)$ no intervalo $[Tm, L+Tm]$, em que m é o *frame* que está a ser analisado, T é o deslocamento temporal da janela e L a duração da janela W. Assim, a TFTC pode ser definida por [5] [3]:

$$\begin{gathered} F(\omega_k, Tm) = \sum_{n=0}^{N-1} W(n-Tm)s(n)e^{-i\omega_k n}, \quad m = 0, 1, ..., \frac{N-1}{T} \\ \omega_k = \frac{2\pi k}{N}, \quad k = 0, ..., L-1 \end{gathered} \tag{2.2}$$

Na equação 2.2 a função $W(n)$ estabelece qual é a porção de $s(n)$ que está a ser analisada. O parâmetro T é escolhido de forma a obter a sobreposição pretendida, por exemplo, para $T = L/2$ há uma sobreposição de 50%. Uma janela arbitrária de duração L pode ser definida da seguinte forma:

$$\begin{aligned} W(n) &= 0, \quad n < 0, n > L \\ W(n) &\neq 0, \quad 0 \leq n \leq L-1 \end{aligned} \tag{2.3}$$

Uma forma eficiente de representar a evolução temporal das frequências de um determinado sinal é o espetrograma. O espetrograma é uma combinação de várias TFTC. Assim, o espetrograma é uma imagem 2D em que no eixo das abcissas está representado o tempo e

no eixo das ordenadas está representada a frequência. De forma a indicar a energia de cada ponto tempo/frequência é definida uma escala de cores, como se pode ver na figura 2.5.

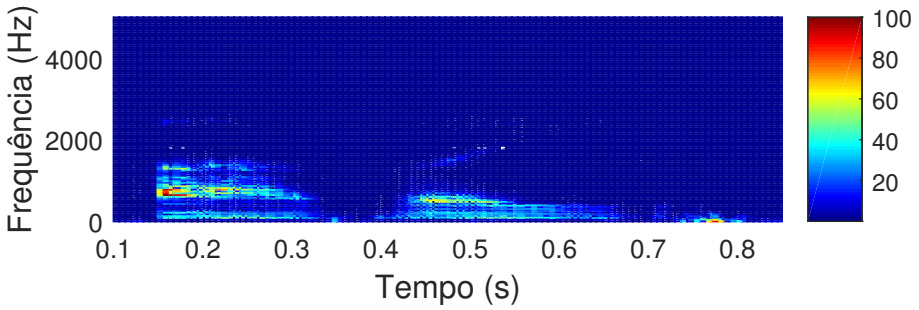


Figura 2.5: Ilustração de um espetrograma da palavra "Aveiro".

2.3 Filtro Pré-Ênfase

Normalmente os sinais produzidos pela fala possuem uma energia demasiado baixa nas frequências altas. Por essa razão, é usual aplicar um filtro de pré-ênfase no sinal antes de ser analisado. O filtro vai aumentar ligeiramente a magnitude das frequências mais altas em relação às mais baixas [6]. Desta forma, obtém-se uma melhor relação sinal-ruído em toda a gama de frequências. O filtro mais utilizado é um filtro passa alto, cuja função de transferência está definida na equação 2.4 e a resposta do filtro em relação à frequência está apresentada na figura 2.6.

$$H(z) = 1 - \alpha z^{-1} \tag{2.4}$$

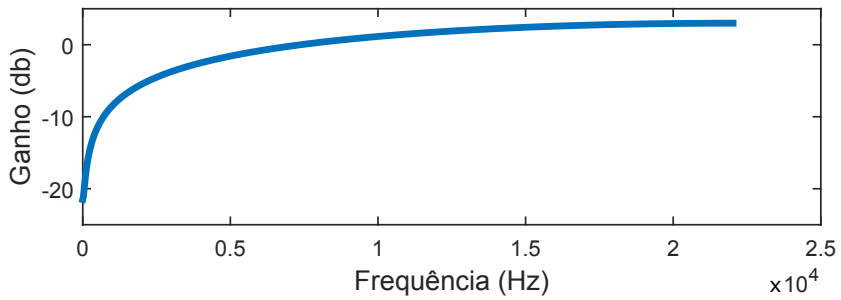


Figura 2.6: Resposta do filtro pré-ênfase no domínio da frequência, com $\alpha = 0.97$ e frequência de amostragem $F_s = 44100Hz$.

2.4 Funções de Janelas

Como já foi referido, no cálculo da transformada de fourier de tempo curto (TFTC) é aplicada uma janela para aumentar a resolução das frequências. A resolução aumenta devido ao fato da janela escolhida diminuir o efeito do vazamento espetral. Este efeito está relacionado com intervalos de observação finita. Uma abordagem intuitiva para o vazamento espetral é em perceber-se que num sinal existem frequências que não são periódicas no intervalo de

observação. Neste sentido, a extensão periódica de um sinal num período não proporcional ao período natural vai gerar descontinuidades. Estas descontinuidades contribuem para o vazamento espetral [7]. Este efeito está ilustrado na figura 2.7.

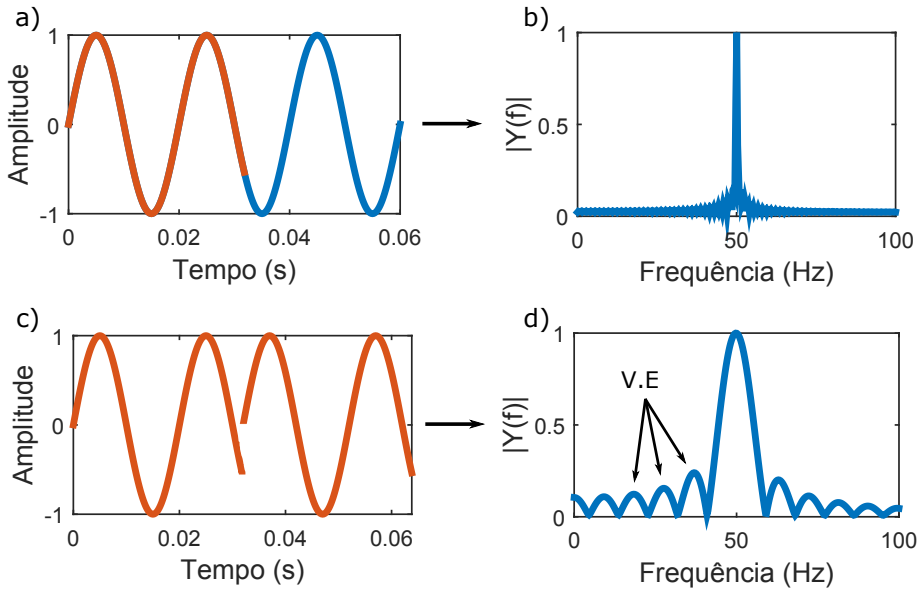


Figura 2.7: Ilustração do vazamento espetral. A imagem a) é o gráfico de um sinal com uma frequência de 50 Hz, cuja transforma de fourier (normalizada) está ilustrada na imagem b). A imagem c) representa o sinal resultante do corte aplicado ao sinal exposto em a). Esse corte está representado com cor laranja. Através da imagem c) pode-se visualizar que o corte vai gerar descontinuidades, que por sua vez vão contribuir para o vazamento espetral (V.E), como se pode observar na imagem d).

Pela figura 2.7 pode-se observar que o vazamento espetral é um espalhamento da energia ao longo da banda de frequências em torno da frequência real.

As janelas são aplicadas aos dados para reduzir a ordem de descontinuidades na fronteira da extensão periódica. Isto é conseguido quando se promove uma correspondência de muitas ordens de derivada junto à fronteira. E como a correspondência mais fácil de atingir é o zero, a janela tende suavemente para zero nos extremos para que a extensão periódica seja contínua no máximo de ordens de derivada [7].

Existe um grande número de janelas que podem ser aplicadas, porém a mais utilizada nas tarefas de parametrização de sinal é a janela Hamming, representada na imagem 2.8. A janela Hamming é definida por:

$$W(n) = 0.57 - 0.46\cos\left(\frac{2\pi n}{L-1}\right), \quad 0 \leq n \leq L-1 \tag{2.5}$$

A TFTC não é a única situação em que se utiliza funções de janelas. Na secção 2.5.1 será mostrada uma outra situação em que se utiliza uma janela.

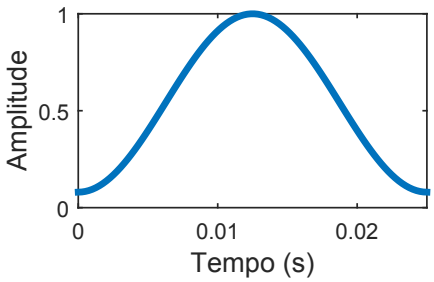


Figura 2.8: Ilustração da janela Hamming.

2.5 Extração de Características

A extração de características é uma das fases mais importantes no processo do reconhecimento da fala automático. Já foi referido que para analisar a evolução temporal, o sinal é decomposto em *frames*. No entanto, os *frames* contêm uma grande quantidade de informação, por exemplo, para um sinal com $F_s = 10000$ Hz, um *frame* de 25 ms contém 250 pontos. Neste sentido, são utilizadas técnicas para condensar a informação num número reduzido de parâmetros (características). Existe uma grande quantidade de técnicas para este efeito, no entanto, os parâmetros mais comuns são os coeficientes *cepstral* derivados da codificação preditiva linear (CPL) e os coeficientes *cepstral* de frequência de mel (CFM). Estas técnicas baseiam-se num modelo em que o sinal da fala pode ser decomposto numa fonte e num filtro. A informação extraída do sinal será maioritariamente derivada das características do filtro, ignorando-se o sinal de excitação. Estas técnicas de extração de características serão explicadas nas próximas subsecções.

2.5.1 Codificação Preditiva Linear

Como já foi referido, a fala é composta por voz vocalizada (ex: vogais) e não vocalizada (ex: a consoante 's'). Neste sentido, a fala pode ser aproximada por um modelo fonte-filtro, representado na figura 2.9. Este modelo faz aproximação da voz vocalizada a um trem de pulsos e a voz não vocalizada a ruído branco, assim, com a aplicação de um filtro nos sinais de excitação, é possível obter uma aproximação do sinal da fala.

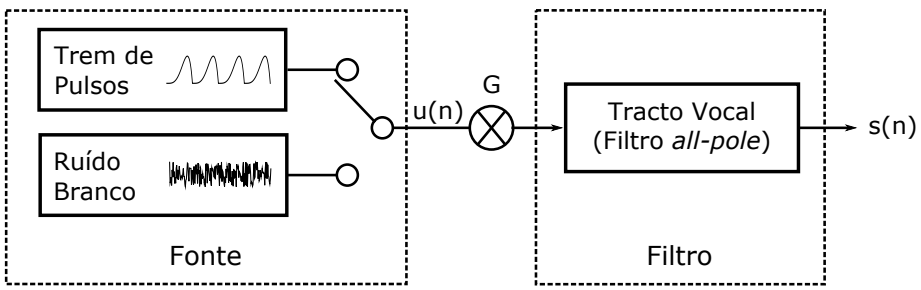


Figura 2.9: Representação de um modelo fonte-filtro simples para a geração de sinais da fala.

A codificação preditiva linear (CPL), é um modelo auto-regressivo de médias móveis utilizado para parametrizar séries temporais. Tendo em conta o modelo considerado para a

geração de sinais da fala, o modelo CPL é simplificado apenas para a componente auto-regressiva. O objetivo da aplicação deste modelo auto-regressivo é encontrar os parâmetros do filtro que modela o envelope espetral do sinal da fala. Do ponto de vista físico, os polos do filtro representam as frequências de ressonância do trato vocal, designados de formantes.

Como já foi referido, o modelo mais geral, é considerar que $s(n)$ é um sinal de saída de um sistema com um sinal de entrada $u(n)$ desconhecido, da seguinte forma [8]:

$$s(n) = -\sum_{k=1}^{p} a_k s(n-k) + G\sum_{l=0}^{q} b_l u(n-l), \quad b_0 = 1 \tag{2.6}$$

em que a_k, b_l e o ganho G são os parâmetros do sistema hipotético. A equação 2.6 pode ser reescrita no domínio da frequência aplicando a transformada Z em ambos os membros. Assim, a função de transferência deste sistema $H(z)$ é dada por:

$$H(z) = \frac{S(z)}{U(z)} = G\frac{1 + \sum_{l=1}^{q} b_l z^{-l}}{1 + \sum_{k=1}^{p} a_k z^{-k}} \tag{2.7}$$

em que

$$S(z) = \sum_{n=-\infty}^{\infty} s(n) z^{-n} \tag{2.8}$$

é a transformada Z do sinal $s(n)$ e de forma equivalente $U(z)$ é a transformada Z do sinal $u(n)$. A função $H(z)$ em 2.7 é o modelo geral designado de *pole-zero* [8]. As raízes dos polinómios do numerador e denominador são os zeros e os polos do modelo, respetivamente. A partir do modelo geral surgem dois casos especiais de interesse: o modelo *all-zero* em que $a_k = 0$, $1 \leq k \leq p$; e o modelo *all-pole* em que $b_l = 0$, $1 \leq l \leq q$ [8].

O modelo *all-pole* é o mais utilizado na análise de séries temporais e também é conhecido como o modelo auto-regressivo [8]. Este será o modelo utilizado nesta dissertação para a parametrização de cada *frame*. No modelo *all-pole* é assumido que o sinal $s(n)$ é dado por uma combinação linear dos valores anteriores e um sinal de entrada $u(n)$.

$$s(n) = -\sum_{k=1}^{p} a_k s(n-k) + Gu(n) \tag{2.9}$$

em que G é o ganho. A função de transferência $H(z)$ é reduzida para:

$$H(z) = \frac{G}{1 + \sum_{k=1}^{p} a_k z^{-k}} \tag{2.10}$$

Dado um sinal $s(n)$, o objetivo é determinar os coeficientes de previsão a_k e o ganho G. Para determinar estes parâmetros pode-se utilizar o método dos mínimos quadrados [8].

Assume-se que o sinal de entrada $u(n)$ é totalmente desconhecido, assim, o sinal $s(n)$ apenas pode ser previsto por uma combinação linear dos valores anteriores. Desta forma, a aproximação de $s(n)$ é dada por $\tilde{s}(n)$ em que

$$\tilde{s}(n) = -\sum_{k=1}^{p} a_k s(n-k) \tag{2.11}$$

O erro entre o valor real $s(n)$ e o valor previsto $\tilde{s}(n)$ é dado por:

$$e(n) = s(n) - \tilde{s}(n) = s(n) + \sum_{k=1}^{p} a_k s(n-k) \tag{2.12}$$

Os parâmetros a_k são obtidos como resultado da minimização da média do erro quadrático (sinal resultante de processo aleatório), ou do total do erro quadrático (sinal determinístico). Porém, neste trabalho apenas será abordada a vertente determinista. Assim, o erro quadrático total é dado por E em que

$$E = \sum_{n} e(n)^2 = \sum_{n} \left(s(n) + \sum_{k=1}^{p} a_k s(n-k) \right)^2 \tag{2.13}$$

A minimização do erro é determinada por:

$$\frac{\partial E}{\partial a_i} = 0, \quad 0 \leq i \leq p \tag{2.14}$$

Das equações 2.13 e 2.14 obtém-se:

$$\sum_{k=1}^{p} a_k \sum_{n} s(n-k)s(n-i) = -\sum_{n} s(n)s(n-i), \quad 1 \leq i \leq p \tag{2.15}$$

A partir da resolução destas equações é possível determinar os coeficientes de previsão a_k que minimizam o erro E.

O erro quadrático total mínimo E_p é obtido expandindo 2.13 e substituindo 2.15:

$$E_p = \sum_{n} s(n)^2 + \sum_{k=1}^{p} a_k \sum_{n} s(n)s(n-k) \tag{2.16}$$

A partir da equação 2.9 pode-se concluir que:

$$e(n) = Gu(n) \Rightarrow E_p = \sum_{n} e(n)^2 = \sum_{n} G^2 u(n)^2 = G^2 \tag{2.17}$$

assumindo que $\sum_{n} u(n) = 1$.

