UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA
DEPARTAMENTO DE COMUNICAÇÕES

RECONHECIMENTO AUTOMÁTICO DE PALAVRAS ISOLADAS : ESTUDO E APLICAÇÃO DOS MÉTODOS DETERMINÍSTICO E ESTOCÁSTICO

Este exemplar corresponde à redação final da tese defendida por NESTOR JORGE BECERRA YOMA 386 e aprovada pela Comissão Julgadora em 22 NOV 1993.

Orientador

por: **NÉSTOR BECERRA YOMA**
Engenheiro Eletrônico (UNICAMP - 1986)

orientador: **PROF. DR. JOÃO MARCOS TRAVASSOS ROMANO**
Professor Livre-Docente do Departamento de Comunicações da Faculdade de Engenharia Elétrica da UNICAMP

Dissertação apresentada à Faculdade de Engenharia Elétrica da UNICAMP como requisito parcial para a obtenção do título de **Mestre** em Engenharia Elétrica.

Campinas, outubro de 1993

Parte deste trabalho foi desenvolvido enquanto o autor era bolsista do Ministério de Educação e Ciência do Governo Espanhol, em Madri.

A meus pais, Néstor e Isabel, e à Luciana

AGRADECIMENTOS

As palavras tornam-se insuficientes para agradecer ao prof. Dr. João Marcos Romano pela confiança, pela orientação na redação deste trabalho e por toda ajuda que me tem dispensado para o prosseguimento da minha carreira.

Ao amigo José Colás, professor do "Grupo de Tecnología del Habla" da Escola de Telecomunicações da Univ. Politécnica de Madri - Espanha- pelo apoio nos momentos difíceis e pela colaboração nas minhas pesquisas bibliográficas.

Ao prof. Javier Sánchez do CSIC de Madri, pela orientação no início do meu trabalho na área de voz.

Ao Prof. Dr. Fábio Violaro e ao doutorando Fernando Runstein pela cessão da base de dados disponível no momento.

À funcionária Janete Sayoko Toma da FEE pelas dicas tão pertinentes e pelo trabalho de impressão.

E a todos aqueles que direta ou indiretamente contribuiram para viabilizar este trabalho.

RESUMO

Esta dissertação objetiva o estudo e implementação das técnicas mais comumente utilizadas em reconhecimento de palavras isoladas, numa abordagem analítica e crítica. Neste sentido, os dois primeiros capítulos foram dedicados à apresentação dos métodos de parametrização e de reconhecimento de padrões acústicos, utilizando um certo rigor matemático, tendo sempre em vista as aplicações.

A seguir, foram comparadas três técnicas de parametrização (coeficientes LPC, LPC-cepstral e Mel-cepstral) no que diz respeito à capacidade de assimilar características intra-locutor e inter-locutor, e quanto à robustez em relação ao ruído interferente. Para implementar estes testes comparativos foi sugerido o algoritmo DTW (método determinístico) que compara diretamente duas elocuções eliminando as diferenças temporais entre elas.

Por último, foi descrita a implementação de um reconhecedor automático de dígitos independente do locutor empregando a técnica HMM (método estocástico) com modelamento por palavra e parametrização Mel-cepstral.

ABSTRACT

In this work, the most commonly used techniques employed in speech recognition for isolated words were studied and implemented. Initially the parametrization and acoustic pattern recognition methods were discussed. In the discussion, we not only maintained the mathematical formalism as suggested in the literature but also sought the easy way for the practical implementation of these techniques.

Three parametrization techniques, namely LPC, LPC-cepstral and Mel-cepstral coefficients, were compared with respect to the assimilation capability of speaker-dependent and independente features, and noise robustness. Particularly, the DTW technique (deterministic analysis) was used for these comparative tests, which is capable of eliminating the time difference between two elocutions.

A speaker independent digit recognizer was implemented employing the HMM techniques (stochastic analysis) with word modelling and Mel-cepstral coefficients.

ÍNDICE

INTRODUÇÃO

I.1. COMUNICAÇÃO HOMEM-MÁQUINA

A comunicação homem-máquina por meio da linguagem oral é um tópico que vem atraindo a atenção dos pesquisadores por várias décadas. A possibilidade da interação entre seres humanos e máquinas por meio de uma linguagem natural e acessível como à que todos estamos acostumados é tão sedutora que na ciência ficção não são raros os diálogos entre personagens e poderosos computadores capazes de raciocinar por si mesmos.

Os processos envolvidos na comunicação oral homem-máquina, a síntese e o reconhecimento de voz, tiveram um desenvolvimento muito grande nos últimos anos sendo que a primeira parece estar mais próxima do estágio desejado. Por outro lado, o reconhecimento de palavras, tema deste trabalho, já foi abordado com diversas técnicas de processamento de sinais, programação dinâmica, cadeias de Markov, redes neurais e inteligência artificial, reduzindo em grande medida as limitações dos primeiros protótipos no que diz respeito ao tamanho do vocabulário, independência do locutor e robustez frente a ruído. Contudo, não é possível ainda o reconhecimento da fala natural em contextos físico (locutor, ruído ambiental) e semântico variáveis.

I.2. ANTECEDENTES, ESTADO ATUAL E PERSPECTIVAS DO RECONHECIMENTO DE VOZ

Nos últimos cinquenta anos foram propostas e implementadas muitas estratégias para o reconhecimento de voz [1]. Estas estrategias cobrem, como já foi mencionado, muitas ciências, tais como processamento de sinal, reconhecimento de formas, inteligência artificial, estatística, teoria da informação, teoria da probabilidade, algoritmos computacionais, psicologia, linguística e mesmo biologia. Embora alguns sistemas mostrem que é possível reconhecer a voz humana eficientemente, estes funcionam impondo pelo menos uma das seguintes restrições: (1) dependência do locutor; (2) palavras isoladas; (3) vocabulário pequeno; (4) gramáticas reduzidas; (5) contexto semântico limitado; e (6)ausência de ruído de fundo. Ainda não existe nenhum sistema que trabalhe sem alguma das res-

trições acima, embora as 3 ou 4 primeiras já tenham sido superadas com técnicas que manipulam uma quantidade muito grande de dados de treinamento [5], tais como os Modelos Ocultos de Markov [1][2].

Atualmente se trabalha em sistemas que integram as técnicas de reconhecimento voz utilizadas até agora com modelos de liguagem natural [2], que introduzem o conhecimento sintático e semântico [3] do idioma guiando o própio reconhecimento de palavras e permitindo lidar com o significado de frases. Estes sistemas, 'Spoken Language Systems' [5], têm por objetivo o diálogo em condições naturais entre usuário e máquina num contexto semântico restrito, por exemplo, acesso a uma base de dados geográficos, reserva de passagens aéreas, etc. Existem, por outro lado, pesquisas [5] viabilizando a tradução automática entre línguas também em contexto semântico limitado.

Simultaneamente, os sistemas de reconhecimento de voz, de maneira geral, devem ser capazes de funcionar em condições de ruído de fundo diferentes das condições de treinamento. Isto força o estudo de técnicas [6] para conseguir a robustez dos sistemas a ambientes adversos, como seria o caso em telefonia móvel, fábricas, aeronáutica, etc.

Os principais avanços começaram a surgir nas últimas duas décadas com o aparecimento de vários sistemas razoavelmente bem sucedidos na tarefa de reconhecer palavras. Estes sistemas, mesmo impondo várias das restrições mencionadas anteriormente, representaram saltos notáveis nas respectivas épocas. Os principais projetos ou trabalhos foram, em ordem cronológica, os seguintes:

(1975) Itakura do NTT introduz o "Dynamic Time Warping" (DTW) para alinhar os quadros da palavra a reconhecer com os da referênça. Este sistema foi experimentado para um só locutor com 200 palavras e atingiu uma taxa de acertos de 97,3%.

(1975) O sistema Hearsay, da Carnegie Mellon University, utilizava uma estrutura com comunicação através da base de dados; isto é, cada fonte de conhecimento se comunica com as demais por meio de uma base de dados que se ajusta a uma determinada forma de representação. Reconhecia com 87% de acerto um vocabulário de 1011 palavras de forma contínua, com dependência do locutor e uma sintaxes limitada.

(1976) O sistema Harpy combinava as vantagens de projetos anteriores (Hearsay e Dragon) e utilizava representação por redes e busca dirigida para melhorar a eficiência. Conseguiu uma taxa de acerto de 97% no mesmo vocabulário de 1011 palavras usado pelo HEARSAY.

(1982) Os laboratórios Bell aplicaram técnica de 'clustering' com o objetivo de criar padrões robustos para o reconhecimento de palavras isoladas independentes do locutor. Wilpon e colaboradores obtiveram uma taxa de acerto de 91% com um vocabulário de 129 palavras para indepedência do locutor.

(1985) O sistema TANGORA da IBM foi o primeiro a trabalhar com vocabulários grandes. Tinha uma taxa de acertos de 97% para reconhecimento de sentenças com palavras isoladas, com dependência do locutor e utilizando um vocabulário de 5000 palavras.

(1987) O sistema BYBLOS da BBN aplicou com sucesso o modelamento de fonemas dependentes do contexto. Embora ele fosse dependente do locutor, tinha um procedimento que pemitia adaptação a um novo locutor com um breve tempo de treinamento.

(1988) Os laboratórios Bell obtiveram a maior taxa de acerto até então, com indepedência do locutor, para reconhecimento de digitos concatenados. Utilizando HMM (Hiden Markov Models) foi conseguida uma taxa de acerto de 97,1% sem uso de gramáticas.

(1988) O sistema SPHINX da CMU conseguiu [7] realizar o reconhecimento de voz contínua independente do locutor para grandes vocabulários. Os princípios básicos em que se baseia o projeto SPHINX são : a)modelo sofisticado porém adaptável da voz; b) inclui o conhecimento atual sobre a voz; c) utiliza unidades de voz que são treináveis e insensíveis ao contexto; d) tem a possibilidade de aprender e se adaptar a locutores individuais. Sobre uma base de dados de 997 palavras consegue uma taxa de acerto de 96% com gramática, e de 82% sem gramática. O modelamento de fonemas e difones é realizado com HMM.

Cada um destes sistemas conseguiu uma boa precisão e teve, ou tem, alguma aplicação prática. Contudo, nenhum deles foi concebido para se comportar satisfatoriamente sem a imposição de pelo menos duas das restrições consideradas. Em particular, os maiores empecilhos têm sido lidar com as variações entre locutores e de ruídos de fundo.

De maneira geral, as dificuldades encontradas na implementação de sistemas de reconhecimento de voz podem ser resumidas como a seguir:

1) Segmentação das unidades acústicas. Não há necessariamente uma separação (silêncios) entre as unidades acústicas comparáveis aos espaços na linguagem escrita (exceto quando se trata de palavras isoladas). Como os espectros mudam continuamente de fonema a fonema devido a interação mútua, resulta muito difícil determinar com precisão os limites dos fonemas e as vezes das próprias palavras.

2) Problemas de redução e de coarticulação. O espectro de um fonema numa palavra ou frase curta está influenciado pelos fonemas vizinhos como consequência da coarticulação. Tal espectro é muito diferente ao de um fonema isolado ou sílaba, dado que os orgãos articulatórios não se movimentam tanto com a fala contínua como com a pronúncia de palavras isoladas. Embora se possa evitar este problema no reconhecimento de palavras isoladas utilizando palavras inteiras como padrões de referência, é indispensável tratá-lo de maneira adequada em reconhecimento de palavras contínuas. Neste caso, a dificuldade vem da ação por parte do locutor de juntar palavras e 'comer' alguns sons.

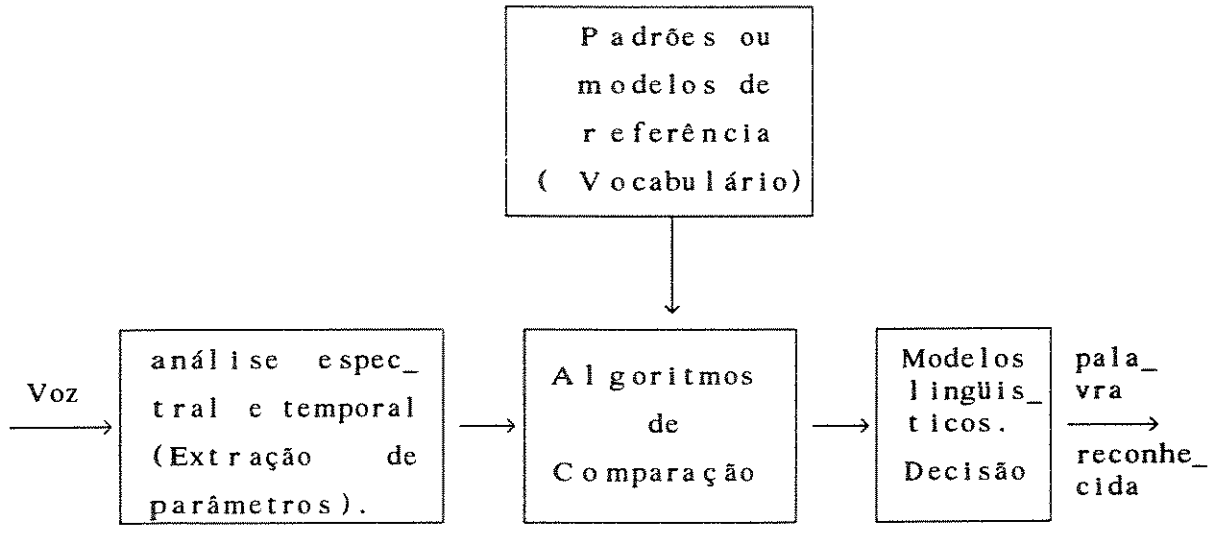
3) Uma grande variabilidade está presente na voz : variabilidade intra-locutor, devida à forma de falar (pausadamente, com melodia, gritando, sussurrando, com resfriado, etc); variabilidade inter-locutor (diferença no timbre, no tom, dependência do sexo, idade, etc); variação no captador do sinal (microfone), ou no ambiente (ruído, interferência no canal transmissor, acústica da sala, etc). Tudo isto faz com que fonemas diferentes ditos por locutores diferentes possam vir a ter espectros semelhantes.

4) Formalização do conhecimento lingüístico. As características físicas da voz nem sempre levam suficiente informação fonética. As frases normalmente se pronunciam e se ouvem com um conhecimento lingüístico, consciente ou inconsciente, e o ouvinte normalmente pode até predizer a próxima palavra, ou mesmo frases, guiado por restrições sintáticas e semânticas; assim, na linguagem oral, uma informação fonética incompleta é compensada com este conhecimento lingüístico.

I.3. OBJETIVO DA TESE

Este trabalho tem por finalidade apresentar um estudo bibliográfico e experimental das técnicas básicas que se utilizam hoje em dia em reconhecimento automático de padrões acústicos. Do ponto de vista de aplicação, foi abordado o problema de reconhecimento de palavras isoladas, dependente e independente do locutor, num vocabulário pequeno (dígitos de 0 a 9). A técnica HMM ('Hidden Markov Models') implementada no capítulo 5 com um modelo por palavra pode ser, quase que diretamente, aplicada a um vocabulário de umas 100 a 200 palavras. Para vocabulários de umas 1000 ou mais palavras os HMM devem ser aplicados a fonemas ou di-fonemas e não a palavras inteiras, embora o conceito se mantenha.

Esquemáticamente, um sistema de reconhecimento automático de palavras isoladas pode ser representado como a seguir:



Uma vez separado dos intervalos de silêncio, o sinal de voz (palavra) é dividido em segmentos de igual duração, quadros, e neles é submetido a uma série de processos (extração de parâmetros) que dão como resultado uma série de coeficientes representativos do intervalo em questão. Estes coeficientes podem vir de uma análise freqüencial (FFT, LPC, etc) ou temporal (cruzamentos por zero, amplitude, etc). Após a parametrização, as palavras são substituídas por uma sequência de vetores de coeficientes e são gerados os padrões ou modelos de referência (treinamento). No reconhecimento comparam-se palavras parametrizadas que não foram utilizadas no treinamento com as referências e, por último, utilizam-se os modelos linguísticos para tomar a decisão final. Uma elocução parametrizada é denominada de 'observação'. Pelo teorema de Bayes, temos então:

$$\text{Prob(palavra/observação)} = \frac{\text{Prob(observação/palavra)} \cdot \text{Prob(palavra)}}{\text{Prob(observação)}}$$

Onde Prob(palavra) é determinada pelo modelo lingüistico empregado, e prob(observação) é irrelevante pois é a probabilidade só da observação, considerada constante.

Nesta dissertação, com exceção do módulo lingüístico, cada bloco da figura acima é estudado e implementado. Apresentamos três exemplos de técnicas para extração de parâmetros freqüenciais: LPC, LPC-cepstral e Mel-cepstral. Os algoritmos de reconhecimento de padrões acústicos implementados foram : o DTW, onde cada padrão de referência é uma elocução de uma palavra, e os HMM, onde cada padrão de referência corresponde a um modelo com informação estatística de várias elocuções de uma mesma palavra. A decisão, no primeiro caso (DTW), se dá por meio da escolha do padrão de referência que apresentar a menor distância em relação ao padrão de teste; no segundo caso (HMM), o reconhecimento se faz escolhendo o modelo que apresentar a maior verossimilhança em relação ao padrão de teste.

Como já foi comentado, a fala tem uma variabilidade muito grande no que diz respeito à informação espectral e temporal (duração de fonemas) do sinal acústico. O reconhecimento automático exige técnicas que possam manipular uma quantidade considerável de elocuções de treinamento para assim tentar lidar com toda a variabilidade possível do sinal de voz. Neste contexto, as ferramentas

estatísticas ganham espaço em relação às determinísticas. Assim, o DTW, que compara diretamente duas elocuções de plavras, foi substituído pelos HMM's que conseguem implementar um levantamento estatístico das informações espectrais e temporais de várias elocuções de uma mesma palavra ou fonema feitas por vários locutores em diferentes contextos.

Cabe mencionar que os HMM (Hidden Markov Models) constituem a técnica que melhor se tem aplicado ao problema de reconhecimento automático de voz até agora, e é utilizada hoje em dia em quase todos os projetos com alguma aplicação prática, seja em reconhecimento de dígitos, seja em máquinas de ditado com vocabulários da orden das 1000 ou 5000 palavras; tanto com palavra isolada quanto com fala contínua.

I.4. CONTEÚDO DA DISSERTAÇÃO

No capítulo 1 apresentamos um resumo das técnicas básicas de processamento digital de sinais utilizadas hoje em dia no problema de reconhecomento automático de palavras : a) janelamento; b)parametrizações freqüenciais (Mel-cepstral, LPC e LPC-cepstral); c) medidas de distância entre quadros; e d) quantização vetorial.

A discussão dos fundamentos básicos das técnicas DTW e HMM é apresentada no capítulo 2. Tenta-se abordar os temas com o maior rigor e pragmatismo possíveis, tendo sempre o espaço como limitador implacável. Alguns tópicos exigiram do autor uma leitura bem maior do que o material apresentado aqui para um entendimento que possibilitasse a implementação e manipulação destas técnicas. Contudo, o leitor interessado em esclarecer algum tópico tem acesso a referências mais específicas no fim do capítulo.

A implementação de um reconhecedor de palavras isoladas dependente do locutor, com vocabulário pequeno (dígitos de 0 a 9) e utilizando o 'Dynamic Time Warping' (DTW), é discutida e apresentada no capítulo 3. Embora esta técnica esteja ultrapassada ela é a mais simples de entender e implementar, e por isto representa um bom 'primeiro passo'. Por outro lado, o algoritmo de Viterbi, fundamento básico do DTW, é atualmente utilizado na técnica dos HMM, como ferramenta essencial em reconhecimento de fala contínua e em complemento com as

redes neurais. Além do exposto acima, a experiência mostrou que um sistema simples como este, que compara uma palavra a reconhecer com outra de referência, é ideal para fazer o estudo de parametrizações pois o padrão de referência é formado somente por uma elocução, e não por um modelo estocástico de várias elocuções (HMM).

No capítulo 4 apresentam-se os resultados de um estudo experimental, empregando o algoritmo DTW desenvolvido no capítulo 3, de três das mais utilizadas parametrizações espectrais : LPC, LPC-cepstral e Mel-cepstral. Foram abordados os seguintes tópicos: capacidade de lidar com características intra-locutor; capacidade de lidar com características inter-locutor; e robustez frente a ruído interferente, executando o algoritmo de reconhecimento em condições diferentes das de treinamento.

A implementação de um sistema de reconhecimento de palavras isoladas independente do locutor, com o mesmo vocabulário de dígitos de 0 a 9, e utilizando os HMM discretos (com quantização vetorial), é apresentada no capítulo 5. Discutiremos mais detalhadamente a teoria básica dos Modelos Ocultos de Markov ('Hidden Markov Models') e alguns detalhes de sua implementação prática. Neste caso foi empregado o algoritmo de Baum-Welch para treinar os HMM. Apresentamos alguns resultados e finalizamos esta seção com uma breve discussão sobre as vantagens e limitações dos HMM.

O capítulo 6 é dedicado às conclusões gerais e à discussão da aplicabilidade das técnicas abordadas nesta dissertação em problemas mais complexos.

I.5. REFERÊNCIAS

[1] Huang, X.D. Ariki, Y Jack, M.A. : "Hidden Markov Models for Speech Recognition". Edinburgh University Press, 1990.

[2] Moore, R. : "State of the Art in Speech". Workshop in Integrating Speech and Natural Language. University College Dublin, 15-17 july 1992.

[3] Seneff, S Meng, H. Zue,V. : "Language Modelling for Recognition and Understanding Using Layered Bigrams". Workshop in Integrating Speech and Natural Language. University College Dublin, 15-17 july 1992.

[4] Peckham,J. : "Spoken Dialogue Systems". Workshop in Integrating Speech and Natural Language. University College Dublin, 15-17 july 1992.

[5] Bahl,L.R. de Souza,P.V. Jellineck,F. Mercer,R.L. Nahamoo,D. Roukos,S. Brown,P.F. : "Solving Language Problems by Statistical Methods". Workshop in Integrating Speech and Natural Language. University College Dublin, 15-17 july 1992.

[6] O'Shaughnessy,D. : "Enhancing Speech Degraded by Additive Noise or Interfering Speakers", IEEE Communications Magazine, February, 1989.

[7] Lee, Kai-Fu : "Large Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System". Phd thesis, Department of Computer Science, Carnegie Mellon University, 1988.

CAPÍTULO 1

TÉCNICAS BÁSICAS DE PROCESSAMENTO DE VOZ

1.1. INTRODUÇÃO

Este capítulo apresenta um apanhado geral das técnicas básicas de PDS (Processamento Digital de Sinais) empregadas pelos algoritmos de reconhecimento. Não são cobertos todos os tópicos pois a quantidade de publicações nos temas é muito grande, o que nos obriga a realizar uma abordagem sucinta dos assuntos.

O trabalho de reconhecimento de voz consiste em recuperar uma mensagem lingüistica que é codificada como um sinal de voz acústico por meio da elocução de palavras. Muitos sistemas e métodos têm sido propostos, testados e abandonados : esta tem sido mais ou menos a história do reconhecimento de voz nos últimos 20 anos. Contudo, avanços em tecnologias computacionais, a disponibilidade de bases de dados de voz e a introdução de poderosas e flexíveis ferramentas estatísticas possibilitaram um grande avanço do tema. Um sistema de reconhecimento de voz consiste, genericamente falando, de três blocos: processamento do sinal, reconhecimento de padrões acústicos e modelamento da linguagem. No estágio de processamento do sinal, o sinal de voz é convertido numa sequência de quadros ou "frames" contendo, cada um, coeficientes espectrais e temporais correspondentes ao próprio intervalo do quadro. O estágio de reconhecimento de padrões acústicos tenta casar esta sequência de quadros com as de possíveis unidades lingüísticas, geralmente palavras ou fonemas. O modelamento de linguagem tenta definir palavras ou frases lingüisticamente válidas.

1.2. EXTRAÇÃO DE PARÂMETROS FREQÜENCIAIS.

O objetivo do processamento do sinal é fornecer um conjunto de parâmetros que sejam representativos dos diferentes intervalos do sinal de voz e que possam ser empregados nos estágios posteriores de reconhecimento de padrões acústicos. Estes parâmetros devem separar, ou pelo menos tentar, os diferentes fonemas ou alofones a partir das características físicas do sinal acústico.

Podem-se utilizar tanto parâmetros temporais como espectrais. Os parâmetros temporais, tais como a energia de quadros ou cruzamentos por zero, lidam diretamente com a forma de onda do sinal de voz e são geralmente simples de implementar. Por outro lado, a abordagem frequencial envolve análises mais complexas e evidencia características importantes que são mais difíceis de visualizar no domínio do tempo. Este tipo de análise, a espectral, é a mais utilizada para a extração de parâmetros do sinal. Vários trabalhos sobre a percepção da voz realizados com alofones sintetizados [2][3], especialmente vogais, mostraram que a identidade fonética está muito mais correlacionada com as características espectrais do sinal do que com as características temporais do mesmo [1]. Isto é, mantendo a densidade espectral de potência e variando a diferença de fase das componentes, o sinal, foneticamente, soa igual.

Associadas aos métodos de análise de sinal, temos as medidas de distância que determinam a semelhança, ou distorção, entre dois quadros resultantes do processo de parametrização do sinal de voz. No geral, a distância Euclideana é muito utilizada pela sua simplicidade. Como uma generalização da distância Euclideana temos a de Mahalanobis, baseada em análise estatística, que pondera cada coeficiente em função de sua importância na classificação. Por último, a distância de Itakura é empregada quando se trabalha com análise LPC. Ela mede a relação entre erros de predição.

Várias técnicas de processamento de sinal e de extração de parametros para reconhecimento de voz têm sido utilizadas. A maior parte destas técnicas, como já comentamos, baseiam-se na representação espectral do sinal através da análise LPC ou de Fourier a curto prazo.

1.2.1. Análise de Fourier a Curto Prazo

Podemos afirmar que o sinal de voz é uma seqüência de intervalos estacionários dentro dos quais a distribuição espectral de potência é mais ou menos constante. Alguns fonemas, como as vogais e outros fonemas sonoros, caracterizam-se por apresentarem intervalos estáveis e densidade espectral bastante caracterizada. Este não é o caso das plosivas (/k/, /p/, etc) que se diferenciam por uma ou mais mudanças abruptas da estacionariedade. Pode-se concluir, então, que o sinal que transporta a informação fonética é intrinsicamente não-estacionário , pois se não houvesse uma dinâmica nas características espec-

trais do sinal, não haveria transferência de informação.

Por outro lado sabemos que há uma correlação estreita entre a densidade espectral de potência e o caráter fonético de muitos fonemas não-oclusivos, o que torna muito atraente a possibilidade de submeter o sinal de voz à análise de Fourier. Mas esta tem que ser realizada em intervalos cuja duração seja menor que a dos intervalos de estacionariedade para garantir uma resolução mínima na representação da dinâmica do sinal. Esta análise de Fourier realizada em intervalos finitos denomina-se de curto prazo.

A análise a curto prazo depende do janelamento aplicado ao sinal para isolar um curto intervalo de tempo onde realizar a análise espectral. Este intervalo de tempo de análise chama-se quadro ("frame"), e a duração do mesmo corresponde à largura da janela. O deslocamento (intervalo entre quadros) e a largura da janela são escolhidos de maneira que se preserve a dinâmica do sinal mantendo um nível mínimo de detalhes nos espectros resultantes. No janelamento tenta-se também evitar mudanças abruptas nos pontos terminais dos quadros atenuando a amplitude do sinal de voz nas extremidades da janela e concentrando a análise no sinal localizado no centro da mesma.

Seja então um sinal de voz contínuo no tempo e uma função janela denotados, respectivamente, por $v(t)$ e $w(t-\tau)$. Logo, o sinal após o janelamento é dado por :

$$x(t,\tau)=v(t)\cdot w(t-\tau) \qquad (1.1)$$

onde τ é o instante em que a janela é aplicada. $x(t,\tau)$ é o sinal resultante do janelamento e é função do tempo t e do instante τ em que a janela é aplicada.

A análise de Fourier a curto prazo é implementada no sinal $x(t,\tau)$ pela transformada de Fourier, que no domínio contínuo é dada por :

$$X(j\omega\ ,\ \tau)=\int_{-\infty}^{\infty} x(t,\tau)\cdot\exp(-j\cdot\omega\cdot t)\cdot dt \qquad (1.2)$$

onde $j=\sqrt{-1}$, $\omega=2\cdot\pi\cdot f$ (f=freqüência). A correspondente inversa da transformada de Fourier, que mapeia do domínio espectral para o domínio do tempo, é definida como:

$$x(t,\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\omega,\tau) \cdot \exp(j \cdot \omega \cdot t) \cdot d\omega \qquad (1.3)$$

Na prática, o sinal contínuo no tempo $x(t,\tau)$ e seu espectro $X(j\omega,\tau)$ são discretizados no tempo e amplitude por meio de amostragem e de digitalização para viabilizar o processamento por computador. Suponhamos que o sinal contínuo $x(t,\tau)$ seja amostrado com um período de amostragem igual a T segundos. Logo, o sinal contínuo $x(t,\tau)$ passa a ser representado por uma sequência de dados discretizados no tempo, $x(i,l)$, onde i e l são inteiros ($t=i \cdot T$; $\tau=l \cdot T$), com i numerando as amostras dentro de uma dada janela e com l indicando o centro da janela de análise de largura N. A correspodente transformada de Fourier para uma sequência de amostras $\{ x(i,l)\}$ é definida como:

$$X(n,l) = \sum_{i=0}^{N-1} x(i,l) \cdot \exp(-i \cdot n \cdot j \cdot 2 \cdot \pi/N) \qquad (1.4)$$

onde $j=\sqrt{-1}$, N é o número de amostras a serem analisadas (largura da janela). A transformada inversa de Fourier é definida por:

$$x(i,l) = \frac{1}{N} \sum_{n=0}^{N-1} X(n,l) \cdot \exp(i \cdot n \cdot j \cdot 2 \cdot \pi/N) \qquad (1.5)$$

De maneira a calcular eficientemente a transformada discreta de Fourier, utilizam-se os algoritmos da FFT (Fast Fourier Transform), normalmente com N sendo potência de 2, diminuindo consideravelmente o número de operações.

Uma questão importante envolve as relações entre a resolução em frequência e a resolução no tempo. A resolução no tempo é definida pelo número de amostras a serem analisadas (largura do quadro ou da janela). Uma vez que uma sequência de quadros com poucos pontos podem representar melhor as características que variam rapidamente no tempo, aumentaremos a resolução no tempo quando os quadros tiverem seus comprimentos diminuídos. Por outro lado, a resolução em frequência é limitada pelo passo em frequência Δf, definido como:

$$\Delta f = \frac{1}{N \cdot T} \qquad (1.6)$$

onde N é comprimento do quadro medido em número de amostras e T é o período de amostragem (a equação (1.6) é facilmente obtida quando comparamos as equações (1.2) e (1.4), fazendo $kT \longrightarrow t$ e $n/(N \cdot T) \longrightarrow n \cdot \Delta f \longrightarrow f$). Da equação (1.6)

fica evidente que um quadro de comprimento N grande aumenta a resolução da análise do ponto de vista frequencial e reduz a resolução do ponto de vista temporal. Um quadro de comprimento pequeno melhora a resolução no tempo, mas piora na frequência.