Especificando o intervalo da soma nas equações 2.13, 2.15, 2.16,podemos obter dois métodos para estimar os parâmetros. O método de auto-correlação (estacionário) e o método de covariância (não estacionário) [8]. Tendo em conta que o sinal da fala é composto por sequências consideradas estacionárias (*frames*), apenas vamos considerar o modelo da auto-correlação. Assim as equações 2.15 e 2.16 são reduzidas em:

$$\sum_{k=1}^{p} a_k R(i-k) = -R(i), \quad 1 \leq i \leq p \tag{2.18}$$

$$E_p = R(0) + \sum_{k=1}^{p} a_k R(k) \tag{2.19}$$

em que

$$R(i) = \sum_{n=0}^{L-1-i} \Big(s(n)W(n)\Big)\Big(s(n+i)W(n+i)\Big) \tag{2.20}$$

é a função de autocorrelação de $s(n)$ e $W(n)$ é uma janela de duração L. A equação 2.18 tem a propriedade de poder ser expandida sob a forma de matiz:

$$\begin{bmatrix} R(0) & R(1) & R(2) & \dots & R(p-1) \\ R(1) & R(0) & R(1) & \dots & R(p-2) \\ R(2) & R(1) & R(0) & \dots & R(p-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R(p-1) & R(p-2) & R(p-3) & \dots & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{bmatrix} = - \begin{bmatrix} R(1) \\ R(2) \\ R(3) \\ \vdots \\ R(p) \end{bmatrix} \tag{2.21}$$

Este sistema de equações pode ser resolvido pelo método de Levinson-Durbin da seguinte forma [8][9]:

$$E_0 = R(0) \tag{2.22}$$

$$k_i = -\frac{\left[R(i) + \sum_{j=1}^{i-1} a_j^{(i-1)} R(i-j)\right]}{E_{i-1}} \tag{2.23}$$

$$a_i^{(i)} = k_i \tag{2.24}$$

$$a_j^{(i)} = a_j + k_i a_{i-j}^{(i-1)}, \quad 1 \leq j \leq i-1 \tag{2.25}$$

$$E_i = (1 - k_i^2) E_{i-1} \tag{2.26}$$

As equações 2.23 - 2.26 são resolvidas recursivamente para $i = 1, 2, \cdots, p$. A solução final é dada por:

$$a_j = a_j^{(p)}, \quad 1 \leq j \leq p \tag{2.27}$$

Nesta subsecção foi derivada parte da teoria da codificação preditiva linear, com vista a extrair parâmetros de modelação do sinal da fala. Estes parâmetros podem ser utilizados tanto para tarefas de reconhecimento da fala como para sintetizar sinais artificialmente. Na próxima subsecção será abordada outra técnica de extração de parâmetros que descrevem o conteúdo do sinal, os coeficientes *cepstral* de frequência de mel (CFM).

2.5.2 Mel-Cepstrum

Segundo a teoria filtro-fonte pode-se considerar o sinal da fala como sendo uma convolução entre um sinal de excitação com um filtro, resultante do trato vocal. Isto é,

$$s(n) = e(n) * h(n) \tag{2.28}$$

A transformada Z da convolução de dois sinais tem a propriedade de ser o produto das suas transformadas Z, isto é

$$S(z) = E(z)H(z) \tag{2.29}$$

O *cepstrum* pode ser visto como uma transformação homomórfica que permite separar a fonte do filtro [10]. Ou seja, converter uma convolução numa soma,

$$s(n) = e(n) * h(n) \quad \rightarrow \quad \hat{s}(n) = \hat{e}(n) + \hat{h}(n) \tag{2.30}$$

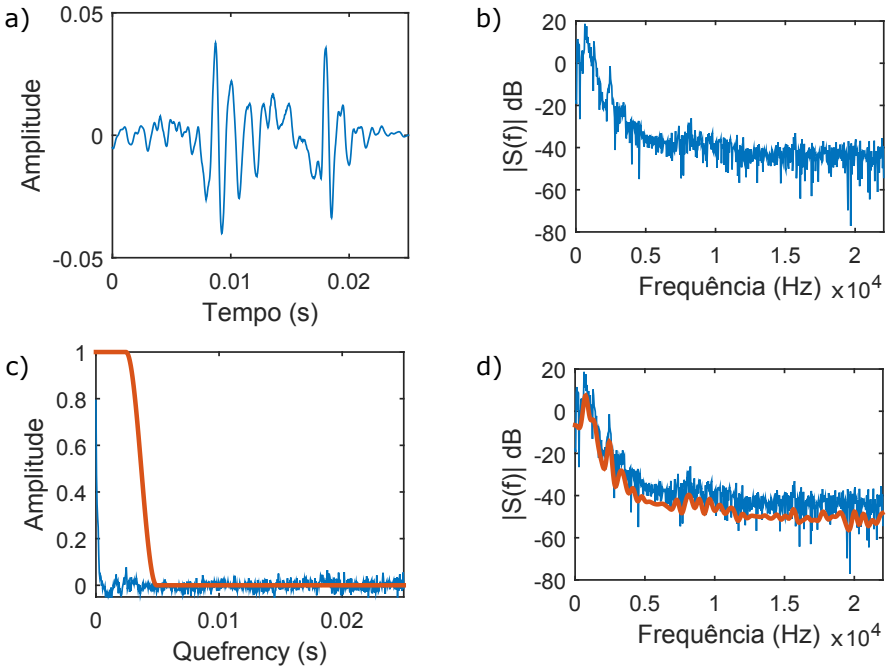


Figura 2.10: Separação filtro-fonte via *cepstrum*. Na imagem a) está apresentado um *frame* de sinal com uma janela *hamming* aplicada e na imagem b) está o seu espetro. Na imagem c) foi aplicado um *lifter* (filtro), representado a laranja, para extrair a componente do filtro $h(n)$. De seguida é aplicada a transformação inversa do *cepstrum* e obtém-se a resposta do filtro $h(n)$ na frequência, representado a laranja na imagem d).

Para um *frame* do sinal da fala de duração L, o *cepstrum* real $\tilde{s}(n)$ pode ser calculado da seguinte forma:

$$\tilde{S}(k) = log\Big|\sum_{n=0}^{L-1} s(n)e^{-j\omega_k n}\Big| \tag{2.31}$$

$$\tilde{s}(n) = \frac{1}{N}\sum_{n=0}^{L-1} \tilde{S}(k)e^{j\omega_k n} \tag{2.32}$$

em que

$$\omega_k = \frac{2\pi k}{L}, \quad 0 \leq k \leq L \tag{2.33}$$

O nome *cepstrum* vem da inversão da primeira sílaba da palavra *spectrum*. De uma forma semelhante, foi dado nome *quefrency* para a variável independente n em $\tilde{s}(n)$, sendo que a *quefrency* tem dimensão de tempo. Tendo em conta que o *cepstrum* transforma uma convolução numa soma, é possível separar a fonte $e(n)$ do filtro $h(n)$ porque a contribuição de cada

componente vai para regiões diferentes da *quefrency*. Deste modo, basta aplicar um *lifter* (filtro) no domínio da *quefrency* para separar as duas componentes, como se pode observar na figura 2.10.

Os coeficientes *cepstral* de frequência de mel (CFM) são uma forma de parametrização de sinais de forma a condensar informação proporcional à energia do espetro. Os coeficientes são baseados em dois conceitos, na escala de Mel e no *cepstrum*.

A escala de Mel, cujo nome vem da palavra *melody*, é uma escala motivada pela perceção de tons [11]. A escala de Mel é ajustada para que $1000mels$ corresponda a $1000Hz$ na escala linear. Um efeito da escala de Mel é a compressão das altas frequências em frequências relativamente baixas, como se pode observar na figura 2.11. A frequência de Mel pode ser aproximada pela seguinte equação [12]:

$$f_{Mel} = 2595\ log_{10}\left(1 + \frac{f}{700}\right) \tag{2.34}$$

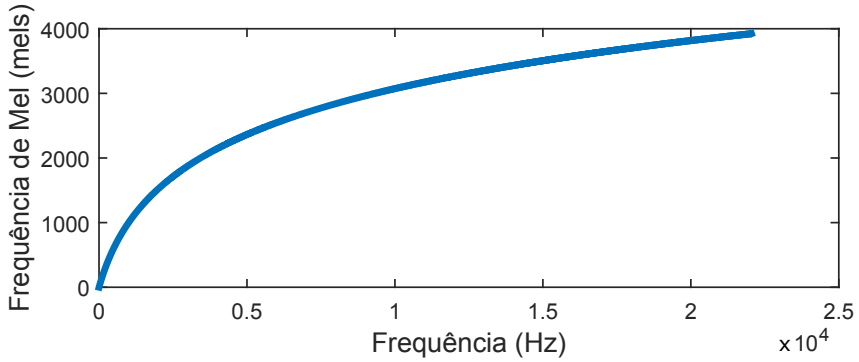


Figura 2.11: Relação entra a escala de Mel e a escala de frequências linear.

Para o cálculo dos coeficientes CFM vão ser utilizados os conceitos acima descritos, assim, o cálculo do *cepstrum* é realizado na escala de frequências de Mel, baseada na perceção auditiva [10]. Dada a transformada de fourier do sinal:

$$S(k) = \sum_{n=0}^{L-1} s(n)e^{-j\omega_k n} \tag{2.35}$$

pretende-se calcular o logaritmo da energia de saída de cada filtro H_m integrante de um banco de filtros triangulares da forma

$$E(m) = log\Big|\sum_{k=0}^{L-1} |S(k)|^2 H_m(k)\Big| \tag{2.36}$$

Cada filtro H_m presente no banco filtros triangulares com M filtros ($m = 1, 2, \cdots, M$), é dado

por:

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \frac{\left(k - f(m-1)\right)}{\left(f(m) - f(m-1)\right)} & f(m-1) \leq k \leq f(m) \\ \frac{\left(f(m+1) - k\right)}{\left(f(m+1) - f(m)\right)} & f(m) \leq k \leq f(m+1) \\ 0 & k > f(m+1) \end{cases} \tag{2.37}$$

Estes filtros vão calcular a média da densidade espetral em torno das frequências centrais, com uma banda crescente [10]. Os filtros estão representados na figura 2.12.

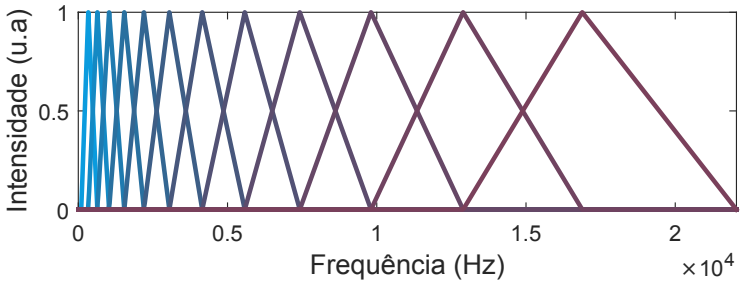


Figura 2.12: Representação de 12 filtros triangulares. Cada filtro está representado com uma cor diferente.

Considere as frequências f_l e f_h como sendo os limites inferior e superior de uma banda de frequência de um banco de filtros com M filtros. As frequências centrais $f(m)$ são definidas uniformemente na escala de Mel [10]:

$$f(m) = f_{Mel}^{-1}\left(f_{Mel}(f_l) + m\frac{f_{Mel}(f_h) - f_{Mel}(f_l)}{M+1}\right) \tag{2.38}$$

em que a escala de Mel inversa f_{Mel}^{-1} é dada por:

$$f_{Mel}^{-1} = 700\left(10^{\frac{f}{2595}} - 1\right) \tag{2.39}$$

A escala de mel foi apontada em [13], como tendo uma taxa de reconhecimento dos sinais da fala superior à escala linear. No entanto, estudos recentes indicam o contrário [14].

É de salientar que a aplicação da equação 2.36 já não é uma transformação homomórfica, só seria se a função logarítmica estivesse dentro do somatório. Porém, segundo [10], concluiu-se de forma empírica que a utilização de 2.36 trás vantagens, na medida em que as energias dos filtros apresentam maior robustez ao ruído.

Por fim, os coeficientes CFM representados por c_k são dados pela transformada de cosseno-II (TC-II) do logaritmo da energia de saída:

$$c_k = \sqrt{\frac{2}{M}}\sum_{m=1}^{M} E(m)cos\left(\frac{k\pi(m+1/2)}{M}\right), \quad 0 \leq k < M \tag{2.40}$$

em que usualmente M varia de 24 a 40. Para tarefas de reconhecimento da fala tipicamente são apenas considerados os primeiros 13 coeficientes [10]. Esta truncatura na sequência dos

coeficientes vai tendencialmente remover detalhes que são foneticamente irrelevantes [4]. É de salientar que é utilizada a TC-II (2.40) em vez da transformada de fourier devido a $E(m)$ ser par [10]. A TC-II tem a propriedade de compactar mais energia do que a TF, ou seja, os seus valores estão mais concentrados em índices menores [15]. Esta propriedade permite descrever um sinal com menos coeficientes. Para além disso, a TC-II tem a propriedade de ser um caso particular da transformada Karhunen-Loéve, que é utilizada na tarefa de decorrelação de sinais [16]. Assim, os coeficientes de saída tendem a ser independentes.

2.5.3 Coeficientes Cepstral derivados da CPL

Nas tarefas de reconhecimento da fala, em geral, os coeficientes CPL não são aplicados diretamente, pois com aplicação de uma simples transformação dos coeficientes com base no *cepstrum*, é possível obter uma melhor taxa de reconhecimento da fala [13]. O *cepstrum* pode ser extraído a partir das funções de transferência racionais de filtros digitais [10]. Assim, aplicando o logaritmo à equação 2.10, obtém-se

$$\log(H(z)) = \log(G) - \log\left(1 + \sum_{l=1}^{p} a_l z^{-l}\right) = \sum_{k=-\infty}^{\infty} c_k z^{-k} \tag{2.41}$$

derivando ambos os lados de ordem a z, obtém-se:

$$-\frac{\sum_{n=1}^{p} n a_n z^{-n-1}}{1 - \sum_{l=1}^{p} a_l z^{-l}} = -\sum_{k=-\infty}^{\infty} k c_k z^{-k-1} \tag{2.42}$$

de seguida multiplica-se ambos os lados por $-z(1-\sum_{l=1}^{p} a_l z^{-l})$ e substitui-se $l = n-k$, obtendo-se

$$\sum_{n=1}^{p} n a_n z^{-n} = \sum_{k=-\infty}^{\infty} n c_n z^{-n} - \sum_{n=k+1}^{p} \sum_{k=-\infty}^{\infty} k c_k a_{n-k} z^{-n} \tag{2.43}$$

Por fim, aplica-se a transformada Z inversa e obtém-se a seguinte recursão:

$$\begin{aligned} c_0 &= \log(G) \\ c_n &= a_n + \sum_{k=1}^{n-1} \frac{k}{n} a_{n-k} c_k \quad 1 \leq n \leq p \end{aligned} \tag{2.44}$$

Os coeficientes *cepstral* (CC) serão utilizados neste trabalho em vez dos coeficientes CPL.

2.5.4 Lifter

Foi mencionado em [17], que a aplicação de um *lifter* (filtro) pode aumentar taxa do reconhecimento da fala. O efeito deste filtro é diminuir a variabilidade indesejada e transformar a medida original numa mais fiável. Um exemplo concreto do efeito deste filtro, é a suavização da resposta das funções de transferência dos filtro calculados a partir da CPL. O *lifter* é definido como uma janela no domínio da *quefrency*. Esta janela é aplicada diretamente por

multiplicação nos vetores dos coeficientes CC ou CFM, isto é, $\tilde{c}_k = c_k \cdot w(k)$. A janela que obteve melhor desempenho, segundo [17], é definida por:

$$w(k) = 1 + \frac{Q}{2} sin(\frac{\pi k}{Q}) \tag{2.45}$$

sendo Q o parâmetro de *lifter* e k indica a componente de um vetor de coeficientes.

2.5.5 Coeficientes Dinâmicos: Delta e Aceleração

Os coeficientes CC e CFM são utilizados para caracterizar o sinal da fala em cada *frame*, desse modo, estes coeficientes são considerados estáticos. Assim, uma parte importante da informação do sinal pode não ter sido captada por estes coeficientes. Com o intuito de captar a informação dinâmica foram desenvolvidos coeficientes dinâmicos com a pretensão de que, juntamente com os coeficientes anteriores, contenham informação mais completa do sinal da fala [18]. Estes coeficientes dinâmicos são calculados a partir da evolução dos coeficientes estáticos ao longo dos *frames*. Os coeficientes dinâmicos podem ser calculados a partir de uma estimativa da derivada numérica, dada pela seguinte equação [19]:

$$\Delta_i = \frac{\sum_{j=1}^{N_F} j(c_{i+j} - c_{i-j})}{2\sum_{j=1}^{N_F} j^2} \tag{2.46}$$

em que i é o *frame* que está a ser analisado e N_F é o número de *frames* utilizados na regressão linear, sendo tipicamente $N_F = 2$. Note-se que a equação 2.46 dá mais peso às diferenças longínquas, do que às mais próximas do instante que se pretende estimar a derivada, ao contrário do que se estaria à espera. No entanto, esta metodologia permite estimar uma derivada suavizada quando aplicada a uma função com ruído. É de salientar que para calcular os coeficientes de aceleração, utiliza-se a mesma equação (2.46), mas aplicada aos coeficientes Δ.