A questão da largura da janela pode ser abordada sob o prisma de dois compromissos : a) a descrição de fonemas não-oclusivos; e b) a descrição de fonemas oclusivos. No caso a) sabemos que a identidade fonética está fortemente relacionada com a densidade espectral de potência e portanto quanto maior a janela mais detalhada será a descrição destes fonemas. Já no caso b) temos que as plosivas se diferenciam por uma mudança de estacionariedade e a resolução no tempo adquire uma conotação muito importante. Em termos quantitativos consideram-se janelas de 12 a 32 mseg de duração e deslocadas de 6 a 15 ms. A sobreposição dos quadros é sempre desejada mas um número muito elevado deles sobrecarrega os algoritmos de reconhecimento de padrões acústicos.

1.2.2. Análise LPC (Linear Predictive Coding)

Yule (1927) propôs que uma série temporal de amostras correlacionadas poderia ser gerada a partir de uma série de amostras estatisticamente independentes (ruído branco) processadas através de um filtro linear. Um tipo de modelamento muito utilizado para processos estocásticos é o AR (auto-regressivo), pois está relacionado ao teorema de Wold e seus parâmetros podem ser determinados pelas equações de Yule-Walker [8]. Segundo o modelamento AR um processo estocástico genérico, x(n), pode ser escrito como :

$$x(i) - \alpha_1 \cdot x(i-1) - \alpha_2 \cdot x(i-2) - \ldots - \alpha_p \cdot x(i-p) = e(i) \qquad (1.7)$$

onde e(i) é um ruído branco e p é a ordem da análise AR. Aplicando a transformada Z a ambos os lados da expressão (1.7) e pondo X(z) em função de E(z), temos:

$$X(z) = H_I(z) \cdot E(z) \qquad (1.8)$$

onde $H_I(z) = \dfrac{1}{\sum_{k=0}^{p} a_k \cdot z^{-k}}$, com $a_0=1$ e $a_k=-\alpha_k$, para $1 \le k \le p$

$H_I(z)$ corresponde portanto a um filtro IIR.

O teorema de Wold diz que qualquer processo estocástico, discreto no tempo e estacionário pode ser decomposto como uma soma de um processo auto-regressivo genérico (regular) e um processo predizível, ambos não-correlacionados. Formalizando, o teorema de Wold diz que [8]:

$$y(i) = x(i) + s(i) \tag{1.9}$$

onde y(i) é um processo estocástico discreto no tempo e estacionário, x(i) e s(i) são processos não correlacionados, x(i) é um processo regular genérico representado por:

$$x(i) = \sum_{k=0}^{\infty} b_k \cdot e(i-k) \tag{1.10}$$

onde b_k são constantes e e(i) é um ruído branco não correlacionado com s(i), e por último s(i) é um processo predizível (determinístico).

Aplicando a transformada z a ambos membros da equação (1.10), temos:

$$X(z) = H_F \cdot E(z) \tag{1.11}$$

onde $H_F(z)$ corrsponde a um filtro FIR (só zeros). Se H_F for um filtro FIR de fase-mínima, ele pode ser substituido por um filtro só de polos (IIR) com a mesma resposta impulsiva [9] , H_I, chegando novamente ao modelamento AR da expressão (1.8).

A análise LPC pode ser vista como o modelamento AR do sinal de voz em intervalos de tempo onde este é considerado estacionário. Como já foi visto, o sinal de voz multiplicado pela janela centrada em l é dado por :

$$x(i,l) = v(i) \cdot w(i-l)$$

onde w(i-l) tem duração de N pontos.

A título de simplificação e como nossa discussão estará restrita ao sinal de voz multiplicado por uma janela centrada num ponto genérico l, a notação x(i,l) será substituida por x(i):

$$x(i) = v(i) \cdot w(i) \tag{1.12}$$

Aplicando o modelamento AR na sequência x(i) chegamos à expressão (1.7) e (1.8). O sinal e(i), da equação (1.7), é também denominado de erro de predição pois ele pode ser visto como a diferença entre a amostra atual x(i) e sua estimativa a partir das amostras anteriores. Por outro lado o denominador da função de transferência H_I, equação (1.8), é denominado de filtro inverso LPC e é representado pelo vetor $(1,-\alpha_1,-\alpha_2,\ldots,-\alpha_p)$, onde p é a ordem da análise LPC e α_k são os coeficientes LPC. Os coeficientes LPC podem ser determinados a partir das equações de Yule-Walker.

A partir da expressão (1.7) definimos a energia do sinal residual E (erro quadrático), como sendo:

$$E = \sum_{i=-\infty}^{\infty} e^2(i) = \sum_{i=-\infty}^{\infty} \left[x(i) - \sum_{k=1}^{p} \alpha_k \cdot x(i-k) \right]^2 \qquad (1.13)$$

com x(i)=0 para i<0 ou i≥N, isto é, fora da janela de análise.

Os coeficientes α_k que minimizam E (minimização do erro quadrático médio) são obtidos pela anulação das derivadas parciais de E em relação a os próprios coeficientes α_k . Isto é:

$$\frac{\partial E}{\partial \alpha_k} = 0 \quad , \quad k=1,2,\ldots,p \qquad (1.14)$$

Temos então p equações lineares:

$$\sum_{i=-\infty}^{\infty} x(i-j)\cdot x(i) = \sum_{k=1}^{p} \alpha_k \cdot \sum_{i=-\infty}^{\infty} x(i-j)\cdot x(i-k), \quad j=1,2,\ldots,p \qquad (1.15)$$

com p incógnitas α_k. Na expressão (1.15), o termo à esquerda corresponde à auto-correlação R(j) a curto prazo de x(i). Dado que a sequência { x(i)} tem duração limitada, podemos dizer que:

$$R(j) = \sum_{i=j}^{N-1} x(i)\cdot x(i-j) \quad , \quad j=1,2,\ldots,p \qquad (1.16)$$

Logo, as equações (1.15) podem ser escritas como :

$$\sum_{k=1}^{p} \alpha_k \cdot R(j-k) = R(j), \quad j=1,2,...,p \tag{1.17}$$

As equações (1.16) e (1.17) são conhecidas como as equações de Yule-Walker [5] e a solução deste sistema leva ao cálculo dos coeficientes α_k. A autocorrelação pode ser calculada para todos os inteiros j, mas, uma vez que R(j) é uma função par, esta pode ser determinada para j=1,2,...,p. Das equações (1.13) e (1.17) temos que a energia residual mínima ou erro de predição mínimo, Ep, para um modelo de p polos, é:

$$Ep = R(0) - \sum_{k=1}^{p} \alpha_k \cdot R(k) \tag{1.18}$$

onde o primeiro termo R(0) é simplesmente a energia de x(i).

Matricialmente, o sistema de equações (1.17) pode ser representado da seguinte forma:

$$\begin{bmatrix} R(0) & R(1) & R(2) & \cdots & R(p-1) \\ R(1) & R(0) & R(1) & \cdots & R(p-2) \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ R(p-1) & R(p-2) & R(p-3) & & R(0) \end{bmatrix} \cdot \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_p \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ \vdots \\ R(p) \end{bmatrix}$$

uma vez que, por simetria da função R(j), R(j-i)=R(i-j). Logo, o sistema de equações (1.17) pode ser escrito na forma matricial como:

$$R \cdot A = r \tag{1.19}$$

onde R é a matriz $p \times p$ de autocorrelação, cujos elementos são definidos como $R(i,j) = R(|i-j|)$, $1 \leq i,j \leq p$. O vetor coluna r corresponde a $(R(1), R(2),..., R(p))^T$; e A é o vetor coluna dos coeficientes LPC, $(\alpha_1, \alpha_2, ..., \alpha_p)$.

Os coeficientes α_k podem ser determinados resolvendo diretamente o sistema de equações (1.19) de Yule-Walker invertendo a matriz R, ou pelo algoritmo recursivo de Levinson-Durbin [7][5], que explora as simetrias da matriz R além do fato desta ser do tipo Toeplitz (os elementos da diagonal principal assim como os das diagonais paralelas são iguais entre si). A seguir, apresenta-se o algoritmo de Levinson-Durbin, onde as seguintes equações são resolvidas recursi-

vamente para m=1,2,...,p :

Algoritmo de Levinson-Durbin

$$k_m = \frac{R(m) - \sum_{k=1}^{m-1} \alpha_k^{m-1} \cdot R(m-k)}{E_{m-1}} \qquad (1.20a)$$

$$\alpha_m^m = k_m \qquad (1.20b)$$

$$\alpha_k^m = \alpha_k^{m-1} - k_m \cdot \alpha_{m-k}^{m-1} \quad , \qquad 1 \le k \le m-1 \qquad (1.20c)$$

$$E_m = (1 - k_m^2) \cdot E_{m-1} \qquad (1.20d)$$

onde inicialmente $E_0 = R(0)$ e $\alpha_0 = 0$.

com α_k^m indicando o coeficiente LPC α_k na m-ésima iteração.

A interpretação mais importante da análise LPC é a que diz respeito à associação entre o modelamento AR e o modelo da produção da fala. Segundo a expressão (1.8), o espectro de sinal de voz X(z) é modelado na análise AR como sendo o produto do filtro H_I pelo erro de predição E(z). Assim, comparando este modelo AR com o modelo da produção da fala [4], o filtro H_I incorpora os efeitos da resposta em freqüencia do trato vocal, da irradiação e do filtro conformador do pulso glotal (para fonemas sonoros), enquanto que E(z) corresponde à excitação do filtro H_I. Esta excitação pode ser ruído branco (fonemas surdos) ou um trem de impulsos (fonemas sonoros).

Introduzindo na expressão (1.8) o ganho G para descrever a intensidade do sinal de voz, temos:

$$X(z) = G \cdot H_I(z) \cdot E(z) \qquad (1.21)$$

onde G é uma constante que pode ser determinada por :

$$G^2 = E_p \qquad (1.22)$$

onde E_p é definido na equação (1.18).

O valor de p, a ordem da análise LPC, depende da frequência de amostragem utilizada na digitalização do sinal de voz ou da largura de banda considerada. Recomenda-se [7] que para f_{am}= 8 Khz p seja fixado em 10.

1.2.3. Coeficientes LPC-Cepstral

Como foi discutido na seção 1.2.2., o modelo básico para a produção do sinal de voz consiste de um filtro H (z), correspondendo ao trato vocal, multiplicado pela transformada z da excitação glotal. No domínio temporal, o sinal excitador pode ser periódico, como nos fonemas sonoros onde ocorre vibração das cordas vocais, ou ruído branco, como nos fonemas surdos. Também no domínio do tempo, o sinal de voz pode ser considerado como o resultado da convolução do sinal excitador com a resposta impulsiva do filtro do trato vocal, pois, em frequência, o espectro do sinal de voz resulta do produto da resposta em frequência de H(z) com o espectro do sinal excitador. Temos então, na produção da voz, uma estrutura que varia lentamente no tempo, correspondendo à configuração do trato vocal (H(z)), e uma outra estrutura que se caracteriza por um sinal que varia rapidamente, correspondendo à excitação (E(z)).

Se, no domínio da frequência, a operação de produto entre H(z) e E(z) for substituída pela operação de soma, pela aplicação da função logaritmo em $H(z)\cdot E(z)$, os dois sinais podem ser separados pelo cálculo da transformada inversa de Fourier de $\log[H(z)\cdot E(z)]=\log[H(z)]+\log[E(z)]$. A transformada inversa de $\log[H(z)\cdot E(z)]$ é denominada de cepstrum e seu domínio de quefrency, uma espécie de parâmetro pseudo-temporal [9]. Nesta análise, denominada de cepstral [7][9], o cepstrum do sinal excitador se localiza em valores múltiplos do período fundamental, sendo que o cepstrum de H(z) cai rapidamente e se concentra em valores baixos de quefrency. A deconvolução de sinal de voz nestas duas componentes é muito vantajosa para o reconhecimento de palavras, pois permite separar a excitação da posição do trato vocal, que é o que determina o

timbre dos fonemas. Mais genericamente, este tipo de processamento, que separa dois padrões convolucionados por meio de transformações que levam à soma dos mesmos, é denominado de homomórfico [9].

Há dois tipos de análise cepstral : a baseada na FFT e a aplicada à análise LPC. No primeiro caso, a FFT é aplicada diretamente ao sinal de voz. A análise mel-cepstral (seção 1.2.4) é uma forma da análise cepstral baseada na FFT. Por outro lado, na análise cepstral LPC a transformada Z é aplicada no sinal de voz modelado pela análise LPC.

Dada a equação (1.8), temos que :

$$\log[X(z)]=\log[H_I(z)]+\log[E(z)] \tag{1.23}$$

onde log é a função logaritmo complexo.

Logo, os coeficientes LPC-cepstrais, c_n, correspondem à transformada z inversa de log[X(z)], sendo que os coeficientes de ordem menor correspondem ao filtro do trato vocal.

Por outro lado, os coeficientes cepstral LPC podem ser calculados recursivamente a partir dos coeficientes da análise LPC (α_k) [4] por meio da seguinte equação:

$$c_n = \alpha_n + \sum_{i=1}^{n-1} \frac{n-i}{n} \cdot \alpha_i \cdot c_{n-i} \quad , \quad n \geq 1 \tag{1.24}$$

onde $\alpha_i=0$ para i>p (p é a ordem da análise LPC).

1.2.4. Mel-Cepstral

De acordo com o item 1.2.3., a análise cepstral pode ser realizada aplicando o logaritmo diretamente na FFT do sinal de voz ou empregando a análise LPC (LPC-cepstral). Em se tratando da FFT o cepstrum corresponde à transformada inversa de fourier de log[X(z)], onde X(z) é obtido pela DFT (transformada discreta de fourier) do sinal de voz.

Uma aplicação da análise cepstral com DFT muito utilizada em reconhecimento de voz é a que emprega os coeficientes Mel-cepstrais. A idéia consiste em dividir a faixa de frequência útil em filtros passo-banda cuja largura é proporcional à escala bark ou mel [7] tentando simular a resposta em frequência da membrana basilar [1]. Determina-se a energia de cada filtro e, após calcular o logaritmo de cada energia, aplica-se a transformada inversa de Fourier. Observe-se que a transformada inversa tem como entrada o logaritmo da energia de cada filtro e não o logaritmo de cada ponto da DFT.

Na implementação apresentada em [6][7], e empregada neste trabalho com algumas modificações, utiliza-se um banco de 20 filtros triangulares com largura de banda constante por abaixo de 1000 hz e proporcional à frequência entre 1000 e 4000 Hz (figura 1.1.). A inclinação de cada filtro é determinada de maneira que o ganho seja sempre 1 na faixa de interesse. Seja E_k a energia em logaritmo na saída de cada filtro k, e supondo que desejamos M coeficientes mel-cepstrais c_n, aplicando a DCT inversa temos:

$$c_n = \sum_{k=1}^{20} E_k \cdot \cos\left[n(k-0.5)\pi/20\right] \qquad \text{para } n=1,2,\ldots,M \qquad (1.25)$$

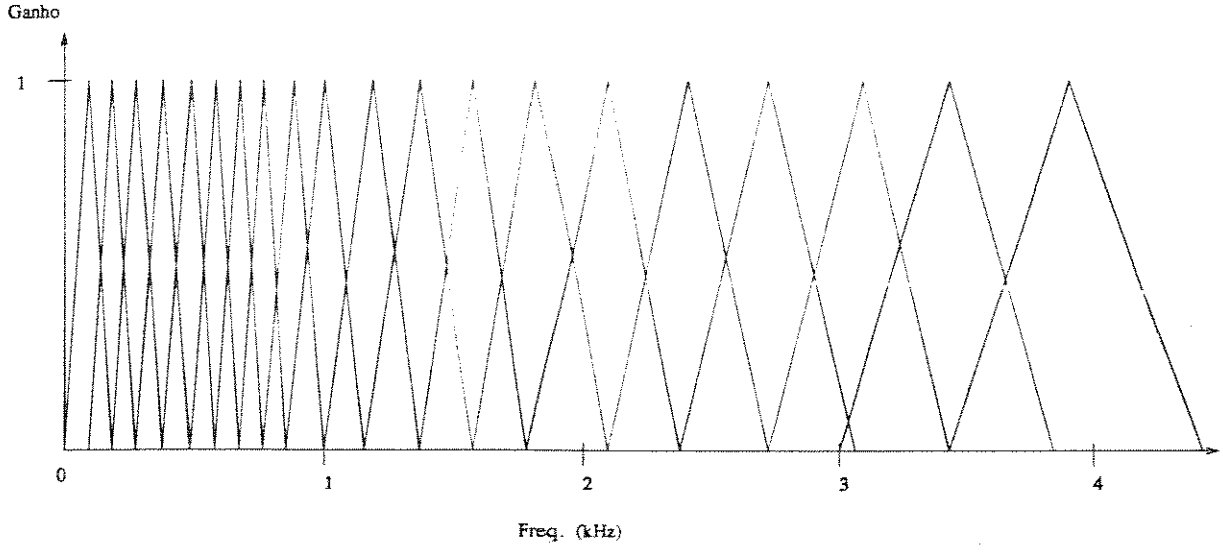


Figura 1.1.: Banco de filtros utilizados para obter os coeficientes mel-cepstral [6][7].

O coeficiente c_0 corresponde à energia média no quadro analizado e é utilizado às vezes como coeficiente de normalização de amplitude; c_1 representa um balanço da energia entre as altas e baixas frequências (valores elevados são típicos de fonemas sonoros enquanto que valores pequenos são representativos de fonemas fricativos). O resto dos coeficientes mel-cepstrum também descrevem detalhes da forma espectral do sinal de voz analisado no quadro, mas, como no caso da análise LPC, resulta difícil relacioná-los de maneira simples e direta com os formantes [7].

1.3. MEDIDAS DE DISTÂNCIA ENTRE QUADROS

A medida de distância, também conhecida como medida de distorção ou de semelhança entre dois conjuntos de coeficientes obtidos do processamento do sinal de voz janelado, tem um papel muito importante em codificação, análise e reconhecimento de voz. Ela tem por objetivo, em última instância, medir a diferença de identidade fonética entre dois segmentos de voz (quadros), de igual duração, por meio da distância numérica, métrica ou não, entre os coeficientes ou parâmetros extraídos do sinal de voz dentro do intervalo em questão. Este intervalo tem uma duração típica de 10 a 30 ms, dentro do qual o sinal de voz é considerado estacionário ou quase-estacionário. Como o carácter fonético da voz está muito correlacionada com a distribuição espectral de potência do sinal, a maior parte dos parâmetros extraídos são frequenciais, e as distâncias, portanto, trabalham com estes coeficientes como uma aproximação numérica razoável para uma caracterização basicamente perceptiva.

A medida de distância mais conhecida, e mais utilizada nos reconhecedores que não utilizam análise LPC, é a euclideana [7]. Suponhamos que de um quadro contendo N amostras de sinal de voz janelado extraímos k parâmetros. Seja :

$Vr_i = (r_1, r_2, \ldots, r_k)$, o vetor de parâmetros do padrão de referência i;

$V_t = (t_1, t_2, \ldots, t_k)$, o vetor de parâmetros do padrão de teste;

define-se então a distância euclideana (d_e) como sendo:

$$d_e^2 = \sum_{j=1}^{k} (r_j - t_j)^2 \qquad (1.26)$$

A distância euclideana é muito utilizada em sistemas de reconhecimento que utilizam parâmetros mel cepstrum ou LPC cepstrum. Neste último caso pode-se provar que [10]:

$$\sum_{j=1}^{k} (c_{r,j}^2 - c_{t,j}^2) = \frac{T}{2\cdot\pi} \int_{-\pi/T}^{\pi/T} d^2(z)\cdot d\omega \qquad (1.27)$$

com :

$$d(z) = \log|x_t(z)|^2 - \log|x_r(z)|^2 \qquad (1.28)$$

onde $x_t(z)$ e $x_r(z)$ são os espectros dos sinais de voz de teste e de referência, respectivamente, nos quadros em questão.

Em outras palavras, as equações (1.27) e (1.28) querem dizer que a distância euclideana entre os coeficientes LPC cepstral de dois quadros (um correspondendo ao padrão de teste e outro ao padrão de referência) é igual à distância entre os espectros dos sinais correspondentes.

Vista como uma generalização da distância euclideana, temos a distância de Mahalanobis, que pondera de maneira diferenciada, por meio de uma matriz de covariâncias, cada componente do vetor de parâmetros. A distância de Mahalanobis (d_m), que tem sua origem na teoria de decisão estatística [7], é definida como:

$$d_m(Vr_i, V_t) = (Vr_i - V_t)^T \cdot W^{-1} \cdot (Vr_i - V_t) \qquad (1.29)$$

com Vr_i o vetor de parâmetros do padrão de referência e V_t o vetor de parâmetros do padrão de teste. W é a matriz de covariância que pondera individualmente cada parâmetro em função de sua importância em identificar um segmento do sinal de voz no espaço vetorial de k parâmetros. Se $W = W^{-1} = I$ (I=matriz identidade), caimos no caso da distância euclidena. Apesar do seu significado teórico, a distância Mahalanobis exige uma quantidade elevada de dados de teinamento e uma carga computacional maior que a distância euclideana, razão pela qual esta última é preferida em muitos sistemas de reconhecimento de

palavras. Neste trabalho a distância Mahalanobis não será utilizada.

Em se tratando dos coeficientes LPC, a distância mais comumente utilizada é a de Itakura-Saito (d_{is}) [4][6][7], definida como:

$$d_{is}(Ar_i,A_t) = \frac{\sigma_r^2 \cdot A\bar{r}_i^T \cdot R_t \cdot Ar_i}{\sigma_t^2 \cdot A_t^T \cdot R_t \cdot A_t} + \log[\sigma_t^2 / \sigma_r^2] - 1 \qquad (1.30)$$

onde R_t é a matriz de autocorrelação do padrão de teste, σ_t e σ_r são os parâmetros de ganho LPC do padrão de teste e de referência, respectivamente; $A\bar{r}_i^T$ = (1, $-\alpha r_1$, $-\alpha r_2$,...,$-\alpha r_p$), corresponde aos coeficientes do filtro inverso LPC do padrão de referência i, onde αr_k são os parâmetros resultante da análise LPC (seção 1.2.2); analogamente, A_t^T= (1, $-\alpha t_1$, $-\alpha t_2$,...,$-\alpha t_p$) é o vetor do filtro inverso LPC do padrão de teste.

A idéia básica por trás da distância de Itakura-Saito é medir a relação entre o ruído residual obtido filtrando o padrão de teste pelo filtro inverso LPC do padrão de referência ($A\bar{r}_i^T \cdot R_t \cdot Ar_i$), e o ruído residual resultante da análise LPC sobre o próprio padrão de teste ($A_t^T \cdot R_t \cdot A_t$).

A distância de Itakura-Saito não leva em conta só o erro residual da filtragem inversa LPC, mas também o ganho dos filtros. Numa simplificação da distância de Itakura-Saito, equação (4.1), os ganhos são eliminados assumindo que as palavras ou fonemas podem ser pronunciados dentro de uma ampla margem de intensidade sem mudar de significado.

1.4. JANELAMENTO DO SINAL

Como foi discutido até agora, o sinal de voz é segmentado em intervalos de igual duração, geralmente sobrepostos, dentro dos quais são calculados uma série de parâmetros, em sua maioria frequenciais, representativos do quadro ou intervalo em questão. Esta operação de selecionar segmentos de sinal é denominada de janelamento e é implementada com funções que tentam eliminar as transições abruptas nos extremos da janela e concentrar a análise no sinal localizado no centro do quadro. Uma das janelas mais comumente utilizada é a de Hamming (JH) definida pela expressão a seguir :

$$JH(n)=\begin{cases} 0.54 - 0.46\cdot\cos[2\cdot\pi\cdot n/(N-1)], & 0 \le n<N \\ 0\ , & n<0 \text{ ou } n\ge N \end{cases} \qquad (1.31)$$

onde N é a largura da janela em número de amostras.

Recomenda-se que a duração da janela seja de 10 a 30 ms, pois neste período o sinal de voz pode ser considerado estacionário ou quase-estacionário, uma vez que a velocidade de articulação do aparelho fonador costuma ser menor do que isto. Como discutido na seção 1.2.1., existe um compromisso entre a resolução temporal e a resolução frequencial da análise : melhorando a resolução no tempo (reduzindo o comprimento da janela) a resolução no domínio da frequência piora, e vice versa.

1.5. PRÉ-ÊNFASE

Antes da análise LPC, o sinal janelado é processado com um filtro passa-altas para compensar a queda em frequência devido à irradiação da onda acústica pelos lábios do locutor, e ao pulso glotal [7]. Esta queda é de aproximadamente 6 dB/oitava e pode ser compensada com um filtro FIR (Pe) de primeira ordem:

$$Pe(z) = 1 - 0.9\cdot z^{-1} \qquad (1.32)$$

1.6. PARÂMETROS TEMPORAIS

Dentre os vários tipos de parâmetros temporais, que podem ser utilizados em complemento com os parâmetros freqüenciais, está a energia E_q por quadro, definida como:

$$E_q = \sum_{i=0}^{N-1} x_i^2 \qquad (1.33)$$

onde x_i são amostras do sinal de voz janelado, sendo N a largura da janela.

Como a percepção da intesidade se aproxima mais de uma função logarítmica do que de uma linear, é interessante trabalhar com o logaritmo da energia, isto é, em dB:

$$E_q(dB)=10\cdot\log\left[\left(\sum_{i=0}^{N-1} x_i\right)^2 / E_{max}\right] \qquad (1.34)$$

onde E_{max} é a energia máxima dentre os quadros da palavra que se analisa.

Além da energia por quadro, há outros tipos de parâmetros temporais. Entre os mais comuns podemos citar a taxa de cruzamentos por zero, a taxa de picos, etc. De maneira geral os parâmetros temporais são mais rápidos de calcular que os espectrais, mas não oferecem uma descrição tão completa quanto estes. Neste trabalho, a atenção estará concentrada nas parametrizações no domínio da freqüência.

1.7. QUANTIZAÇÃO VETORIAL

A quantização é um processo de aproximação pelo qual um sinal cuja amplitude apresenta valores contínuos fica representado por um novo sinal cuja amplitude só pode assumir valores discretos. A quantização é uma ferramenta muito útil para codificação pois o número de bits para representar uma variável pode ser reduzido quanto se quiser, cumprindo a exigência de distorção máxima permitida. No caso de termos uma só variável, a quantificação é escalar. Contudo, quando temos mais de um parâmetro (como os obtidos da análise LPC, mel-cepstrum, etc...) a quantificação é dita vetorial.

Suponhamos que trabalhamos com k coeficientes obtidos a partir da parametrização do sinal de voz em quadros de igual duração. Os coeficientes de cada quadro $T=(t_1, t_2, t_3,..., t_k)$ são representados por um ponto no espaço R^k de k dimensões. A idéia da quantização vetorial é dividir este espaço de k dimensões em L células (figura 1.2), sendo que cada célula (C_i) é representada por um centróide z_i, i=1, 2, 3,...,L . Mede-se então a distância entre os parâmetros do quadro e cada um dos centróides, e substitui-se o valor dos coeficientes do quadro pelo centróide mais próximo. Denominando o processo de quantização vetorial por q(), temos :

$$q(T) = z_i \mid d(X, z_i) \leq d(X, z_j) \quad , i \neq j \tag{1.35}$$

podemos então dizer que X está dentro da célula C_i.

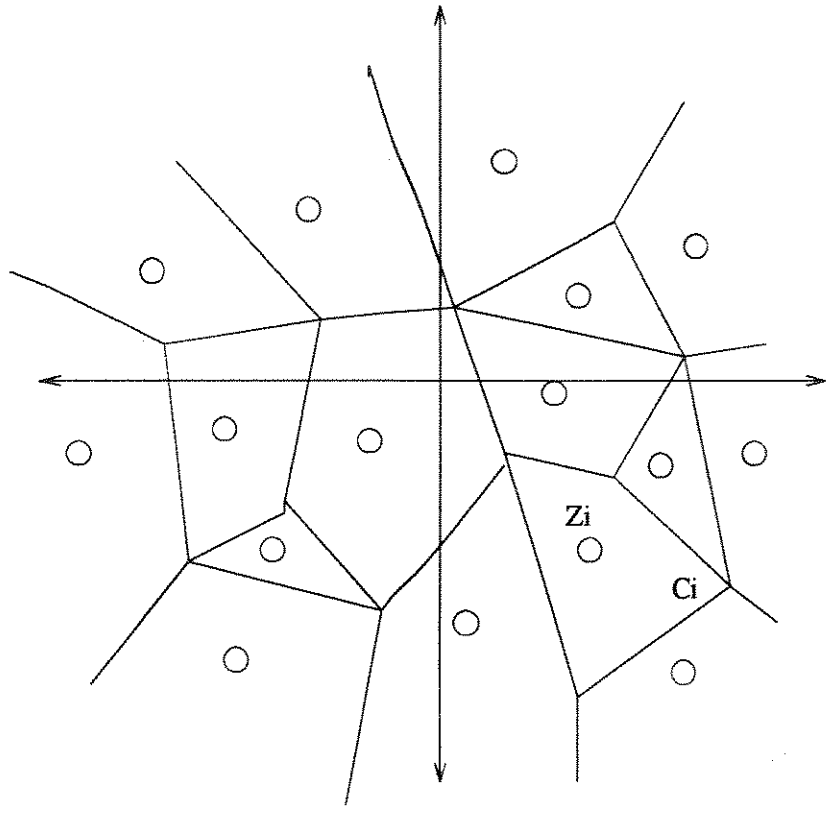


Figura 1.2. : Exemplo de quantificação vetorial num espaço de duas dimensões dividido em 16 células.

O conjunto de todos os z_i , $1 \leq i \leq L$ é denominado de codebook (livro código), e cada z_i é um codeword (palavra código).

Um dos métodos mais comumente utilizados para a elaboração do codebook é o algoritmo "k-means" [10]. O critério utilizado por esta técnica é a minimização da distorção global (D) [11], definida como :

$$D = \sum_{i=1}^{L} D_i \tag{1.36}$$

onde D_i é a distorção dentro da célula i,

$$D_i = \frac{1}{N_i} \cdot \sum_{T \in C_i} d(T, z_i) \qquad (1.37)$$

onde $d(T,z_i)$ é a distância entre um padrão de teste localizado na célula C_i e o centróide z_i da célula; N_i é o número de elementos T dentro da célula C_i.

Dado o critério da minimização da distorção global, que conduz à minimização de D_i, $1 \leq i \leq L$, pode-se provar que os centróides podem ser determinados por [10]:

$$z_i = \frac{1}{N_i} \cdot \sum_{T \in C_i} T \qquad (1.38)$$

isto é, a média aritmética dos elementos dentro da célula.

O algoritmo das "K-means", que gera um codebook ótimo minimizando iterativamente a distorção global média, é o seguinte:

Algoritmo "K-means"

Passo 1 : Inicialização. Escolher, utilizando um método adequado, um codebook inicial (z_i, $1 \leq i \leq L$);

Passo 2 : Classificação. Classificar cada vetor T (quadro) dos dados de treinamento numa das L células, escolhendo o centróide mais próximo :

$$q(T) = z_i \mid d(T, z_i) \leq d(T, z_j) \quad , \quad i \neq j$$

(classificação por mínima distância);

Passo 3 : Atualização do Codebook . re-calcular cada codeword através da média aritmética dos elementos no interior da célula correspondente;

Passo 4 : Teste de Convergência . Se a redução da distorção global média for menor que um certo limiar, **PARAR**. Caso contrário, retornar ao passo 2.

1.8. CONCLUSÃO

Foram abordadas neste capítulo as técnicas básicas de processamento digital aplicáveis ao sinal de voz após a segmentação palavra-silêncio e antes do reconhecimento de padrões acústicos. Como resultado final temos que o sinal no tempo é substituído por uma sequência de vetores de parâmetros("frames" ou quadros), geralmente obtidos por meio de análises espectrais em intervalos de igual duração do sinal em questão. Esta sequência de vetores de parâmetros também pode ser denominada de sequência de observação, pois é formada por coeficientes gerados a partir de medidas físicas realizadas sobre o sinal que transporta a informação fonética.

Por último, foi visto que cada "frame" ou quadro pode ser associado (quantizado), pelo critério de distância mínima, a um padrão ("codeword") pertencente a um conjunto finito ("code-book") de elementos. Assim, o quadro é substituído pelo padrão que a ele mais se assemelha e a seqüência de observação, que era formada por vetores de parâmetros, resulta numa sequência de números inteiros. Cada número corresponde ao "codeword" associado a um "frame". Este processo de quantificação vetorial será utilizado na implementação dos HMM discretos.

1.9. REFERÊNCIAS

[1] Flanagan,J.L. : "Speech Analysis Synthesis and Perception". Second Edition, Springer-Verlag, 1972.

[2] Fant,G. : "Vowels, Production and Perception". Dept. of Speech Communication, Royal Institute of Technology (KTH), S-100 44, Stockolm 70, Sweden, 1975.

[3] Carlson,R. Fant,G. Granstrom,B. : "Two Formant Models, Pitch and Vowel Perception", Dpt. of Speech Communication, Royal Institute of Technology (KTH), Auditory Analysis and Perception of Speech, Edited by G.Fant and M.A.A. Tatham, Academic Press, 1975.

[4] Rabiner,L.R. Schafer,R.W. : "Digital Processing of Speech Signals". Prentice Hall, 1978.

[5] Dautrich,B.A. Rabiner,L.R. Martin,T.B. :"On the Effects of Varying Filter Bank Parameters on Isolated Word Recognition". IEEE Trans. ASSP, vol.31, pp. 793-806, 1983.

[6] Davis,S.B. and Mermelstein,P. : "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences", IEEE Trans. ASSP, vol.28, pp.357-366, 1980

[7] O'Shaughnessy, D. : "Speech Communication, Human and Machine",INRS-Telecomunications, Addison-Wesley Publishing Company,1987.

[8] Haykin, S. : "Adaptive Filter Theory", Prentice Hall, Englewood Cliffs, New Jersey, 1986.

[9] Oppenheim, A.V. Schafer,R.W. : "Digital Signal Processing". Prentice Hall, Englewood Cliffs, New Jersey 1975.

[10] Huang,X.D. Ariki,Y. Jack,M.A. : "Hidden Markov Models for Speech recognition", Edinburgh University Press, 1990.

[11] Gersho, A. : "On the Structure of Vector Quantizers", IEEE Trans. Information Theory, vol.IT-28, pp.157-166, 1982.

CAPÍTULO 2

RECONHECIMENTO DE PADRÕES ACÚSTICOS : ANÁLISE DETERMINÍSTICA E ANÁLISE ESTOCÁSTICA

2.1. INTRODUÇÃO

Até agora foram apresentadas as técnicas básicas mais comumente utilizadas no processamento do sinal de voz prévio ao reconhecimento. Como resultado deste processamento, as palavras são divididas em segmentos de igual duração onde se extraem uma série de parâmetros, que devem estar intimamente correlacionados com a identidade fonética do segmento em questão. Sendo assim, cada palavra, após esta parametrização, fica representada por uma sequência temporal de vetores de parâmetros. Como estes coeficientes são obtidos por meio de medidas físicas sobre o sinal de voz, podemos dizer também que cada palavra fica representada por uma sequência temporal de vetores de observação.

O reconhecimento de padrões acústicos, e nesta tese também de palavras, consiste em associar uma seqüência de vetores de parâmetros ou de observação, relativos a uma palavra pronunciada, a uma das seqüências ou modelos de referência. A elocução a ser reconhecida não pode fazer parte, é claro, da base de dados utilizada para obter as seqüências ou modelos de referências (treinamento)

O avanço do reconhecimento de voz nos últimos 20 anos se deve ao advento de três técnicas básicas : o DTW ("Dynamic Time Warping"), os HMM ("Hidden Markov Models") e, mais recentemente, as redes neurais. Todos os sistemas de reconhecimento automático de voz, comerciais ou em pesquisa, trabalham com uma ou mais destas três técnicas, sendo que hoje em dia quase a totalidade dos sistemas mais modernos utilizam os HMM. O grande mérito desta última foi modelar estatisticamente a ocorrência dos vetores de parâmetros e a duração das palavras ou fonemas.

O DTW tem o mérito de ser a primeira técnica utilizada com sucesso em reconhecimento de palavras isoladas com vocabulário pequeno, tendo como ponto forte a capacidade em alinhar seqüências de vetores de parâmetros (palavras) com diferentes durações. As redes neuronais, por último, constituem uma técnica

promissora pela flexibilidade em assimilar superfícies de decisão formadas por estatísticas complexas, mas têm como desvantagem a dificuldade em lidar com variações temporais. Neste sentido as TDNN ("Time Delay Neural Nets") constituem uma tentativa de corrigir esta deficiência [3].

Nesta dissertação utilizaremos o algoritmo clássico do DTW como ferramenta de estudo de algumas parametrizações, e os HMM como técnica para implementar o reconhecimento automático de palavras isoladas, independente do locutor, num vocabulário pequeno (dígitos de 0 a 9).

2.2. DTW ("DYNAMIC TIME WARPING" OU ALINHAMENTO NÃO-LINEAR)

O "Dynamic Time Warping" [1][2] foi introduzido para contornar o problema causado pelas variações na velocidade de pronunciação das palavras no reconhecimento automático. Na sua versão 'solo', o DTW surgiu nos finais dos anos 70, sendo que também é utilizado hoje em dia nos HMM, onde é mais conhecido como algoritmo de decodificação de Viterbi [3].

Suponhamos que temos duas seqüências (R e T) de vetores de parâmetros ou de observação, cada seqüência correspondendo a uma palavra. Assim sendo:

$$X = X_1 , X_2 , X_3 , \ldots , X_{Tx}$$

$$Y = Y_1 , Y_2 , Y_3 , \ldots , Y_{Ty}$$

onde X_i e Y_j correspondem, cada um, a um vetor de parâmetros (coeficientes LPC, Mel-cepstral, etc) calculados dentro de um quadro. A duração da seqüência X é Tx, e a duração da seqüência Y é Ty, em número de quadros. Suponhamos que, para efeito de raciocínio, Ty > Tx ou seja que Y tem duração maior que X. Consideraremos também que os vetores X_i e Y_j são unidimensionais, para facilitar a visualização gráfica do método. Na figura 2.1(a) temos a seqüência X e na figura 2.1(b) a seqüência Y, para o caso unidimensional. Embora discretizadas no tempo, as seqüencias X e Y estão representadas como funções contínuas por simplificação.

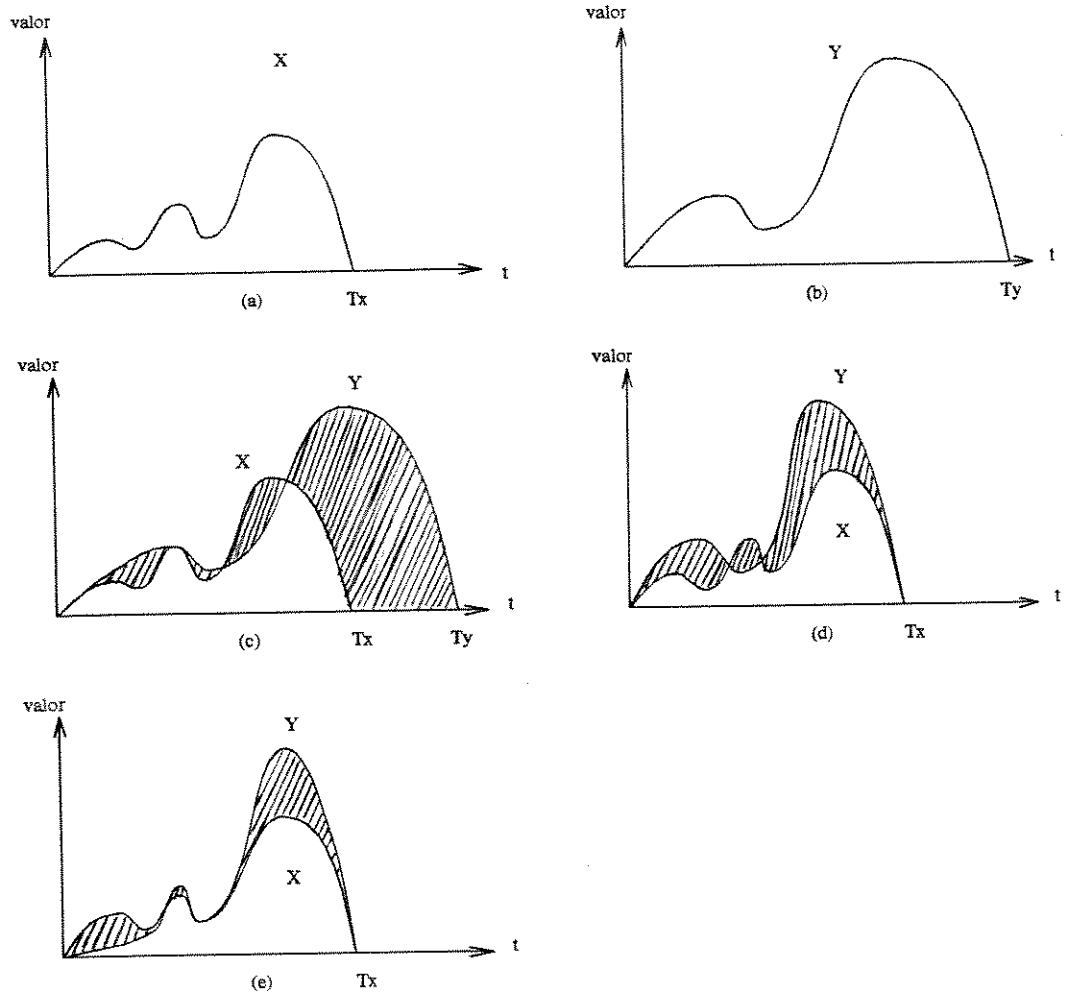


Figura 2.1.: Padrões acústicos X e Y, e os diferentes métodos de comparação: (a) seqüência X; (b) seqüência Y; (c) comparação sem alinhamento; (d) alinhamento linear; e (e) alinhamento não-linear [3].

Se quisermos comparar os dois padrões X e Y, temos três opções. A primeira consistiria em comparar diretamente as duas seqüências, computando a distância entre quadros $d(X_i,Y_i)$ para $1 \leq i \leq Tx$. Para $Tx < i \leq Ty$, X_i pode ser consiserado como 0. Sendo assim, a distância total ao comparar as duas seqüências, D(X,Y), ficaria representada pela área sombreada da figura 2.1 (c).

O segundo método é o conhecido como alinhamento linear e comprime li-

nearmente a seqüência Y, fazendo Tx = Ty; em seguida, calcula-se a distância total D(X,Y) como no caso anterior (área sombreada da figura 2.1(d)).

O terceiro método é o denominado de alinhamento não-linear, que comprime e dilata de maneira não-linear as seqüências X e Y, comparando-as ao longo do eixo temporal resultante. A distância total D(X, Y) para o alinhamento não-linear é representada na figura 2.1 (e). Comparando a área sombreada das figuras 2.1(d) e 2.1(e) podemos ver que a distância acumulada no último caso é menor que no anterior.

De uma maneira geral, a realização acústica de uma mesma palavra (elocução) por um mesmo locutor (ainda mais em locutores diferentes) pode variar significativamente, modificando a velocidade de articulação do aparelho fonador. Assim, uma palavra pode ser pronunciada mais ou menos rapidamente, encurtando ou alongando os períodos estacionários do sinal de voz, enquanto que os períodos não-estacionários se mantêm quase sempre constantes. No DTW o que se procura é eliminar estas diferenças na duração dos períodos estacionários, que são insignificantes do ponto de vista fonético e semântico.

Consideraremos a comparação entre dois padrões acústicos X e Y com alinhamento temporal num plano de duas dimensões, figura 2.2., onde cada seqüência X e Y se localiza num dos eixos ortogonais do plano (X em x e Y em y, por exemplo). Seja i o índice da seqüência X, e j o índice da seqüência Y, os pares **c(k)=(i(k), j(k))** indicam os elementos de X e Y que são emparelhados no alinhamento. Assim, a seqüência

$$F = c(1), c(2), \ldots, c(K)$$

corresponde ao caminho ou à função de alinhamento temporal.

As seqüências X e Y de quadros ou vetores de parâmetros são posicionadas (veja figura 2.2) de maneira a que os primeiros quadros de cada seqüência se localizem no canto inferior esquerdo do gráfico. Os vetores seguintes de X e Y são posicionados segundo a direção do eixo x e y, respectivamente. A inclinação do caminho de alinhamento é uma medida da compressão de X ao ser comparada com Y. Por exemplo, um passo vertical significa que dois quadros de Y são emparelhados com um mesmo quadro de X, e um passo horizontal corresponde a dois quadros de X comparados com um mesmo quadro de Y.

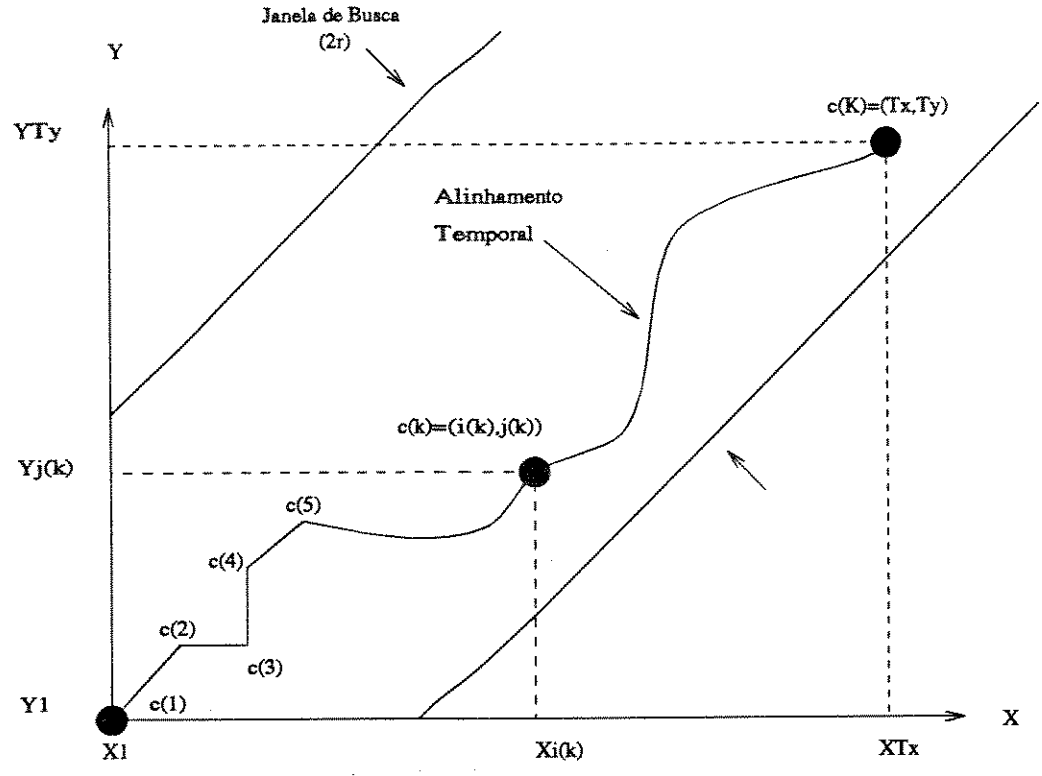


Figura 2.2.: Alinhamento não-linear entre palavras no DTW. Cada palavra é representada por uma seqüência de vetores de parâmetros [1].

A distância total é então a soma ponderada de todas as distâncias locais entre quadros $d(c(k)) = d(X_{i(k)}, Y_{j(k)})$ ao longo do caminho de alinhamento, podendo ser expressa como:

$$D(X,Y,F) = \frac{\sum_{k=1}^{K} d(c(k)) \cdot w(k)}{\sum_{k=1}^{K} w(k)} \qquad (2.1)$$

onde:

$$d(c(k)) = d(X_{i(k)}, Y_{j(k)}), \qquad (2.2)$$

K é o comprimento total do caminho F e w(k) é o peso que pondera diferenciadamente cada medida de distância local, sendo função da inclinação do caminho F imediatamente anterior a k. A idéia de w(k) [1] é penalizar os pontos que se afastam da diagonal principal.

Formalizando, o problema a resolver no DTW consiste em achar o caminho F que minimize a função distância total D(X,Y,F), dadas as restrições de coincidência dos pontos terminais, continuidade e monotonicidade [1][3], e de que o caminho F se encontre dentro da janela de busca definida por $2 \cdot r$ na figura 2.2. A variável r define a diferença máxima (em módulo) permitida no comprimento ou duração das palavras que estão sendo comparadas [1]. Temos então que :

$$D(X,Y) = \min_{F} \frac{\sum_{k=1}^{K} d(c(k)) \cdot w(k)}{\sum_{k=1}^{K} w(k)} \qquad (2.3)$$

Dadas as seguintes restrições para F [1]:

1. Coincidências dos pontos terminais;

   No ponto inicial: $i(1)=j(1)=1$ (2.4)

   No ponto final : $i(K)=Tx$ e $j(k)=Ty$ (2.5)

2. Continuidade e Monotonicidade ;

   Para garantir a monotonicidade entre pares de quadros consecutivos c(k) e c(k-1), estabelecemos que :

   $$0 \leq i(k)-i(k-1) \quad , \quad 0 \leq j(k)-j(k-1) \qquad (2.6)$$

   e, para a continuidade:

   $$i(k)-i(k-1) \leq 1 \quad , \quad j(k)-j(k-1) \leq 1 \qquad (2.7)$$

   Logo, c(k-1) pode ser expresso como :

$$c(k-1) = \begin{bmatrix} (i(k),\ j(k)-1) \\ (i(k)-1,\ j(k)-1) \\ (i(k)-1,\ j(k)) \end{bmatrix} \qquad (2.8)$$

A relação acima pode ser chamada de restrição local e estabelece que, dado um ponto c(k) no caminho de alinhamento, só existem três possíveis c(k-1), dadas as restrições de monotonicidade e de continuidade.

3. Janela de busca.
   A fim de evitar um caminho de alinhamento F pouco razoável e de limitar a carga computacional na escolha de F que minimize D(X,Y), estabelece-se uma região onde é permitida a busca do F ótimo. Esta região pode ser limitada por duas linhas paralelas espaçadas por $2 \cdot r$, como proposto inicialmente por Sakoe-Chiba [1] (figura 2.2), ou por um paralelogramo, como proposto por Rabiner et al. [2].

O denominador da expressão (2.3) é o fator de normalização da distância total, como veremos a seguir, em relação ao comprimento de X e Y [1], numa tentativa de medir a diferença ou a semelhança entre palavras, independentemente de suas durações. Isto é coerente também com a idéia de eliminar as diferenças temporais relacionadas com os períodos estacionários da voz. Pode-se definir w(k) de duas maneiras :

1. Simétrica :

$$w(k) = (i(k)-i(k-1))+(j(k)-j(k-1)) \qquad (2.9)$$

utilizando a definição (2.9) na expressão do denominador da equação (2.3), temos:

$$\sum_{k=1}^{K} w(k) = Tx + Ty \qquad (2.10)$$

2. Assimétrica :

$$w(k) = i(k)-i(k-1), \qquad (2.11)$$

O que conduz a:

$$\sum_{k=1}^{K} w(k) = Tx \tag{2.12}$$

Consequentemente, a distância mínima total ao alinhar os dois padrões, utilizando a definição simétrica para w(k) [1], é:

$$D(X,Y) = \min_{F} \frac{\sum_{k=1}^{K} d(c(k)) \cdot w(k)}{Tx + Ty} \tag{2.13}$$

A solução do problema de minimizar (2.13) selecionando o melhor caminho de alinhamento $F = c(1), c(2), c(3),\ldots,c(K)$, pode ser vista, em primeira instância, como um processo de decisão em forma de árvore com K estágios. Contudo, utilizando ferramentas de programação dinâmica, é possível tratar este problema como uma seqüência de K processos de decisão de um estágio só. Graças a esta decomposição, é possível reduzir o tempo de computação requerido para achar o caminho ótimo.

Seja G(c(K)) a mínima distância D(X,Y) sem o denominador Tx + Ty da expressão (2.13). Como G(c(k)) representa a mínima distância acumulada entre quadros de k=1 a k=K, temos que:

$$G(c(K)) = G(Tx,Ty) = \min_{c(1),\ldots,c(K-1)} \sum_{k=1}^{K} d(c(k)) \cdot w(k) , \tag{2.14}$$

onde c(K) é fixo. A expressão acima pode ser expandida novamente :

$$G(c(K)) = G(Tx,Ty) = \min_{c(1),\ldots,c(K-1)} \left[\sum_{k=1}^{K-1} d(c(k)) \cdot w(k) + d(c(K)) \cdot w(K) \right] = \tag{2.15}$$

$$= \min_{c(K-1)} \left[\min_{c(1),\ldots,c(K-2)} \left[\sum_{k=1}^{K-1} d(c(k)) \cdot w(k) \right] + d(c(K)) \cdot w(K) \right]$$

O primeiro termo dentro do parênteses externo pode ser substituído por G(c(K-1)). Logo a expressão pode ser reescrita como :

$$G(c(K)) = \min_{c(K-1)} \left[G(c(K-1)) + d(c(K)) \cdot w(K) \right] \qquad (2.16)$$

Generalizando, substituindo K por k ($1 \leq k \leq K$), temos:

$$G(c(k)) = \min_{c(k-1)} \left[G(c(k-1)) + d(c(k)) \cdot w(k) \right] \qquad (2.17)$$

A expressão acima indica que a seqüência de K processos de decisão de um estágio substitui o processo inicial de K estágios. Esta é precisamente a expressão matemática para o princípio de otimização, no qual se baseia a programação dinâmica:

"Um critério ótimo tem a propriedade de que qualquer que seja o estado inicial e a decisão inicial, as decisões restantes devem constituir uma estratégia ótima com relação ao estado resultante da decisão anterior" [4].

Empregando as expressões (2.8) e (2.9), a expressão (2.17) pode ser escrita como [1]:

$$G(i,j) = \min \begin{bmatrix} G(i,\ j-1) + d(X_i, Y_j) \\ G(i-1,\ j-1) + 2 \cdot d(X_i, Y_j) \\ G(i-1,\ j) + d(X_i, Y_j) \end{bmatrix} \qquad (2.18)$$

com:

$$g(1,1) = 2 \cdot d(1,1)$$

e a seguinte restrição envolvendo a janela de busca:

$$j-r \leq i \leq j+r \qquad (2.19)$$

O DTW é um método não-paramétrico, isto é, cada padrão de referência corresponde a uma palavra pronunciada, o que conduz à necessidade de se ter vários padrões de uma mesma palavra, na intenção de cobrir todas as variações possíveis na sua elocução, por um ou mais locutores. Isto tem como conseqüência um

aumento da carga computacional e de memória. Se levarmos em conta que a comparação de duas palavras pelo alinhamento não-linear exige por si só um tempo de cálculo elevado, a utilização do DTW é inviável para grandes vocabulários (>5000 palavras) e extremamente difícil para vocabulários médios (>1000 palavras). Neste sentido, tornou-se interessante o emprego de um método paramétrico, como o dos HMM, que permite acumular informações de várias elocuções de uma mesma palavra nos parâmetros de um só modelo. Contudo, como veremos a seguir, o DTW pode ser visto como um caso particular do HMM, uma vez que este utiliza o algoritmo de Viterbi, muito parecido com a algoritmo do DTW que apresentamos aqui.

Por outro lado, o DTW é apresentado nesta dissertação como uma ferramenta interessante para o estudo de parametrizações (LPC, LPC-cepstral, Mel-cepstral, etc). A extrema dependência do DTW em relação à parametrização é utilizada para estudar as próprias parametrizações em questão. Do ponto de vista prático podemos mencionar as seguintes vantagens deste procedimento:

1. Nos HMM é preciso treinar os modelos com as elocuções de treinamento e, no caso dos HMM discretos, otimizar o "code-book" a cada troca de parâmetro. No DTW basta parametrizar novamente a base de dados. Isto, para uma seqüência exaustiva de testes, é um ponto importante.

2. Nos HMM é preciso uma quantidade razoável de dados para treinamento dos modelos, mesmo para testes dependentes de locutor. Com o DTW pode-se comparar uma elocução de uma palavra (padrão de referência) com várias elocuções da mesma palavra (padrões de teste), uma de cada vez, feitas por um ou vários locutores. Assim, o DTW é extremamente dependente da parametrização utilizada e, por coseqüência, da medida de distância entre quadros. Logo, o resultado da análise pelo DTW é uma ótima medida da qualidade da parametrização empregada.

2.3. HMM ("HIDDEN MARKOV MODELS" OU MODELOS OCULTOS DE MARKOV)

Os HMM têm sido a melhor técnica aplicada ao problema de reconhecimento de palavras faladas. Em primeiro lugar, por ser uma ferramenta baseada em modelamento estocástico, adapta-se muito bem ao fenômeno da fala uma vez que esta apresenta uma variabilidade intra e inter-locutor muito grande. Estas diferen-

ças são principalmente do tipo espectral, relacionadas com a produção dos fonemas, e temporais, relacionadas com a duração dos mesmos. Fora isto, há também uma variabilidade considerável de coarticulação que depende do contexto em que aparecem os fonemas, mesmo se tratando de um mesmo locutor.

Em segundo lugar, os HMM são muito flexíveis e permitem modelar uma palavra inteira, ou mesmo fonemas ou difones. Adapta-se, portanto,ao problema de reconhecimento de palavras isoladas ou contínuas, com vocabulários pequenos ou grandes.

Por último, os HMM conseguem assimilar a informação de uma grande quantidade de dados de treinamento nos parâmetros dos modelos, permitindo, na operação de reconhecimento, lidar com muitas elocuções de referência ao mesmo tempo.

2.3.1. Processos de Markov

Em muitos processos, um evento atual é determinado, ou influenciado pelos eventos anteriores. Na linguagem escrita [7] ou natural, por exemplo, uma letra pode ser prevista em função das anteriores, ou mesmo uma palavra omitida pode ser deduzida em função do contexto. Podemos definir então [3] a propriedade de Markov segundo a qual para qualquer seqüência temporal de eventos, a densidade de probabilidade condicional de um evento atual depende somente dos j eventos mais recentes. Um processo que satisfaz esta propriedade é dito de Markov, sendo que no caso genérico da definição dissemos que é de ordem j. Contudo, restringiremos nosso estudo aos processos markovianos de primeira ordem pois são estes que se utilizam em reconhecimento de palavras.

As cadeias de Markov consideradas aqui se constituem de um número finito de estados e transições entre estes, sendo que cada transição se caracteriza por uma probabilidade. As cadeias de Markov têm muita importância em estudos de comunicações e foram abordadas nos artigos já clássicos de Shannon [6] [7]. Por sua vez, nas cadeias ocultas de Markov não se tem acesso ao estado atual da cadeia, mas sim às medidas físicas que são funções estatísticas do estado em questão. Baum, Petri e colaboradores [8] desenvolveram um algoritmo baseado na maximização da verossimilhança para determinar os parâmetros das cadeias ocultas de Markov (probabilidade das transições entre estados e funções de probabilidade de observação de medidas físicas em cada estado), e sugeriram certas aplicações

como ecologia e predição das condições meteorológicas, sem mencionar sequer o problema de reconhecimento de voz. Jellinek, da IBM, aplicou com sucesso ferramentas estatísticas ao problema de reconhecimento de palavras [9] e foi um dos primeiros, se não o primeiro, a aplicar a técnica dos modelos ocultos de Markov neste campo. Nos finais dos anos 80 estas técnicas foram a tal ponto 'popularizadas' que se tornou quase impossível falar em reconhecimento de voz sem falar nos modelos de Markov.

Como um exemplo de um processo de Markov de primeira ordem, suponhamos que temos três símbolos (a, b, c). A probabilidade do símbolo "a" ser seguido por qualquer um dos três "a", "b" ou "c" é de 1/3. A probabilidade do símbolo "b" ser seguido pelo mesmo "b" é de 1/2, e por qualquer um dos outros ("a" ou "c") é de 1/4. Finalmente a probabilidade do símbolo "c" ser seguido por "c" é de 1/2, e por qualquer um dos outros é também de 1/4. Temos então:

$$\begin{aligned} &Pr\ (a/a) = 1/3\ ,\ Pr(b/a) = 1/3\ ,\ Pr(c/a) = 1/3 \\ &Pr\ (a/b) = 1/4\ ,\ Pr(b/b) = 1/2\ ,\ Pr(c/b) = 1/4 \\ &Pr\ (a/c) = 1/4\ ,\ Pr(b/c) = 1/4\ ,\ Pr(c/c) = 1/2 \end{aligned} \qquad (2.20)$$

Na figura 2.3. cada linha direcionada é uma transição de um estado para um outro, sendo que sua probabilidade é indicada pelo número próximo da linha. Por exemplo, Pr(a/b) corresponde à linha que vai do estado "b" para o estado "a" e indica a probabilidade de transição de 1/4 entre estes dois estados. Este tipo de modelo de Markov pode ser utilizado para descrição do estado meteorológico. Seja o estado "a" correspondendo a 'ensolarado', o "b" a 'nublado', e o "c" a chuvoso. Dado que hoje está ensolarado, a probabilidade de amanhã estar também ensolarado é a mesma de estar chuvoso ou nublado. Mas, se hoje está ou chuvoso ou nublado, há 50% de probabilidade de continuar no mesmo estado amanhã. Deste modo, o processo estocástico relacionado ao estado meteorológico pode ser descrito, numa primeira aproximação, como um processo markoviano.

Se ordenamos os estados "a", "b" ou "c" com os números 1, 2 e 3, respectivamente, o grafo da figura 2.3 pode ser substituido pela matriz de transições A, definida como:

$$A = \begin{bmatrix} Pr\ (a/a) & Pr(b/a) & Pr(c/a) \\ Pr\ (a/b) & Pr(b/b) & Pr(c/b) \\ Pr\ (a/c) & Pr(b/c) & Pr(c/c) \end{bmatrix} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/4 & 1/2 & 1/4 \\ 1/4 & 1/4 & 1/2 \end{bmatrix} \qquad (2.21)$$

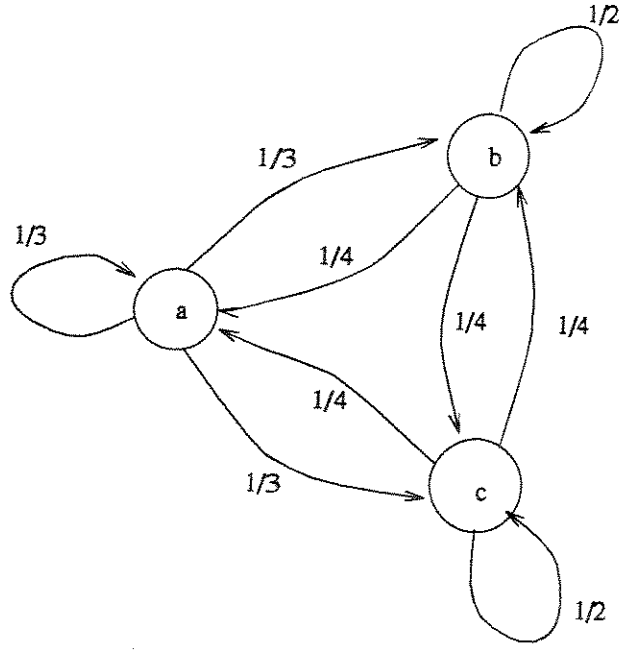


Figura 2.3 :O modelo de Markov definido pelas probabilidades de (2.20) [3].