2.6 Fonética

A fonética é a área que estuda a natureza física da produção e perceção dos sinais da fala [20]. A unidade sonora de uma língua que estabelece significado para diferenciar palavras é o fonema. Por exemplo, a diferença entre as palavras cato e gato, quando faladas, está apenas no primeiro fonema: /k/ na primeira e /g/ na segunda. A forma mais comum de representar os fonemas pelos linguistas é através do Alfabeto Fonético Internacional (AFI), desenvolvido pela Associação Internacional de Fonética (IPA) [21].

Um algoritmo com aplicação ao ensino de línguas, idealmente deverá conseguir identificar os fonemas que estão a ser pronunciados pelo utilizador. Para além disso, deverá indicar os fonemas que estão a ser pronunciados erradamente e sugerir formas objetivas de como atingir a pronúncia pretendida.

Nas tabelas 2.6 e 2.6 estão representados os fonemas utilizados na língua portuguesa [21].

Vogais orais	Exemplo	Ditongos orais	Exemplo
/i/	v<u>i</u>	/ɛi/	an<u>éi</u>s
/e/	v<u>ê</u>	/ɐ̃i/	c<u>em</u>
/ɛ/	s<u>é</u>	/ai/	s<u>ai</u>
/a/	v<u>á</u>	/ɐi/	s<u>ei</u>
/ɔ/	s<u>ó</u>	/ɔi/	m<u>ói</u>
/o/	s<u>ou</u>	/oi/	m<u>oi</u>ta
/u/	m<u>u</u>d<u>o</u>	/ui/	an<u>ui</u>s
/ɐ/	p<u>a</u>gar	/iu/	vi<u>u</u>
/ə/	p<u>e</u>gar	/eu/	me<u>u</u>
Vogais Nasais	Exemplo	/ɛu/	v<u>éu</u>
/ĩ/	vi<u>m</u>	/au/	ma<u>u</u>
/ẽ/	<u>en</u>tro	Ditongos Nasais	Exemplo
/ɐ̃/	<u>an</u>tro	/õi/	an<u>õe</u>s
/õ/	s<u>om</u>	/ũi/	m<u>ui</u>ta
/ũ/	m<u>un</u>do	/ɐ̃u/	m<u>ão</u>
		/ɐ̃u/	m<u>ão</u>

Tabela 2.1: Representação dos fonemas das vogais e dos ditongos portugueses.

Consoantes	Exemplo	Consoantes	Exemplo	Consoantes	Exemplo
/m/	<u>m</u>ato	/ɾ/	pi<u>r</u>a	/s/	<u>s</u>aca
/n/	<u>n</u>ato	/ʃ/	<u>ch</u>ato	/ʁ/	<u>r</u>ato
/t/	<u>t</u>ato	/ʒ/	<u>j</u>ato	/z/	<u>z</u>ebra
/d/	<u>d</u>ata	/ʎ/	ga<u>lh</u>o	/g/	<u>g</u>ato
/f/	<u>f</u>aca	/k/	<u>c</u>ato	/v/	<u>v</u>inha
/p/	<u>p</u>ato	/l/	ga<u>l</u>o		
/b/	<u>b</u>ato	/ɲ/	pi<u>nh</u>a		

Tabela 2.2: Representação dos fonemas das consoantes portuguesas.

Neste capítulo, foram abordadas várias técnicas de processamento de sinal com vista a extrair a informação mais importante dos sinais da fala. No próximo capítulo serão abordados dois algoritmos que utilizam esta informação compacta com o intuito de identificar segmentos dos sinais da fala.

Capítulo 3

Algoritmos de Classificação Dinâmicos

Neste capítulo serão abordados dois algoritmos de classificação dinâmicos, o primeiro é baseado na distorção temporal dinâmica e o segundo em modelos de Markov ocultos. Estes algoritmos são dinâmicos, pois foram desenvolvido com a intenção de contemplar a evolução temporal de um sistema. No âmbito desta dissertação serão utilizados para identificar os segmentos do sinal da fala.

3.1 Distorção Temporal Dinâmica

A distorção temporal dinâmica (DTD) é um método para calcular a distância entre duas sequências temporais. A distância entre séries temporais pode servir de métrica para tarefas de classificação. Tendo em conta que a informação fonética de um sinal da fala não é proporcional à sua duração, a aplicação de uma normalização temporal linear não é aconselhável [22]. A DTD leva isso em conta, permitindo obter um algoritmo que pode ser utilizado para reconhecimento da fala, por meio de comparação de séries temporais.

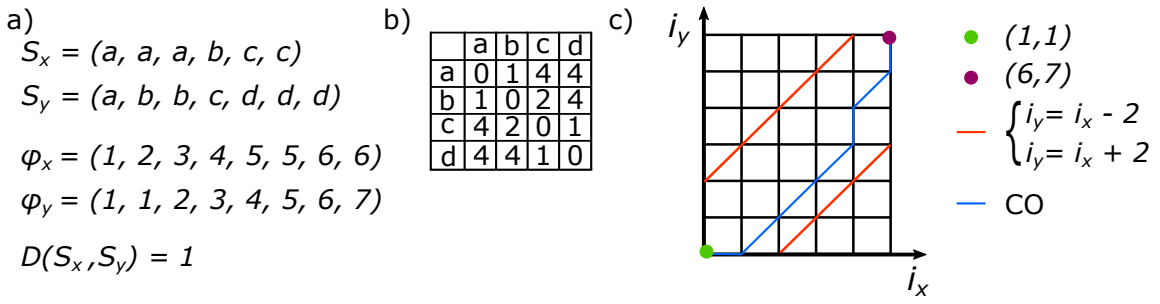
Considere duas séries temporais $\boldsymbol{S_x} = (\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_{T_x}})$ e $\boldsymbol{S_y} = (\boldsymbol{y_1}, \boldsymbol{y_2}, \cdots, \boldsymbol{y_{T_y}})$, em que $\boldsymbol{x_{i_x}}$ e $\boldsymbol{y_{i_y}}$ são vetores de características que representam as séries temporais nos instantes i_x e i_y respetivamente. Note-se que T_x e T_y não necessitam de ser iguais. De forma a normalizar as sequências, é realizado um mapeamento para uma dimensão temporal, por meio de uma função $\phi(k)$, em que as duas sequências têm a mesma duração. Isto é,

$$i_x = \phi_x(k), \quad i_y = \phi_y(k) \tag{3.1}$$

em que $k = 1, 2, \cdots, T$ e onde T é uma duração comum nas duas sequências. Estas funções associam a cada índice da sequência $\boldsymbol{S_x}$ um índice da sequência $\boldsymbol{S_y}$. Este processo está ilustrado na imagem 3.1 a). De forma a calcular a distorção total entre as duas sequências, é calculada a média das distâncias de todos os instantes entre as duas séries. Isto é,

$$d_\phi(S_x, S_y) = \sum_{k=1}^{T} \frac{w(k) \cdot d\Big(\phi_x(k), \phi_y(k)\Big)}{W_\phi}, \quad W_\phi = \sum_{k=1}^{T} w(k) \tag{3.2}$$

em que $w(k)$ é o peso de cada passo k, W_ϕ é uma constante de normalização e $d(\cdot, \cdot)$ é a distância entre dois pontos, sendo que usualmente utiliza-se a distância euclidiana.



	a	b	c	d
a	0	1	4	4
b	1	0	2	4
c	4	2	0	1
d	4	4	1	0

Figura 3.1: Exemplo de uma distorção dinâmica. Na imagem a) estão representadas duas séries temporais S_x e S_y, constituídas por símbolos discretos. Para além disso, estão representadas as funções de mapeamento ϕ_x e ϕ_y que foram utilizadas para a normalização temporal ótima, segunda a métrica representada na imagem b). Na imagem c), o caminho ótimo (CO) está representado numa grelha por uma linha azul.

Naturalmente se chega à conclusão que existe um vasto conjunto de funções $\phi(k)$ possíveis, com normalizações temporais distintas. Desta forma, o objetivo é encontrar o par de funções $\big(\phi_x(k), \phi_y(k)\big)$ que minimizam a distorção global entre as séries temporais. No entanto, para o alinhamento temporal das sequências ter sentido, é necessário impor algumas condições nas funções de distorção $\phi(k)$. As condições de fronteira são as mais importantes. Assim, apesar das séries temporais terem durações diferentes, no processo de normalização temporal os extremos são fixados. Isto é,

$$\begin{aligned} \phi_x(1) = 1, \quad \phi_y(1) = 1 \\ \phi_x(T) = T_x, \quad \phi_y(T) = T_y \end{aligned} \tag{3.3}$$

A partir da imagem 3.1 a) é possível observar que as funções de distorção (ϕ) possuem o mesmo tamanho.

Os caminhos são obrigatoriamente monotonamente crescentes. Esta condição elimina a possibilidade de distorção temporal reversa.

$$\phi(k) \leq \phi(k+1) \tag{3.4}$$

De forma a assegurar que nenhum segmento que contenha informação importante é perdido, são introduzidas condições de continuidade local [22] [23].

$$\phi(k+1) - \phi(k) \leq 1 \tag{3.5}$$

Também é claro que as funções de distorção (ϕ) apresentadas na imagem 3.1 a) respeitam esta restrição.

São inseridas condições globais do caminho no sentido de condicionar o desfasamento temporal entre as duas séries temporais [23].

$$|\phi_x(k) - \phi_y(k)| \leq r \tag{3.6}$$

em que r é um inteiro positivo que deve ser adequado. É de salientar que um efeito secundário desta condição é o aumento da velocidade do algoritmo, visto que o número de caminhos

possíveis é reduzido. As condições globais de caminho são mais fáceis de se visualizar quando o caminho é introduzido numa grelha, como apresentado na imagem 3.1 c). Nessa imagem, as condições estão representadas por linhas de cor laranja e possuem o parâmetro $r = 2$.

Para garantir que o caminho não crie uma correspondência pouco realista entre as duas sequências, uma condição para o declive é imposta. Uma forma de o fazer é: se o caminho $\phi(k)$ se deslocar na direção horizontal ou vertical m vezes, então $\phi(k)$ não se pode deslocar mais nessa direção sem pelo menos se deslocar n vezes na diagonal [23]. A intensidade da condição do declive pode ser dada por:

$$P = \frac{n}{m} \tag{3.7}$$

De acordo com [23], após ter sido experimentado um conjunto de valores para o declive, a melhor taxa de reconhecimento da fala foi obtida utilizando a condição $P = 1$. No entanto, esta condição pode impedir o cálculo da distorção total nas situações em que as duas séries temporais possuem um tamanho muito diferente. Na imagem 3.1 c) é claro que a condição de declive imposta é de $P = 1$, assim, sempre que o caminho se descolar uma vez vertical ou horizontalmente, terá que no instante a seguir se deslocar na diagonal. O deslocamento na diagonal é livre.

Como já foi referido, é utilizada uma função peso no cálculo da distorção total. Existem duas definições típicas para o peso, a forma simétrica e a forma assimétrica. No entanto, a forma simétrica é a mais plausível para se utilizar no reconhecimento da fala. Ou seja, dar mais peso a uma série em relação à outra parece levar a uma normalização temporal pouco realista. De acordo com [23], a forma simétrica é a que oferece melhor precisão no reconhecimento, como era de esperar. O peso simétrico é dado por,

$$w(k) = \big(\phi_x(k) - \phi_x(k+1)\big) + \big(\phi_y(k) - \phi_y(k+1)\big) \tag{3.8}$$

o que implica que $W_\phi = T_x + T_y$. Pode-se concluir a partir da equação 3.8 que o peso na diagonal será 2 e na vertical ou horizontal será 1.

3.1.1 Programação Dinâmica Para Cálculo da Distorção Total

Um exemplo simples para o cálculo da distorção total pode ser definido por:

1. Condição Inicial:
$$D(1,1) = w(1)d(1,1) = 2d(x_1, y_1) \tag{3.9}$$

2. Equação de Programação Dinâmica:
$$D(i_x, i_y) = min \begin{bmatrix} D(i_x - 1, i_y - 2) + 2d(\boldsymbol{x_{i_x}}, \boldsymbol{y_{i_y - 1}}) + d(\boldsymbol{x_{i_x}}, \boldsymbol{y_{i_y}}) \\ D(i_x - 1, i_y - 1) + 2d(\boldsymbol{x_{i_x}}, \boldsymbol{y_{i_y}}) \\ D(i_x - 2, i_y - 1) + 2d(\boldsymbol{x_{i_x - 1}}, \boldsymbol{y_{i_y}}) + d(\boldsymbol{x_{i_x}}, \boldsymbol{y_{i_y}}) \end{bmatrix} \tag{3.10}$$

3. Sujeito à condição:
$$i_y - r \leq i_x \leq i_y + r \tag{3.11}$$

4. A distorção total do caminho ótimo:
$$D(S_x, S_y) = \frac{D(T_x, T_y)}{W_\phi}, \quad W_\phi = T_x + T_y \tag{3.12}$$

O algoritmo inicia com o cálculo da distância entre os vetores de características do primeiro instante das séries temporais. O peso atribuído neste ponto é indiferente, mas para ser consistente com a equação 3.10, é atribuído peso 2. Pela equação 3.10, pode-se observar que, em cada instante, a distância é calculada a partir do mínimo de 3 opções possíveis. A condição de declive e os pesos estão incluídos na equação. Assim, quando existe um deslocamento horizontal ou vertical, obrigatoriamente existe um deslocamento na diagonal. Em relação ao peso, é utilizada a forma simétrica. A aplicação recorrente da equação 3.10 calcula a distorção total entre as duas séries quando se atingir o ponto (T_x, T_y).

3.2 Modelos de Markov

Os modelos de Markov são modelos capazes de caracterizar as propriedades estatísticas de séries temporais de variadas durações. Por definição, um modelo de Markov é um processo cuja probabilidade de o sistema estar em determinado estado em dado período de observação depende apenas do estado no período de observação imediatamente anterior [24].

Estes modelos têm bastante sucesso nas tarefas de reconhecimento da fala automático, sendo essa a razão de serem introduzidos neste trabalho. Como já foi referido, será utilizado reconhecimento da fala para avaliar a dicção do utilizador, da aplicação que se pretende desenvolver.

3.2.1 Modelos de Markov Ocultos

Considere um sistema que a qualquer altura pode ser definido por um conjunto de N estados $S = (S_1, S_2, \cdots, S_N)$. Por exemplo, em processamento de áudio é frequente associar estados a fonemas, as unidades fonéticas elementares que compõem uma palavra. Num modelo de Markov, em cada instante t o sistema muda de estado (possivelmente para o mesmo) com uma probabilidade associada. O estado no instante t é representado por q_t [25]. É considerado que a descrição probabilística deste sistema é resumida ao seu estado atual e ao seu antecessor, ou seja:

$$P(q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k \cdots) = P(q_t = S_j | q_{t-1} = S_i) \tag{3.13}$$

Para além disso, é considerado que esta descrição probabilística é independente do tempo [25]. Assim, as probabilidades de transição de estados a_{ij} podem ser definidas da seguinte forma:

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i), \quad 1 \leq i, j \leq N \tag{3.14}$$

sendo que os coeficientes de transição de estado possuem as seguintes características:

$$a_{ij} \geq 0 \tag{3.15}$$

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{3.16}$$

pois obedecem às condições padrão de processos estocásticos [25].

Vamos considerar um exemplo de um modelo de Markov com três estados, em que cada estado representa um fonema $f_i, i = 1, \cdots, 3$ tal como ilustrado na figura 3.2.

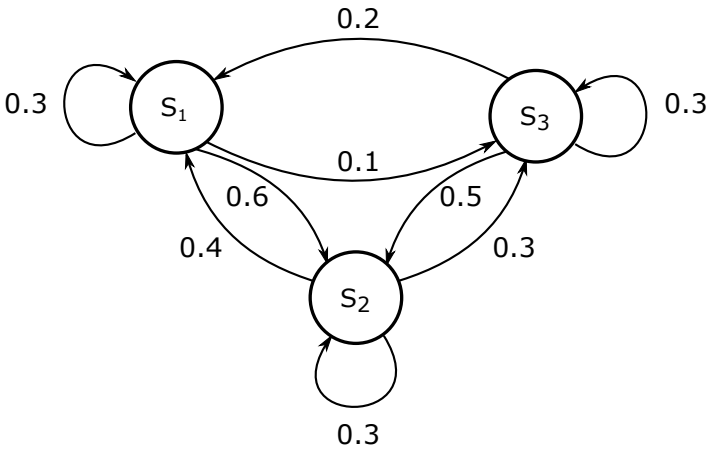


Figura 3.2: Exemplo de um modelo de Markov com três estados, com as respetivas probabilidades de transição.