Podemos observar que na matriz A, a soma dos elementos em qualquer linha é sempre igual a 1, devido à lei da probabilidade total.

2.3.2. Definição dos Modelos Ocultos de Markov (HMM)

Para entender o conceito dos HMM, consideramos o seguinte exemplo [3] ilustrado na figura 2.4. Uma pessoa realiza um experimento atrás de uma cortina. Há N=3 cestas contendo várias bolas coloridas, sendo que há L=4 cores diferentes. Inicialmente uma cesta é escolhida aleatoriamente. Desta cesta, uma bola é escohida também aleatoriamente. A pessoa diz em voz alta a cor da bola e a coloca novamente na cesta de orígem e uma nova cesta é escolhida sob as mesmas condições, repetindo o processo. Este experimento gera uma seqüência finita de bolas coloridas que são retiradas da cesta, sem se saber, a priori, de que

cestas foram retiradas. Este processo se compõe então de uma informação observável, que é a cor das bolas retiradas das cestas, e de uma informação oculta, as cestas de onde foram retiradas as bolas. Os HMM são uma técnica que visam modelar ou estabelecer a probabilidade de uma seqüência de cestas (informação oculta) dada uma seqüência de bolas coloridas (informação disponível).

Para formalizar o conceito dos HMM, utilizaremos as seguintes notações:

D = comprimento da seqüência de observação, $O_1, O_2, ..., O_D$; (número de bolas coloridas observadas no nosso experimento)

N = número de estados no modelo (números de cestas);

O_t = vetor de observação resultante de medidas físicas no quadro t;

L = número de símbolos observáveis (número de cores diferentes);

S = {s}, um conjunto de estados. Por simplificação, o estado i no instante t pode ser denominado $s_t = i$;

V = $\{v_1, v_2, ..., v_L\}$, conjunto finito de possíveis símbolos observáveis. O_t é relacionadoa um dos elementos do conjunto V por meio da quantização vetorial;

A = $\{ a_{ij} | a_{ij} = Pr(s_{t+1}=j | s_t=i)\}$, distribuição de probabilidade de transição entre estados; a_{ij} indica a probabilidade de transição do estado i para o j;

B = $\{b_j(O_t) | b_j(O_t) = Pr(O_t | s_t=j)\}$. Para cada estado há uma correspondente função de probabilidade de saída (função de distribuição de probabiliade discreta ou contínua); todas estas funções representam variáveis aleatórias ou processos estocásticos a serem modelados. Nos HMM discretos, referimo-nos à probabilidade de gerar um dado símbolo discreto V_k no estado j, denotada por $b_j(V_k)$, onde V_k é obtido por quantificação vetorial de O_t. Nos HMM contínuos, faz-se referência diretamente à função de densidade de probabilidade de emissão de observações O_t. Neste caso O_t não é quantificado. Dadas

estas diferenças, os algoritmos de re-estimação dos parâmetros dos HMM discretos e dos HMM contínuos são diferentes, embora as deduções das equações de re-estimação sejam parecidas.

π = $\{\pi_i | \ \pi_i = Pr(s_1 = i)\}$, função de distribuição de probabilidade para o estado inicial;

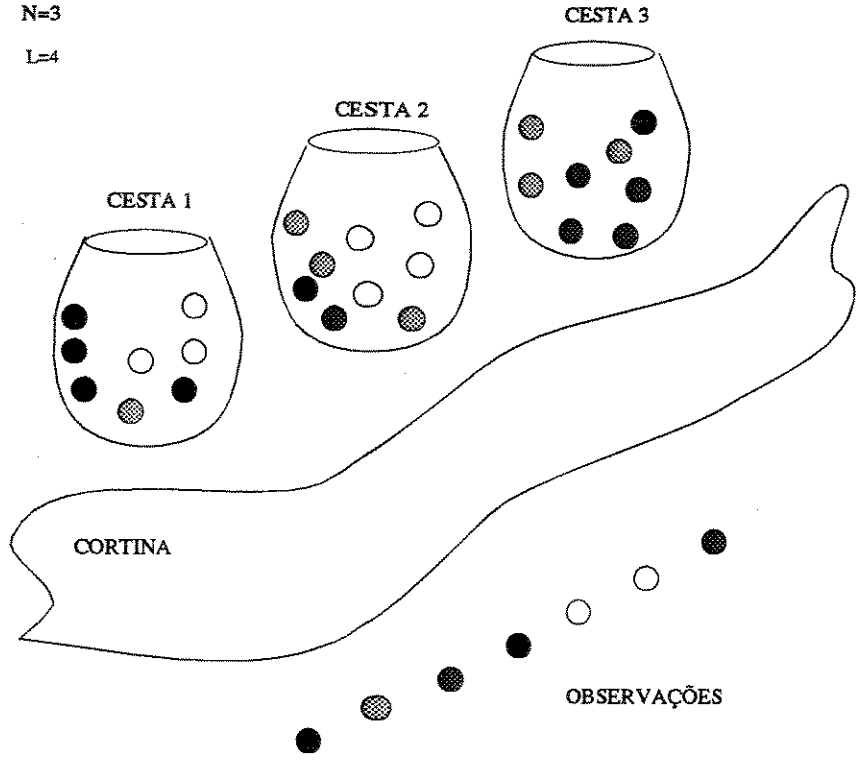


Figura 2.4.: Experimento das cestas e das bolas coloridas realizado atrás de uma cortina (modificado segundo [3]).

Assim sendo, um HMM pode ser representado por $\lambda=(A, B, \pi)$. A especificação de um HMM envolve a escolha de um número de estados, N, o número de símbolos discretos L (para o caso de HMM discreto), e a especificação de três densidades de probabilidades indicadas pelas matrizes A, B, e π. Também podemos fixar um

conjunto de estados iniciais e finais, S_I e S_F respectivamente. Por exemplo, em se tratando de reconhecimento de voz, é razoável supor que uma palavra termine e comece em silêncio. Assim, as transições devem começar num dos estados S_I e terminar num dos estados S_F. Na prática, o número de estados iniciais e finais, N_I e N_F, são frequentemente fixados em 1.

O experimento das bolas coloridas e das cestas pode ser modelado com um HMM utilizando as definições acima : cada estado, i, corresponde a uma cesta; as probabilidades de saída $b_i(O)$ são definidas pelas bolas coloridas observáveis em cada estado, isto é, a distribuição de probabilidade de bolas coloridas em cada urna. O símbolo observado v_k, é a cor da bola selecionada de uma das cestas. A escolha das cestas é modelada pela distribuição de estado inicial e pela distribuição de probabilidade de transição entre estados. A figura 2.5 mostra um HMM modelando este experimento. As probabilidades de transição são mostradas na figura, assim como as funções de probabilidade de saída associadas a cada estado. Este modelo (do experimento) pode ser considerado como gerador de uma seqüência de experimentos, uma vez que a cadeia de Markov gera uma seqüência de estados, a partir de um estado inicial, e a função de probabilidade de saída se encarrega de gerar a bola colorida correspondente a cada estado. A seqüência de observações, ou de bolas coloridas, dá uma idéia da seqüência de estados e dos parâmetros do modelo.

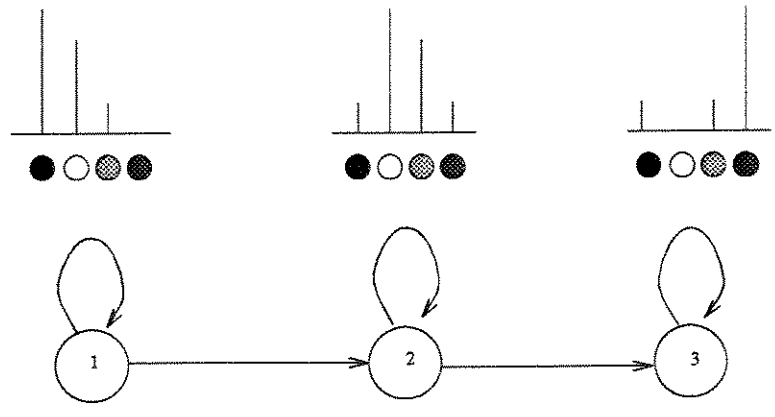


Figura 2.5.: Exemplo de um modelo de um HMM para o experimento das cestas e das bolas coloridas (modificado segundo [3]).

2.3.3. Algoritmos Básicos para os HMM

Dadas as definições dos parâmetros envolvidos nos HMM, discutiremos os três problemas básicos desta ténica:

1. **Problema de Avaliação:** dada a seqüência de observação $O = O_1, O_2, ..., O_T$, onde T é tamanho da seqüência O, e o modelo $\lambda=(A, B, \Pi)$, o problema é computar $Pr(O/\lambda)$, isto é, a probabilidade da observação ter sido gerada pelo modelo. Este problema também pode ser visto como parte de um mais geral: dados vários modelos concorrentes e uma seqüên cia de observação, como escolher o modelo que melhor se casa com a seqüência em questão, tendo em vista a classificação e reconhecimento da própria se qüência. O reconhecimento é portanto a escolha do modelo que maximize $P(O/\lambda)$;

2. **Problema de Estimação** : dada a seqüência de observação O, como se deve ajustar os parâmetros $\lambda = (A, B, \Pi)$ do modelo de maneira maximizar a probabilidade $Pr(O/\lambda)$. Em outras palavras, trata-se de otimizar os parâmetros do modelo de modo a melhor descrever a seqüência de observação que se lhe atribui;

3. **Problema de Decodificação:** Dada uma seqüência de observação O, qual é seqüência de estados $S=s_1, s_2, ..., s_T$, de maior verossimilhança. Trata-se de recuperar a informação oculta do modelo.

A seguir apresentaremos o tratamento matemático e os algoritmos necessários para a resolução destes problemas.

2.3.3.1 Algoritmo Forward-Backward

A maneira mais direta de computar a probabilidade de uma seqüência de observação é enumerando toda possível seqüência de estados de comprimento T (o número de observações). Para uma dada seqüência de estados $S = s_1, s_2, ..., s_T$, a probabilidade de uma seqüência de observação O, $Pr(O/S,\lambda)$, é :

$$Pr(O/S,\lambda) = b_{s_1}(O_1) \cdot b_{s_2}(O_2) \ldots b_{s_T}(O_T) \qquad (2.22)$$

Por outro lado, a seqüência S tem probabilidade $Pr(S/\lambda)$ dada por :

$$Pr(S/\lambda) = \pi_{s_1} \cdot a_{s_1 s_2} \cdot a_{s_2 s_3} \cdot \ldots \cdot a_{s_{T-1} s_T} \qquad (2.23)$$

Por simplicidade de notação chamaremos π_{s_1} de $a_{s_0 s_1}$.

O estado s_0 é fictício. A expressão (2.23) fica:

$$Pr(S/\lambda) = a_{s_0 s_1} \cdot a_{s_1 s_2} \cdot a_{s_2 s_3} \cdot \ldots \cdot a_{s_{T-1} s_T} \tag{2.24}$$

A probabilidade conjunta de O e S, isto é a probabilidae de O e S ocorrerem simultaneamente, é simplesmente o produto das expressões (2.22) e (2.24):

$$Pr(O,S/\lambda) = Pr(O/S,\lambda)Pr(S/\lambda) \tag{2.25}$$

A probabilidade $Pr(O/\lambda)$ é a somatória da equação (2.25) sobre todas as possíveis seqüências de estados:

$$Pr(O/\lambda) = \sum_{\substack{\text{todas as} \\ \text{seq. S}}} Pr(O/S,\lambda)Pr(S/\lambda) =$$

$$= \sum_{\substack{\text{todas as} \\ \text{seq. S}}} \prod_{t=1}^{T} a_{s_{t-1} s_t} \cdot b_{s_t}(O_t) \tag{2.26}$$

Da equação (2.26) podemos ver que uma transição começa em um estágio inicial (instante t=1) com probabilidade $a_{s_0 s_1}$, gerando o símbolo O_1 com a probabilidade de saída $b_{s_1}(O_1)$ no correspondente estado s_1, e a transição ocorre então do estado inicial s_1 para o estado s_2 com probabilidade $a_{s_1 s_2}$, gerando o símbolo O_2 com probabilidade de saída $b_{s_2}(O_2)$, correspondendo ao estado s_2. O processo continua até a última transição do estado s_{T-1} para o último estado s_T, com probabiliade de transição $a_{s_{T-1} s_T}$, e emitindo o último símbolo O_T com probabilidade de saída $b_{s_T}(O_T)$.

Pode-se mostrar [5] que para calcular $Pr(S/\lambda)$ pela expressão (2.26) são necessárias $2 \cdot T \cdot N^T$ operações. Como há somente N possíveis estados a cada instan-

te t=1,2,..., T, o cálculo de $Pr(S/\lambda)$ pode ser bastante simplificado pelos algoritmos Forward-Backward [3][5].

Definamos primeiro a variável forward, α_t , como:

$$\alpha_t(i) = Pr(O_1, O_2, ..., O_t, s_t=i/\lambda) \qquad (2.27)$$

Esta é a probabilidade de observar uma parte da seqüência de observação, de O_1 até O_t, e do estado no instante t ser $s_t=i$, para um determinado modelo de Markov. As probabilidades $\alpha_t(i)$ podem ser calculadas por indução por meio do algoritmo Forward, mostrado a seguir:

Algoritmo Forward

Passo 1 : $\alpha_1(i) = \pi_i b_i(O_1)$, para todos os estados i (se $i \in S_I$, então:

$\pi_i = 1/N_I$, caso contrário $\pi_i = 0$

(geralmente é feito $\pi_1 = 1$ e $\pi_i = 0$ para $i \neq 1$)

Passo 2 : Cálculo de $\alpha()$ ao longo do eixo temporal para t=2,3,...,T, e todos os estados j, através da seguinte equação indutiva:

$$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) \cdot a_{ij}\right] \cdot b_j(O_t) \qquad (2.28)$$

Passo 3: A probabilidade final é dada por :

$$Pr(O/\lambda) = \sum_{i \in S_F} \alpha_T(i) \qquad (2.29)$$

No algoritmo acima, o passo 1 inicializa as probabiliades forward com a probabiliade inicial para todos os estados. A eq. (2.28) mostra que o estado j pode ser alcançado a partir de qualquer um dos estados i (i=1,2,...,N) no instante t-1. Observe que $\alpha_{t-1}(i)$ é a probabilidade conjunta de observar a seqüência O_1, O_2,...,O_{t-1}, e de que o último estado seja i: assim o produto $\alpha_{t-1}(i) \cdot a_{ij}$ é a probabilidade conjunta de observar O_1, O_2,...,O_{t-1}, e que o estado j seja alcançado no instante t a partir do estado i no instante t-1. Fazendo a somatória do produto $\alpha_{t-1}(i) \cdot a_{ij}$ para todos os possíveis estados i, temos

a probabiliade do estado j ser alcançado no instante t através de todas as observações anteriores. A multiplicação por $b_j(O_t)$ leva à probabilidade do estado j ser atingido no instante t e de observar a seqüência O_1, O_2,...,O_t. Lembremos que $b_j(O_t)$ é a probabilidade de observar O_t no estado j.

O passo 3 fornece a probabilidade $Pr(O/\lambda)$ que estávamos procurando, fazendo a somatória de todas as variáveis finais α_T sobre todos os possíveis estados finais. Isto porque $\alpha_T = Pr(O_1,O_2,...,O_T,\ s_T=i/\lambda)$ e as transições devem terminar num dos estados finais.

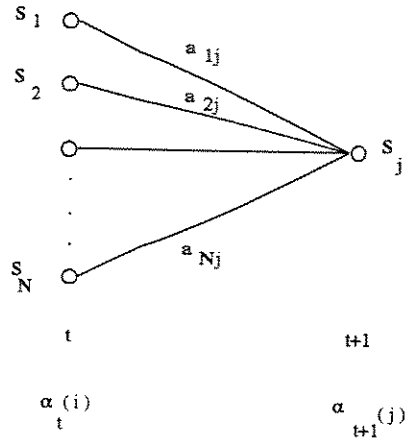


Figura 2.6. : Ilustração do algoritmo Fordward [5].

Pode-se mostrar [5] que, pelo algoritmo fordward, a probabilidade $Pr(O/\lambda)$ pode ser calculada com $N^2 \cdot T$ operações aritméticas, e não com $2 \cdot T \cdot N^T$ como seria o caso se a expressão (2.26) fosse utilizada.

De maneira semelhante, podemos definir a variável Backward $\beta_t(i)$ como:

$$\beta_t(i) = Pr(O_{t+1},\ O_{t+2},\ O_{t+3},...,O_T\ /\ s_t = i,\ \lambda)\ , \qquad (2.30)$$

isto é, a probabilidade de se observar a seqüência de O_{t+1} a O_T, dado que no instante t o estado era i, e dado o modelo λ. Esta variável pode ser determinada indutivamente pelo algoritmo Backward para todos os estados e instantes, de maneira análoga ao algoritmo Fordward. O algoritmo Backward é mostrado a seguir:

Algoritmo Backward

Passo 1 : $\beta_T(i) = 1/N_F$, para todos os estados $i \in S_F$, caso contrário $\beta_T(i)=0$;

Passo 2 : Cálculo das variáveis $\beta()$ ao longo do eixo temporal para t=T-1, T-2, T-3,...,1 , e todos os estados j, com a seguinte expressão:

$$\beta_t(j) = [\sum_{i=1}^{N} a_{ji} \cdot b_i(O_{t+1}) \cdot \beta_{t+1}(i)] \qquad (2.31)$$

Passo 3 : Cálculo da probabilidade final dada por:

$$Pr(O/\lambda) = \sum_{i \in S_I} \pi_i \cdot b_i(O_1) \cdot \beta_1(i) \qquad (2.32)$$

O passo 1 define arbitrariamente $\beta_T(i)$ como sendo $1/N_F$, para todos os estados finais. O passo 2 calcula a probabilidade de observar a seqüência O_{t+1} a O_T, dado o estado j no instante t. Repare que se faz a somatória sobre sobre todos os possíveis estados em t+1. A figura 2.7 mostra o algoritmo backward.

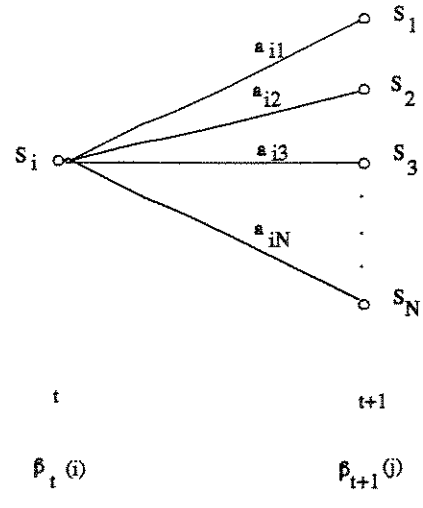


Figura 2.7. :Ilustração do algoritmo Backward [5].

Como pode ser observado, a probabilidade $Pr(O/\lambda)$ pode ser computada tanto pelo algoritmo Backward como pelo Forward, que oferecem uma complexidade computacional semelhante. O primeiro problema relacionado aos HMM, o da avaliação, fica então resolvido. Como mostraremos a seguir estes dois algoritmos têm também um papel importantíssimo na solução do segundo problema, o da estimação dos parâmetros dos modelos de Markov dadas as seqüências de observação.

2.3.3.2 Algoritmo de Viterbi

A informação oculta dos HMM, isto é, a seqüência de estados não pode ser recuperada, mas pode ser estimada pelo critério de probabilidade máxima. O algoritmo de Viterbi encontra a seqüência de estados que tem maior probabilidade de ter gerado a seqüência de estados observada, isto é, maximiza $Pr(O,S/\lambda)$. O algoritmo de Viterbi é muito semelhante ao DTW discutido na seção 2.2 deste capítulo no sentido de que este tenta encontrar a seqüência de alinhamento que minimize a distância global ao comparar duas seqüências de vetores de observação.

O algoritmo de Viterbi oferece uma alternativa a mais, além dos algoritmos Forward e Backward, para o problema de avaliação de $Pr(O/\lambda)$. A vantagem consiste em um menor esforço computacional, apresentando, contudo, uma robustez menor [3][5].

Repare-se que no algoritmo de Viterbi, mostrado a seguir, não se determina somente a probabilidade $Pr(O,S/\lambda)$ máxima, mas também a seqüência de estados correspondente. Com isto, o terceiro problema dos HMM, o da decodificação, fica superado. Na próxima seção trataremos do algoritmo de Baum-Welch que resolve o problema de reestimação dos parâmetros dos modelos de Markov em função das seqüências de observação.

Algoritmo de Viterbi

Passo 1: Inicialização. Para todos os estados i,

$$\delta_1(i) = \pi_i \cdot b_i(O_1)$$
$$\psi_1(i) = 0$$

Passo 2: Recursão. Do instante t=2 até T, para todos os estados j:

$$\delta_t(j) = \text{Max}\,[\delta_{t-1}(i)\cdot a_{ij}]\cdot b_j(O_t)$$
$$\psi_t(j) = \text{argmax}[\delta_{t-1}(i)\cdot a_{ij}]$$

Passo 3: Finalização. (O asterisco "*" indica os resultados ótimos.

$$P^* = \text{Max}\,[\delta_T(s)]$$
$$s \in S_F$$

$$s_T^* = \text{argmax}\,[\delta_T(s)]$$
$$s \in S_F$$

Passo 4 :Determinação do caminho ótimo, para atrás. De t = T-1 a t=1 :

$$s_t^* = \psi_{t+1}(s_{t+1}^*)$$

No passo 2 do algoritmo de Viterbi, a notação $\text{argmax}[\delta_{t-1}(i)\cdot a_{ij}]$ indica o estado i que maximiza $\delta_{t-1}(i)\cdot a_{ij}$.

2.3.3.3 *Algoritmo de Baum-Welch para Reestimação dos Parâmetros dos HMM.*

O problema mais difícil nos HMM diz respeito a como ajustar os parâmetros do modelo $(A,B.\pi)$ de maneira a maximizar a probabilidade $P(O/\lambda)$. Não existe uma solução analítica para este problema, mas se dispõe de um algoritmo iterativo, conhecido como de Baum-Welch, baseado na técnica de maximização EM (maximização da esperança da verossimilhança). O algoritmo de Baum-Welch também pode ser visto como um caso particular da técnica do gradiente.

Apresentaremos a dedução das equações de re-estimação segundo a referência [3], que segue a linha da publicação original [8] e que, embora um pouco mais sofisticada, é muito útil para estudos mais avançados em HMM. Antes porém faremos algumas definições a serem utilizadas na própria dedução.

Considere um HMM qualquer com parâmetros $\lambda=(A,B,\pi)$ todos diferentes de zero. A probabilidade de transição $\gamma(i,j)$ do estado i para o estado j, depois de dada seqüência de observação, é dada por:

$$\gamma_t(i,j) = Pr(s_t=i,\ s_{t+1}=j \ / \ O,\lambda) \tag{2.33}$$

A expressão (2.33) pode ser colocada em função das variáveis Backward e Forward como:

$$\gamma_t(i,j) = \frac{\alpha_t(i)\cdot a_{ij}\, b_j(O_{t+1})\cdot \beta_{t+1}(j)}{Pr(O/\lambda)} =$$

$$= \frac{\alpha_t(i)\cdot a_{ij}\cdot b_j(O_{t+1})\cdot \beta_{t+1}(j)}{\sum_{k\in S_F} \alpha_T} \tag{2.34}$$

Como ilustrado na figura 2.8, $\gamma_t(i,j)$ é a probabilidade de um caminho passar pelo estado i no instante t e fazer uma transição para o estado j no instante t+1, dada a seqüência de observações e o modelo.

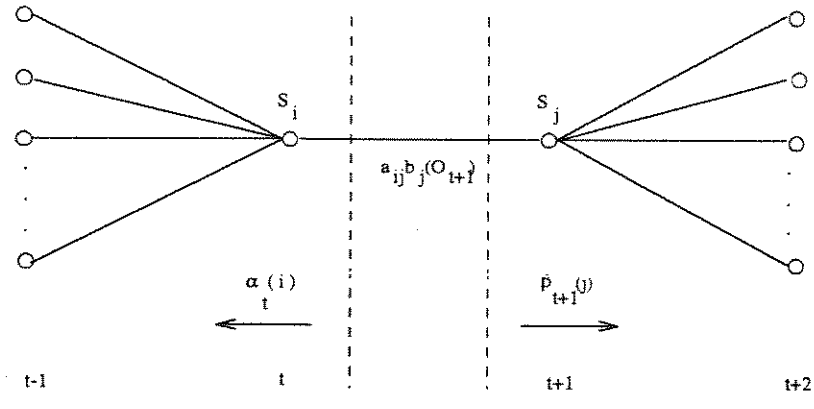


Figura 2.8. : Ilustração do cálculo das variáveis gama.

De maneira semelhante definimos a probabilidade a posteriori de estar no estado i no instante t, $\gamma_t(i)$, dada a seqüência de observações e o modelo:

$$\gamma_t(i) = Pr(s_t=i \ / \ O,\lambda) =$$

$$= \frac{\alpha_t(i) \cdot \beta_t(i)}{\sum_{k \in S_F} \alpha_T} \qquad (2.35)$$

sendo que $\gamma_t(i)$ também pode ser determinada a partir $\gamma_t(i,j)$:

$$\gamma_t(i) = \sum_{j=1}^{N} \gamma_t(i,j) \quad , \text{ para } \quad t < T \qquad (2.36)$$

A expressão (2.36) tem a vantagem em relação à (2.35) de exigir somente somas.

A prova do algoritmo de Baum-Welch apresentada em [8] previa um número finito de estados e uma distribuição de probabilidade de saída genérica para cada estado [3]. Seja então λ os parâmetros de um HMM, e $\bar{\lambda}$ os parâmetros do mesmo HMM após uma reestimação. Define-se $Q(\lambda, \bar{\lambda})$ como :

$$Q(\lambda,\bar{\lambda}) = \frac{1}{Pr(O/\lambda)} \cdot \sum_{\substack{\text{todos} \\ \text{os estados}}} Pr(O,S/\lambda) \cdot \log[Pr(O,S/\bar{\lambda})] \qquad (2.37)$$

Aqui $Q(\lambda,\bar{\lambda})$ é uma função de $\bar{\lambda}$, pois λ são os parâmetros atuais do HMM, antes da seguinte re-estimação de parâmetros.

Pode-se provar [3][8][11], pela concavidade da função logaritmo, que :

$$\log[\frac{Pr(O/\bar{\lambda})}{Pr(O/\lambda)}] \geq Q(\lambda,\bar{\lambda}) - Q(\lambda,\lambda) \qquad (2.38)$$

A partir da expressão (2.38) fica evidente que o novo modelo reestimado $\bar{\lambda}$ maximiza $Pr(O/\bar{\lambda})$ se e somente se maximiza também a função $Q(\lambda,\bar{\lambda})$.

Seja $\bar{\lambda}=(\bar{A}, \bar{B}, \bar{\pi})$ e utilizando as expressões (2.22), (2.24)e (2.25), aplicadas a uma seqüência de estados $S=s_1, s_2,...,s_T$, temos :

$$\log[Pr(O,S/\bar{\lambda})] = \log[\pi_1] + \sum_{t=1}^{T-1} \log[\bar{a}_{s_t s_{t+1}}] + \sum_{t=1}^{T} \log[\bar{b}_{s_t}(O_t)] \qquad (2.39)$$

Substituindo a expressão (2.39) na expressão (2.37) e reagrupando os termos correspondentes às probabilidades das transições e às probabilidades de observação de símbolos, chegamos a:

$$Q(\lambda,\bar{\lambda}) = \sum_i \sum_j c_{ij} \cdot \log[\bar{a}_{ij}] + \sum_j \sum_{k=1}^{L} d_{jk} \cdot \log[\bar{b}_j(k)] + \sum_i e_i \cdot \log[\bar{\pi}_i] \qquad (2.40)$$

onde :

$$c_{ij} = \sum_{t=1}^{T-1} \gamma_t(i,j) \qquad (2.41)$$

$$d_{jk} = \sum_{t \in O_t = v_k} \gamma_t(j) \qquad (2.42)$$

$$e_i = \gamma_1(i) \qquad (2.43)$$

De acordo com o apêndice B, $Q(\lambda,\bar{\lambda})$ pode ser maximizada se e somente se:

$$\bar{a}_{ij} = \frac{c_{ij}}{\sum_j c_{ij}} = \frac{\sum_{t=1}^{T-1} \gamma_t(i,j)}{\sum_{t=1}^{T-1} \sum_j \gamma_t(i,j)} \qquad (2.44)$$

$$\bar{b}_j(k) = \frac{d_{jk}}{\sum_k d_{jk}} = \frac{\sum_{t \in O_t = v_k} \gamma_t(j)}{\sum_j \gamma_t(j)} \qquad (2.45)$$

$$\bar{\pi}_i = \frac{e_i}{\sum_i e_i} = \gamma_1(i) \qquad (2.46)$$

As expressões (2.44), (2.45) e (2.46) são conhecidas como as equações de reestimação de Baum-Welch, para o caso em que a distribuição de probabilidade de saída é discreta. Para o caso de se utilizar funções de densidade de probabilidade contínuas (HMM contínuos) , as correspondentes equações de reestimação podem ser deduzidas a partir da mesma função $Q(\bar{\lambda},\lambda)$ da expressão (2.37)[3]. No caso particular desta tese, trabalharemos com os HMM discretos para o qual foram deduzidas as equações de atualização de parâmetros.

2.4. APLICAÇÃO DOS HMM A RECONHECIMENTO DE PALAVRAS FALADAS

As cadeias de Markov têm um espectro de atuação bastante amplo no problema de reconhecimento de fala. A linguagem natural, seja escrita ou oral, tem uma redundância muito grande de informação, sendo possível prever elementos (letras, fonemas, palavras, etc) em função dos elementos anteriores [7]. Em outras palavras, o modelamento por processos Markovianos se aplica bastante bem, de maneira geral, à linguagem.

Nesta dissertação, abordaremos o problema do reconhecimento automático de palavras isoladas utilizando as cadeias (ocultas) de Markov para o modelamento de seqüências de quadros. Assim, cada palavra, pronunciada e digitalizada, do nosso vocabulário (dígitos de 0 a 9) é dividida em intervalos de tempo de igual duração, extraindo dentro de cada intervalo (quadro) parâmetros freqüenciais (quadro) do sinal de voz. Utilizando as definições da seção 2.3.2., cada quadro também pode ser chamado de vetor observação. Temos assim para cada palavra uma seqüência de observação O_1, O_2,...,O_T. Onde cada O_t corresponde a um quadro com parâmetros de um intervalo do sinal de voz. A duração da palavra, em número de quadros, é T. Após a quantização vetorial, classificamos cada O_t numa das categorias de $V = V_1, V_2,..., V_k$.

Uma parte da base de dados é utilizada para treinar os modelos e a outra para reconhecer. A cada palavra do vocabulário atribuímos um HMM e separamos as elocuções de treinamento em função da palavra a que pertencem. Uma vez que as elocuções sejam parametrizadas e quantificadas, procedemos ao treinamento. Escolhemos para cada HMM um jogo de parâmetros inicial e a partir daí utilizamos as expressões (2.43), (2.44) e (2.45) para reestimar os parâmetros dos HMM,utilizando a correspondente base de dados (seqüências de observação ou quadros) de

treinamento. A atualização dos parâmetros se executa até que o aumento de $Pr(O/\lambda)$ seja menor que um certo limiar.

Uma vez treinados os HMM (um para cada palavra do vocabulário), podemos executar o reconhecimento. Este se faz calculando para cada elocução de teste, parametrizada e quantificada, a probabilidade $Pr(O/\lambda_i)$, onde λ_i corresponde aos parâmetros do HMM do i-ésima palavra do vocabulário. A palavra correspondente ao HMM que der maior $Pr(O/\lambda)$ será a reconhecida. Esta probabilidade pode ser calculada pelos algoritmo Forward, Backward ou de Viterbi (escolhendo a seqüência de maior verossimilhança).

Com esta técnica, um HMM por palavra, podemos lidar com um vocabulário de até 200 ou 300 palavras. Para vocabulários de tamanho médio, é necessário aplicar os HMM a nível de fonemas ou trifones [10].

2.5. CONCLUSÃO

O problema de reconhecimento de padrões acústicos pode ser abordado por dois tipos de análises : determinística ou estocástica. Um exemplo de análise determinística seria o DTW que compara diretamente duas elocuções de palavras pronunciadas e parametrizadas, eliminando as diferenças de duração dos períodos estacionários. Como exemplo de análise estocástica estão os HMM que conseguem assimilar informações de muitas elocuções nos parâmetros de um só modelo. Dada a grande variabilidade do sinal de voz e às limitações das parametrizações utilizadas hoje em dia, os HMM têm se tornado a técnica mais popular em reconhecimento de voz, tanto para palavra isolada como contínua.

Este capítulo teve por objetivo explicar a teoria e os fundamentos nos quais se baseiam os algoritmos DTW e HMM implementados nos capítulos posteriores. O primeiro foi utilizado para realizar um estudo comparativo de parametrizações espectrais, sendo que o segundo foi empregado para implementar um reconhecedor de dígitos independente do locutor.

2.6. REFERÊNCIAS

[1] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition", IEEE Trans. on ASSP, vol. ASSP-26, No. 1, Feb. 1978, pp. 43-49.

[2] Myers,C. Rabiner,L.R. Rosenberg,A.E. : "Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition", IEEE Trans. ASSP, no. 6, Dec. 1980.

[3] Huang,X.D. Ariki,Y. Jack,M.A. : "Hidden Markov Models for Speech recognition", Edinburgh University Press, 1990.

[4] Pierre,D.A. : "Optimization Theory with Applications". Dover Editions, 1986.

[5] Rabiner,L.R. : "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". Proceedings of the IEEE, vol.77, No.2, February, 1989.

[6] Shannon,C.E. :"A Mathematical Theory of Communications". Bell System Technical Journal. Vol.27, pp.379–423, 623-656, 1948.

[7] Shannon, C.E. : "Prediction and Entropy of Printed English". Bell System Technical Journal, vol.30,pp.50-64, 1951.

[8] Baum,L.E . Petrie,T . Soules,G. Weiss,N. : "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains". The Annals of Mathematical Statistics, 1970, vol.41, No.1, pp.164-171.

[9] Jelinek,F. :"Continuous Speech Recognition by Statistical Methods", Proc. IEEE, vol.64, pp.532-536, april, 1976.

[10] Lee, Kai-Fu : "Large Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System". Phd thesis, Department of Computer Science, Carnegie Mellon University, 1988.

[11] Dempster,A.P. Laird,N.M. Rubin,D.B. :"Maximum - likelihood from incomplete data via the EM algorithm". J. Royal Statistic Soc. Serie B, vol.39, pp.1-38, 1977.

CAPÍTULO 3

IMPLEMENTAÇÃO DO ALGORITMO DTW PARA RECONHECIMENTO DEPENDENTE DO LOCUTOR

3.1. INTRODUÇÃO

Como foi mencionado no capítulo anterior, o DTW ("Dynamic Time Warping") compara diretamente duas elocuções, após a parametrização, tentando eliminar as diferenças na velocidade de pronunciação de uma mesma palavra pronunciada pelo mesmo locutor. Isto é equivalente a eliminar as diferenças de duração dos períodos estacionários do sinal de voz.

Paralelamente, ja foi discutido que o sinal da fala apresenta uma grande variabilidade no que diz respeito à informação espectral e temporal, tanto intra-locutor quanto inter-locutor. Se o DTW elimina as diferenças temporais na pronúncia das palavras comparando diretamente duas elocuções, resulta evidente que esta técnica é muito sensível à parametrização utilizada, e se esta não for boa, a mínima distância resultante entre duas elocuções de uma mesma palavra feitas pelo mesmo locutor será grande e propiciará as condições de confusão no reconhecimento. Esta forte dependência do DTW em relação aos parâmetros utilizados é que dificulta sua utilização em reconhecimento independente do locutor, em sua versão original, mas será a principal razão de utilizá-lo no próximo capítulo como um medidor da qualidade de parametrizações.

Cabe mencionar que, além de sua importância no capítulo seguinte como ferramenta de teste de parâmetros espectrais, o DTW pode ser visto como uma forma do algoritmo de Viterbi de decodificação, que é empregado nos HMM implementados no capítulo 5, no algoritmo "Level Building" para reconhecimento de palavra contínua [4], e em conjunto com redes neurais [9].

Serão apresentados e comentados neste capítulo os detalhes de implementação do DTW clássico com a técnica Mel-cepstral.

3.2. ALINHAMENTO NÃO LINEAR NO TEMPO

Para eliminar as diferenças de duração nas elocuções das palavras, recorre-se ao alinhamente não-linear (seção 2.2.) para comprimir ou dilatar alguns trechos dos sinais de modo a minimizar a distância global (equação (2.1)) das duas sequências (a de teste e a de referência). Esta distância mínima é obtida ao longo do caminho ótimo de comparação. A título de exemplo, a figura 3.1. ilustra a comparação de duas elocuções para o caso unidimensional (cada "frame" dá origem a um só coeficiente paramétrico).

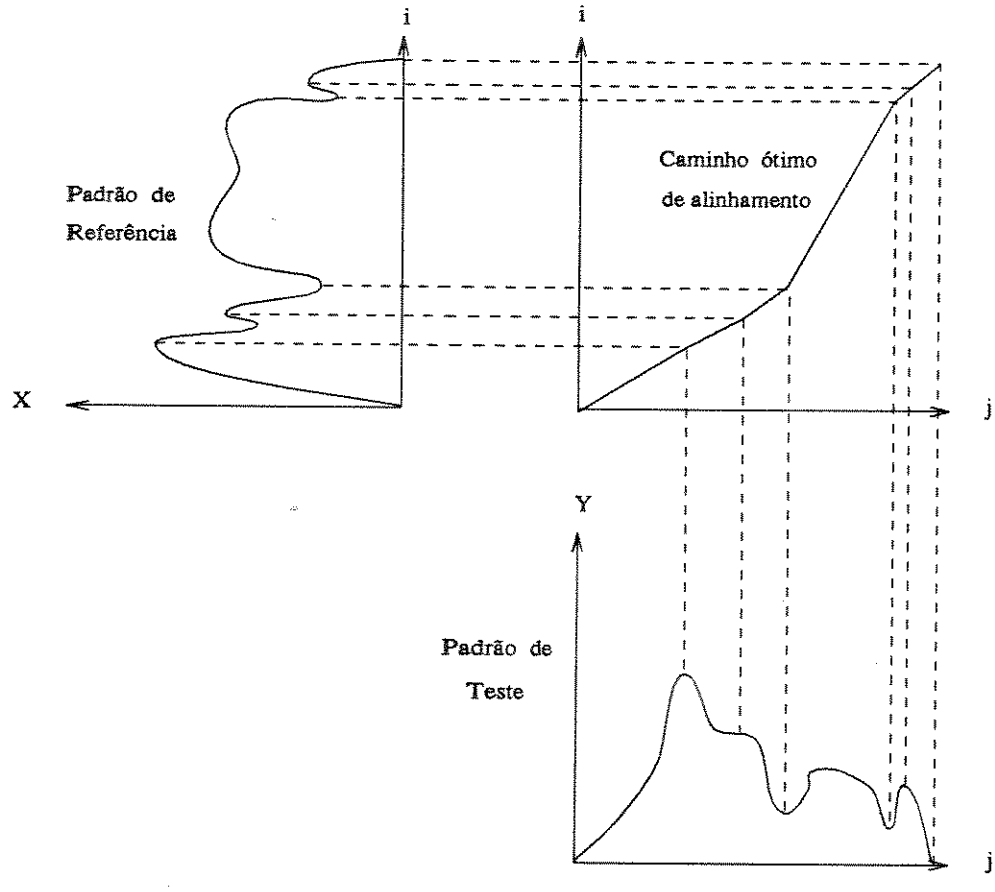
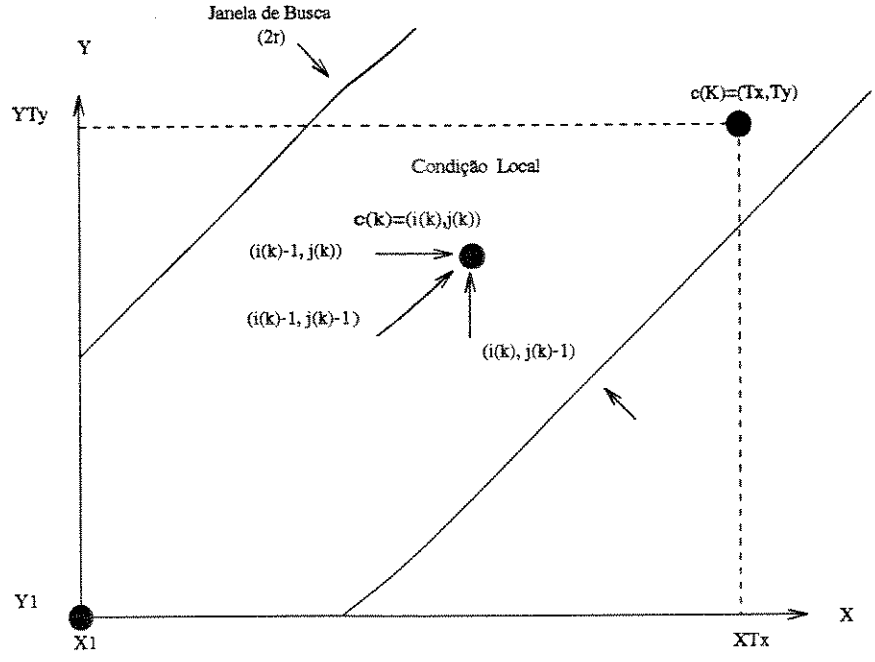


Figura 3.1.: Alinhamento temporal entre a elocução referência e a elocução a reconhecer [1].

As elocuções de referência e de teste parametrizadas, sequências X e Y respectivemente, são posicionadas (veja figura 3.2) de maneira que os primei-

ros quadros de cada sequência se localizem no canto inferior esquerdo do gráfico. Os vetores seguintes de X e Y são posicionados segundo a direção do eixo x e y, respectivamente. A inclinação do caminho de alinhamento é uma medida da compressão de X ao ser comparada com Y. Por exemplo, um passo vertical significa que dois quadros de Y são emparelhados com um mesmo quadro de X, e um passo horizontal corresponde a dois quadros de X comparados com um mesmo quadro de Y.

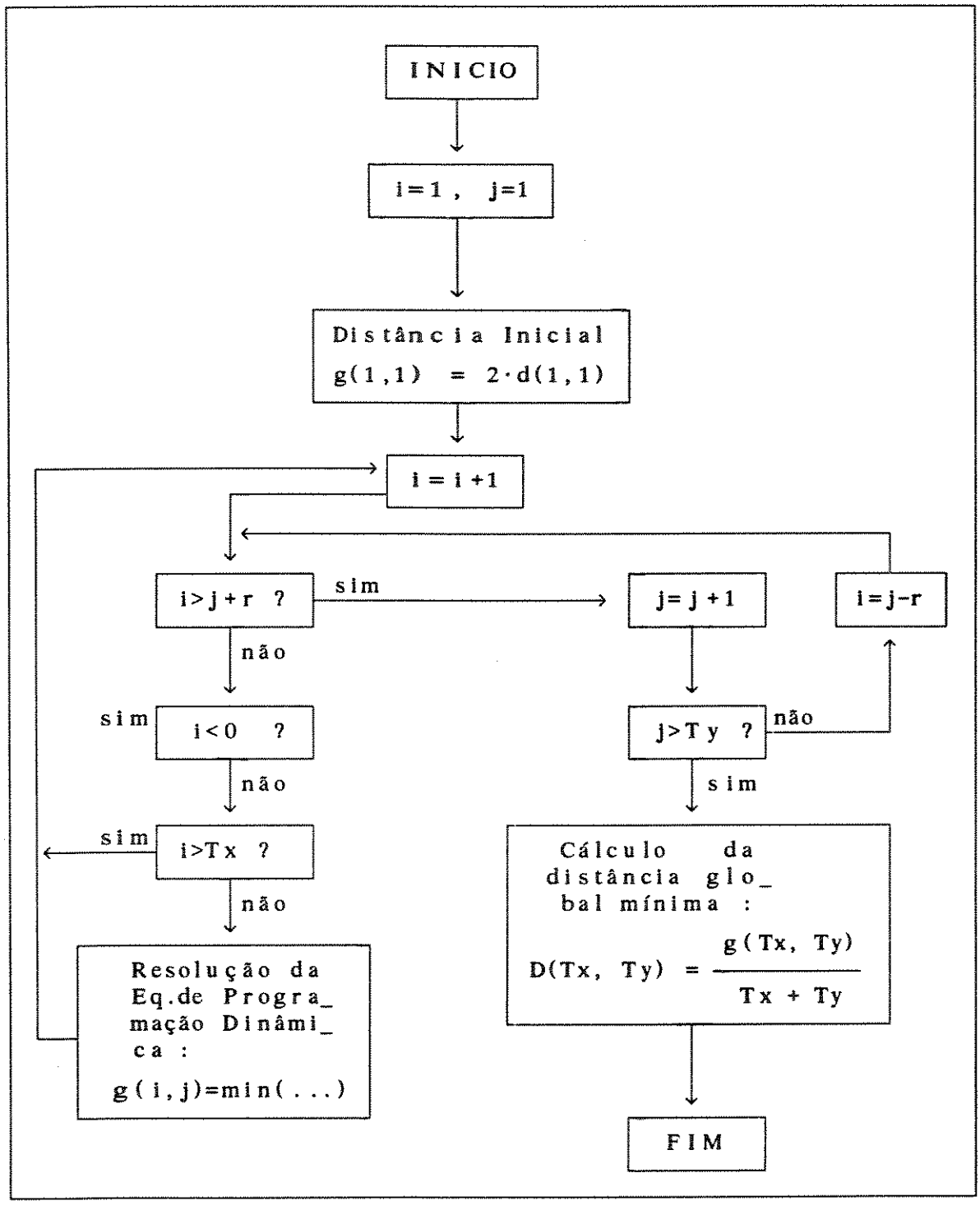


Fígura 3.2.: Definição do caminho de alinhamento, da janela de busca e da condição local.

Seja i o índice da sequência X, e j o índice da sequência Y, os pares c(k)=(i(k), j(k)) indicam os elementos de X e Y que são emparelhados no alinhamento. Assim, de acordo com as restrições da seção 2.2. do capítulo anterior, supõe-se que o início e o fim das palavras estão corretamente determinados, e se impõe que coincidam no caminho de alinhamento. Isto é, o caminho de alinhamento começa em (i(1), j(1)) e termina em (i(Tx), j(Ty)), onde Tx e Ty são, respectivamente, as durações em número de frames dos padrões X e Y. Também impõe-se que o caminho ótimo deve estar dentro da janela de busca de largura 2r (figura 3.2.), onde r é função da máxima diferença de duração permitida entre elocuções da mesma palavra. Esta imposição é feita para limitar o esforço computacional e a memória exigidos.

O algoritmo implementado neste trabalho é o apresentado por Sakoe-Chiba [2] e mostrado a seguir. Ele corresponde ao alinhamento ilustrado na figura 3.2.

Algoritmo DTW (segundo [2])



O algoritmo se inicia pelo cálculo da distância acumulada g(i,j) para i=1 e j=1, isto é, a distância entre os primeiros frames das duas sequências : g(1,1) = 2·d(1,1). A partir daí, calcula-se a distância acumulada para cada par

(i,j), dentro da janela de busca (condição global), por meio da equação 2.18, que determina o caminho "mais curto" para chegar ao par (i,j) a partir de um conjunto finito de estados anteriores (condição local). Esta equação de programação dinâmica leva em conta, para cada par (i,j), somente as distâncias acumuladas nos possíveis estados anteriores a (i, j) e não como se chegou a estes. Este procedimento se repete até chegar em i=Tx ou j=Ty onde é determinada a distância global final entre as duas elocuções. Deve-se observar a grande semelhança entre o DTW e o algoritmo de Viterbi de decodificação que, ao invés de minimizar a distância acumulada, lida com a maximização da verossimilhança.

Dentre as possíveis condições locais, foi utilizada a condição local mostrada na figura 3.3., que exige menos cálculos que as demais apresentadas em [2] e fornece bons resultados.

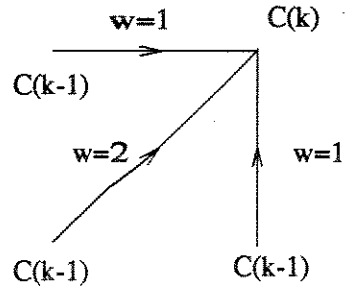


Figura 3.3. :Condição local empregada no algoritmo DTW [2].

3.3. IMPLEMENTAÇÃO DA PARAMETRIZAÇÃO MEL-CEPSTRUM

Os testes de reconhecimento dependentes do locutor apresentados neste capítulo foram obtidos com o DTW em conjunto com a parametrização Mel-cepstrum (seção 1.2.2.).

Para obter os 10 coeficientes Mel-cepstrum por quadro foi utilizada uma janela de Hamming de 128 pontos. O deslocamento entre duas janelas consecutivas

foi fixado em 100 pontos. Dado que a frequência de amostragem era de 8 Khz, a largura e o deslocamento da janela foram, respectivamente, 16.0 e 12.5 ms.

Foram empregados 15 filtros triangulares que, ao contrário de [5], foram situados diretamente na escala mel e não na escala linear de frequência. Assim, a largura dos filtros foi determinada de modo a manter constante o ganho na banda de interesse, fixada entre 2.0 e 17.0 barks, o que corresponde aproximadamente a 200 e 3700 hz respectivamente. A energia dentro de cada filtro foi determinada por meio do módulo da FFT, calculada com 256 pontos (128 pontos do sinal janelado + 128 zeros). A FFT com 128 pontos apresentou taxas de reconhecimento pouco satisfatórias devido à amostragem no domínio da freqüência ser muito espaçada para a determinação da energia dentro dos filtros abaixo dos 1000 hz. Assim, para diminuir a distância entre os pontos da FFT e não aumentar a largura da janela, dado que isto reduz a resolução temporal na descrição dos fonemas oclusivos, foram adicionados 128 zeros aos 128 pontos do sinal em questão.

De posse das energias em dB dentro dos filtros, estas foram normalizadas em relação à máxima energia no quadro com uma variação de 50 dB. Isto é, fixando a maior energia em 0dB, a menor não pode ser inferior a -50dB. Esta normalização das energias de um mesmo quadro tenta modelar o CAG (controle automático de ganho) do sistema auditivo periférico [7] e o fato dele responder melhor aos picos que aos vales espectrais [8].

Uma vez determinadas as energias em dB normalizadas de cada banda, os 10 coeficientes Mel-cepstrum foram obtidos com a equação 1.25.

As distâncias entre quadros de coeficientes mel-cepstrum foram calculadas através da distância euclideana. Esta foi preferida pela sua simplicidade, embora outra opção interessante fosse dividir cada ítem da distância euclideana pela variância do respectivo coeficiente.

3.4. BASE DE DADOS E EXPERIMENTOS

O vocabulário utilizado para testar os algoritmos de treinamento e reconhecimento, neste capítulo e nos seguintes, foi constituído pelos dígitos de 0 a 9 do português. Cada palavra (dígito) foi pronunciada 4 vezes por 8 locutores (4 homens e 4 mulheres) resultando em $10 \cdot 4 \cdot 8 = 320$ elocuções ao todo. Esta base de

dados foi gravada no Laboratório de Processamento de Voz, do DECOM, na Faculdade de Engengaria Elétrica da UNICAMP, pelo prof. Dr. Fábio Violaro e o doutorando Fernando Runstein.

A freq. de amostragem foi 8 Khz e cada amostra codificada linearmente com 12 bits. Cada locutor pronunciou 4 sequências de dígitos de 0 a 9. Nas duas primeiras, a pronúncia foi em ordem natural de contagem progressiva e nas duas seguintes a ordem foi: "três", "oito", "cinco", "dois", "zero", "nove", "um", "sete", "quatro" e "seis". Cada seqüência de dígitos foi posteriormente dividida nas 10 elocuções que as compunham, sendo que o início e fim de cada elocução foi determinada visualmente eliminando assim o silêncio entre palavras. Optou-se por realizar este tipo de segmentação manual, pois os algoritmos de segmentação palavra/silêncio introduzem um erro na delimitação do início e fim dos sinais da fala e o intuito deste trabalho era mais fazer um estudo de técnicas do que apresentar uma aplicação totalmente acabada.

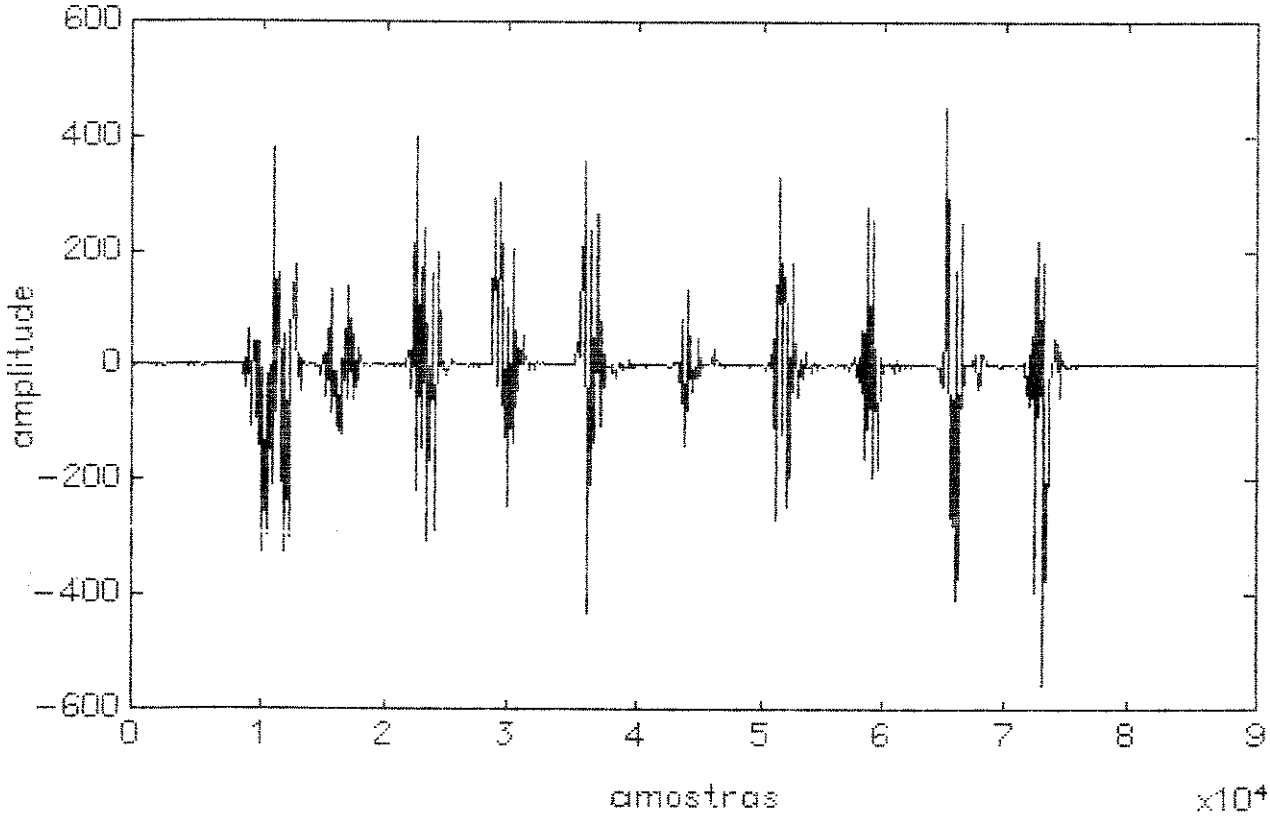


Figura 3.4.: Sequências de dígitos de 0 a 9 em ordem de contagem progressiva.

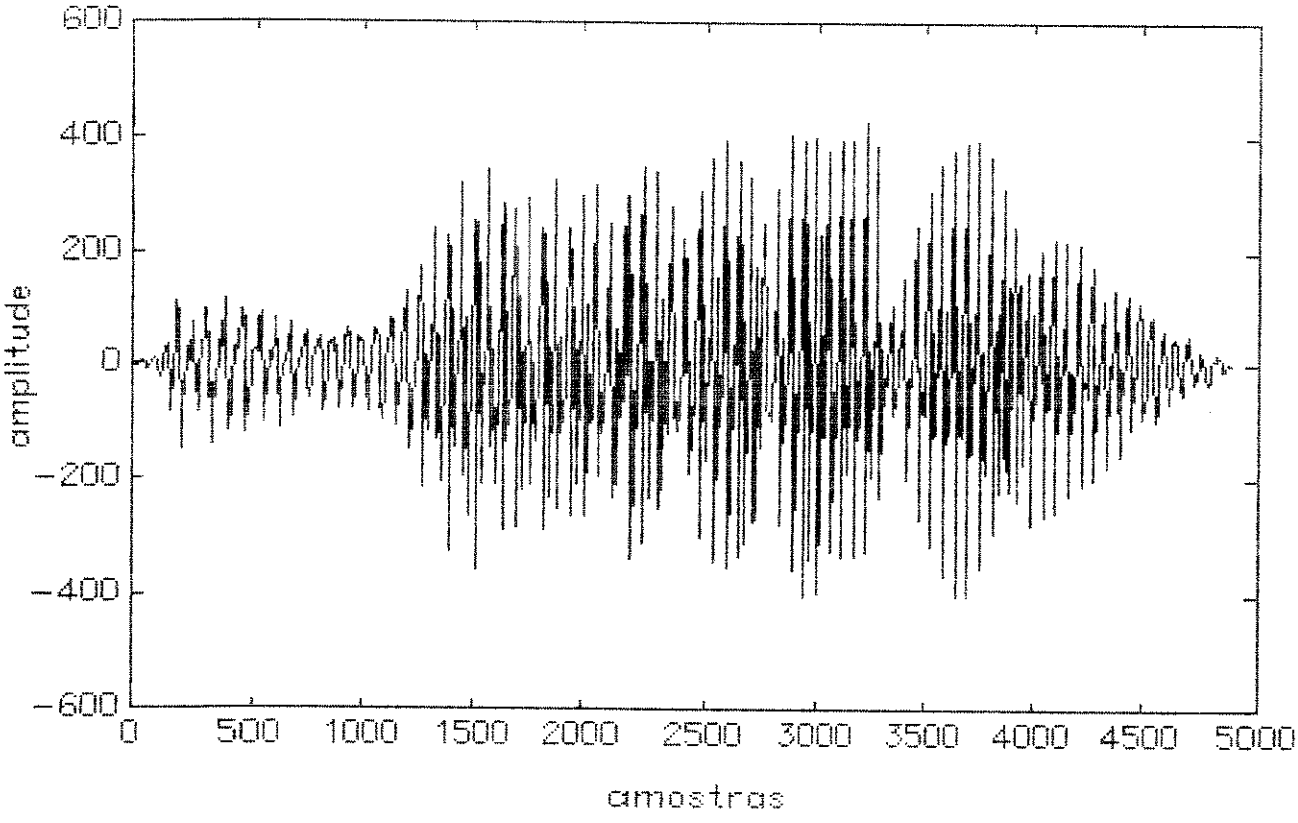


Figura 3.5.: Elocução da palavra "zero" isolada da sequência da figura 3.5.

Tendo 4 elocuções por palavra por locutor, os experimentos dependentes do locutor foram realizados escolhendo uma das quatro elocuções como referência e usando as outras três para reconhecer. Este procedimento foi repetido 4 vezes fazendo um rodízio nas 4 elocuções na hora de escolher a de referência. Seja :

eloc(l, n, i) ≡ a elocução i, da palavra n, do locutor l, onde $1 \le l \le 8$, $0 \le n \le 9$ e $1 \le i \le 4$;

ref(l, eloc_ref) ≡ o conjunto das elocuções de referência, para um dado locutor, formado por eloc(l, n, i), onde $0 \le n \le 9$ e i= ref. Deduz-se que o conjunto ref(l, eloc_ref)tem 10 elementos;

test(l) ≡ o conjunto da elocuções eloc(l,n,i) a reconhecer, para o mesmo locutor do conjunto ref(l, eloc_ref), onde $1 \leq i \leq 4$ e $i \neq eloc\_ref$, e $0 \leq n \leq 9$;

Os testes de reconhecimento dependente do locutor foram executados então medindo a distância de cada elocução eloc(l,nt,it) de test(l) com cada uma das elocuções de referência eloc(l, nr, ir), ir=eloc_ref e $0 \leq n \leq 9$, de ref(l, eloc_ref), por meio do algoritmo DTW. A elocução eloc(l,nt, it) era corretamente reconhecida se :

menor dist[eloc(l,nr,ir), eloc(l,nt,it)] se nr=nt

Para reconhecer uma elocução é necessário então executar o DTW 10 vezes: uma para cada padrão de referência.

Como a cada locutor correspondiam 40 elocuções, sendo que 10 eram utilizadas para formar o conjunto ref(l, eloc_ref), para cada eloc_ref correspondiam então 30 experimentos de reconhecimento. Fazendo, sucessivamente, eloc_ref = 1, 2, 3 e 4, temos então $4 \cdot 30 = 120$ experimentos de reconhecimento por locutor e $120 \cdot 8$ locutores = 960 experimentos de reconhecimento ao todo.

3.5. RESULTADOS

Como ja foi dito, o DTW foi desenvolvido para eliminar as diferenças de duração nos períodos estacionários do sinal de voz. A janela de busca, mostrada na figura 3.2., corresponde à máxima diferença na duração das elocuções a ser comparada. Este limite é estabelecido para reduzir o tempo de processamento mas introduz um erro ao supor que o caminho ótimo de comparação está dentro da mencionada janela.

A tabela 3.1. apresenta o erro de reconhecimento para um locutor masculino em função da largura da janela de busca ($2 \cdot r$). Quanto maior r, maior a probabilidade do caminho ótimo estar situado dentro da área permitida, porém maior é o tempo de cômputo das distâncias entre os padrões.