A partir da figura 3.2, podemos definir a matriz de probabilidade de transição de estados (A) para um instante t, como:

$$A = \{a_{ij}\} = \begin{bmatrix} 0.3 & 0.6 & 0.1 \\ 0.4 & 0.3 & 0.3 \\ 0.2 & 0.5 & 0.3 \end{bmatrix}$$

Com base nesta matriz é possível calcular a probabilidade de ocorrência de uma determinada sequência de fonemas, por exemplo numa palavra. Por exemplo, se for pronunciada uma sequência de fonemas $(f_1, f_3, f_1, f_2, f_3, f_1)$ a probabilidade da sua ocorrência pode ser calculada através de:

$$\begin{aligned} P(O|Modelo) &= P(S_1, S_3, S_1, S_2, S_3, S_1|Modelo) \\ &= P(S_1)P(S_3|S_1)P(S_1|S_3)P(S_2|S_1)P(S_3|S_2)P(S_1|S_3) \\ &= \pi_1 \cdot a_{13} \cdot a_{31} \cdot a_{12} \cdot a_{23} \cdot a_{31} \\ &= 1 \cdot 0.1 \cdot 0.2 \cdot 0.6 \cdot 0.3 \cdot 0.2 \\ &= 7.2 \times 10^{-4} \end{aligned}$$

em que é utilizada a notação

$$\pi_i = P(q_1 = S_i), \quad 1 \leq i \leq N \tag{3.17}$$

para definir as probabilidades do estado inicial.

Neste modelo de Markov cada estado corresponde a uma observável (fonema). Porém, este modelo é demasiado simplista, pois os estados podem ter manifestações diversas e estados diferentes podem levar às mesmas consequências. Por exemplo, no caso precedente os fonemas nem sempre são pronunciados da mesma forma e por outro lado dois fonemas podem ser confundidos, isto é, percecionados da mesma forma. Assim nos modelos de Markov ocultos introduziu-se uma diferença entre a variável estado e a observável. A informação a que temos acesso será a observável, sendo no caso precedente definida pelos coeficientes que caracterizam o sinal áudio. Por outro lado, o estado torna-se uma variável conceptual (oculta) responsável

por estabelecer a correta transição entre estados diferentes e as observações mensuráveis. Assim, tal como se indica na figura 3.3 à luz dos modelos de Markov ocultos, um sinal áudio pode ser visto como o resultado de um modelo com uma sequência de estados com probabilidades de transição associadas e cada estado pode gerar um vetor de características que caracteriza o sinal áudio num determinado instante.

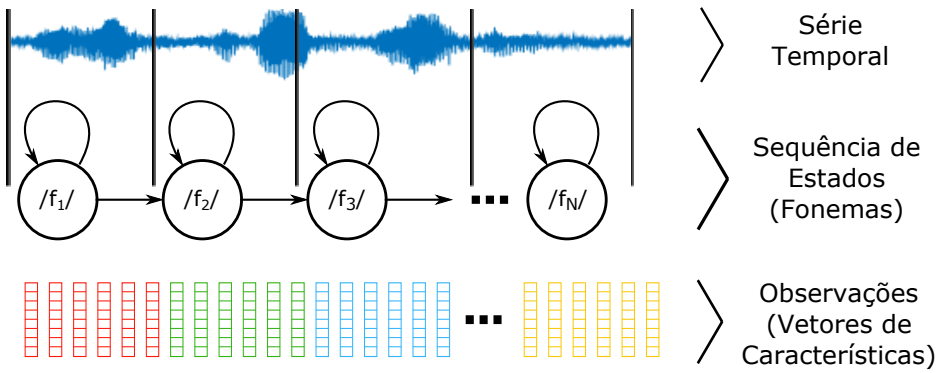


Figura 3.3: Descrição de sinais audio em termos de modelos de Markvov ocultos. Nestes modelos, é considerada uma direção pois os fonemas estão encadeados temporalmente. Cada estado representa um fonema (indicado por $/\mathbf{f_n}/$) dando origem a diferentes vetores de características, que caracterizam o som pronunciado.

Os MMOs genérica e formalmente podem ser caracterizado da seguinte forma [25]:

1. N é o número de estados do modelo. Cada estado individualmente é definido por $S = (S_1, S_2, \cdots, S_N)$, e um estado no instante t é definido por q_t. Embora os estados estejam ocultos, na maior parte dos casos existe um significado físico associado a cada estado ou a um conjunto de estados. No exemplo apresentado anteriormente, cada estado está associado a um fonema.

2. M é o número de símbolos discretos que se pode observar. Cada símbolo individualmente é definido por $V = (v_1, v_2, \cdots, v_M)$. O símbolo corresponde à observável física de saída do sistema que está a ser modelado. No exemplo apresentado anteriormente, cada símbolo está associado a um vetor de características.

3. A distribuição de probabilidade da transição de estados é dada por $A = \{a_{ij}\}$ em que:

$$a_{ij} = P(q_{t+1} = S_j | q_t = S_i), \quad 1 \leq i, j \leq N \tag{3.18}$$

   Para o caso em que cada estado pode transitar para qualquer estado, então $a_{ij} > 0$. Para outros tipos de MMO podem existir alguns pares (i, j) para os quais $a_{ij} = 0$.

4. A distribuição de probabilidade da observação de símbolos no estado j é dada por $B = \{b_j(k)\}$, em que:

$$\begin{aligned} b_j(k) = P(v_k | q_t = S_j), \quad & 1 \leq j \leq N \\ & 1 \leq k \leq M \end{aligned} \tag{3.19}$$

5. A distribuição de probabilidade dos estados iniciais é dada por $\pi = \{\pi_i\}$, em que:

$$\pi_i = P(q_1 \leq S_i), \quad 1 \leq i \leq N \tag{3.20}$$

Por conveniência, um MMO é geralmente caracterizado pela notação compacta

$$\lambda = (A, B, \pi) \tag{3.21}$$

para indicar o conjunto de parâmetros do modelo.

Dado um MMO, existem dois problemas que devem ser resolvidos para que o modelo seja útil para ser aplicado em reconhecimento da fala automático [25]. Assim, dada uma sequencia de observações O (série temporal) e um modelo $\lambda = (A, B, \pi)$, pretende-se calcular eficientemente $P(O|\lambda)$. Para tal, vamos considerar uma sequência de estados:

$$Q = (q_1, q_2, \cdots, q_T) \tag{3.22}$$

em que q_1 é o estado inicial. A probabilidade da sequência O para a sequência de estados Q é dada por:

$$P(O|Q, \lambda) = \prod_{t=1}^{T} P(O_t|q_t, \lambda) \tag{3.23}$$

em que é assumida independência estatística entre observações. Assim, a equação 3.23 pode ser reescrita da seguinte forma:

$$P(O|Q, \lambda) = b_{q_1}(O_1) \cdot b_{q_2}(O_2) \ \cdots \ b_{q_T}(O_T) \tag{3.24}$$

A probabilidade da sequência de estados Q pode ser escrita como:

$$P(Q|\lambda) = \pi_{q_1} \cdot a_{q_1 q_2} \cdot a_{q_2 q_3} \ \cdots \ a_{q_{T-1} q_T} \tag{3.25}$$

A probabilidade de O e Q ocorrerem em simultâneo é dada pelo produto dos dois termos:

$$P(O, Q|\lambda) = P(O|Q, \lambda) \cdot P(Q, \lambda) \tag{3.26}$$

A probabilidade de O dado o modelo λ pode ser calculada a partir da soma de todas as probabilidades conjuntas de todas as sequências de estados Q possíveis e é dada por:

$$\begin{aligned} P(O|\lambda) &= \sum_{\forall Q} P(O|Q, \lambda) \cdot P(Q|\lambda) \\ &= \sum_{q_1, q_2, \cdots, q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \cdots a_{q_{T-1} q_T} b_{q_T}(O_T) \end{aligned} \tag{3.27}$$

O cálculo da probabilidade apresentada na equação 3.27 é muito dispendioso computacionalmente se for calculada diretamente ($2TN^T$ cálculos) [25]. Desta forma, é utilizado o algoritmo *Forward-Backward* que realiza esta tarefa eficientemente [26].

Considere a variável *forward* $\alpha_t(i)$ definida por:

$$\alpha_t(i) = P(O_1, O_2, \cdots O_t, q_t = S_i|\lambda) \tag{3.28}$$

como sendo a probabilidade da sequência de observações parcial $O_1, O_2, \cdots O_t$ a ocorrer em simultâneo com o estado S_i no instante t, dado o modelo λ. A partir da equação 3.28 pode-se deduzir que:

$$\begin{aligned} \alpha_1(i) &= P(O_1, q_1 = S_i) \\ &= P(O_1|q_1 = S_i)P(q_1 = S_i) \\ &= b_i(O_1)\pi_i \quad 1 \leq i \leq N \end{aligned} \tag{3.29}$$

Em relação ao resto dos instantes t, é possível derivar uma equação recursiva para o cálculo de $\alpha_t(i)$ da seguinte forma:

$$\begin{aligned} \alpha_t(i) &= P(O_1, \cdots, O_t, q_t = S_i) = P(O_1, \cdots, O_t|q_t = S_i)P(q_t = S_i) \\ &= P(O_1, \cdots, O_{t-1}|q_t = S_i)P(O_t|q_t = S_i)P(q_t = S_i) \\ &= P(O_t|q_t = S_i)P(O_1, \cdots, O_{t-1}, q_t = S_i) \\ &= b_i(O_t)\sum_{j=1}^{N} P(O_1, \cdots, O_{t-1}, q_{t-1} = S_j, q_t = S_i) \\ &= b_i(O_t)\sum_{j=1}^{N} P(O_1, \cdots, O_{t-1}, q_t = S_i|q_{t-1} = S_j)P(q_{t-1} = S_j) \\ &= b_i(O_t)\sum_{j=1}^{N} P(O_1, \cdots, O_{t-1}|q_{t-1} = S_j)P(q_t = S_i|q_{t-1} = S_j)P(q_{t-1} = S_j) \\ &= b_i(O_t)\sum_{j=1}^{N} P(O_1, \cdots, O_{t-1}, q_{t-1} = S_j)a_{ji} \\ &= b_i(O_t)\sum_{j=1}^{N} \alpha_{t-1}(j)a_{ji} \quad 2 \leq t \leq T, \quad 1 \leq i \leq N \end{aligned} \tag{3.30}$$

Por fim, calcula-se $P(O|\lambda)$ da seguinte forma:

$$P(O|\lambda) = \sum_{i=1}^{N} P(O_1, \cdots, O_T, q_T = S_i) = \sum_{i=1}^{N} \alpha_T(i) \tag{3.31}$$

Utilizando este algoritmo o número de cálculos é reduzido para $\sim N^2T$ [25].

De forma equivalente, vamos considerar a variável $\beta_t(i)$ definida por:

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \cdots O_T|q_t = S_i, \lambda) \tag{3.32}$$

como a probabilidade da sequência de observações parcial $O_{t+1}, O_{t+2}, \cdots O_T$ dado o estado S_i no instante t e o modelo λ. Novamente, pode-se resolver $\beta_t(i)$ através de uma equação

recursiva, da forma:

$$\begin{aligned}
\beta_t(i) &= P(O_{t+1}, \cdots, O_T | q_t = S_i) \\
&= \sum_{j=1}^{N} P(O_{t+1}, \cdots, O_T, q_{t+1} = S_j | q_t = S_i) \\
&= \sum_{j=1}^{N} P(O_{t+1}, \cdots, O_T | q_t = S_i, q_{t+1} = S_j) P(q_{t+1} = S_j | q_t = S_i) \\
&= \sum_{j=1}^{N} P(O_{t+2}, \cdots, O_T | q_{t+1} = S_j) P(O_{t+1} | q_{t+1} = S_j) a_{ij} \\
&= \sum_{j=1}^{N} \beta_{t+1}(j) b_{t+1}(j) a_{ij}, \quad t = T-1 \cdots 1, \quad 1 \leq i \leq N
\end{aligned} \tag{3.33}$$

E é definida a seguinte condição inicial:

$$\beta_T(i) = 1, \quad 1 \leq i \leq N \tag{3.34}$$

Note-se que a parte *backward* não é necessária para o cálculo de $P(O|\lambda)$, mas será útil na resolução do nosso próximo problema.

Dada uma sequencia de observações O, pretende-se ajustar os parâmetros do modelo $\lambda = (A, B, \pi)$ de forma a maximizar $P(O|\lambda)$. Este é um problema bastante difícil porque não existe nenhuma forma analítica conhecida de o resolver. No entanto, é possível determinar λ de forma que $P(O|\lambda)$ seja maximizado localmente. Assim, é utilizado um método iterativo de reestimação, que a cada iteração irá estimar os parâmetros do modelo $\bar{\lambda}$. Este método é designado de Baum-Welch [27] [28].

Considere a variável $\xi(i, j)$ definida por

$$\xi(i, j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda) \tag{3.35}$$

como sendo a probabilidade de estar no estado S_i no instante t e no estado S_j no instante $t + 1$, dada uma sequência de observações O e um modelo λ. É possível reescrever a equação 3.35 utilizando as variáveis *forward* e *backward* da seguinte forma [25]:

$$\begin{aligned}
\xi(i, j) &= P(q_t = S_i, q_{t+1} = S_j | O) \\
&= P(O | q_t = S_i, q_{t+1} = S_j) P(q_t = S_i, q_{t+1} = S_j) / P(O) \\
&= \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) / P(O) \\
&= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}
\end{aligned} \tag{3.36}$$

Considere agora a variável $\gamma_t(i)$ definida por

$$\gamma_t(i) = P(q_t = S_i | O, \lambda) \tag{3.37}$$

que é a probabilidade de estar no estado S_i no instante t, dada uma sequência de observações e o modelo. A equação 3.37 pode ser reescrita utilizando as variáveis *forward* e *backward* da

forma [25]:

$$\begin{aligned}\gamma_t(i) &= P(q_t = S_i|O) \\ &= P(O|q_t = S_i)P(q_t = S_i)/P(O) \\ &= \alpha_t(i)\beta_t(i)/P(O) \\ &= \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)}\end{aligned} \tag{3.38}$$

em que $P(O|\lambda)$ é um fator de normalização para que $\sum_{i=1}^{N} \gamma_t(i) = 1$.

As variáveis $\gamma_t(i)$ e $\xi_t(i,j)$ estão relacionadas da forma:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{3.39}$$

Se for realizada a soma de $\gamma_t(i)$ ao longo do tempo t, obtém-se o número esperado de transições feitas a partir do estado S_i. De forma semelhante, a soma de $\xi_t(i,j)$ ao longo do tempo t, pode ser interpretada como o número esperado de transições do estado S_i para o estado S_j. Assim,

$$\sum_{t=1}^{T-1} \gamma_t(i) \equiv \text{Número esperado de transições a partir do estado } S_i \tag{3.40}$$

$$\sum_{t=1}^{T-1} \xi_t(i,j) \equiv \text{Número esperado de transições a partir do estado } S_i \text{ para o estado } S_j \tag{3.41}$$

Utilizando as equações 3.40 e 3.41 é possível reestimar os parâmetros do MMO da seguinte forma [25]:

$$\begin{aligned}\bar{\pi}_i &\equiv \text{Número esperado de vezes no estado } S_i \text{ no instante } t = 1 \\ &= \gamma_1(i)\end{aligned} \tag{3.42}$$

$$\begin{aligned}\bar{a}_{ij} &\equiv \frac{\text{Número esperado de transições a partir do estado } S_i \text{ para o estado } S_j}{\text{Número esperado de transições a partir do estado } S_i} \\ &= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}\end{aligned} \tag{3.43}$$

$$\begin{aligned}\bar{b}_i(k) &\equiv \frac{\text{Número esperado de vezes no estado } S_j \text{ a observar o símbolo } v_k}{\text{Número esperado de vezes no estado } S_j} \\ &= \frac{\sum_{\substack{t=1 \\ s.a\ O_t = v_k}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(i)}\end{aligned} \tag{3.44}$$

Utilizando as equações acima definidas obtém-se então uma nova estimação do modelo $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$. E foi provado por Baum et al [29] [30], que o modelo $\bar{\lambda}$ é mais provável de produzir a sequência de observações do que o modelo λ, ou seja, $P(O|\bar{\lambda}) > P(O|\lambda)$. Por outro lado, se o modelo λ é um ponto crítico da função de verosimilhança, então $\bar{\lambda} = \lambda$.