Tabela 3.1.: Erro de reconhecimento para um locutor da base de dados em função da largura da janela de busca (2·r). Foram empregados 10 coeficientes mel cepstrum, janela de Hamming de 128 pontos, e FFT também de 128 pontos.

r num.de quadros	Erro de Rec. em %
0	12.5
1	13.0
2	10.8
3	9.2
4	7.5
5	1.7
6	1.7
7	1.7
8	1.7
9	0.8
10	0.0
11	0.0
12	0.0

Fixando r em 12 quadros foi calculada a porcentagem de erro para cada uns dos 8 locutores em função da largura da janela de Hamming e do número de pontos da FFT. Os resultados destas simulações são apresentados na tabela 3.2. Para cada locutor corresponderam 120 experimentos de reconhecimento, resultando ao todo em 120·8=969 experimentos. Como pode ser observado, a menor porcentagem de erro foi obtida com janela de Hamming e FFT de, respectivamente, 128 e 256 pontos.

Tabela 3.2.: Porcentagem de erro média com 10 coeficientes mel-cepstrum por quadro e r=12, em função da largura da janela de Hamming e do número de pontos da FFT.

	Jan. de Hamming: 128 pontos FFT : 128 pontos	Jan. de Hamming 128 pontos FFT : 256 pontos	Jan. de Hamming: 256 pontos FFT : 256 pontos
Erro de Rec. em %	0.94	0.52	0.94

3.6. CONCLUSÕES

As análises comparativas deste capítulo resumem-se à escolha da janela de busca (2·r) do algoritmo DTW, utilizado no capítulo seguinte para estudo de parametrizações, da largura da janela de Hamming e do número de pontos da FFT na parametrização Mel-cepstrum.

A janela de busca do algoritmo DTW foi fixada em r = 12, pois este valor mostrou ser um bom compromisso no que diz respeito ao erro de reconhecimento e ao tempo de processamento. O tempo de processamento por experimento de reconhecimento, que aumenta com r, foi um fator importante de decisão dadas as quantidades de simulações executadas no capítulo seguinte e a infraestrutura disponível (PC AT de 16 MHz com co-processador numérico). A janela de busca é função da máxima diferença de duração entre as elocuções de uma mesma palavra pronunciadas pelo mesmo locutor e, portanto, o aumento de r poderia diminuir ainda mais a menor taxa de erro (tabela 3.2.) obtida com DTW e Mel-cepstrum. Contudo, o valor de 0.52% foi considerado bastante razoável quando comparado com os erros de reconhecimento apresentados em [2] e [3] (0.2 a 0.6 %, utilizando uma banda de freqüência maior e somente locutores masculinos). Como o intuito deste trabalho era utilizar o DTW como ferramenta de comparação de parametrizações espectrais, não houve então motivo para procurar um r ainda maior.

Por último, pode ser observado nas tabela 3.2. que um compromisso razoável entre a descrição da dinâmica temporal do sinal e a transformada de Fourier a curto prazo é obtido com uma janela de Hamming e FFT de 128 e 256 pontos, respectivamente. Para este caso a probabilidade de erro média foi de 0.52% contra 0.94% para as duas outras combinações.

3.7. REFERÊNCIAS

[1] Poza Lara,M.J. Villarubia,L. Siles,J.A. : "Teoría y Aplicaciones del Reconocimiento Automático del Habla". Comunicaciones de Telefónica I+D, Vol. 2, No. 3, janeiro-junho, 1991.

[2] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition", IEEE Trans. on ASSP, vol. ASSP-26, No. 1, Feb. 1978, pp. 43-49.

[3] Myers,C. Rabiner,L.R. Rosenberg,A.E. : "Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition", IEEE Trans. ASSP, no. 6, Dec. 1980.

[4] Rabiner,L.R. : "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". Proceedings of the IEEE, vol.77, No.2, February, 1989.

[5] Davis,S.B. and Mermelstein,P. : "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences", IEEE Trans. ASSP, vol.28, pp.357-366, 1980

[6] Scharf,B. : "Critical Bands". Department of Psychology, Northeastern University, Boston, Massachusetts. Foundations of Modern Auditory Theory. Edited by Jerry V. Tobias, Academic Press, 1970.

[7] Seneff,S. :"A computational Model for the Peripheral Auditory System: Application to Speech Recognition Research". ICASSP-86, pp. 1983-1986.

[8] Sánchez,J. Becerra,N. : "Hacia la Forma Espectral Relevante en el Timbre de los Alófonos Sonoros del Español (Castellano), Symposium URSI, Cáceres, Spain, September 1991.

[9] Haffner,P. Franzini,M. Waibel,A. : "Integrating Time Alignment and Neural Networks for High Performance Continuous Speech Recognition". ICASSP 91, S2.17, vol1, pp. 105-108.

CAPÍTULO 4

ESTUDO COMPARATIVO DE PARAMETRIZAÇÕES ESPECTRAIS

4.1. INTRODUÇÃO

Após o janelamento, o sinal de voz é dividido em segmentos superpostos dentro dos quais são calculados uma série de coeficientes, que se espera estejam fortemente correlacionados com o significado fonético do intervalo em questão. Este procedimento, denominado parametrização, é de vital importância para o correto funcionamento dos sistemas de reconhecimento, mesmo se tratando daqueles que funcionam com técnicas estocásticas de modelamento (HMM e Redes Neurais).

Estudos em psico-acústica [1][2][3] mostraram que a densidade espectral de potência do sinal acústico da fala, mais do que a forma de onda, está muito relacionada com a identidade fonética, em especial no que se refere às vogais e outros fonemas estacionários. Isto leva a preferir as análises realizadas no domínio frequêncial às realizadas no domínio temporal. Mas, a grande variabilidade existente na pronúncia de um mesmo fonema em contextos diferentes pelo mesmo locutor, e mais ainda por locutores diferentes, torna muito difícil estabelecer uma relação direta entre as características físicas do sinal e seu significado lingüístico. Na figura 4.1 são mostrados os fonemas vocálicos da língua inglesa em função dos dois primeiros formantes. Observe-se a região ampla correspondente a cada elemento e a intersecção entre elas.

Por outro lado, além de ser sensível à densidade espectral de potência do sinal de voz, o sistema auditivo periférico apresenta certas sutilezas na manipulação da informação acústica que não são triviais de modelar. Estas peculiaridades, tais como não-linearidades e mascaramento ("masking")[4][5], tornam a percepção auditiva muito robusta a variações externas, como características de locutor e ruídos interferentes.

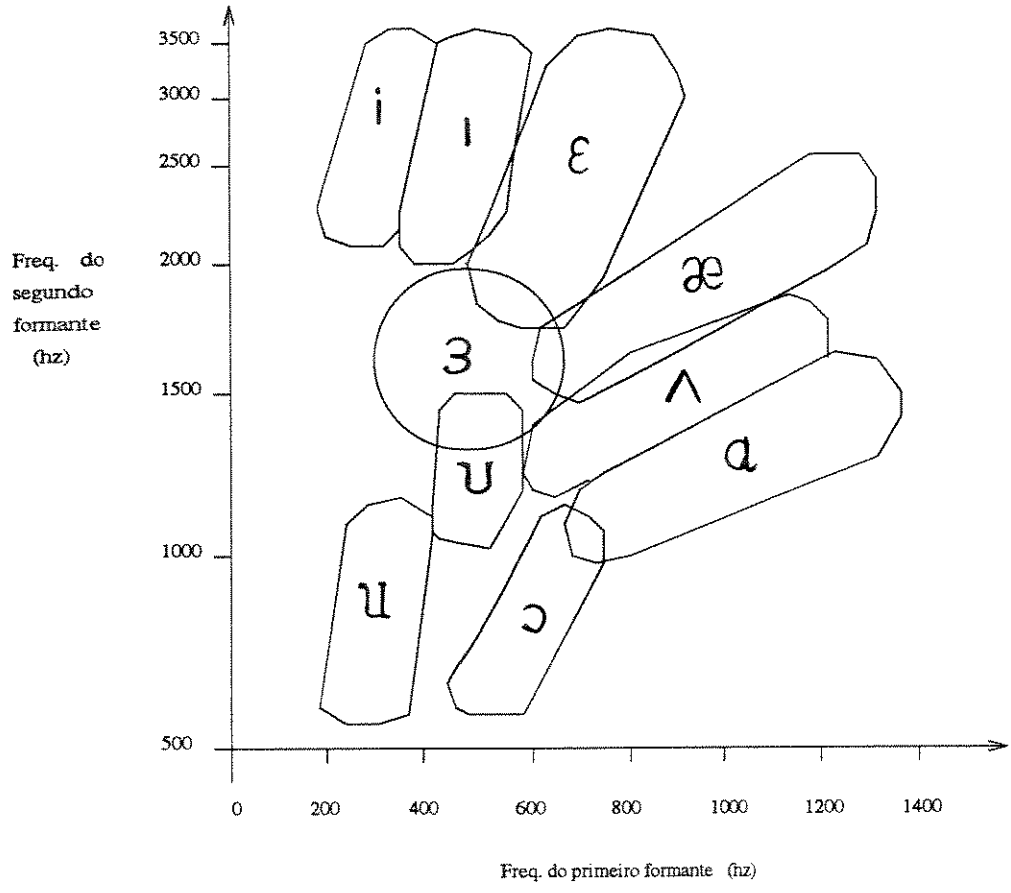


Figura 4.1.: Fonemas vocálicos em função dos primeiros dois formantes [5].

É importante então ter conhecimento, se possível quantitativo, das parametrizações a utilizar num sistema de reconhecimento de voz, tendo em mente o nível de exigência a que o sistema será exposto. Neste sentido, os resultados do estudo experimental de três das mais utilizadas parametrizações espectrais (Mel-cepstral, LPC, LPC-cepstral) serão apresentados neste capítulo. Antes, contudo, é necessário adiantar que nenhuma destas técnicas é capaz de extrair características inter-locutor e de, por assim dizer, "limpar" o sinal do ruído interferente. São estas fragilidades que tornaram extremamente bem sucedidos os métodos estocásticos de modelamento de padrões (HMM).

Por último, é interessante destacar que o método sugerido para medir a qualidade das técnicas de parametrização pode ser extendido ao estudo da utilização de parâmetros espectrais em conjunto com parâmetros temporais e os respectivos coeficientes de ponderação.

4.2. O ESTUDO DAS PARAMETRIZAÇÕES E O DTW

As aplicações dos sistemas de reconhecimento de voz em ambientes adversos são numerosas : telefonia móvel, aquelas que utilizam o canal telefônico como meio transmissor, indústrias, etc. Mesmo situações bem menos radicais, como pessoas conversando a média distância, podem causar interferências. Dado este panorama, é evidente a importância de saber onde as técnicas de parametrização falham e como isto pode ser resolvido. Esta última questão foge do assunto desta dissertação, mas será parcialmente respondida em termos de qual técnica é mais ou menos adaptada para tal contexto.

Como já foi mencionado nas introduções deste trabalho e deste capítulo, o modelamento estocástico (HMM) tornou-se tão popular em reconhecimento de voz nos últimos 5 anos que ficou praticamente impossível falar em reconhecimento sem mencionar os HMM. Isto porque as técnicas atuais de parametrização falham em não extrair características que independam do locutor e em não tornar as análises robustas aos ruídos interferentes.

Contudo, os HMM precisam de uma quantidade muito grande de dados de treinamento, e, de fato, o reconhecimento de voz também precisa lidar com uma base de dados grande. Mas a manipulação de muitas elocuções acaba escondendo as deficiências das técnicas de parametrização e torna o estudo destas, em certa medida, camuflado.

Por outro lado, como poderá ser visto no capítulo seguinte, onde será discutida a implementação de reconhecimento independente do locutor, os HMM exigem o treinamento dos modelos, a elaboração do code-book e quantização vetorial, estes dois últimos em se tratando de HMM's discretos. Tudo isto torna o trabalho experimental mais extenso.

Já o DTW, que compara diretamente duas elocuções parametrizadas eliminado as diferenças de duração dos segmentos onde o sinal é estacionário, não precisa

de uma quantidade elevada de dados de treinamento para funcionar corretamente, com todas as suas limitações é claro. Além disto, do ponto de vista experimental, ele só precisa das elocuções parametrizadas, o que torna bem menos trabalhosos os processos de comparação entre as técnicas de prametrização. A tabela 4.1. resume a análise comparativa entre o HMM e o DTW em relação ao estudo das técnicas de extração de parâmetros.

Tabela 4.1.: Comparação entre as técnicas DTW e HMM no que diz respeito ao estudo das parametrizações.

HMM	DTW
-Exige um número elevado de elocuções de treinamento para estimar os coeficientes dos modelos. O treinamento com poucas elocuções torna incompleto o modelamento das funções de probabilidade.	-Na sua versão mais simples compara diretamente duas elocuções parametrizadas, eliminando as diferenças de duração nos intervalos onde os sinais são estacionários.
-Uma vez parametrizadas, as elocuções de treinamento devem atualizar os HMM's com o algoritmo de Baum-Welch. No caso dos HMM's discretos, as elocuções de treinamento devem gerar um codebook e ser quantificadas antes da estimação dos modelos.	-O treinamento consiste apenas na parametrização das elocuções e na escolha daquelas que servirão como padrões de referência.

Assim, como o DTW é muito dependente da parametrização utilizada e manipula com eficiência as informações temporais, ele é uma boa ferramenta para estudar comparativamente técnicas de parametrização. Além disto, o treinamento consiste apenas na parametrização das elocuções e na escolha daquelas que servirão como referência. Tudo isto não quer dizer que o DTW seja a melhor técnica para

implementar um reconhecedor de palavras, muito pelo contrário, em sua versão "solo" é considerada ultrapassada em relação aos HMM no que diz respeito à flexibilidade, esforço computacional na operação de reconhecimento e capacidade de assimilar as variações temporais e espectrais do sinal de voz.

É importante destacar que para uma dada técnica de parametrização há um tipo de distância que melhor se lhe adapta. Assim, enquanto que para a análise Mel-cepstral e LPC-cepstral foi utilizada a distância euclideana, para a análise LPC foi empregada a distância de Itakura simplificada (seção 1.3.), que calcula a relação entre os erros de predição sem levar em conta os ganhos dos filtros preditores.

4.3. COEFICIENTES DE SELETIVIDADE DE RECONHECIMENTO

Conforme apresentado no capítulo 3, o reconhecimento de uma elocução por meio do algoritmo DTW se faz medindo a distância entre esta elocução, que não se sabe a priori a que palavra pertence, e as elocuções de referência, uma para cada palavra do vocabulário de reconhecimento e cujas origens são conhecidas. O reconhecimento consiste então em determinar o padrão de referência cuja distância ao padrão a reconhecer é a menor de todas. A figura 4.2. ilustra este procedimento.

Para cada elocução de teste são necessárias, no nosso caso particular, 10 medidas de distâncias obtidas por meio do algoritmo DTW. Se compararmos este sistema como um todo com um filtro de sinais, resultaria interessante poder quantificar com mais precisão numérica a seletividade de reconhecimento. Isto é, quão bem reconhece a palavra correta e quão bem rejeita as outras elocuções, numa clara analogia com os parâmetros banda de passagem, banda de rejeição, etc, característicos dos filtros de sinais.

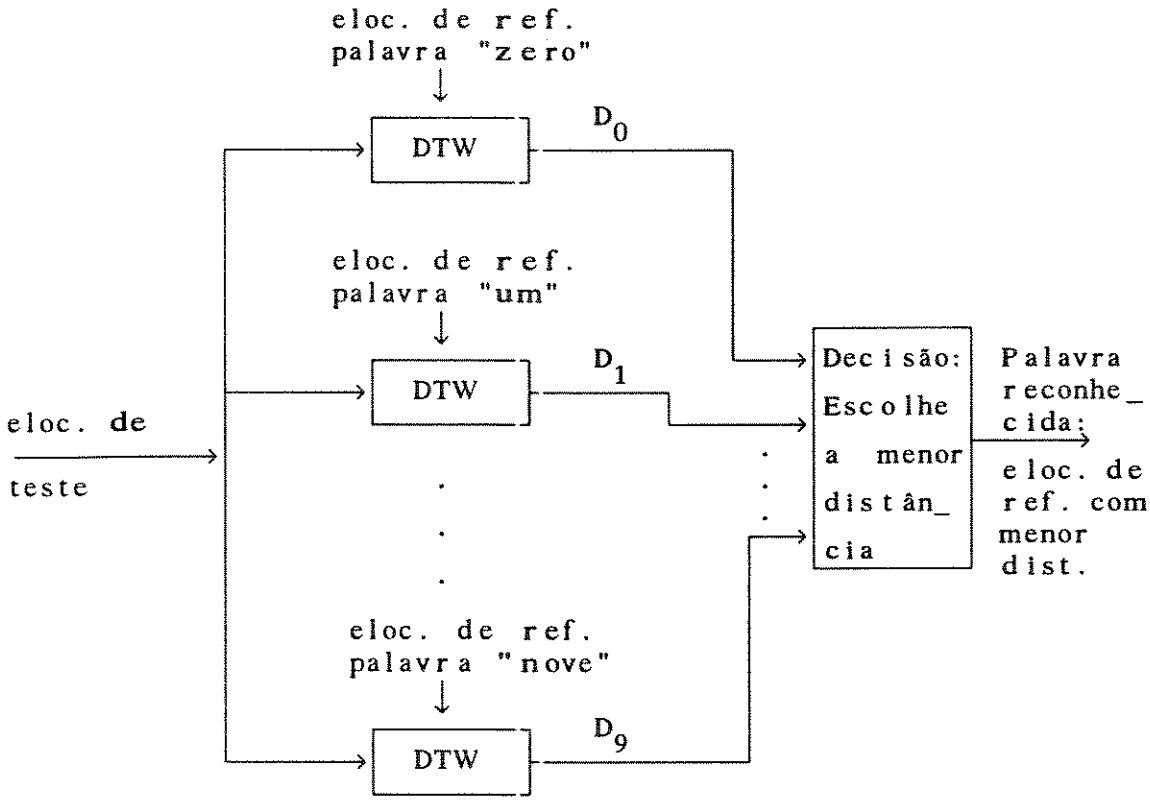


Figura 4.2.: Esquema de um reconhecedor de palavras baseado na técnica DTW.

Seja:

D_n ≡ distância entre a elocução de teste e a elocução da referência n, $0 \leq n \leq 9$;

D_n^1 ≡ a menor das distâncias D_n, $0 \leq n \leq 9$;

n_1 ≡ elocução de referência mais próxima da eloc. de teste;

D_n^2 ≡ a segunda menor distância;

$$\bar{D} \equiv \frac{\sum_{n=0;n\neq n_1}^{9} D_n}{9}$$

Definimos então, os seguintes coeficientes de seletividade se a elocução foi corretamente reconhecida:

$$C_1 = D_n^1$$

$$C_2 = D_n^2 - D_n^1$$

$$C_3 = \bar{D} - D_n^1$$

O primeiro coeficiente, C_1, é a menor distância entre a elocução de teste e as elocuções de referência. Dado que estes coeficientes são válidos se a elocução de teste foi corretamente reconhecida, tanto esta como a de referência pertencem à mesma palavra. Portanto, quanto menor C_1, melhor a qualidade de reconhecimento, e vice-versa.

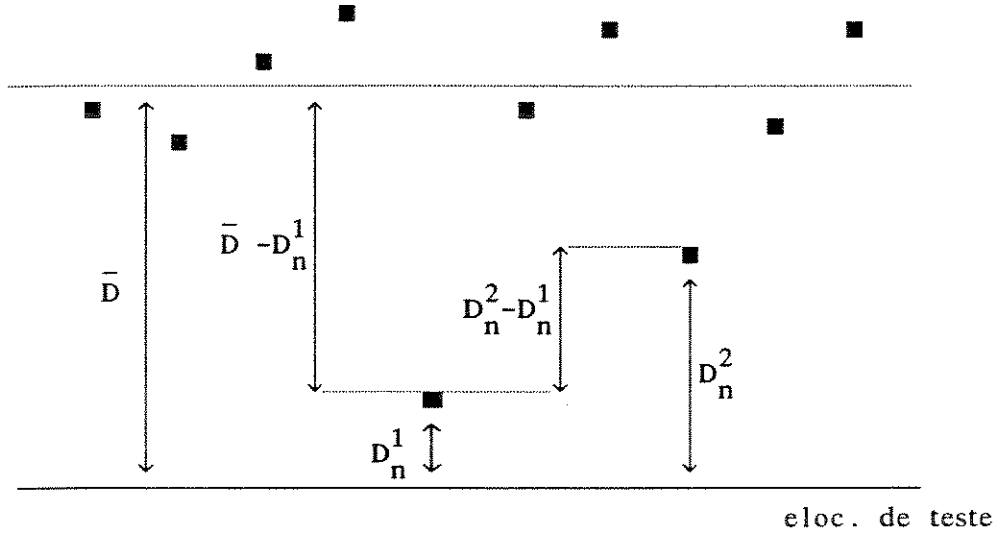


Figura 4.3.: Visualização dos coeficientes de seletividade. O símbolo (■) indica elocução de referência.

O segundo coeficiente é a diferença entre a segunda menor distância e a menor distância de todas entre a elocução de teste e as de referência. Portanto, quanto maior C_2, melhor a qualidade de reconhecimento, e vice-versa.

O terceiro coeficiente é a diferença entre a média das distâncias e a menor distância de reconhecimento entre a elocução de teste e as de referência que não pertencem à mesma palavra da elocução de teste. Portanto, quanto maior C_3, melhor a qualidade de reconhecimento, e vice-versa.

Como as parametrizações abordadas neste trabalho geram coeficientes nume-

ricamente diferentes, o resultado numérico da distância entre duas elocuções, determinada por meio do algoritmo DTW, é também diferente, dado que esta é soma das distâncias entre frames ao longo do caminho ótimo de comparação. Definiremos então coeficientes de seletividade que nos permitam comparar a performance de parametrizações distintas.

Dados os coeficientes C_1, C_2 e C_3, definimos os seguintes coeficientes normalizados:

$$Sel_1 = \frac{C_2}{C_1}$$

$$Sel_2 = \frac{C_3}{C_1}$$

Dado que o DTW determina a distância entre duas elocuções por meio da soma das distâncias entre frames ao longo do caminho ótimo de comparação, e que esta soma é normalizada em relação à duração dos padrões de teste e de referência, é muito razoável supor que os coeficientes acima são uma boa medida da qualidade da parametrização em si mesma. Isto porque, de acordo com o que foi discutido nos capítulos 2 e 3, o DTW elimina as diferenças de duração dos períodos estacionários do sinal de voz e, depois, normaliza a distância final em relação à duração das elocuções, eliminando dessa forma as informações temporais. Por outro lado, a normalização de C_2 e C_3 em relação a C_1 na determinação de Sel_1 e Sel_2, elimina as diferenças numéricas entre as diferentes parametrizações e permite a comparação direta dos diferentes procedimentos de extração de parâmetros. Da maneira em que Sel_1 e Sel_2 foram definidos, quanto maiores eles forem, melhor será a qualidade ou seletividade do reconhecimento.

4.4. PARAMETRIZAÇÕES E MEDIDAS DE DISTÂNCIAS ESTUDADAS

Foram implementados três processos de extração de parâmetros espectrais : Mel-cepstral, LPC e LPC-cepstral.

Os coeficientes Mel-cepstral (seção 1.2.4.) foram calculados como descrito na seção 3.3. e distância entre frames determinada com a distância euclideana.

A janela de Hamming foi fixada em 128 pontos e a FFT, calculada com 256 pontos. Cada quadro é formado por 10 coeficientes Mel-cepstral.

A análise LPC foi implementada com o algoritmo de Levinson-Durbin (seção 1.2.2.), obtendo 10 coeficientes LPC por quadro (recomendado para freq. de amostragem de 8 khz [5]). A distância entre quadros foi calculada com uma versão simplificada da distância de Itakura-Saito (seção 1.3.) onde não se levam em conta os ganhos dos filtros preditores, supondo que a intensidade do sinal de voz não interfere na mensagem fonética [5]. Esta simplificação conduz a :

$$d_{is}(Ar_i, A_t) = \frac{A\overset{T}{r}_i \cdot R_t \cdot Ar_i}{A_t^T \cdot R_t \cdot A_t} \qquad (4.1)$$

Por último, para determinar os coeficientes LPC-cepstral a partir dos coeficientes LPC, foi empregada a equação recursiva da seção 1.2.3., utilizando a distância euclideana para comparar quadros [7]. Neste caso também foram obtidos 10 coeficientes por quadro, a fim de compatibilizar a comparação com a análise LPC.

4.5. EXPERIMENTOS

As três parametrizações implementadas foram comparadas no que tange a três pontos essenciais do projeto de sistemas de reconhecimento de palavras. Estes são :

i) capacidade de reconhecer elocuções de um mesmo locutor;

ii) capacidade de distinguir elocuções de locutores diferentes;

iii) capacidade de distinguir elocuções em condições de ruído diferentes das condições de treinamento;

Para estudar i) foi realizado o mesmo experimento dependente do locutor do capítulo 3, onde os padrões de teste e de referência pertencem ao mesmo locutor, mas trocando a parametrização Mel-cepstral pelos coeficientes LPC e poste-

riormente pelos coeficientes LPC-cepstral. Para cada parametrização foram executados:

4(conj.eloc.de ref.)·30(num.eloc.de teste/locutor)·8(locutor)=
960 experimentos de reconhecimento

A comparação segundo o critério ii) foi implementada realizando o reconhecimento independente do locutor, onde as elocuções de referência e de teste correspondem a locutores diferentes. Isto é, com um conjunto de padrões de teste por locutor foram reconhecidas os padrões de teste dos outros 7 locutores. Ao todo, foram implementados

8(loc)·1(conj.eloc.teste/loc)·7(loc. a rec)·40(eloc./loc.a rec)=
2240 experimentos de rec por parametrização

No que diz respeito ao item iii), as técnicas de extração de parâmetros foram comparadas por meio do reconhecimento dependente do locutor, como no item i), mas adicionando ruído aos padrões de teste antes da parametrização. Este ruído adicionado ao sinal de voz foi gaussino e branco e sua potência fixada para três valores de SNR (10, 20 e 30dB, aproximadamente). A energia média do sinal de voz foi determinada diretamente da base de dados, gravada na forma de sequências de dígitos (figura 3.4). A distribuição gaussiana da amplitude do ruído foi aproximada com a convolução de 5 funções de probabilidade planas de média nula ("Teorema do Limite Central" [6]). Assim, a esperança (μ) e a variância (σ^2) do ruído são dadas por :

$$\mu = 5 \cdot \mu_p = 0 \tag{4.2}$$

$$\sigma^2 = 5 \cdot \sigma_p^2 \tag{4.3}$$

onde μ_p e σ_p são a esperança e a variância das funções de probabilidade planas utilizadas para aproximar a distribuição gaussiana.

Todos estes experimentos de comparação, segundo os itens i), ii) e iii), foram levados adiante com o algoritmo DTW de medida de distância entre elocuções, fixando a janela de busca em r = 12 (seção 3.6).

4.6. RESULTADOS

As tabela 4.1. apresenta os resultados (erro de reconhecimento, $Selec_1$ e $Selec_2$) dos testes de reconhecimento dependente do locutor para as três parametrizações estudadas. Foram realizados 960 testes de reconhecimento por parametrização.

A tabela 4.2. mostra os resultados dos experimentos de reconhecimento independente do locutor. Foram realizados 2240 experimentos de reconhecimento.

Tabela 4.1.: Resultados de testes de reconhecimento dependente do locutor.

	Mel-cepstral com dist.eucl.	LPC com dist. de Itak.simpl.	LPC-cepstral com dist.eucl
Erro de rec. em %	0.52	0.42	0.52
Sel_1	0.65	1.61	0.72
Sel_2	1.17	3.21	1.33

Tabela 4.2.: Resultados de testes de reconhecimento independente do locutor.

	Mel-cepstral com dist.eucl.	LPC com dist. de Itak.simpl.	LPC-cepstral com dist.eucl
Erro de rec. em %	24.8	31.0	30.8
Sel_1	0.18	0.32	0.18
Sel_2	0.46	0.88	0.47

As tabelas 4.3. a 4.5 apresentam os resultados de reconhecimento quando adicionado ruído branco gaussiano ao padrões de teste. Foram empregados três ruídos com potências difrentes: SNR = 10.0 dB; SNR = 20.2 dB; e SNR = 31.3. Os valores obtidos de SNR não são inteiros dado que a função de distribuição gaussiana da amplitude foi aproximada por meio da convolução de 5 variáveis aleatórias discretas independentes e identicamente distribuídas. Para cada

Para cada parametrização e cada relação sinal ruído foram executados 960 testes de reconhecimento.

Tabela 4.3.: Resultados de testes de reconhecimento dependente do locutor com SNR = 10.0 dB.

	Mel-cepstral com dist.eucl.	LPC com dist. de Itak. simpl.	LPC-cepstral com dist.eucl
Erro de rec. em %	62.6	68.2	70.9
Sel_1	0.12	0.19	0.13
Sel_2	0.54	0.78	0.56

Tabela 4.4.: Resultados de testes de reconhecimento dependente do locutor com SNR = 20.2 dB.

	Mel-cepstral com dist.eucl.	LPC com dist. de Itak. simpl.	LPC-cepstral com dist.eucl
Erro de rec. em %	25.6	32.6	38.1
Sel_1	0.21	0.31	0.18
Sel_2	0.63	1.07	0.62

Tabela 4.5.: Resultados de testes de reconhecimento dependente do locutor com SNR = 31.3 dB.

	Mel-cepstral com dist.eucl.	LPC com dist. de Itak. simpl.	LPC-cepstral com dist.eucl
Erro de rec. em %	3.03	3.22	4.37
Sel_1	0.43	0.82	0.39
Sel_2	0.92	1.94	0.90

As figuras 4.1. a 4.3. resumem os resultados das tabelas 4.3. a 4.5. Para SNR de 50 dB foram considerados os resultados dos testes dependentes do locutor sem ruído interferente, considerando apenas o ruído de quantização.