O tratamento precedente pressupõe que as observações são representadas por símbolos pertencentes a um conjunto finito, o que permite a associação de uma densidade de probabilidade discreta a cada estado [25]. Porém, em tarefas de reconhecimento da fala, as observações são vetores de características associados a cada *frame*, que podem conter um conjunto contínuo de valores. É possível utilizar MMOs discretos utilizando técnicas de quantização vetorial [31], que associam um conjunto finito de símbolos a regiões do espaço das características.

A ligação entre estados e observáveis pode ser modelada associando a cada estado uma função de densidade de probabilidade que pode ser obtida a partir de uma mistura de gaussianas de acordo com:

$$b_j(\boldsymbol{O}) = \sum_{m=1}^{M} c_{jm}\, \mathcal{N}(\boldsymbol{O}, \boldsymbol{\mu_{jm}}, \boldsymbol{\Sigma_{jm}}), \quad 1 \leq j \leq N \tag{3.45}$$

em que $\boldsymbol{O}$ é o vetor multidimensional que está a ser modelado e c_{jm} é o coeficiente de mistura da m-ésima mistura no estado j. Os parâmetros $\boldsymbol{\mu_{jm}}$ e $\boldsymbol{\Sigma_{jm}}$ são o vetor média e matriz de covariância, para a m-ésima componente de mistura no estado j, respetivamente. Note-se que os coeficientes c_{jm} são positivos e obedecem a

$$\sum_{m=1}^{M} c_{jm} = 1, \quad 1 \leq j \leq N \tag{3.46}$$

para garantir que a função de distribuição de probabilidade está normalizada.

Tendo em conta que a distribuição real é desconhecida, espera-se que uma mistura pesada de distribuições gaussianas multidimensionais se aproxime da realidade. É por esta razão que os MMOs necessitam de uma grande quantidade de dados de treino para se gerar um modelo.

Neste trabalho, foi considerado o caso mais simples, em que cada fonema é representado por um estado. Para além disso, cada estado é associada uma gaussiana. Foram utilizados estes parâmetros porque a quantidade de dados disponível é bastante reduzida.

Escalamento

Existe um detalhe técnico na aplicação dos algoritmos apresentados nesta secção, o escalamento das variáveis α e β de forma a prevenir o *underflow*. Isto é, os valores atingidos nestes cálculos podem ser tão baixos que deixam de estar nos limites de precisão da máquina [32]. Após o cálculo de α_t utilizando a equação 3.30 em cada instante t, é aplicado um coeficiente de escalamento c_t da forma [25]:

$$c_t = \frac{1}{\sum_{i=1}^{N} \alpha_t(i)} \tag{3.47}$$

Para o caso da variável $\beta_t(i)$ utiliza-se os mesmos coeficientes utilizados para o escalamento do $\alpha_t(i)$, pois estas variáveis apresentam uma magnitude semelhante.

No cálculo de $P(O|\lambda)$ não se pode simplesmente somar os coeficientes $\alpha_T(i)$, mas pode-se utilizar a seguinte propriedade [25]:

$$\prod_{t=1}^{T} c_t \sum_{i=1}^{N} \alpha_T(i) = 1 \tag{3.48}$$

Assim, temos [25]:

$$\log[P(O|\lambda)] = -\sum_{t=1}^{T} \log(c_t) \tag{3.49}$$

O logaritmo da probabilidade pode ser calculado, mas não a probabilidade, pois esta está fora dos limites de precisão da máquina.

Neste capítulo foram abordados dois algoritmos de classificação dinâmicos. Estes algoritmos têm a função de classificar os segmentos do sinal da fala, sendo que a aplicação destes algoritmos será explicada com maior detalhe no capítulo 5. No próximo capítulo serão abordados mais dois algoritmos de classificação, mas têm a característica de serem estáticos. Isto é, não possuem a capacidade de modelar uma evolução temporal de um sistema.

Capítulo 4

Algoritmos de Classificação Estáticos

Neste capítulo vão ser abordados dois algoritmos de classificação estáticos muitos conhecidos, Máquinas de Vetores de Suporte (*Support Vector Machines*) e Florestas Aleatórias (*Random Forests*). Estes algoritmos são considerados estáticos, pois não foram desenvolvidos para contemplar a evolução temporal de um sistema. No âmbito desta dissertação estes algoritmos serão utilizados para a classificação das séries temporais.

4.1 Máquinas de Vetores de Suporte

As MVSs apresentam como função separar um conjunto de dados em diferentes classes. Neste trabalho vão ser apresentadas inicialmente tarefas de classificação binária (duas classes), posteriormente as MVSs multi-classe e por fim MVSs de uma classe.

4.1.1 MVS de Classificação Binária

A figura 4.1 ilustra um exemplo em que se pretende classificar dados em duas classes. O objetivo do método é definir um hiperplano que permite estabelecer esta classificação. Neste caso os dados são linearmente separáveis, pelo que conseguem ser perfeitamente separados através de um hiperplano.

A tarefa de classificação pode ser formalizada da seguinte forma [33]. Dadas l amostras de treino $\boldsymbol{x_i}$ ($i = 1, \cdots, l$), onde cada amostra é definida num espaço a N dimensões ($x_i \in \Re^N$) e para as quais são conhecidas as classificações $y_i \in \{-1, 1\}$, pretende-se determinar uma fronteira de decisão, definida por um hiperplano cuja equação é dada por,

$$\boldsymbol{w^T} \cdot \boldsymbol{x} + b = 0 \tag{4.1}$$

onde $\boldsymbol{w^T}$ e b são parâmetros a determinar.

Os parâmetros $\boldsymbol{w}$ e b são redimensionados de forma a que os elementos de treino satisfaçam as seguintes condições:

$$\boldsymbol{w^T} \cdot \boldsymbol{x_i} + b \geq 1 \text{ se } y_i = 1 \tag{4.2}$$

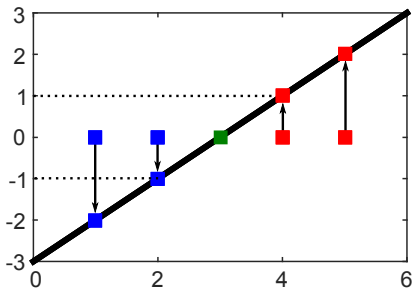


Figura 4.1: Representação de um espaço de decisão a partir de vetores de entrada em $\Re^1$, com duas classes diferentes (azul representa a classe -1 e vermelho representa a classe 1). Como os vetores de entrada pertencem a $\Re^1$, o hiperplano é reduzido a um ponto (representado a verde). A reta preta representa a função de decisão.

$$\boldsymbol{w^T} \cdot \boldsymbol{x_i} + b \leq -1 \text{ se } y_i = -1 \tag{4.3}$$

Estas duas inequações podem ser expressas conjuntamente da seguinte forma:

$$y_i(\boldsymbol{w^T} \cdot \boldsymbol{x_i} + b) \geq 1 \tag{4.4}$$

Há pelo menos um ponto de cada classe para os quais a inequação (4.4) se torna uma igualdade, exatamente devido ao reescalamento realizado.

A determinação de $\boldsymbol{w}$ e b, pode ser vista como um problema de otimização em que se pretende maximizar a margem (M) que separa ambas as classes.

$$\begin{aligned} M &= \min_{x:y=1} \left(\boldsymbol{x} \cdot \frac{\boldsymbol{w}}{||\boldsymbol{w}||} \right) - \max_{x:y=-1} \left(\boldsymbol{x} \cdot \frac{\boldsymbol{w}}{||\boldsymbol{w}||} \right) \\ M &= \left(\frac{1-b}{||\boldsymbol{w}||} \right) - \left(\frac{-1-b}{||\boldsymbol{w}||} \right) \end{aligned} \tag{4.5}$$

Conclui-se pelas equações 4.4 e 4.5 que $M = \frac{2}{||\boldsymbol{w}||}$. O objetivo é maximizar a margem o que é equivalente a minimizar $\frac{||\boldsymbol{w}||}{2}$, ou ainda, $\frac{||\boldsymbol{w}||^2}{2}$, o que nos permite traduzir o problema num problema de programação quadrática (PQ):

$$\begin{aligned} &\min_{\boldsymbol{w},b} \tfrac{1}{2}||\boldsymbol{w}||^2 \\ &\text{s.a } y_i(\boldsymbol{w^T} \cdot \boldsymbol{x_i} + b) \geq 1 \end{aligned} \tag{4.6}$$

As restrições impostas asseguram que não existam elementos de treino entre as margens de separação das classes e, por essa razão, denominam-se estes métodos como MVSs de margens rígidas.

A função objetivo que está a ser minimizada é convexa e os pontos que satisfazem as restrições formam um conjunto convexo, assim, o problema possui apenas um mínimo global [33]. Os problemas deste tipo podem ser resolvidos com a introdução de uma função de

Lagrange, que reúne as restrições à função objetivo, associadas a parâmetros designados de multiplicadores de Lagrange (α_i). Temos então a seguinte função de Lagrange:

$$\mathcal{L}(\boldsymbol{w}, b, \alpha) = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_i \alpha_i[y_i(\boldsymbol{w^T} \cdot \boldsymbol{x_i} + b) - 1] \tag{4.7}$$

A partir da equação anterior, podemos definir as condições de Karush-Kuhn-Tucker (KKT):

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \Leftrightarrow \boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x_i} \tag{4.8}$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Leftrightarrow \sum_i \alpha_i y_i = 0 \tag{4.9}$$

$$y_i(\boldsymbol{w^T} \cdot \boldsymbol{x_i} + b) \geq 1 \tag{4.10}$$

$$\alpha_i[y_i(\boldsymbol{w^T} \cdot \boldsymbol{x_i} + b) - 1] = 0 \tag{4.11}$$

A condição KKT (4.11) certifica que quando a inequação (4.10) é uma igualdade, $\alpha_i \neq 0$. Porém, quando existe desigualdade na inequação (4.10), $\alpha_i = 0$. Desta forma os pontos que assumem importância estão na margem, designando-se de vetores de suporte, enquanto, os demais serão irrelevantes. Consequentemente, verifica-se que o vetor $\boldsymbol{w}$ é apenas a combinação linear de todos os vetores de suporte. Substituindo as equações (4.8) e (4.9) na equação (4.7), podemos definir o problema dual da seguinte forma:

$$\begin{gathered} \max_{\alpha} \sum_{i=1}^{l} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j (\boldsymbol{x^T} \cdot \boldsymbol{x}) \\ \text{s.a } \sum_i^l \alpha_i y_i = 0 \\ \alpha_i \geq 0, \forall i \end{gathered} \tag{4.12}$$

A forma dual é vantajosa, na medida em que, apresenta restrições mais simples e permite a definição do problema de otimização em termos de produtos internos entre os dados, o que será útil nas MVSs não lineares, que será apresentado na secção 4.1.1. Como resultado final, obtém-se a seguinte função de decisão:

$$f(x) = sign\left(\sum_i \alpha_i y_i (\boldsymbol{x_i} \cdot \boldsymbol{x}) + b\right) \tag{4.13}$$

Na maior parte das situações os dados não são perfeitamente separáveis linearmente, como se considerou anteriormente. A generalização das MVSs para casos mais gerais tornou-se crucial para tornar o método aplicável a situações práticas. Isso foi conseguido usando duas ideias principais: o relaxamento das restrições do problema de otimização, permitindo que alguns pontos violem as margens e fazendo com que estas deixem de ser tão rígidas; a definição de um mapeamento não linear dos dados, permitindo transformar conjuntos de pontos não linearmente separáveis em conjuntos linearmente separáveis. Na secção seguinte discute-se como podem ser introduzidas margens suaves nas MVSs.

MVSs de Margens Suaves

Para introduzir margens suaves, considera-se que alguns dos pontos sejam classificados erradamente. Neste caso, o problema de programação quadrática é modificado, acrescentando variáveis de folga, para cada x_i. Estas novas condições possibilitam que um elemento esteja na margem de erro, caso se constate que $0 \leq \varepsilon_i \leq 1$ ou então seja classificado erradamente, se $\varepsilon_i \geq 1$. Isto pode ser implementado adicionando um termo $C \sum_i \varepsilon_i$ à função objetivo de forma a penalizar as classificações erradas, embora permitindo-as. Consequentemente, o problema PQ assume-se desta forma:

$$\begin{gathered} \min_{\boldsymbol{w},b} \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_i \varepsilon_i \\ \text{s.a } y_i(\boldsymbol{w^T} \cdot \boldsymbol{x_i} + b) \geq 1 - \varepsilon_i \\ \varepsilon_i \geq 0, \forall i \end{gathered} \tag{4.14}$$

A constante C pode ser considerada um parâmetro ajustável, dado que, para valores altos de C, impõe-se que todos os dados de instrução sejam classificados corretamente, enquanto que para valores baixos de C, o hiperplano torna-se mais flexível, minimizando a margem de erro para cada elemento[33].

A função de Lagrange para resolver este problema de otimização pode ser obtida seguindo passos muito semelhantes aos apresentados anteriormente no caso das MVSs de margens rígidas. O resultado pode ser descrito com o seguinte problema dual:

$$\begin{gathered} \max_{\alpha} \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j (\boldsymbol{x^T} \cdot \boldsymbol{x}) \\ \text{s.a } \sum_{i}^{l} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C, \forall i \end{gathered} \tag{4.15}$$

MVSs Não Lineares

Nem sempre é possível separar os dados de forma linear, como se pode observar na figura 4.2. No entanto, Cortes e Vapnik [33] introduziram um truque que permitiu fazer esta separação numa grande generalidade dos casos. Foi este o passo decisivo que tornou as MVSs uma técnica muito popular. A ideia consiste em mapear os pontos iniciais (vetores $\boldsymbol{x}$ de $\Re^N$) num espaço de maiores dimensões (vetores $\phi(\boldsymbol{x})$ de $\Re^{\tilde{N}}$). A função de mapeamento $\phi(\boldsymbol{x})$ terá que ser escolhida especificamente para que os dados de treino possam ser separados linearmente.[33]. A aplicação deste procedimento é fundamentada pelo teorema de Cover[34]. Este teorema afirma que a probabilidade das classes serem linearmente separáveis aumenta quando as características são não-linearmente mapeadas para um espaço (de características) de maior dimensão.

O objetivo do problema de otimização continua a ser minimizar $\frac{||\boldsymbol{w}||^2}{2}$, contudo, como consequência do que foi referido no parágrafo anterior, a equação (4.8) utilizada anteriormente, altera-se para:

$$\boldsymbol{w} = \sum_i \alpha_i y_i \phi(\boldsymbol{x_i}) \tag{4.16}$$

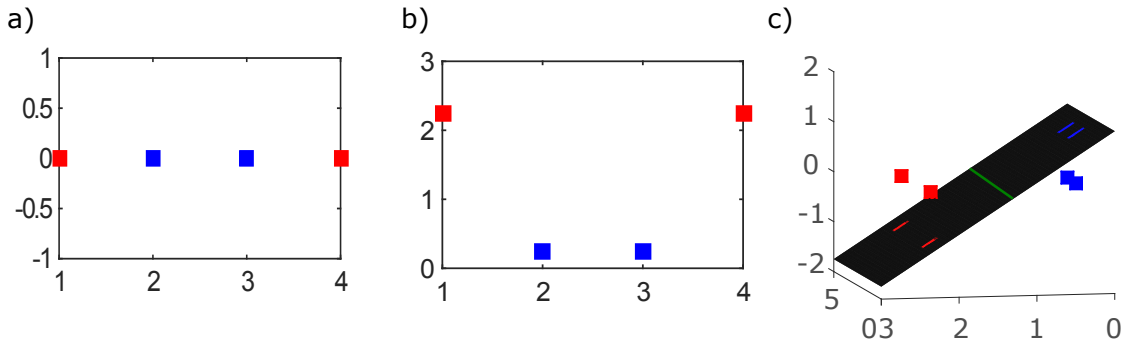


Figura 4.2: Nesta figura está representado o processo de separação dos dados de forma linear, após um mapeamento dos mesmos num espaço de maior dimensão. Na imagem a) estão representados os vetores de entrada $\Re^1$, pertencentes a duas classes diferentes (azul representa a classe 1 e vermelho representa a classe -1). Estes vetores são mapeados num espaço $\Re^2$, através da função $\Phi(x) = [x, (x-2.5)^2]$, como se pode observar pela imagem b). Após este mapeamento, já é possível separar os dados linearmente como se pode observar na imagem c). Através da imagem c), observa-se que o hiperplano está a reduzido a uma reta (representada a verde). A função de decisão está definida em $\Re^3$ e é representada pelo plano de cor preta.