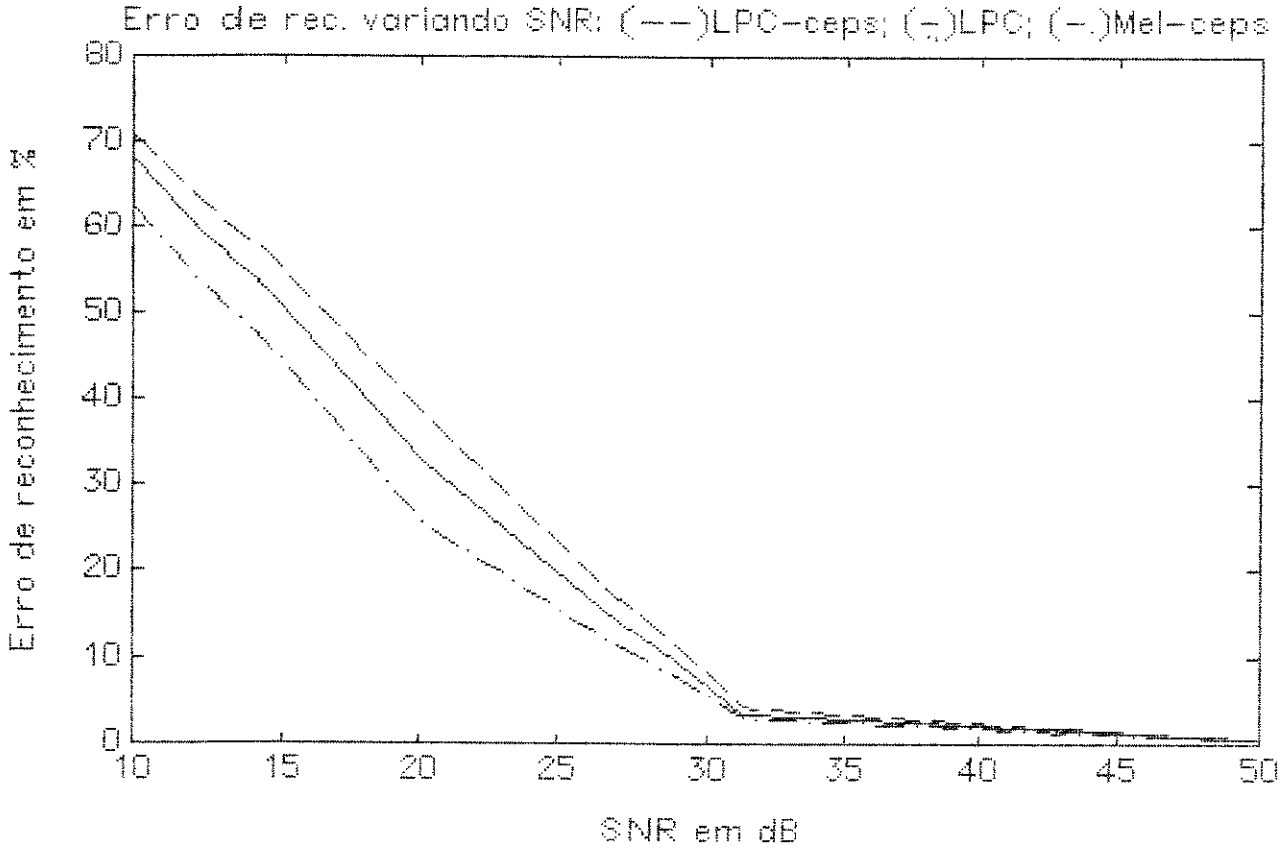


Figura 4.1.: Erro de reconhecimento em função da relação sinal-ruído nas elocuções a reconhecer para as três parametrizações.

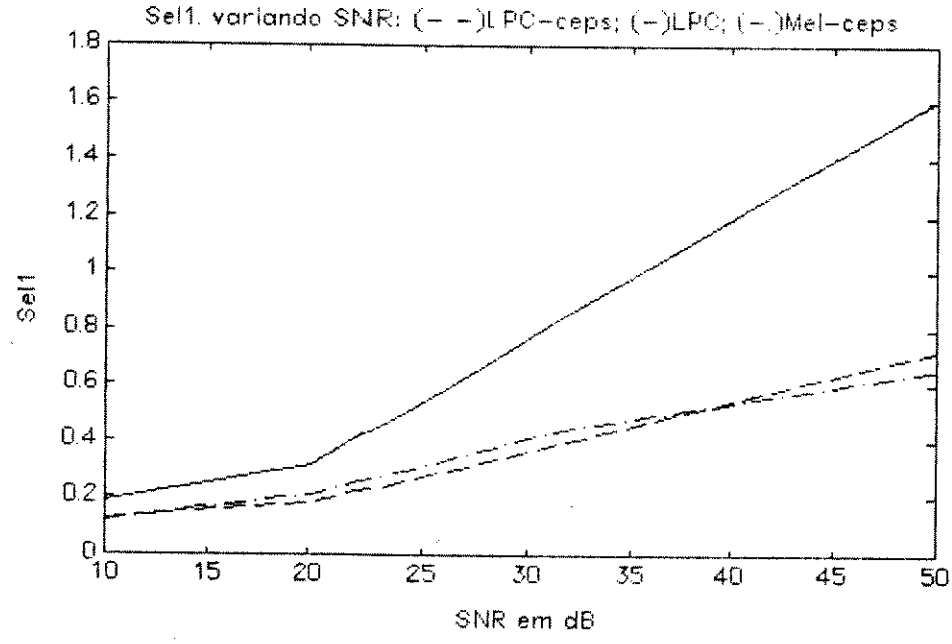


Figura 4.2.: Coeficiente Sel1 de seletividade de reconhecimento em função da relação sinal-ruído nas elocuções a reconhecer para as três parametrizações.

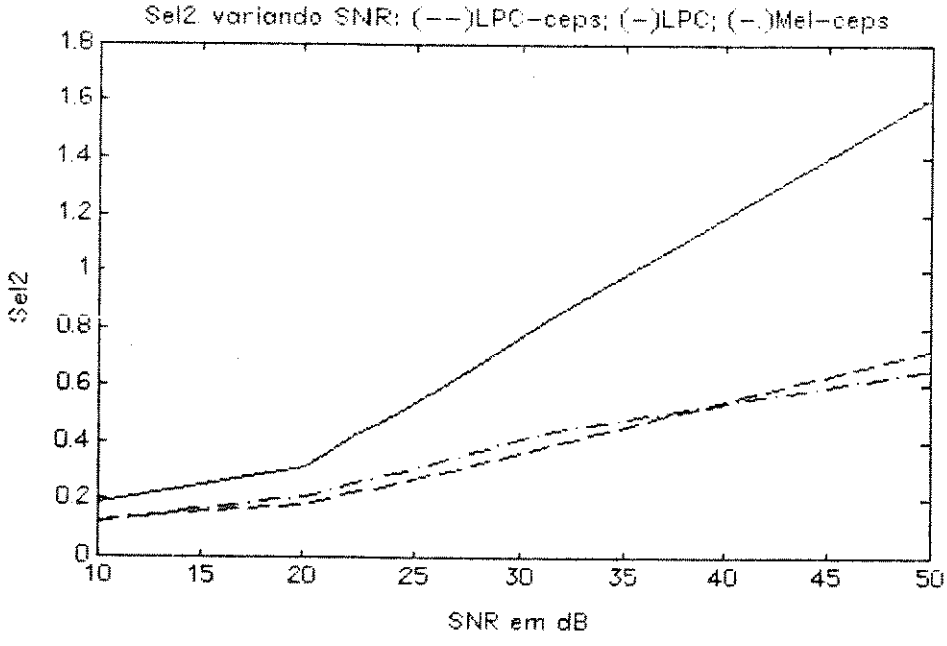


Figura 4.3.: Coeficiente Sel2 de seletividade de reconhecimento em função da relação sinal-ruído nas elocuções a reconhecer para as três parametrizações.

4.7. CONCLUSÕES

Certamente o leitor com alguma intimidade em processamento de voz, ou mesmo em PDS, terá como sugestão várias outras parametrizações para submeter a uma análise comparativa. Em especial, a energia por quadro (seção 1.6) e a variação temporal dos parâmetros espectrais [8] são informações que, de maneira geral, são muito utilizadas em sistemas modernos e reduzem em até 50 ou 70% a taxa de erro [9], quando utilizadas em conjunto com os parâmetros estáticos (coeficientes paramétricos comparados nesta seção). Contudo, o estudo exaustivo de todas as técnicas foge do escopo deste capítulo e merece a atenção de um trabalho a parte.

Antes de analisar os resultados é interessante observar que entropia igual 1, isto é, total imprevisibilidade em relação à informação fonética recuperada a partir das análises frequenciais, corresponde a uma taxa de erro de 90% , dado que há 10 palavras no vocabulário de reconhecimento. Em outros termos, a probabilidade de acertar a origem de uma elocução sem considerar nenhuma informação acústica, "a cegas", é de 10%.

Pela tabela 4.1. pode se concluir que a análise LPC, em conjunto com a distância de Itakura-Saito, assimila melhor as características dependentes do locutor que as análises Mel-cepstral e LPC-cepstral, pois apresenta menor taxa de erro e maiores (o dobro) coeficientes de seletividade. Contudo, a diferença na taxa de erro entre estas duas últimas técnicas e a análise LPC corresponde só a 0.1% ou 1 erro a mais. A semelhança entre os resultados da análise LPC-cepstral e a Mel-cepstral deve-se ao uso da distância euclideana que, aplicada sobre coeficientes cepstral, corresponde à distância entre espectros (seção 1.3).

Nos resultados da tabela 4.2. pode-se observar que as taxas de erros são extremamente elevadas, o que explica a inviabilidade do DTW ser aplicado como aqui em sistemas de reconhecimento independente do locutor. Contudo, o objetivo deste teste era verificar qual das três técnicas consegue assimilar melhor as características inter-locutor e, neste sentido, a análise Mel-cepstral apresenta uma taxa de erro 25% menor. Isto é, ela se apróxima um pouco melhor da percepção do sinal acústico no sistema auditivo periférico, uma vez que a escala mel é um modelo da resposta em frequência da membrana basilar [4][5]. A análise LPC e LPC-cepstral têm a mesma taxa de erro dado que elas utilizam a escala linear de frequência.

Os experimentos de robustez frente a ruído, cujos resultados são apresentados nas tabelas 4.3. a 4.5. e nas figuras 4.1. a 4.3, revelam que para SNR igual 10 e 20 dB há uma degradação muito grande na informação extraída do sinal de voz pelas três técnicas de parametrização. Contudo, segundo as taxas de erros, os parâmetros Mel-cepstral são 27 e 49% mais robustos que os das análises LPC e LPC-cepstral, respectivamente, para SNR=20 dB. Com SNR igual a 10 e 30 db, os coeficientes Mel-cepstral são também mais imunes à interferência, mas a diferença relativa é menor nestas situações. A maior robustez desta última técnica, no que diz respeito ao ruído aditivo, deve-se a que ela utiliza a energia de filtros passa-bandas, enquanto que as análises LPC e LPC-cepstral utilizam polinômios preditores cujos polos são muito susceptíveis ao ruído.

Tentando resumir estes resultados, podemos dizer que a análise LPC, em conjunto com a distância de Itakura-Saito, apresenta uma seletividade (Sel1 e Sel2) muito boa em relação às outras duas técnicas, quase sempre o dobro. Isto, em conjunto com fato de ela apresentar uma rejeição maior nos testes de independência do locutor, sugere que ela deve ser apropriada para aplicações de reconhecimento do locutor. Uma particularidade a mais da análise LPC, que não está registrada nestes resultados, é que ela é muito mais rápida que a análise Mel-cepstral implementada neste trabalho (baseada na FFT) e facilita sua implementação em tempo real. Contudo, os coeficientes Mel-cepstral apresentam uma robustez bastante melhor em relação ao ruído e assimilam melhor as características inter-locutor, em função da escala mel. Por último, os parâmetros LPC-cepstral, segundo foram aqui implementados, não apresentam nenhuma característica que os detaquem dos demais. Assim, pelos resultados relativos à robustez ao ruído interferente e à influência da escala mel, o reconhecedor de palavras independente do locutor com HMM do capítulo seguinte foi implementado urilizando os parâmetros Mel-cepstral.

Por último, além destes resultados, o autor espera ter contribuído com uma discussão até certo ponto original a respeito de técnicas de reconhecimento de padrões (DTW e HMM) e de como o DTW, apesar de ser ultrapassado, constitue-se numa ferramenta bastante apropriada para o estudo de parametrizações.

4.8. REFERÊNCIAS

[1] Flanagan,J.L. : "Speech Analysis Synthesis and Perception". Second Edition, Springer-Verlag, 1972.

[2] Fant,G. : "Vowels, Production and Perception". Dpt. of Speech Communication, Royal Institute of Technology (KTH), S-100 44, Stockolm 70, Sweden, 1975.

[3] Carlson,R. Fant,G. Granstrom,B. : "Two Formant Models, Pitch and Vowel Perception", Dpt. of Speech Communication, Royal Institute of Technology (KTH), Auditory Analysis and Perception of Speech, Edited by G.Fant and M.A.A. Tatham, Academic Press, 1975.

[4] Allen,J.B. : "Cochlear Modeling", IEEE ASSP Magazine, janeiro, 1985

[5] O'Shaughnessy, D. :"Speech Communication, Human and Machine", INRS-Telecomunications, Addison-Wesley Publishing Company,1987.

[6] Papopulis, A. : "Probability, Random Variables, and Stochastic Processes". McGraw-Hill, third edition.

[7] Davis,S.B. and Mermelstein,P. : "Comparison os Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences", IEEE Trans. ASSP, vol.28, pp.357-366, 1980

[8] Furui,S. : "Speaker-Independent Isolated Word Recognition Using Dynamic Features of Speech Spectrum".IEEE Trans.ASSP, vol.34, pp.52-59, Feb.1986.

[9] Dubois,D. : "Comparison of Time-Dependent Acoustic Features for a Speaker-Independent Speech Recognition System". Proceedings Eurospeech 91, Genova, Italy, vol.3, pp. 935-938.

CAPÍTULO 5

RECONHECIMENTO INDEPENDENTE DO LOCUTOR E HMM

5.1. INTRODUÇÃO

A comparação direta de elocuções parametrizadas por meio do algoritmo DTW permite realizar o reconhecimento de palavras pronunciadas pelo mesmo locutor com razoável precisão. Contudo, foi visto no capítulo 4 que se reconhecermos elocuções de um locutor com padrões de referência de outro, a taxa de erro aumenta a níveis nada aceitáveis para as três parametrizações estudadas : LPC, LPC-cepstral e Mel-cepstral. Apesar de que esta última apresentasse uma taxa de reconhecimento um pouco melhor, em função da escala mel, ela ainda está longe de ser considerada aceitável.

Esta limitação das parametrizações levou à necessidade da inclusão de algoritmos de comparação de padrões baseados na manipulação estatística de muitas elocuções para gerar padrões ou modelos robustos de referência. A técnica que melhor vem se aplicando em reconhecimento de voz são os HMM ("Hidden Markov Models") que permitem o modelamento estocástico da informação espectral (resultado das parametrizações) e temporal (duração dos fonemas, palavras, etc).

No capítulo 2 foram abordados os principais tópicos teóricos relacionados com os HMM, como o cálculo da verossimilhança e o treinamento dos modelos. Neste capítulo serão tratados os detalhes de implementação dos algoritmos relativos ao HMM tendo como aplicação o reconhecimento independente do locutor com a mesma base de dados utilizada nos capítulos 3 e 4.

5.2. RECONHECIMENTO DE PALAVRAS ISOLADAS COM HMM

Como foi discutido no capítulo 3, o modelamento por processos markovianos é muito apropriado à linguagem, de maneira geral. Para o caso restrito de reconhecimento de padrões acústicos, são as sequências de quadros que serão modeladas por meio dos modelos ocultos de Markov. O nome "ocultos" é empregado por

que cada palavra é modelada por uma seqüência de estados não observáveis, correspondendo a cada estado densidades de probabilidade de vetores de observação (quadros).

Utilizando a notação do capítulo 2, temos que o sinal de voz é segmentado em janelas de igual duração onde são extraídos uma série de parâmetros, representativos do segmento em questão, por meio de análises realizadas basicamente no domínio da frequência. Temos então que o sinal no tempo dá lugar a uma sequência $O=\{O_1, O_2, O_3, ..., O_{T-1}, O_T\}$ onde O_t é vetor de parâmetros correspondentes à janela t, e T a duração da palavra em número de quadros ou janelas. Cada vetor O_t vem sendo chamado neste trabalho de quadro, "frame", ou de vetor observação pois ele é o resultado de medidas físicas realizadas sobre o sinal acústico que transporta a informação fonética.

Em se tratando de HMM discretos, os implementados neste trabalho, cada vetor informação é quantificado vetorialmente (seção 1.7.) e se lhe atribui um codeword (v_l) do codebook, que contém ao todo L símbolos. Isto é:

$$q(O_t) = V_i \mid d(O_t, V_i) \leq d(O_t, V_j) \quad , i \neq j, \ 1 \leq i, j \leq L \qquad (5.1)$$

onde q() indica o operador de quantização e d() a medida de distância entre quadros (seção 1.3.).

Após a quantização vetorial, a sequência $O=\{O_1, O_2, O_3, ..., O_{T-1}, O_T\}$ dá origem então à sequência $V=\{V_1, V_2, V_3, ..., V_{T-1}, V_T\}$.

Os HMM's discretos foram preferidos em relação aos contínuos em função da base de dados disponível. De fato, os HMM's contínuos apresentam melhores taxas de acerto que os discretos [2] quando as funções de densidade de probabilidade são misturas de várias gaussianas, o que exige uma quantidade considerável de elocuções para treinar todos os parâmetros. Contudo, para uma mistura de poucas gaussianas os HMM's discretos são superiores aos contínuos [1]. Além do mais, o algoritmo de Baum-Welch é mais fácil de implementar para o caso discreto e exige menos esforço computacional. Por outro lado, as condições iniciais são muito menos importante nos HMM's discretos do que nos contínuos no que diz respeito à convergência do treinamento [2].

A escolha da topologia dos HMM é função do tipo de modelamento que se deseja fazer. Para reconhecimento de palavras isoladas com vocabulário pequeno (até algumas centenas de vocábulos), cada palavra pode ser modelada com um só HMM. Contudo, para o reconhecimento de vocabulários médio e grande (acima de 1000 palavras), ou de palavra contínua, o modelamento mais apropriado é o de fonemas ou difones [5].

Em nosso caso, como o vocabulário é pequeno (dígitos de 0 a 9), foi utilizado o modelamento por palavras inteiras. Para esta situação, a topologia mais empregada é a "esquerda-direita", onde não é permitida a transição de um estado para um anterior ou pular mais de dois adjacentes numa única transição [2]. A figura 5.1. mostra algumas configurações comuns.

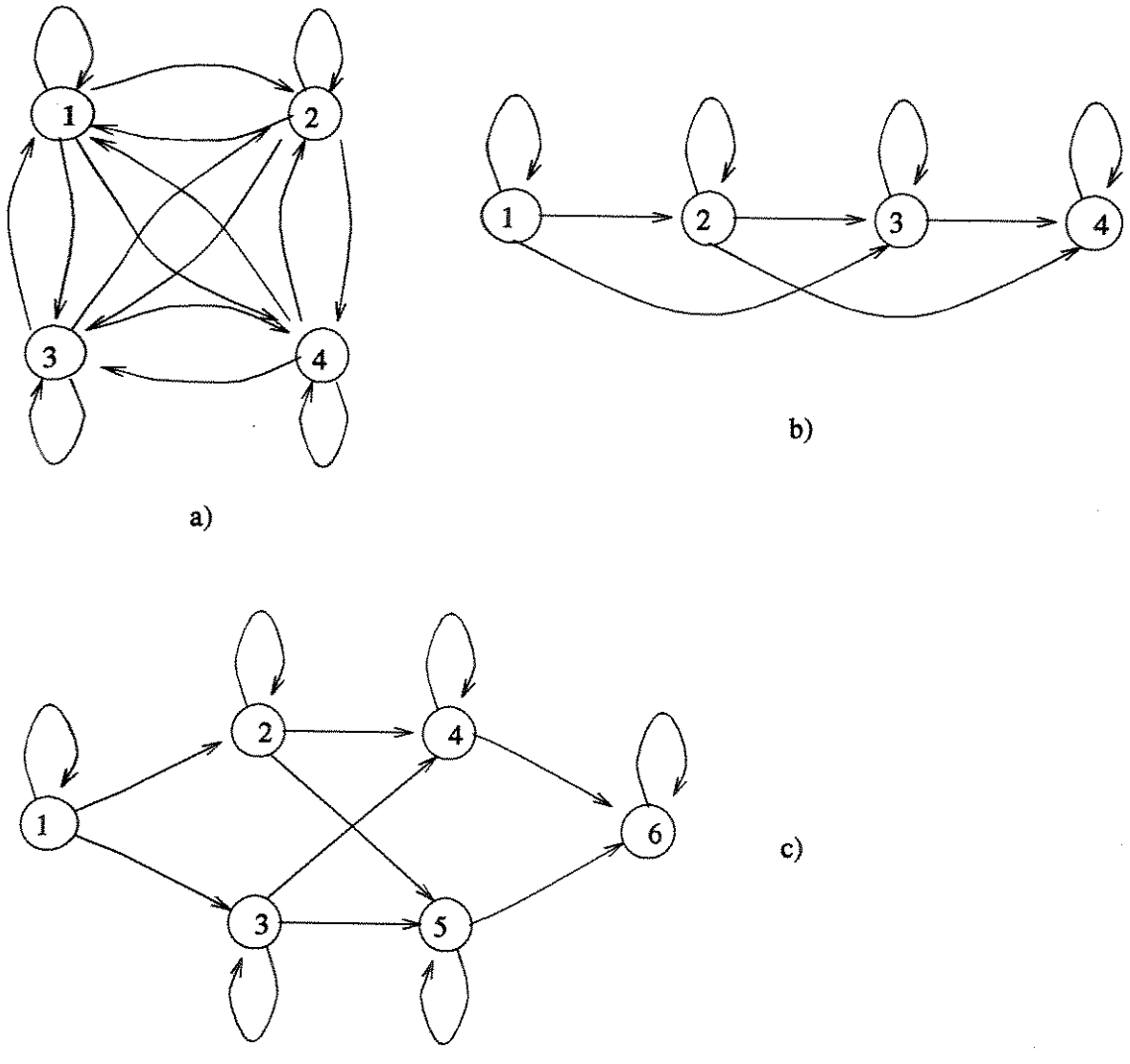


Figura 5.1.: Topologias de HMM. a) ergódica; b) esquerda-direita; c)paralela esquerda-direita.

Uma vez definida a topologia, cada modelo (que corresponde a uma palavra do vocabulário) é definido pelo número de estados, pelas probabilidades de transição entre os estados (matriz A), pela probabilidade de cada vetor de observação em cada estado (matriz B) e pela a probabilidade de cada estado ser o primeiro de cada sequência (vetor π).

Não existe nenhuma relação direta entre os fonemas e os estados dos modelos, mas recomenda-se que o número destes seja maior ou igual ao maior número de fonemas das palavras do vocabulário de reconhecimento [2]. No nosso caso, o número de estados foi fixado em 5. A figura 5.2. mostra a topologia "esquerda-direita" com 5 estados.

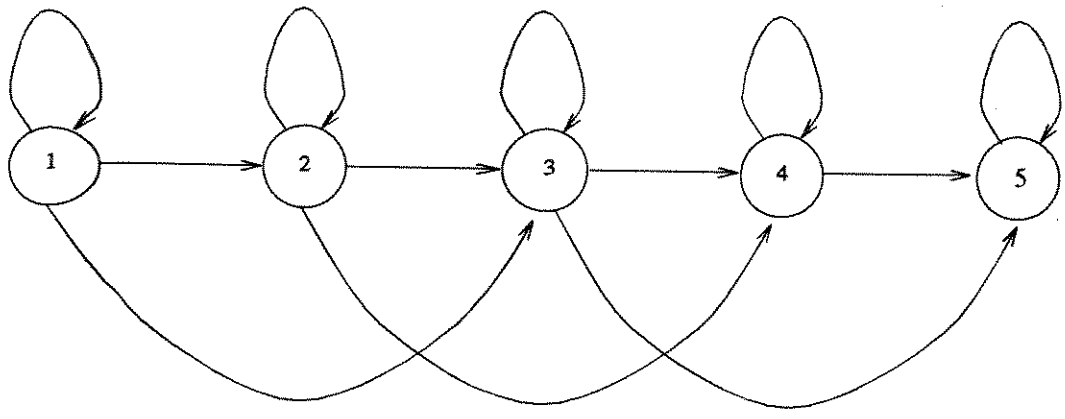


Figura 5.2.: HMM com topologia esquerda-direita e 5 estados.

Dado o número de estados, temos então que cada modelo é definido por:

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & 0 & 0 \\ 0 & a_{2,2} & a_{2.3} & a_{2.4} & 0 \\ 0 & 0 & a_{3.3} & a_{3.4} & a_{3.5} \\ 0 & 0 & 0 & a_{4.4} & a_{4.5} \\ 0 & 0 & 0 & 0 & a_{5.5} \end{pmatrix}$$

$$B = \begin{pmatrix} b_1(V_1) & b_1(V_2) & b_1(V_3) & \ldots & b_1(V_L) \\ b_2(V_1) & b_2(V_2) & b_2(V_3) & \ldots & b_2(V_L) \\ b_3(V_1) & b_3(V_2) & b_3(V_3) & \ldots & b_3(V_L) \\ b_4(V_1) & b_4(V_2) & b_4(V_3) & \ldots & b_4(V_L) \\ b_5(V_1) & b_5(V_2) & b_5(V_3) & \ldots & b_5(V_L) \end{pmatrix}$$

$$\Pi = \begin{pmatrix} \pi_1 & \pi_2 & \pi_3 & \pi_4 & \pi_5 \end{pmatrix}$$

onde :

$A = \{ a_{ij} \mid a_{ij} = Pr(s_{t+1}=j \mid s_t=i) \}$, distribuição de probabilidade de transição entre estados, a_{ij} indica a probabilidade de transição do estado i para o j;

$B = \{b_j(V_l) \mid b_j(V_l) = Pr(O_t=V_l \mid s_t=j)\}$, função de distribuição de probabilidade de saída dos estados, indica a probabilidade de se observar o elemento V_l no quadro do instante t (após a quantização vetorial), dado que neste instante se está no estado j;

$\Pi = \{\pi_i \mid \pi_i = Pr(s_1 = i)\}$, é a função de distribuição de probabilidade para o estado inicial, que nesta dissertação foi fixada em $\pi_1=1$ e $\pi_i=0$ para $i \neq i$ no modelo inicial;

Temos então que um modelo, λ, é definido pelos parâmetros A, B e Π. Dados estes parâmetros, é possível então calcular $P(O/\lambda)$, isto é, a probabilidade ou verossimilhança de uma dada sequência de observação $O=\{O_1, O_2, O_3, \ldots, O_{T-1}, O_T\}$, que após a quantificação vetorial dá origem à sequência $V=\{V_1, V_2, V_3, \ldots, V_{T-1}, V_T\}$, ter sido gerada pelo modelo $\lambda=(A, B, \Pi)$.

O processo de reconhecimento como um todo se dá nos seguintes passos :

i) atribui-se um modelo $\lambda^i=(A^i, B^i, \Pi^i)$ a cada palavra i do vocabulário de reconhecimento, no caso deste trabalho, dígitos de "zero" a "nove";

ii) a base de dados é dividida em base de treinamento e de reconhecimento; para realizar os testes de reconhecimento independente do locutor, as elocuções de sete locutores foram empregadas para treinar os modelos de cada palavra, e as do oitavo para reconhecer. Este procedimento foi repetido para todos os locutores; o treinamento é realizado com o algoritmo de Baum-Welch e consiste na estimação dos parâmetros A^i, B^i, Π^i que maximizam a verossimilhança das elocuções da palavra i dos locutores que não são o de reconhecimento;

iii) após a estimação dos parâmetros A^i, B^i, Π^i, o reconhecimento propriamente dito consiste em calcular as verossimilhanças de uma elocução, que não se sabe a priori a que palavra pertence, do locutor que não foi utilizado para treinar os modelos; a palavra escolhida é aquela cujo modelo apresenta a maior verossimilhança em ter gerado a sequência de observação da elocução (veja figura 5.3.);

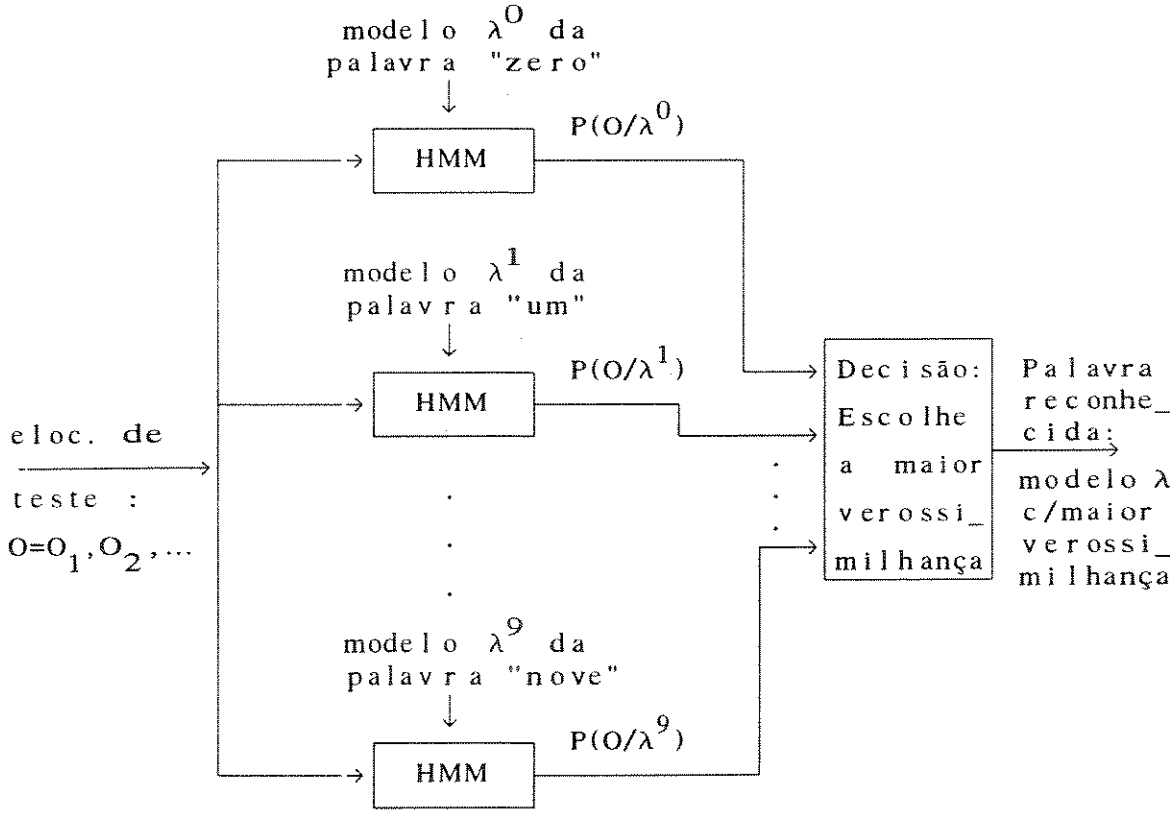


Figura 5.3.: Esquema de um reconhecedor de palavras baseado na técnica HMM.

5.3. IMPLEMENTAÇÃO

5.3.1. Condições Iniciais

O passo i) consiste basicamente na escolha de parâmetros iniciais para A^i, B^i e Π^i antes de iniciar o treinamento. Uma escolha apropriada do modelo inicial, em especial de B^i, favorece a covergência e melhora a performance do algoritmo [1][2]. Neste trabalho foram testadas algumas configurações iniciais para A^i mantendo B^i plana no modelo inicial. Foram obtidos bons resultados com $a_{i,i}$ elevados e $a_{i,j}$ $(i \neq j)$ pequenos. Estes resultados certamente não são tão bons quanto os conseguidos modificando B^i, porém são mais fáceis de se obter.