No mesmo sentido, a equação (4.13) será modificada ficando:

$$f(x) = sign\left(\sum_i \alpha_i y_i K(\boldsymbol{x_a}, \boldsymbol{x_b}) + b\right) \tag{4.17}$$

em que $K(\boldsymbol{x_a}, \boldsymbol{x_b})$ designa-se por *kernel* e pode ser definido pela seguinte expressão:

$$K(\boldsymbol{x_a}, \boldsymbol{x_b}) = \phi(\boldsymbol{x_a}) \cdot \phi(\boldsymbol{x_b}) \tag{4.18}$$

Assim, para encontrar o hiperplano ótimo procede-se da mesma forma que foi explicada anteriormente.

A escolha dos *kernels* adequados a cada caso torna-se crucial para uma boa classificação dos dados. Os exemplos de *kernels* mais utilizados são os seguintes:

$$K(\boldsymbol{x_a}, \boldsymbol{x_b}) = (\boldsymbol{x_a^T} \cdot \boldsymbol{x_b} + c)^P \ \ Polinomial \tag{4.19}$$

$$K(\boldsymbol{x_a}, \boldsymbol{x_b}) = exp\left(-\frac{||\boldsymbol{x}_a^T \cdot \boldsymbol{x}_b||^2}{\gamma}\right) \ \ Gaussiano \tag{4.20}$$

O *kernel* adequado deve ser escolhido com base nos dados a que se tem acesso, dependendo de caso para caso. Uma nota importante, é que com a utilização dos *kernels*, o espaço original não é efetivamente mapeado num espaço de maiores dimensões. Isto é, os dados são mapeados implicitamente, pois lida-se apenas com os produtos internos dos vetores mapeados. Os mapeamentos nem sempre são possíveis de realizar ao contrário dos *kernels*, como por exemplo, no caso do *kernel* gaussiano que mapeia implicitamente os dados num espaço $\Re^\infty$.

4.1.2 MVS Multi-classe

Existem vários métodos para o problema das MVS de multi-classe, como por exemplo, *one-against-all, one-against-one* e *DAGSVM* [35]. Porém, neste trabalho, o único método que se vai abordar é designado de *one-against-one*.

Dado que existem k classes, então são construídos $k(k-1)/2$ classificadores (hiperplanos), e cada um é treinado com dados de duas classes. Para os dados de treino das classes i e j é resolvido o seguinte problema de classificação binária [35] [36]:

$$\begin{aligned} \min_{\boldsymbol{w^{ij}},b^{ij},\varepsilon^{ij}} \ & \tfrac{1}{2}||\boldsymbol{w^{ij}}||^2 + C\sum_t \varepsilon_t^{ij} \\ \text{s.a } (\boldsymbol{w}^{ij})^T \cdot \phi(\boldsymbol{x}_t) + b^{ij}) &\geq 1 - \varepsilon_t^{ij} \quad \text{se } y^{ij} = i \\ (\boldsymbol{w}^{ij})^T \cdot \phi(\boldsymbol{x}_t) + b^{ij}) &\leq -1 + \varepsilon_t^{ij} \quad \text{se } y^{ij} = j \\ \varepsilon_t^{ij} &\geq 0, \forall t \end{aligned} \tag{4.21}$$

Na classificação é utilizada uma estratégia de votação, ou seja, se $\text{sign}\big((\boldsymbol{w}^{ij})^T \cdot \phi(\boldsymbol{x}_t) + b^{ij})\big) > 0$ então é acrescentado um voto para a classe i. Caso contrário, é acrescentado um voto para a classe j. Por fim, é atribuída a $\boldsymbol{x}$ a classe que tiver maior votação [35].

4.1.3 MVS de Uma Classe

As máquinas de vetores de suporte de uma classe são algoritmos de deteção de anomalias. A função destes algoritmos é detetar conjuntos de dados que sejam semelhantes ou diferentes dos dados de treino.

Considere-se um conjunto de dados de treino constituído por l amostras $\boldsymbol{x_i}$ $(i = 1, \cdots, l)$, onde cada amostra é definida num espaço a N dimensões ($\boldsymbol{x_i} \in \Re^N$), designado de espaço de entrada I [37]. A função não linear Φ mapeia os dados $\boldsymbol{x_i}$ num espaço de maior dimensão, designado de espaço das características F. Este mapeamento é realizado por meio de um *kernel*.

O objetivo é definir uma função f que tome valor 1 numa determinada região que contenha a maior parte dos dados de treino e valor -1 no espaço restante [37]. Neste sentido, é definido um hiperplano no espaço das características F da seguinte forma [37]:

$$\boldsymbol{w} \cdot \Phi(\boldsymbol{x}) - \rho = 0 \tag{4.22}$$

Em que $\boldsymbol{w}$ e ρ são parâmetros do hiperplano que se pretende determinar. De forma a encontrar o hiperplano ótimo que afasta maximamente os dados de treino da origem, é utilizado o seguinte problema de programação quadrática [37]:

$$\begin{aligned} \min_{\boldsymbol{w},\rho,\varepsilon} \ & \tfrac{1}{2}||\boldsymbol{w}||^2 + \tfrac{1}{\nu l}\sum_i \varepsilon_i - \rho \\ \text{s.a } (\boldsymbol{w} \cdot \Phi(\boldsymbol{x_i})) &\geq \rho - \varepsilon_i \\ \varepsilon_i &\geq 0, \forall i \end{aligned} \tag{4.23}$$

Como em muitos casos os dados não são completamente separáveis, são introduzidas variáveis de folga ε de modo a permitir que alguns dos dados sejam classificados erradamente, ou seja, como anomalias. Porém, existe o parâmetro $\nu \in]0, 1]$ que penaliza essas variáveis de folga na função objetivo. O parâmetro ν pode ser visto como o limite superior da fração de anomalias [37].

Neste problema de otimização é introduzido um Lagrangeano, para facilitar a sua resolução.

$$\mathcal{L}(\boldsymbol{w}, \rho, \alpha, \beta) = \frac{1}{2}||\boldsymbol{w}||^2 + \frac{1}{\nu l}\sum_i \varepsilon_i - \rho - \sum_i \alpha_i(\boldsymbol{w}\cdot\Phi(\boldsymbol{x_i}) - \rho + \varepsilon_i) - \sum_i \beta_i\varepsilon_i \tag{4.24}$$

Para resolver o Lagrangeano é necessário colocar as derivadas de ordem às variáveis w, ε, ρ a zero.

$$\frac{\partial\mathcal{L}}{\partial w} = 0 \Leftrightarrow \boldsymbol{w} = \sum_i \alpha_i\Phi(\boldsymbol{x_i}) \tag{4.25}$$

$$\frac{\partial\mathcal{L}}{\partial\varepsilon} = 0 \Leftrightarrow \alpha_i = \frac{1}{\nu l} - \beta_i \tag{4.26}$$

$$\frac{\partial\mathcal{L}}{\partial\rho} = 0 \Leftrightarrow \sum_i \alpha_i = 1 \tag{4.27}$$

Substituindo 4.25, 4.26, 4.27 em 4.24 obtém-se o seguinte problema dual [37]:

$$\begin{aligned} \max_{\alpha} & -\tfrac{1}{2}\sum_{ij}\alpha_i\alpha_j\Phi(\boldsymbol{x_i})\cdot\Phi(\boldsymbol{x_j}) \\ & \text{s.a } \sum_i \alpha_i = 1 \\ & 0 \le \alpha_i \le \tfrac{1}{\nu l}, \forall i \end{aligned} \tag{4.28}$$

A introdução do Lagrangeano trás vantagens nomeadamente pelo fato de contemplar apenas restrições nos multiplicadores de Lagrange (α_i) e os dados de treino $\boldsymbol{x_i}$ vão aparecer sob a forma de produtos internos, que é uma condição necessária para utilizar os *kernels*.

Por fim, a função de decisão que vai classificar os dados é dada por [37]:

$$f(\boldsymbol{x}) = sign\left(\sum_i \alpha_i K(\boldsymbol{x_i}, \boldsymbol{x}) - \rho\right) \tag{4.29}$$

4.2 Florestas Aleatórias

As Florestas Aleatórias (FA) são um método de classificação e regressão baseado num conjunto de árvores de decisão. A floresta cresce à medida que são construídas as árvores, com base em decisões binárias que vão segmentar o espaço de entrada [38]. Note-se que neste trabalho apenas vai ser apresentada a componente de classificação de variáveis contínuas.

Considere um espaço com N dimensões $X = (X_1, X_2, \cdots, X_N)$ que representa as variáveis de entrada e Y a classificação. O objetivo é encontrar uma função de previsão $f(\boldsymbol{x})$ para prever Y. As árvores de decisão são os classificadores individuais $h(\boldsymbol{x})$ que combinados vão construir a função de decisão $f(\boldsymbol{x})$. A função de decisão estabelece que a classe prevista é a mais frequente, e é definida por [39]:

$$f(\boldsymbol{x}) = \underset{y\ \in\ \mathscr{Y}}{\arg\max}\left[\sum_{j=1}^{J} I_y(h_j(\boldsymbol{x}))\right] \tag{4.30}$$

em que $I(\cdot)$ é a função indicadora, J é o número total de árvores e $\mathscr{Y}$ é o conjunto de todas as classes. As árvores de decisão bem como as estratégias usadas na construção das FAs serão discutidas de seguida.

4.2.1 Construção de Árvores de Decisão

As árvores de decisão são construidas usando uma partição binária recursiva. Neste sentido, as árvores segmentam o espaço de entrada utilizando uma sequência de partições binárias nas variáveis de entrada [39]. Cada nodo é dividido em dois nodos descendentes, um para a esquerda e outro para a direita, dependendo do valor de uma das variáveis de entrada. Para uma variável contínua, a divisão é determinada por um ponto de divisão P_d numa das variáveis de entrada X_i [39]. Um exemplo deste mecanismo está exposto na figura 4.3. Assim, os vetores de entrada $\boldsymbol{X}$ que possuem um valor nessa variável mais baixo que o ponto de divisão vão para o nodo da esquerda e os restantes para o nodo da direita [39]. Este processo é repetido até se atingir um nodo terminal, também designado de folha.

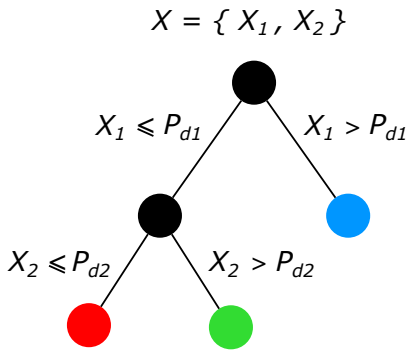


Figura 4.3: Representação de uma árvore de decisão arbitrária. Neste exemplo, o espaço de entrada X é constituído por duas variáveis, X_1 e X_2. A classificação é realizada a partir de dois pontos de divisão arbitrários, P_{d1} e P_{d2}. Neste caso, as folhas (classes) estão representadas por cores: azul, verde e vermelho.

Para realizar uma partição é tido em conta todas as partições possíveis de todas a variáveis, e é escolhida a melhor de acordo com um critério. No contexto de classificação é utilizado o índice de Gini. Dado que existem K classes, a frequência de uma classe estar num nodo pode ser dada por

$$p_k = \frac{n_k}{\sum_{k=1}^{K} n_k} \tag{4.31}$$

em que n_k é o número de elementos da classe k que existe nesse nodo. Desta forma, pode-se definir o índice de Gini como:

$$G = 1 - \sum_{k=1}^{K} p_k^2 \tag{4.32}$$

O índice de Gini oferece uma medida de impureza do nodo. O objetivo é encontrar um par $\{X_i, P_d\}$ que minimize a impureza dos dois nodos descendentes, o que pode ser feito através da minimização [39]:

$$\min_{X_i, P_d} \left[G_{N_1} + G_{N_2} \right] \tag{4.33}$$

em que G_{N_1} e G_{N_2} são o índice de Gini dos subconjuntos de vetores de entrada $\boldsymbol{X}$ presentes no nodo da esquerda e da direita, respetivamente. Os dois subconjuntos são definidos da

forma:

$$N_1 = \{\boldsymbol{X} | X_i \leq P_d\}, \quad N_2 = \{\boldsymbol{X} | X_i > P_d\} \tag{4.34}$$

Após ser realizada a partição, utiliza-se o mesmo processo para os nodos subsequentes. Pode-se estabelecer que o algoritmo para quando a impureza for inferior a um valor ε pré-estabelecido.

4.2.2 Construção da Floresta Aleatória

Nas Florestas Aleatórias a $j-$ésima árvore de decisão é definida por $h_j(X, \Theta_j)$ em que Θ_j é uma componente aleatória, sendo essa componente independente para cada árvore. Na prática a componente Θ é inserida implicitamente de duas formas, com a aplicação do Bootstrap Aggregating (Bagging) e uma limitação do número de variáveis utilizadas para as divisões dos nodos [39].

O Bagging utiliza um procedimento de amostragem dos vetores de entrada designado de *bootstrap* [40]. Neste tipo de amostragem são selecionados aleatoriamente e com reposição vetores do conjunto de entrada e inseridos num novo subconjunto. Deste modo, são construídos diferentes subconjuntos a partir do conjunto de entrada original. Como consequência da reposição, existe a possibilidade de alguns vetores de entrada se repetirem e outros não serem escolhidos. Note-se que estes subconjuntos apresentam o mesmo tamanho que o conjunto de entrada original. Os vetores de entrada não selecionados por Bagging formam o conjunto *out-of-bag* (oob) e são utilizados para estimar o erro de generalização [40]. O erro pode ser estimado da forma [39]:

$$E_{oob} = \frac{1}{L} \sum_{i=1}^{L} I_{\neq y_i}(f_{oob}(\boldsymbol{x}_i)) \tag{4.35}$$

em que L é o número de vetores de entrada e $f_{oob}(\boldsymbol{x}_i)$ é a previsão *out-of-bag* para o vetor $\boldsymbol{x}_i$.

Como foi referido, é acrescentada mais uma componente de aleatoriedade na construção das árvores, dado que, apenas se considera em cada nodo uma parte das variáveis dos vetores de entrada selecionados por Bagging. Isto é, considerando N variáveis, é comum serem selecionadas $\sqrt{N}$ aleatoriamente [38].

O Bagging mais a componente de aleatoriedade na construção das árvores impede o sobreajuste dos dados, pois na construção das árvores de decisão nunca é utilizada toda a informação dos dados de treino.

Neste capítulo foram abordados dois algoritmos de classificação estáticos, e consequentemente, não permitem vetores de entrada com tamanho variável. No entanto, foi utilizada uma estratégia que permitiu obter vetores de entrada com o mesmo tamanho e ao mesmo tempo que contemplam a evolução temporal. As estratégias utilizadas serão apresentadas no próximo capítulo.

Capítulo 5

Simulações: Avaliação do Desempenho do Utilizador

Todas as técnicas descritas nos capítulos anteriores serão aplicadas com vista a avaliar o desempenho fonético em exercícios de dicção. Neste capítulo será descrita a aplicação das técnicas de processamento de sinal e de classificação com vista a atingir este objetivo.

5.1 Descrição dos testes realizados

De forma a testar a capacidade de reconhecimento dos algoritmos de classificação foram realizados testes com três níveis de complexidade diferentes. Primeiro testou-se a capacidade de reconhecimento de um conjunto de vogais orais (/a/, /ɛ/, /i/, /ɔ/, /u/). De seguida, testou-se a capacidade de reconhecimento de palavras isoladas (dígitos de zero até cinco) e depois de palavras não isoladas. Neste último teste, utilizaram-se os conjuntos de palavras "Engenharia Física" (EF) e "Universidade de Aveiro" (UA), em que o objetivo foi detetar a presença das palavras integrantes de cada conjunto.

Os exercícios foram realizados por sete indivíduos do sexo masculino, tendo cada exercício sido repetido quatro vezes. Os exercícios de cinco dos indivíduos foram utilizados para treinar os algoritmos e os exercícios dos outros dois indivíduos para testar.

Os primeiros dois testes pretenderam fazer uma avaliação da capacidade do algoritmo não fazer uma classificação errada, por exemplo confundindo vogais. Nesse sentido, aos diversos algoritmos foram apresentados 8 exemplos de cada uma das 5 vogais e 8 exemplos de cada dígito, tendo sido contabilizada a taxa de acerto independentemente do dígito ou da vogal apresentada.

No terceiro teste os diversos algoritmos procuraram determinar a localização de cada uma das palavras no conjunto apresentado. Foram apresentados 8 exemplos de cada conjunto. A localização foi considerada acertada se a região definida pelo algoritmo contivesse a palavra que se pretende encontrar. A região correta das palavras foi determinada manualmente através da audição do registo áudio. A taxa de acerto de cada algoritmo considerou também o número de localizações corretamente atribuídas, independentemente da palavra considerada.

5.2 Processamento de Sinal

Cada registo áudio foi realizado com o software Audacity. Posteriormente, importaram-se os dados para o MATLAB® e converteram-se em vetores de características, como se ilustra na figura 5.1.

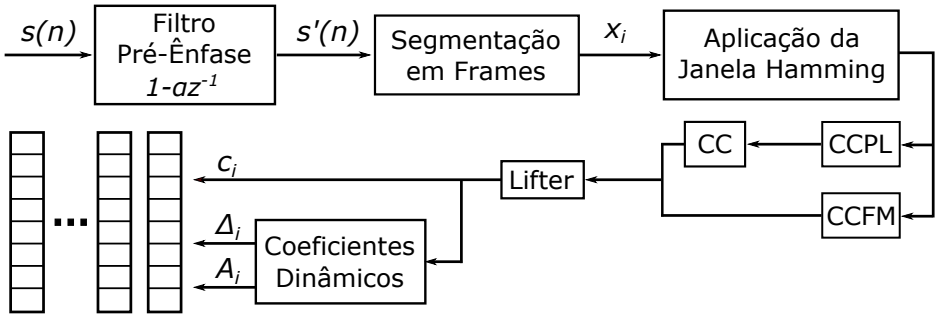


Figura 5.1: Representação da análise espetral realizada para a extração das características das séries temporais.

O processamento inicia-se com a aplicação do filtro pré-ênfase, com o objetivo de uniformizar o espetro de frequência, tal como referido na secção 2.3. Foi utilizado um valor típico, $\alpha = 0.97$. Posteriormente, o sinal é segmentado em *frames*, tendo sido aplicada uma janela Hamming para reduzir efeito de vazamento espetral. Neste caso, foram utilizadas janelas de 25 ms e um deslocamento temporal de 10 ms, o que resulta numa sobreposição 60%. De seguida, são extraídos coeficientes estáticos em cada *frame*. Existem várias técnicas de parametrização, mas neste trabalho apenas foram abordados os coeficientes *cepstral* (CC) derivados da codificação preditiva linear (CPL) e os coeficientes *cepstral* de frequência de Mel (CCFM). É de salientar que os CC e CCFM não são utilizados em simultâneo, são duas formas alternativas de parametrização do sinal. Neste trabalho foram utilizados 13 coeficientes (CC ou CCFM) para parametrizar cada *frame*. Após a extração dos coeficientes estáticos é aplicado um *lifter*. O valor do parâmetro de *lifter* escolhido foi igual ao número de coeficientes, ou seja, 13 [17]. Por fim, foram extraídos os coeficientes dinâmicos (CD), delta e aceleração, para contemplar a evolução temporal dos coeficientes estáticos. Os coeficientes CC ou CCFM, com ou não evolução dinâmica (CD) definem os vetores de características utilizados nos diversos métodos. Para o cálculo dos coeficientes CPL e CFM foram utilizadas rotinas disponibilizadas em [41] [42].

5.3 Aplicação das Técnicas de Classificação

5.3.1 Aplicação da Distorção Temporal Dinâmica

No caso dos exercícios para identificar as vogais orais e palavras isoladas, utilizou-se a distorção temporal dinâmica (DTD) como métrica discriminante: para cada exercício de teste calculou-se a distância média aos exercícios de treino de cada classe, atribuindo-se ao exercício a classe (y) à menor distância, como se ilustra na figura 5.2 e é traduzido na fórmula:

$$y = \underset{c \in \mathscr{C}}{\arg\min} \left[\langle D_c(S) \rangle \right] \tag{5.1}$$

em que $\langle D_c(S)\rangle$ é a distância média entre um exercício de teste S e os exercícios de treino da classe c e $\mathscr{C}$ é o conjunto de todas as classes.

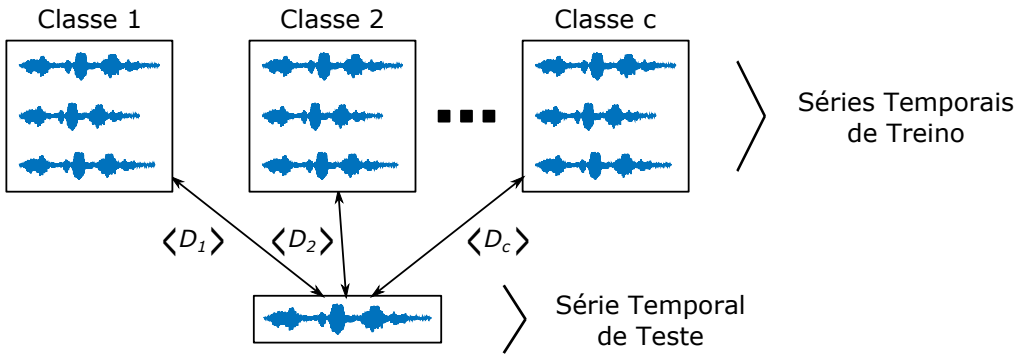


Figura 5.2: Ilustração do processo de classificação utilizando a DTD como métrica discriminante. É calculada a distância média $\langle D_c\rangle$ entre um exercício de teste e os exercícios de treino das várias classes e atribuída a classe à menor distância média.

No caso das palavras não isoladas, a posição de cada palavra foi encontrada usando uma janela móvel de tamanho igual ao tamanho médio (L_w) da palavra que se pretende detetar e determinando a janela que minimizou a distância (distorção) média:

$$T_c = \underset{i \in I}{\arg\min} \left[\langle D_{i,c}(W_i)\rangle\right], \quad I = 1, 2, \cdots, L_S - L_w \tag{5.2}$$

em que T_c é a posição da palavra que define a classe c, $\langle D_{i,c}(W_i)\rangle$ é a distância média entre a janela (W_i) com inicio em i e as palavras de treino de classe c e L_S é o tamanho total do exercício de teste.

Na tabela 5.1 estão apresentados os resultados obtidos com os diferentes coeficientes, cepstral (CC) e cepstral de frequência de mel (CCFM), e usando ou não, coeficientes dinâmicos (CD), delta e aceleração. Conforme se pode apreciar nestes resultados, os coeficientes CCFM possuem uma taxa de acerto superior aos CC e a introdução dos CDs leva a uma diminuição da taxa de acerto. Esta técnica demonstrou ser bastante imprecisa, principalmente no teste do conjunto UA.

Coeficientes	Coeficientes Dinâmicos	Taxa de Acerto			
		Vogais	Dígitos	EF	UA
CC	Não	28%	33%	87%	32%
CC	Sim	25%	33%	85%	30%
CCFM	Não	85%	94%	82%	42%
CCFM	Sim	85%	92%	78%	38%

Tabela 5.1: Resultados obtidos utilizando a DTD no reconhecimento de vogais, dígitos e as palavras que compõem os conjuntos Engenharia Física (EF) e Universidade de Aveiro (UA).

5.3.2 Aplicação das Máquinas de Vetores de Suporte

Na aplicação das Máquinas de Vetores de Suporte (MVSs) enfrentou-se uma grande dificuldade, isto é, o algoritmo assume que as observações são representadas por vetores de

entrada de dimensão fixa. Para contornar este problema, foi utilizado o processo de extração de vetores de entrada que está ilustrado na figura 5.3. Nesse sentido, cada exercício foi dividido em cinco partes iguais, sendo que foi extraído um vetor de características de cada uma das partes para a construção de cada vetor de entrada. Cada um dos vetores de entrada contém informação à cerca da evolução temporal do exercício.

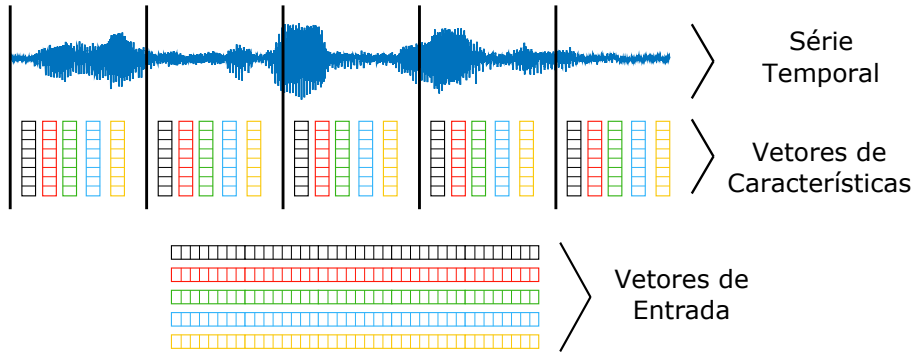


Figura 5.3: Ilustração do processo da extração de vetores de entrada. Neste exemplo, a série está dividida em cinco partes iguais. Em cada uma das partes foi extraído um vetor de características para a construção de cada vetor de entrada. Note-se que o número de partes em que se divide a série temporal não é sempre igual ao número de vetores de entrada.

No caso dos exercícios para identificar as vogais orais e palavras isoladas, a classificação atribuída foi dada pela classe dos vetores de entrada mais frequente.

No caso das palavras não isoladas, a posição de cada palavra foi encontrada usando uma janela móvel de tamanho igual ao tamanho médio (L_w) da palavra que se pretende detetar e determinando a janela que maximiza a soma dos valores de decisão médios:

$$T_c = \arg\max_{i \in I} \left[\sum_{k=1}^{K} C_k \langle DV_{i,c,k} \rangle \right], \quad I = 1, 2, \cdots, L_S - L_w \tag{5.3}$$

em que $\langle DV_{i,c,k} \rangle$ é o valor de decisão k médio da janela com em início em i que contribui diretamente na atribuição da classe c e $C_k \in \{-1, 1\}$ indica se $\langle DV_{i,c,k} \rangle$ contribui positiva ou negativamente em relação à classe c. O valor de K varia dependendo do número de classes que se considera num determinado conjunto de palavras. Concretamente, o valor de K no conjunto EF é igual a 1 e no conjunto UA é igual a 2.

Na aplicação das MVSs de uma classe, em todos os testes, foi construída uma função de decisão para cada classe. Foi considerado que se fosse obtido um valor de decisão positivo dada uma função de decisão, então haveria uma deteção. No entanto, na aplicação das MVSs de uma classe não foi possível obter qualquer reconhecimento. Uma possível razão é o facto da quantidade de dados de treino ser reduzida, resultando na construção de uma função de decisão não apropriada. Em todos os testes, foram utilizadas as rotinas disponibilizadas em [43].

Nas tabelas 5.2 e 5.3 estão apresentados os resultados obtidos com os diferentes coeficientes, cepstral (CC) e cepstral de frequência de mel (CCFM), e usando ou não, coeficientes dinâmicos (CD), delta e aceleração. O *kernel* utilizado em todos os testes foi o gaussiano.

Conforme se pode apreciar nestes resultados, os coeficientes CCFM possuem uma taxa de acerto superior aos CC e a introdução dos CDs leva a uma diminuição da taxa de acerto.

Coeficientes	Coeficientes Dinâmicos	γ	Taxa de Acerto
CC	Não	0.1	50%
CC	Sim	0.1	45%
CCFM	Não	0.001	90%
CCFM	Sim	0.001	90%

Tabela 5.2: Resultados das MVSs na classificação de vogais orais. O *kernel* utilizado foi o gaussiano, sendo que o parâmetro γ utilizado para cada teste está indicado na tabela.

Coeficientes	Coeficientes Dinâmicos	Taxa de Acerto		
		Digitos	EF	UA
CC	Não	75%	87%	63%
CC	Sim	71%	87%	58%
CCFM	Não	92%	100%	75%
CCFM	Sim	88%	100%	79%

Tabela 5.3: Resultados obtidos com a aplicação das MVSs na classificação de dígitos e das palavras que compõem os conjuntos EF e UA. O *kernel* utilizado foi o gaussiano, sendo que foi utilizado o parâmetro $\gamma = 0.001$ em todos os testes.

5.3.3 Aplicação das Florestas Aleatórias

Na aplicação das Florestas Aleatórias enfrentou-se a mesma dificuldade das MVSs em relação à assunção do algoritmo de que as observações são representadas por vetores de entrada de dimensão fixa. Consequentemente, foi utilizado o mesmo processo na extração de vetores de entrada, ilustrado na figura 5.3.

Na fase de treino das FAs foi construída uma função de decisão utilizando mil árvores. No caso dos exercícios para identificar as vogais orais e palavras isoladas, a classificação atribuída foi dada pela classe dos vetores de entrada mais frequente.

No caso das palavras não isoladas, a posição de cada palavra foi encontrada usando uma janela móvel de tamanho igual ao tamanho médio (L_w) da palavra que se pretende detetar e determinando a janela que maximiza a probabilidade:

$$T_c = \underset{i \in I}{\arg\max} \left[\langle P_{i,c} \rangle\right], \quad I = 1, 2, \cdots, L_S - L_w \tag{5.4}$$

em que $\langle P_{i,c} \rangle$ é a probabilidade média da janela, com inicio em i, pertencer à classe c.

Na tabela 5.4 estão apresentados os resultados obtidos com os diferentes coeficientes, cepstral (CC) e cepstral de frequência de mel (CCFM), e usando ou não, coeficientes dinâmicos (CD), delta e aceleração. A partir dos resultados obtidos, verifica-se que os coeficientes CCFM possuem uma taxa de acerto superior aos CC e que a introdução dos CDs leva a uma diminuição da taxa de acerto. Esta técnica demonstrou ter as melhores taxas de acerto.

Coeficientes	Coeficientes Dinâmicos	Taxa de Acerto			
		Vogais	Digitos	EF	UA
CC	Não	50%	75%	100%	71%
CC	Sim	40%	63%	94%	67%
CCFM	Não	90%	96%	100%	96%
CCFM	Sim	90%	83%	100%	92%

Tabela 5.4: Resultados obtidos com a aplicação das FAs na classificação de dígitos e das palavras que compõem os conjuntos EF e UA.

5.3.4 Aplicação dos Modelos de Markov Ocultos

Na aplicação dos Modelos de Markov Ocultos (MMOs) foi considerado que cada modelo (λ) apresenta uma estrutura igual à ilustrada na figura 5.4. Nesse sentido, considerou-se que os estados representam os fonemas que são sequencialmente pronunciados, e por essa razão, as sequências de estados apenas podem evoluir numa direção. Devido ao facto da quantidade de dados de treino ser reduzida, os modelos foram construídos da forma mais simples possível, ou seja, associando apenas um estado a cada fonema e apenas uma distribuição gaussiana a cada estado.

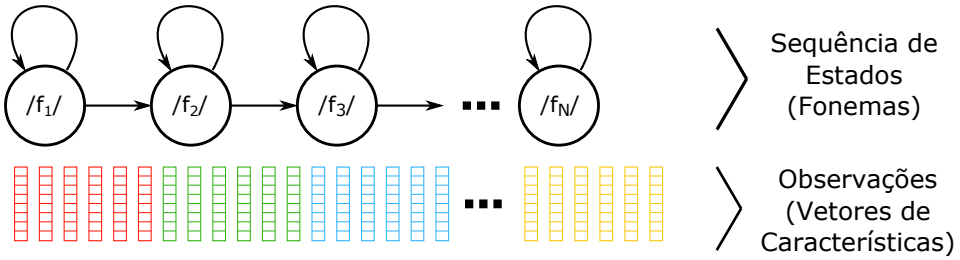


Figura 5.4: Ilustração da estrutura utilizada na construção dos modelos de Markvov ocultos. Nestes modelos, é considerada uma direção pois os fonemas estão encadeados temporalmente. Cada estado representa um fonema (indicado por /$\mathbf{f_n}$/). As observações (vetores de características) associadas a cada fonema estão ilustradas por cores diferentes.