5.3.2. Treinamento

Determinado o modelo inicial e escolhidas as elocuções de treinamento, os parâmetros do modelo $\lambda^i=(A^i, B^i, \Pi^i)$ são re-estimados pelas equações 2.44, 2.45 e 2.46 de Baum-Welch, que são uma forma do algoritmo EM [5]. Como há várias sequências de treinamento e não uma só, estas equações devem ser modificadas. Seja $O^M = [O^1, O^2, ..., O^m]$ o conjunto de todas as seqüências de observação a serem utilizadas no treinamento, onde $O^n=\{O_1^n, O_2^n, O_3^n, ..., O_{Tn-1}^n, O_{Tn}^n\}$ é a n-ésima sequência de treinamento com Tn observações. Considerando que estas sequências são independentes umas das outras, a estimação dos parâmetros do modelo é então baseada na maximização de

$$\log Pr(O^M | \lambda) = \sum_{n=1}^{m} \log Pr(O^n | \lambda) \qquad (5.2)$$

Seja :

$\sum_{t=1}^{Tn-1} \gamma_t^n(i,j)$, o número esperado de transições do estado i para o estado j estimado a partir da sequência O^n;

Assim, o número médio esperado de transições do estado i para o estado j é a somatória $\sum \gamma_t^n(i,j)$ para todas as sequências O^n, com $1 \leq n \leq m$. As equações de re-estimação tornam-se então :

$$\bar{a}_{ij} = \frac{\sum_{n} \sum_{t=1}^{Tn-1} \gamma_t^n(i,j)}{\sum_{n} \sum_{t=1}^{Tn-1} \sum_{j} \gamma_t^n(i,j)} \tag{5.3}$$

$$\bar{b}_j(V_l) = \frac{\sum_{n} \sum_{t \in O_t^n = V_l} \gamma_t^n(j)}{\sum_{n} \sum_{t=1}^{Tn} \gamma_t^n(j)} \tag{5.4}$$

$$\bar{\pi}_j = \sum_{n} \gamma_1^n(j) \tag{5.5}$$

Seja λ um modelo a treinar e $\bar{\lambda}$ uma re-estimação do modelo λ após uma interação, o processo de treinamento é então repetido até que $\log Pr(O^M|\ \bar{\lambda}) - \log Pr(O^M|\ \lambda)$ seja menor ou igual que um dado limiar de convergência. A figura 5.4. apresenta a curva de convergência de $\log Pr(O^M|\ \lambda)$ para um determinado modelo. É importante lembrar que as equações de Baum-Welch garantem que $\log Pr(O^M|\ \lambda)$ sempre vai aumentar de uma interação para outra, exceto se $\bar{\lambda}=\lambda$, isto é, se λ for um máximo local [1][2][3].

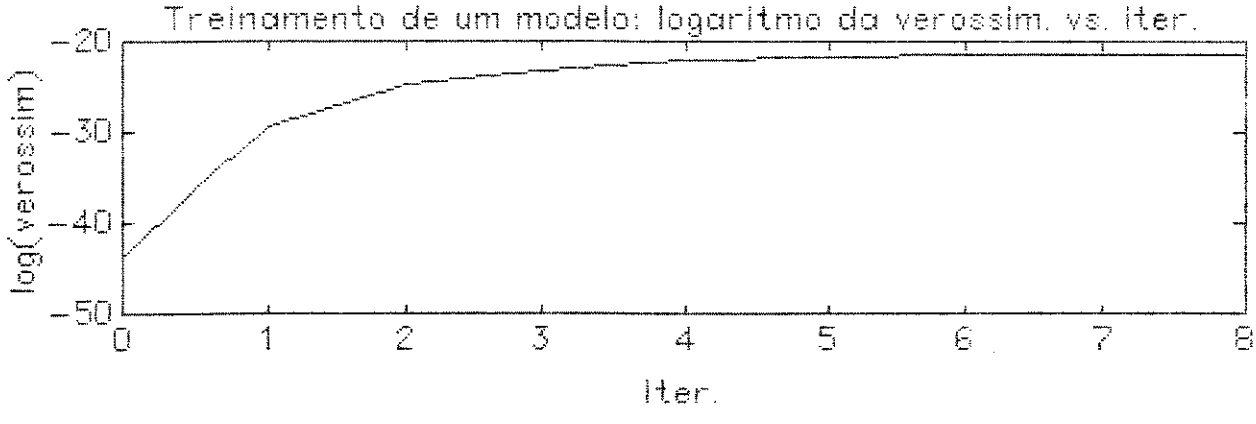


Figura 5.4.: Curva de adaptação dos parâmetros monitorada atravésde $P(O^M|\lambda)$.

5.3.3. Quantização Vetorial

Os HMM são uma técnica essencialmente estatística e, para tanto, exigem uma razoável base de dados de treinamento para estimar eficientemente todas as variáveis dos modelos. Para o caso dos HMM discretos, em [6] é sugerido que uma relação mínima aceitável entre o número de total de quadros das elocuções de treinamento e o número de variáveis a estimar seja da ordem de 10.

Posto que a base de dados disponível não foi elaborada para trabalhar com HMM, isto é, havia poucas elocuções para cada palavra do vocabulário, foi preciso trabalhar com um codebook bastante reduzido. Como o número médio de quadros por elocução era de aproximadamente 40 e cada modelo foi treinado com 4(eloc.por locutor)·7(locutores)=28 seqüências, tinha-se ao todo 40·28=1120 quadros ou vetores de observação por modelo a ser treinado. Dado que cada modelo consistia de 5 estados, e cada estado possuía L (número de codewords) probabilidades de saída, tinha-se que a relação entre o número de quadros de treinamento e o número de variáveis a estimar era igual a 1120/(5·L). Para garantir que esta relação fosse da ordem de 10, L foi fixado em 16 (seguindo a convenção de potência em relação a 2) o que dá uma relação de 14 quadros por variável a estimar. Um codebook ainda menor implicaria num aumento excessivo da distorção provocada pela quantificação vetorial.

Foram empregados 10 coeficientes Mel-cepstral por quadro, sendo que o codebook inicial foi determinado a partir das elocuções de um locutor e a otimização dos codeword's realizada com o algoritmo "K-means" (capítulo 1) sobre toda a base de dados. O processo foi finalizado quando a taxa de convergência foi menor que 1%.

5.3.4. Escalonamento

No cômputo das variáveis $\alpha()$ e $\beta()$ (seção 2.3.3.), utilizadas para o cálculo de $P(O/\lambda)$, e das variáveis $\gamma(i,j)$, empregadas nas equações de re-estimação de parâmetros, há um fator multiplicativo $b_j(O_t)$ por cada quadro da sequência de observação, onde $1 \le t \le T$ e T é o comprimento em frames da elocução. Assim, o valor das variáveis $\alpha()$ e $\beta()$ tendem a zero em proporção geométrica à medida que o comprimento da sequência de observação aumenta, podendo sair fora do domínio das variáveis "float" ou mesmo "double" do programa.

O princípio do escalonamento é multiplicar $\alpha_t(i)$ e $\beta_t(i)$ por um coeficiente de escalonamento para elimianar o risco de "underflow". Estes coeficientes devem ser eliminados no fim do cálculo de maneira a não modificar o algoritmo de Baum-Welch.

Os coeficientes $\alpha_t(i)$ são calculados pela equação 2.28 e em seguida multiplicados pelo coeficiente c_t, dado por:

$$c_t = [\sum_i \alpha_t(i)]^{-1} \quad , \tag{5.6}$$

de maneira a que $\sum_i c_t \alpha_t(i)=1$ para $1 \le t \le T$.

Os coeficientes $\beta_t(i)$ também podem ser multiplicados por c_t para $1 \le t \le T$ e $1 \le i \le N$, onde N é o número de estados. Como as variáveis forward e backward são computadas recursivamente de maneira exponencial, no instante t o fator de escalonamento total sobre a variável $\alpha_t()$ é :

$$C_t = \prod_{n=1}^{t} c_n \tag{5.7}$$

e o escalonamento total sobre as variáveis $\beta_t()$ é:

$$D_t = \prod_{n=t}^{T} c_n \tag{5.8}$$

Denotando por $\alpha'_t()$, β'_t e $\gamma'_t()$ as variáveis escalonadas obtidas a partir de $\alpha_t()$, $\beta_t()$ e $\gamma_t()$, respectivamente, temos que :

$$\begin{aligned} \sum_{i \in S_F} \alpha'_T(i) &= C_T \cdot \sum_{i \in S_F} \alpha_T(i) \\ &= C_T \cdot P(O/\lambda) \end{aligned} \tag{5.9}$$

A variável escalonada $\gamma'_t(i,j)$ pode ser escrita como:

$$\gamma'_t(i,j) = \frac{C_t \cdot \alpha_t(i) \cdot a_{ij} \cdot b_j(O_{t+1}) \cdot \beta_{t+1}(j) \cdot D_{t+1}}{C_T \cdot \sum_{i \in S_F} \alpha_T(i)} \qquad (5.10)$$

$$= \gamma_t(i,j)$$

A expressão 5.10 mostra que no cômputo das variáveis $\gamma'_t(i,j)$ os coeficientes C_t e D_{t+1} do numerador cancelam-se com C_T do denominador, tornando $\gamma'_t(i,j)$ igual a $\gamma_t(i,j)$ e mantendo válidas as equações de re-estimação 5.3, 5.4 e 5.5. Só $P(O/\lambda)$ deve ser computada segundo a expressão 5.9.

5.3.5. O Problema de Dados de Treinamento Insuficientes

A maior limitação dos HMM diz respeito à base de dados utilizada para treinar os modelos. Foi discutido na seção 5.3.3. que o tamanho do codebook era função do número de elocuções de treinamento, dado que para cada codeword há uma probabilidade em cada estado de cada modelo. Poucas sequências de treinamento deixam com frequência alguns elementos $b_j(V_l)$, a probabilidade de observar o elemento V_l dado que o estado atual é j, com valores iguais a zero. Isto porque a quantização vetorial força a associação de cada vetor observação O_t a um único codeword V_l. Uma medida paliativa é fixar um patamar mínimo para as probabilidades $b_j(V_l)$:após o treinamento os $b_j(V_l)$ menores que ε são igualados a ε,

$$\text{se } b_j(V_l) < \varepsilon \Rightarrow \quad b_j(V_l) = \varepsilon$$

onde ε é geralmente igual a 0.00001 [2][5].

A fixação do patamar mínimo evita que $P(O/\lambda^i)$ seja zero, pois se $P(O/\lambda^i)$ for zero o modelo λ^i é descartado de vez como possível gerador da sequência O, isto é, de que a palavra reconhecida seja i. Contudo, evitar que ocorram probabilidades iguais a zero não é suficiente para solucionar o problema da falta de dados de treinamento.

Uma solução bastante mais geral e eficiente consiste em considerar o modelo ideal como uma situação intermediária entre o modelo normal e um modelo incompleto com poucas variáveis a re-estimar, ambos treinados com a mesma base de dados. O modelo incompleto, por possuir poucos parâmetros, é considerado ro-

busto e bem treinado, enquanto que o modelo normal é considerado completo porém mal treinado [4].

Seja $b'_j(V_l)$ a distribuição de probabilidade do modelo incompleto (λ'), e $b_j(V_l)$, a distribuição do modelo completo (λ). Uma opção bastante frequente é considerar o modelo incompleto como uma função de probabilidade $b'_j(V_l)$ plana, isto é, para um mesmo estado j, a probabilidade de observar cada um dos elementos V_l do codeword, $1 \le l \le L$, é igual a 1/L, onde L é o número de codewords. O novo modelo ótimo, λ'', pode então ser considerado como uma interpolação dos modelos incompleto e completo:

$$b''_j(V_l) = k \cdot b_j(V_l) + (1-k) \cdot b'_j(V_l) \qquad (5.11)$$

onde $b''_j(V_l)$ é a distribuição de probabilidade do modelo λ''.

Um método sugerido para determinar as constantes k de interpolação é o denominado "Delete Interpolation", sugerido por Jelinek [1][2][4]. A idéia é considerar a capacidade dos modelos em reconhecer sequências de observação até então não-observadas. Por exemplo, se o modelo completo estiver bem treinado, ao determinar a verossimilhança de uma sequência (O) que não faz parte da base de treinamento, teremos que $P(O/\lambda) > P(O/\lambda')$. Por outro lado, se o modelo completo estiver mal treinado teremos que $P(O/\lambda) < P(O/\lambda')$. Assim, k será próximo de 1 se λ estiver bem treinado (λ melhor que λ'), e será próximo de 0 se λ estiver mal treinado (λ' melhor que λ)

A técnica "Delete Interpolation" sugere que a base de treinamento seja dividida em N blocos. Desses N blocos, uma parte (N-1, por exemplo) é utilizada para treinar um modelo incompleto e o restante dos blocos (y_i) para medir a capacidade de predição do modelo treinado. O coeficiente de interpolação é então obtido com a seguinte equação recursiva :

$$\bar{k} = \frac{1}{N} \sum_{i=1}^{N} \frac{k \cdot P(y_i|\lambda)}{k \cdot P(y_i|\lambda) + (1-k) \cdot P(y_i|\lambda')} \qquad (5.12)$$

onde $\bar{k}$ indica uma nova estimação de k, λ o modelo treinado com todos os blocos de dados exceto o bloco y_i; e λ' o modelo incompleto.

5.4. EXPERIMENTOS E RESULTADOS

Foram treinados para cada locutor da base de dados (4 homens e 4 mulheres) 10 modelos correspondentes a cada uma das palavras do vocabulário (dígitos de 0 a 9 do português). Cada modelo foi treinado por meio das equações (5.3), (5.4) e (5.5) de re-estimação, com 7 locutores (excluindo aquele a ser reconhecido) e utilizando 4 eloc./(palavra·locutor) · 7 locutores = 28 elocuções por palavra do vocabulário de reconhecimento. De acordo com o que foi mencionado na seção 5.3.1., no modelo inicial foi escolhida a distribuição plana de probabilidade para $b_j(V_l)$, enquanto que as probabilidades de transição foram determinadas da seguinte maneira :

$$a_{i,i} = 0.9 \; ; \quad a_{i,i+1} = a_{i,i+2} = 0.05$$

Uma vez treinados os modelos, foram reconhecidas as elocuções do locutor que não participou no treinamento dos modelos (reconhecimento independente do locutor) por meio do algoritmo de Viterbi (seção 2.3.3.2.). Este procedimento foi repetido para todos os locutores. Como cada um pronunciou cada palavra 4 vezes, foram realizados ao todo

4(eloc./palavra-locutor)·10(palavras do vocab)·8(locutores)=
320 experimentos de reconhecimento

Com este procedimento, de treinar os modelos com as elocuções de sete locutores e reconhecer logo em seguida as elocuções do oitavo, foram obtidos 293 acertos contra 27 erros ou, em porcentagem, 8.4% de taxa de erro.

Dado que a base de dados possuía poucas elocuções por locutor, foi necessário recorrer à técnica "Delete Interpolation" para melhorar a taxa de erro. Foram corrigidos somente aqueles modelos que propiciaram erro. Por exemplo, se a palavra 3 do locutor 2 teve alguma elocução mal reconhecida, então o modelo correspondente à palavra 3 treinado com as elocuções da palavra 3 dos outros locutores foi corrigido com a técnica "Delete Interpolation". Após este procedimento, aplicado a todos os locutores foram, conseguidos 302 acertos contra 19 erros ou, em porcentagem, 5.9% de taxa de erro. A tabela 5.1. resume estes resultados.

Tabela 5.1.: Resultados dos experimentos de reconhecimento com independência do locutor.

	Sem Del. Int.	Com Del. Int
Num.de Exper.	320	320
Num.de Erros	27	18
Taxa de Erro	8.4%	5.9%

5.5. CONCLUSÕES

A maior vantagem dos HMM's é que eles podem lidar com muitas elocuções de uma mesma palavra e assimilar toda a informação nos parâmetros de um só modelo. Contudo, a maior desvantagem dos HMM é que eles precisam de um número bastante elevado de elocuções de uma mesma palavra para conseguir treinar corretamente todas as funções de probabilidade do respectivo modelo. No caso particular da implementação deste capítulo, a base de dados de que se dispunha não foi elaborada para ser utilizada com HMM e possuía poucas elocuções por dígito. Isto forçou a diminuir o tamanho do codebook , aumentado assim a distorção média da quantização vetorial. Além disto, foi necessário recorrer à técnica "Delete Interpolation" para conseguir a taxa de erro de 5.6%. Este valor, quando comparado com o 3.7% apresentado em [2], parece bastante razoável levando em consideração os tamanhos das respectivas bases de dados de treinamento: 7 locutores, 4 eloc. por locutor por palavra (28 eloc. por dígito), contra 100 locutores com 1 eloc. por locutor por palavra (100 eloc. por dígito). Por outro lado, a inclusão de coeficientes dinâmicos (variação no tempo dos parâmetros espectrais) reduz a taxa de erro em 50 a 70% [7].

Uma outra limitação dos HMM diz respeito ao modelamento da duração dos estados. A probabilidade de continuar num mesmo estado por n quadros é $a_{i,i}^{n} \cdot (1-a_{i,i})$, isto é, corresponde a uma distribuição geométrica. Contudo, esta distribuição não é apropriada para descrever a duração dos períodos estacionários do sinal de voz, havendo portanto, versões de HMM onde a duração dos esta-

dos é modelada com funções de probabilidades mais apropriadas [1][2]. Neste sentido, a escolha de $a_{i,i}$ elevado, tipicamente 0.9, no modelo inicial, antes do treinamento, parece ser uma opção interessante, sendo que alguns testes preliminares apresentaram bons resultados quando comparada com valores menores como $a_{i,i} = a_{i,i+1} = a_{i,i+2} = 1/3$.

5.6. REFERÊNCIAS

[1] Huang,X.D. Ariki,Y. Jack,M.A. : "Hidden Markov Models for Speech recognition", Edinburgh University Press, 1990.

[2] Rabiner,L.R. : "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". Proceedings of the IEEE, vol.77, No.2, February, 1989.

[3] Baum,L.E . Petrie,T . Soules,G. Weiss,N. : "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains". The Annals of Mathematical Statistics, 1970, vol.41, No.1, pp.164-171.

[4] Lee, Kai-Fu : "Large Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System". Phd thesis, Department of Computer Science, Carnegie Mellon University, 1988.

[5] Dempster,A.P. Laird,N.M. Rubin,D.B. :"Maximum - likelihood from incomplete data via the EM algorithm". J. Royal Statistic Soc. Serie B, vol.39, pp.1-38, 1977.

[6] O'Shaughnessy, D. : "Speech Communication. Human and Machine". Adison-Wesley Publishing Company, 1987.

[7] Dubois,D. : "Comparison of Time-Dependent Acoustic Features for a Speaker-Independent Speech Recognition System". Proceedings Eurospeech 91, Genova, Italy, vol.3, pp. 935-938.

CAPÍTULO 6

DISCUSSÃO FINAL

6.1. RESUMO GERAL DO TRABALHO

O problema do reconhecimento automático de palavras é um dos maiores desafios da engenharia hoje em dia, envolvendo uma quantidade muito grande de ferramentas e um conhecimento multidisciplinar. Tentou-se, ao longo deste trabalho, estudar, discutir e experimentar as técnicas mais utilizadas e populares em reconhecimento de voz nos últimos anos e medir suas performances e limitações. Em primeiro lugar, foram estudadas três das mais empregadas parametrizações espectrais (LPC, LPC-cepstral e Mel-cepstral) com o algoritmo DTW, que compara elocuções eliminando as diferenças de duração entre elas. As técnicas de parametrização foram comparadas segundo a capacidade de assimilar características intra-locutor e inter-locutor e quanto à robustez frente a ruído, ficando patente nos resultados a extrema fragilidade destes processos quando comparados com o sistema auditivo periférico do ser humano. A análise Mel-cepstral é a que apresentou a melhor assimilação de características inter-locutor (em função da escala mel) e maior robustez frente a ruído, enquanto que a análise LPC, além de exigir menos cálculos, teve maior seletividade de reconhecimento.

Por último, implementou-se um reconhecedor de dígitos isolados independente do locutor com HMM e Mel-cepstral. Os HMM revelaram ser uma técnica extremamente cômoda de se trabalhar, pois cada modelo consegue assimilar informações relativas a muitas elocuções e a convergência da operação de treinamento é garantida pelo algoritmo de Baum-Welch. Contudo, esta técnica é muito dependente do tamanho da base de dados de treinamento pois, se esta for pequena, as funções de probabilidade dos modelos não são estimadas corretamente.

Com este trabalho o autor espera ter contribuído com uma discussão e uma visão críticas, até certo ponto originais, do que é o reconhecimento de voz hoje em dia. Nenhuma técnica foi testada exaustivamente, o que daria certamente um trabalho a parte para cada método, mas tentou-se analisar os fundamentos teóricos e práticos que permitissem aos leitores interessados entender como e porque utilizar uma dada ferramenta. Assim, sempre que possível foram consultadas refe-

rências de autores renomados para fixar um ou vários parâmetros dos algoritmos utilizados nas implementações práticas. Por outro lado, mais do que fechar assuntos, esta tese tenta mostrar os pontos frágeis do reconhecimento de palavras e por isto abre um leque de temas a pesquisar em futuras teses de mestrado e doutorado, e que era justamente um dos objetivos do autor. Finalmente, é interessante mencionar que este trabalho tem o mérito de ser um dos primeiros realizados em reconhecimento de palavras com a língua portuguesa do Brasil, e por tanto, o autor enfrentou uma série de empecilhos práticos como a falta de uma base de dados suficientemente robusta.

6.2. PERSPECTIVAS

Hoje em dia o que se consegue fazer nos principais laboratórios do mundo em termos de reconhecimento de voz, está ainda bastante longe de imitar a capacidade de comunicação entre os seres humanos (via fala). Sendo assim, este campo de estudo, além de ser novo no Brasil, oferece um leque bastante diversificado de linhas de pesquisa, como por exemplo, aumentar o vocabulário, passar de palavra isolada a palavra contínua, melhorar a robustez em relação ao ruído, reconhecimento em ambientes adversos, introdução de conhecimento lingüístico, e pesquisa de novas técnicas tais como as redes neurais. Neste sentido, poderíamos dizer que esta dissertação sugere de maneira imediata a abordagem das três primeiras linhas de pesquisa citadas.

A grande popularidade do modelamento por HMM deve-se principalmente à sua capacidade de lidar com uma quantidade elevada de informação e à sua flexibilidade. O modelamento por palavra, empregado na implementação do capítulo 5, é viável para um vocabulário de até algumas centenas de palavras, contudo, exige uma quantidade excessiva de elocuções de treinamento para vocabulários da ordem de mil palavras. Assim, a primeira saída encontrada para vocabulários grandes é modelar cada fonema com um HMM. Como o número de fonemas é inferior a 100, o modelamento destes é muito mais fácil e é possível modelar qualquer palavra como uma sequência de HMM de fonemas. Porém, como os fonemas são muito influenciados pelos fonemas vezinhos (coarticulação), o reconhecimento empregando HMM para estas unidades tem pior desempenho que para palavras inteiras [1]. Uma alternativa muito bem sucedida é modelar os fonemas de acordo com o contexto, como no caso dos trifones [2]. Nesse caso, o modelo colhe informação do fonema anterior

e do posterior (veja figura 6.1). Assim, um mesmo fonema em contextos diferentes pode ser representado por trifones diferentes.

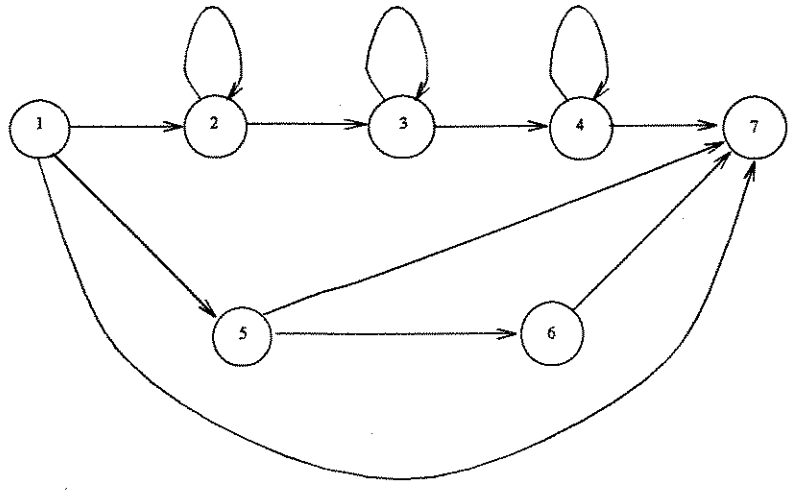


Figura 6.1.: Topologia de HMM para modelar um fonema [2].

Para o problema de palavra contínua os HMM também foram utilizados com bastante êxito. O algoritmo "Level-Building" [3] utiliza modelamento com HMM por palavra e a sequência de palavras é determinada pelo algoritmo de Viterbi. O "Level-Building" é apropriado para vocabulários pequenos e já foi testado com dígitos concatenados. Contudo, quando o vocabulário aumenta, é necessário empregar modelamento por fonemas (ou trifones) e técnicas sub-ótimas ("Beam-Search") [4]para determinar a melhor sequência de modelos, além de utilizar gramáticas [2].

No que diz respeito a melhorar a robustez frente a ruído, ficou patente no capítulo 4 que as parametrizações que se empregam hoje em dia em sistemas de reconhecimento são extremamente susceptíveis a ruídos interferentes. Isto priviligia ainda mais o modelamento de padrões acústicos por HMM, pois ele

permite treinar os modelos com elocuções corrompidas por diversos níveis de interferência [5], dado que ao comparar uma elocução "limpa" com uma corrompida por ruído, ambas da mesma palavra e do mesmo locutor, elas podem se tornar muito diferentes quando parametrizadas com as técnicas atuais. Assim, o estudo de processos de parametrização mais robustos é, sem dúvida, uma área muito interessante de pesquisa.

O reconhecimento da fala pode ser considerado como um motor propulsor para vários outros campos de pesquisa pois exige um conhecimento multidisciplinar, desde ferramentas de PDS e análise estocástica até biologia, passando pela lingüística e inteligência artificial. As opções de pesquisa são tão diversificadas que resulta difícil conceber grupos reduzidos de trabalho neste campo, deixando para trás o conceito do pesquisador solitário.

6.3. REFERÊNCIAS

[1] Bahl,L.R. Brown,P.F. de Souza,P.V. Mercer,R.L. : "Acoustic Markov Models used in the Tangora Speech Recognition System". Proc.ICASSP, 1988.

[2] Lee, Kai-Fu : "Large Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System". Phd thesis, Department of Computer Science, Carnegie Mellon University, 1988.

[3] Rabiner,L.R. J.G.Wilpon Soong,F.K. : "High Performance Connected Digit Recognition Using HMM". Proc. ICASSP, 1988.

[4] Ney,H. Mergel,D. Noll,A Paeseler,A.: "A Data-Driven Organization of the Dynamic Programming Beam Search for Continuous Speech Recognition". Proc. ICASSP, 1987

[5] Dubois,D. : "Comparison of Time-Dependent Acoustic Features for a Speaker-Independent Speech Recognition System". Proceedings Eurospeech 91, Genova, Italy, vol.3, pp. 935-938.

APÊNDICE A

IMPLEMENTAÇÃO EM LINGUAGEM "C" DO ALGORITMO FORWARD

Cada HMM (um por cada palavra do vocabulário) é definido pelas seguintes estruturas:

```
/*Cada estado da topologia*/
struct estado {
        short num_saidas;
        short num_entradas;
        short flag_inic;
        short flag_fim;
        float prob_saida[16];
        int arcos_saida[3];
        int arcos_entrada[3];
        };

/*Cada transição da topologia*/
struct trans  {
        int destino;
        int origem;
        float prob_trans;
        };


/*HMM*/
struct hmm {
        int num_estados;
        int num_trans;
        int num_estados_inic;
        int num_estados_fim;
        int num_prob_saida;
        float pi[5];
        struct estado estados[5];
        struct trans transitions[12];
        };
```

O conjunto dos dez (número de palavras do vovabulário) HMM's é definido pela seguinte estrutura:

```
struct hmm_geral {
        struct hmm modelo[10];
        char nome_hmm[8], locutor_ref;
        };
```

Dadas as declarações acima o algoritmo Forward é escrito como a seguir. Nesta implementação o algoritmo tem dois parâmetros de entrada: o número de quadros (int num_quadros) da seqüência de observação a analisar; e o modelo (int palavra) utilizado.

```
struct hmm_geral hmm_locutor;
double alfa[100][5];          /*variáveis α */
double beta[100][5];          /*variáveis ß */
double escal[100];            /*Coeficientes de escalonamento*/
int sec_qv_trein[100];        /*seqüência de treinamento após a
                                    quantif. vetorial*/

/*Algoritmo forward*/
forward(int num_quadros, int palavra)
    {
    int stop, cptr, i, s;
    float o_prob;
    struct trans tptr;
    for(i=0; i<100; i++)
        {
        escal[i]=0.0;
        for(s=0; s<hmm_locutor.modelo[palavra].num_estados; s++)
            {
            alfa[i][s]=0.0;
            }
        }

    for(s=0; s<hmm_locutor.modelo[palavra].num_estados; s++)
        {
        alfa[0][s]=hmm_locutor.modelo[palavra].pi[s]*
           hmm_locutor.modelo[palavra].estados[s].prob_saida[sec_qv_trein[0]];
        escal[0]+=alfa[0][s];
        }

    for(s=0; s<hmm_locutor.modelo[palavra].num_estados; s++)
        {
        alfa[0][s]/=escal[0];
        }

    for(i=1; i<num_quadros; i++)
        {
        for(s=0; s<hmm_locutor.modelo[palavra].num_estados; s++)
            {
            stop=hmm_locutor.modelo[palavra].estados[s].num_entradas;
            o_prob=hmm_locutor.modelo[palavra].estados[s].
                    prob_saida[sec_qv_trein[i]];
            for(cptr=0; cptr<stop; cptr++)
                {
                tptr=hmm_locutor.modelo[palavra].
                      transitions[hmm_locutor.modelo[palavra].estados[s].
                      arcos_entrada[cptr]];
                alfa[i][s]+=alfa[i-1][tptr.origem]*(tptr.prob_trans);
                }
```

```
                alfa[i][s]*=o_prob;
                escal[i]+=alfa[i][s];
                }
        for(s=0; s<hmm_locutor.modelo[palavra].num_estados; s++)
                alfa[i][s]/=escal[i];
        }
}
```

APÊNDICE B

B.1. Seja a seguinte restrição linear sobre um conjunto de variáveis x_i:

$$f_1(X) = \sum_{i=1}^{n} x_i = 1 \qquad (b.1)$$

e dados os parâmetros c_i $(1 \le i \le n)$, determinar x_i $(1 \le i \le n)$, tal que a função

$$f(X) = \sum_{i=1}^{n} c_i \cdot \log(x_i) \qquad (b.2)$$

seja maximizada.

Utilizando o método dos multiplicadores de Lagrange [1], definimos a função expandida como sendo:

$$f_a(X) = f(x) + \lambda \cdot f_1(X) \qquad (b.3)$$

onde λ é denominado multiplicador de Lagrange.

Temos então, que [1]:

$$\frac{\partial f_a(X)}{\partial x_i} = \frac{\partial f(X)}{\partial x_i} + \frac{\partial f_1(X)}{\partial x_i} = 0 \qquad (b.4)$$

Substituindo $f(X)$ e $f_1(X)$ pelas expressões (b.1) e (b.2) na expressão (b.4), temos:

$$c_i + \lambda \cdot x_i = 0 \qquad (b.5)$$

Fazendo a somatória em ambos membros da expressão (b.5) para $(1 \le i \le n)$ chegamos a :

$$\sum_{i=1}^{n} (c_i + \lambda \cdot x_i) = 0$$

$$\Rightarrow \sum_{i=1}^{n} c_i + \lambda = 0$$

$$\Rightarrow \lambda = - \frac{1}{\sum_{i=1}^{n}} \qquad \text{(b.6)}$$

Portanto, substituindo λ nas expressões (b.5) para $1 \leq i \leq n$, temos:

$$x_i = \frac{c_i}{\sum_{i=1}^{n} c_i} \qquad \text{(b.7)}$$

B.2. REFERÊNCIA

[1] Haykin, S.: "Adaptive Filter Theory". Prentice Hall, Englewood Cliffs, New Jersey