Na fase de treino dos MMOs, são determinados os parâmetros do modelo (λ_c) mais adequados para cada classe c. Como foi mencionado na secção 3.2.1, os MMOs necessitam de uma solução inicial dos parâmetros do modelo, para depois se utilizar o método de reestimação. Neste sentido, foi utilizada uma solução inicial para a matriz de probabilidade transição de estados da forma,

$$A = \begin{bmatrix} 0.9 & 0.1 & 0 & \dots & 0 \\ 0 & 0.9 & 0.1 & \dots & 0 \\ 0 & 0 & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & 0.9 & 0.1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.5}$$

e um vetor da probabilidade de estados iniciais da forma,

$$\pi = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{5.6}$$

Foi escolhida esta solução inicial para os parâmetros A e π pois parece ser adequada tendo em conta o encadeamento temporal dos fonemas. É de salientar que as probabilidade de transição de estados são um indicador da duração de cada fonema. A solução inicial dos parâmetros ($\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$) das distribuições gaussianas é dada por um vetor de características e uma matriz identidade, respetivamente.

No caso dos exercícios para identificar as vogais orais e palavras isoladas, a cada exercício de teste foi atribuída a classe c de acordo com o modelo (λ_c) ao qual tinha a maior probabilidade de pertencer.

No caso das palavras não isoladas, a posição de cada palavra foi encontrada usando uma janela móvel de tamanho igual ao tamanho médio (L_w) da palavra que se pretende detetar e determinando a janela que maximiza o logaritmo da probabilidade:

$$T_c = \underset{i \in I}{\arg\max} \left[\log(P_i(O|\lambda_c) \right], \quad I = 1, 2, \cdots, L_S - L_w \tag{5.7}$$

em que $P_i(O|\lambda_c)$ é a probabilidade da janela, com inicio em i, pertencer ao modelo λ_c. Para aplicação dos MMOs foram utilizadas rotinas disponibilizadas em [44] [45].

Na tabela 5.5 estão apresentados os resultados obtidos com os diferentes coeficientes, cepstral (CC) e cepstral de frequência de mel (CCFM), e usando ou não, coeficientes dinâmicos (CD), delta e aceleração. Durante as simulações apenas se obteve convergência do algoritmo de reestimação utilizando coeficientes estáticos. No caso particular das vogais apenas se obteve convergência para os coeficientes CFM. Apreciando os resultados, conclui-se que os coeficientes CCFM possuem uma taxa de acerto superior aos CC.

Coeficientes	Taxa de Acerto			
	Vogais	Dígitos	EF	UA
CC	-	42%	88%	38%
CCFM	75%	83%	93%	75%

Tabela 5.5: Resultados obtidos com a aplicação dos MMOs na classificação de vogais, de dígitos e das palavras que compõem os conjuntos EF e UA.

5.4 Discussão de Resultados

A partir dos resultados obtidos verifica-se que os coeficientes que atingiram taxas de acerto mais elevadas foram os CCFM. Para além disso, a introdução dos coeficientes dinâmicos nos vetores de características, de forma geral, não melhorou o desempenho dos algoritmos. Contrariamente ao que se previa, a introdução destes coeficientes diminuiu, em termos globais, a taxa de acerto. Esta diminuição pode ter tido várias razões, dependendo da técnica que se está a considerar. No caso da Distorção Temporal Dinâmica, uma possível explicação pode

ter sido o facto de terem sido consideradas ponderações iguais entre coeficientes estáticos e dinâmicos no cálculo das distâncias. De facto, no artigo original em que se propuseram os coeficientes dinâmicos, as ponderações foram diferente [18]. Em relação às Máquinas de Vetores de Suporte e Florestas Aleatórias, concluiu-se que tendencialmente a taxa de acerto diminui com o aumento do número de dimensões dos vetores de entrada. Assim, a introdução dos coeficientes dinâmicos pode piorar os resultados por essa razão. Em relação aos MMOs, a introdução dos coeficientes dinâmicos aumentou consideravelmente o número de dimensões da distribuição da gaussiana, resultando na impossibilidade de convergência do algoritmo de reestimação durante a construção dos modelos. A explicação mais lógica para este resultado deve-se ao fato de que um aumento do número de dimensões da distribuição gaussiana requer uma maior quantidade de dados de treino de forma a parametrizar a distribuição devidamente. Nesse sentido, os dados de treino foram suficientes para parametrizar distribuições gaussianas de 13 dimensões, mas tornaram-se insuficientes para as distribuições de 39 dimensões.

O método de classificação baseado na DTD possui a vantagem de lidar com a evolução dinâmica de uma série temporal de uma forma natural. No entanto, este método apresenta duas grandes desvantagens. Em primeiro lugar, demonstrou-se pelos resultados obtidos que um algoritmo de classificação baseado na sua utilização levará a precisões baixas. Em segundo lugar, o método tem a desvantagem de não contemplar uma fase de treino, tornando o método relativamente lento na fase de teste.

Os algoritmos MVSs e FAs, têm a grande desvantagem de não lidarem com a evolução dinâmica de uma série temporal. Para contornar este problema, foi utilizada uma estratégia em que uma série temporal é mapeada num conjunto de vetores que contêm informação acerca da evolução temporal. No entanto esta estratégia tem uma desvantagem, pois não descreve de uma forma explícita quais as componentes do vetor de entrada que estão associadas a um determinado instante. Consequentemente, os algoritmos podem assumir dependências nas variáveis de entrada, durante a construção da função de decisão, que na realidade não existem. No entanto, uma vantagem que estes algoritmos apresentam é que ao gerarem um função de decisão na fase de treino, permitem obter uma classificação rápida na fase de teste. Analisando os resultados obtidos, verifica-se que estes métodos obtiveram as melhores taxas de classificação de uma forma geral.Nesta consideração foram ignoradas as MVSs de uma classe que não permitiram qualquer reconhecimento das palavras. Este último resultado deve-se ao facto dos dados terem sido insuficientes para construir a função de decisão devidamente.

Os MMOs, à semelhança da DTD, têm a vantagem de lidar com a evolução dinâmica de uma série temporal de uma forma natural. Outra vantagem é que permitem construir um modelo com base na estrutura fonética da série temporal. Isto é, associa-se à sequência de fonemas, uma sequência de estados. No entanto, este método possui algumas limitações. A maior limitação é a assunção de que observações (vetores de características) sucessivas são independentes. Outra limitação é a assunção de que as observações podem ser bem representadas por uma mistura de distribuições gaussianas. Por fim, a assunção de Markov, em que a probabilidade de estar num estado num determinado instante depende apenas do estado anterior, é inapropriada pois os sons podem conter dependências em mais estados. De uma forma geral podemos concluir que as experiências realizadas permitiram obter resultados aceitáveis, apesar da pequena quantidade dos dados de treino utilizados.

Capítulo 6

Conclusões e Trabalho Futuro

Neste trabalho foram apresentadas diferentes metodologias para classificar séries temporais. Abordaram-se duas formas diferentes de extração de coeficientes estáticos e uma forma de extrair a evolução dinâmica desses coeficientes. Para além disso foram expostas várias técnicas de classificação com vista a identificar os exercícios considerados.

Ao nível do processamento dos sinais, os coeficientes estáticos cepstral de frequência de Mel foram os que apresentaram melhor desempenho. Para além disso, chegou-se à conclusão de que a introdução dos coeficientes dinâmicos não melhorou os resultados.

A distorção temporal dinâmica (DTD) demonstrou ser uma técnica pouco precisa para a classificação de séries temporais. Os algoritmos de classificação usando máquinas de vetores de suporte (MVSs) e florestas aleatórias (FAs) foram os que obtiveram melhor desempenho a nível global. Porém, para o tipo de reconhecimento que se pretende, estes algoritmos de classificação não são os mais indicados. Isto é, num software de ensino de uma nova língua, é necessário ter uma metodologia que detete determinadas palavras e ignore tudo o resto. Ou seja, é necessária uma metodologia que se aproxime das MVSs de uma classe ou então de modelos generativos como o caso dos modelos de Markov ocultos (MMOs). No entanto, não foi possível obter reconhecimento aplicando MVSs de uma classe, o que torna a sua aplicação inviável no atual estado. Os MMOs mostraram ser uma abordagem que pode atingir bons resultados, mas têm o inconveniente de requererem uma grande uma grande quantidade dados para serem treinados. Tendo em conta que neste tipo de reconhecimento é necessário que os modelos sejam treinados com um determinado grau de perfeição fonética isto torna a aplicação dos MMOs um pouco difícil.

Considerando os resultados obtidos com os métodos utilizados, não é clara qual metodologia mais apropriada para o tipo de reconhecimento que se pretende. Neste sentido, é necessária mais investigação nesta área, para determinar qual o impacto da quantidade de dados para o treino dos modelos. Para além disso, existem novas abordagens para processamento de sinal [14] e para a classificação de séries temporais que merecem atenção. Um exemplo de uma metodologia mais recente para a classificação é a combinação de modelos generativos como os MMOs e modelos discriminativos como as MVSs, que foi indicada como sendo um passo em frente neste tipo de tarefa [46].

Uma componente também muito importante para o desenvolvimento de um software para o ensino de uma língua é o feedback. A maior parte dos softwares que existe, tem medidas de desempenho do utilizador que são pouco claras, não mostrando de uma forma objetiva como melhorar a pronúncia. Assim, esta será uma componente para investigação futura.

Referências

[1] Abdul J Jerri. "The Shannon sampling theorem—Its various extensions and applications: A tutorial review". Em: *Proceedings of the IEEE* 65.11 (1977).

[2] Claude E Shannon. "Communication in the presence of noise". Em: *Proceedings of the IRE* 37.1 (1949), pp. 10–21.

[3] *Short time fourier transform*. 2015. URL: `http://cnx.org/content/m10570/latest/`.

[4] Wendy Holmes. *Speech synthesis and recognition*. CRC press, 2001.

[5] Jont B Allen e Lawrence R Rabiner. "A unified approach to short-time Fourier analysis and synthesis". Em: *Proceedings of the IEEE* 65.11 (1977), pp. 1558–1564.

[6] Geoff Lawday, David Ireland e Greg Edlund. *A signal integrity engineer's companion: real-time test and measurement and design simulation*. Pearson Education, 2008.

[7] Fredric J Harris. "On the use of windows for harmonic analysis with the discrete Fourier transform". Em: *Proceedings of the IEEE* 66.1 (1978), pp. 51–83.

[8] John Makhoul. "Linear prediction: A tutorial review". Em: *Proceedings of the IEEE* 63.4 (1975), pp. 561–580.

[9] James Durbin. "The fitting of time-series models". Em: *Revue de l'Institut International de Statistique* (1960), pp. 233–244.

[10] Xuedong Huang et al. *Spoken language processing: A guide to theory, algorithm, and system development*. Prentice Hall PTR, 2001.

[11] Stanley S Stevens e John Volkmann. "The relation of pitch to frequency: A revised scale". Em: *The American Journal of Psychology* (1940), pp. 329–353.

[12] Todor Ganchev, Nikos Fakotakis e George Kokkinakis. "Comparative evaluation of various MFCC implementations on the speaker verification task". Em: 1 (2005), pp. 191–194.

[13] Steven B Davis e Paul Mermelstein. "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences". Em: *Acoustics, Speech and Signal Processing, IEEE Transactions on* 28.4 (1980), pp. 357–366.

[14] Iosif Mporas et al. "Comparison of speech features on the speech recognition task". Em: *Journal of Computer Science* 3.8 (2007), pp. 608–616.

[15] Alan V Oppenheim, Ronald W Schafer, John R Buck et al. *Discrete-time signal processing*. Vol. 2. Prentice-hall Englewood Cliffs, 1989.

[16] Vladimir Britanak, Patrick C Yip e Kamisetty Ramamohan Rao. *Discrete cosine and sine transforms: general properties, fast algorithms and integer approximations*. Academic Press, 2010.

[17] Biing-Hwang Juang, Lawrence R Rabiner e Jay G Wilpon. "On the use of bandpass liftering in speech recognition". Em: *Acoustics, Speech and Signal Processing, IEEE Transactions on* 35.7 (1987), pp. 947–954.

[18] Sadaoki Furui. "Speaker-independent isolated word recognition using dynamic features of speech spectrum". Em: *Acoustics, Speech and Signal Processing, IEEE Transactions on* 34.1 (1986), pp. 52–59.

[19] Steve Young et al. *The HTK book*. Vol. 2. Entropic Cambridge Research Laboratory Cambridge, 1997.

[20] Francisco da Silva Borba. *Introdução aos estudos lingüísticos*. Vol. 3. Companhia editora nacional, 1970.

[21] Madalena Cruz-Ferreira. "European Portuguese". Em: *Journal of the International Phonetic Association* 25 (02 dez. de 1995), pp. 90–94. ISSN: 1475-3502.

[22] Lawrence Rabiner e Biing-Hwang Juang. "Fundamentals of speech recognition". Em: (1993).

[23] Hiroaki Sakoe e Seibi Chiba. "Dynamic programming algorithm optimization for spoken word recognition". Em: *Acoustics, Speech and Signal Processing, IEEE Transactions on* 26.1 (1978), pp. 43–49.

[24] Bernard Kolman, David Ross Hill e Alessandra Bosquilha. *Introdução à Álgebra Linear com aplicações*. LTC, 2006.

[25] Lawrence R Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition". Em: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286.

[26] Leonard E Baum. "An equality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes". Em: *Inequalities* 3 (1972), pp. 1–8.

[27] Leonard E Baum et al. "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains". Em: *The annals of mathematical statistics* (1970), pp. 164–171.

[28] Leonard E Baum e Ted Petrie. "Statistical inference for probabilistic functions of finite state Markov chains". Em: *The annals of mathematical statistics* (1966), pp. 1554–1563.

[29] James K Baker. "The DRAGON system–An overview". Em: *Acoustics, speech and signal processing, IEEE transactions on* 23.1 (1975), pp. 24–29.

[30] Leonard E Baum, George R Sell et al. "Growth transformations for functions on manifolds". Em: *Pacific J. Math* 27.2 (1968), pp. 211–227.

[31] John Makhoul, Salim Roucos e Herbert Gish. "Vector quantization in speech coding". Em: *Proceedings of the IEEE* 73.11 (1985), pp. 1551–1588.

[32] Stephen E Levinson, Lawrence R Rabiner e Man Mohan Sondhi. "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition". Em: *Bell System Technical Journal, The* 62.4 (1983), pp. 1035–1074.

[33] Corinna Cortes e Vladimir Vapnik. “Support-vector networks”. Em: *Machine learning* 20.3 (1995), pp. 273–297.

[34] Thomas M Cover. “Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition”. Em: *Electronic Computers, IEEE Transactions on* 3 (1965), pp. 326–334.

[35] Chih-Wei Hsu e Chih-Jen Lin. “A comparison of methods for multiclass support vector machines”. Em: *Neural Networks, IEEE Transactions on* 13.2 (2002), pp. 415–425.

[36] Chih-Chung Chang e Chih-Jen Lin. “LIBSVM: A library for support vector machines”. Em: *ACM Transactions on Intelligent Systems and Technology (TIST)* 2.3 (2011), p. 27.

[37] Bernhard Schölkopf et al. “Estimating the support of a high-dimensional distribution”. Em: *Neural computation* 13.7 (2001), pp. 1443–1471.

[38] Leo Breiman. “Random forests”. Em: *Machine learning* 45.1 (2001), pp. 5–32.

[39] Adele Cutler, D Richard Cutler e John R Stevens. “Random forests”. Em: *Ensemble Machine Learning*. Springer, 2012, pp. 157–175.

[40] Leo Breiman. “Bagging predictors”. Em: *Machine learning* 24.2 (1996), pp. 123–140.

[41] Kamil Wojcicki. “Htk mfcc matlab”. Em: *MATLAB Central File Exchange* (2011).

[42] Malcolm Slaney. “Auditory toolbox: a Matlab toolbox for auditory modelling work”. Em: *Tech. Rep. 45, Apple Technical Report*. Apple Computer Inc, 1994.

[43] Chih-Chung Chang e Chih-Jen Lin. “LIB: A library for support vector machines”. Em: *ACM Transactions on Intelligent Systems and Technology* 2 (3 2011). Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`, 27:1–27:27.

[44] Kevin Murphy. “Hidden Markov model (HMM) toolbox for Matlab, 1998”. Em: *URL: `http://people.cs.ubc.ca/~murphyk/Software/HMM/hmm.html`* ().

[45] I Nabney e C Bishop. “Netlab toolbox”. Em: *Internet, June* (2004).

[46] Tommi Jaakkola, David Haussler et al. “Exploiting generative models in discriminative classifiers”. Em: *Advances in neural information processing systems* (1999), pp. 487–493.