ANDRÉ PASQUALE ROCCO SCAVONE

RECONHECIMENTO DE PALAVRAS POR MODELOS OCULTOS DE MARKOV

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Engenharia.

SÃO PAULO
1996

ANDRÉ PASQUALE ROCCO SCAVONE

RECONHECIMENTO DE PALAVRAS POR MODELOS OCULTOS DE MARKOV

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Engenharia.

Área de concentração:
Sistemas eletrônicos

Orientador:
Geraldo Lino de Campos

SÃO PAULO

1996

“E o dono foi perdendo a voz...
E disse: Minha voz, se vós não sereis minha
Vós não sereis de mais nínguém.”

A voz do dono e o dono da voz
Chico Buarque

a meus pais,

Tê e Nino,

ao
Prof. Geraldo,

Martinha,

Fernando,
Mary,
Ana Elisa,
Cris, Luísa,
Maria Alice,
Orlando Legname,
Sérgio, Guilhermo,

Neusa e
colegas da biblioteca do DEE,

pela indispensável colaboração neste trabalho,
meus sinceros agradecimentos.

Sumário

Lista de Figuras

Lista de Tabelas

Resumo

O uso de modelos ocultos de Markov (HMM) na tarefa de reconhecimento de voz tem sido objeto de extensa pesquisa. Esses modelos utilizam dois processos estatísticos inter-relacionados: enquanto um modela a variabilidade dos ritmos de emissão, o outro representa a diversidade dos fenômenos acústicos da fala. Este segundo processo permitiria também absorver as caracteríticas de diferentes vozes.

Este trabalho estuda o uso dos modelos ocultos de Markov através da implementação de um sistema de reconhecimento de vocabulário restrito. O sistema utiliza a análise por coeficientes de predição linear e quantização vetorial para representar o sinal de voz por uma seqüência de símbolos que estima os parâmetros dos modelos.

Os resultados obtidos com um locutor confirmam a capacidade de representação desses modelos. No entanto, o desempenho do sistema se reduz consideravelmente quando aplicado a diversos locutores. Algumas alternativas são propostas no sentido de melhorar o desempenho do sistema, sem atingir grande êxito. As soluções para a independência do locutor apontam para métodos adaptativos que preservem a versatilidade dos HMM.

Abstract

The use of hidden Markov models (HMM) in automatic speech recognition has been object of extensive research. These models use two inter-related statistics processes, one modeling the variability of the utterances rythms while the other describes the diversity of speech acoustical phenomena. The latter would also be able to manage the features of different individual voices.

This work studies the use of hidden Markov models through a recognition system for a restricted vocabulary. The system uses the linear coeficient prediction analysis and vector quantization to represent the voice signal by sequences of symbols to estimate model parameters.

The results achieved with a single speaker support the representation capability of these models. However, system performance drops significantly when applied to different voices. Some alternatives are proposed to improve the system in this way, without achievieng better results. The solutions for speaker-independent tasks points to adaptative methods preserving HMM versatility.

1 Introdução

Quando escutamos alguém falar, um conjunto de perturbações causadas pela emissão refletem-se em impulsos nervosos transmitidos ao cérebro pelo nosso sistema auditivo. O que percebemos não é um conjunto confuso de sons que de alguma forma faz sentido, mas entes significativos: as palavras.

Esta é uma abordagem da psicologia ao problema da percepção: o estudo do reconhecimento das formas (LINDSAY, e NORMAN, [1980]), que busca a compreensão do processo pelo qual os sinais externos percebidos pelos órgãos sensorais são transformados em experiências perceptivas dotadas de sentido.

O "aspecto dos gabaritos" é o esquema mais simples aplicado ao reconhecimento das formas. Assim, cada forma é reconhecida pela máxima semelhança com um gabarito interno pré-existente. No entanto, a flexibilidade da percepção humana (se consideramos, p. ex., a capacidade de isolarmos um discurso no meio de diversas vozes), e a capacidade da percepção de novas formas (o reconhecimento de uma palavra que não conhecemos) indicam a insuficiência desta proposta se considerada isoladamente.

A identificação de gabaritos é apenas uma componente da percepção: o tratamento da informação *orientado por dados*. Uma outra componente atua tratando a informação *por conceitos*. Isto ocorre, por exemplo, quando, na conversação, uma certa expectativa deduzida do contexto permite que interpretemos corretamente as palavras de mesmo som, como "voz" e "vós".

*

Esta breve digressão serve para introduzir o problema de reconhecimento de voz por máquinas. A abordagem da engenharia é a de um típico problema de reconhecimento de padrões que envolve duas etapas: o treinamento, ou aprendizado, quando geramos um modelo daquilo que se está analisando, e uma segunda etapa, o reconhecimento propriamente dito, ou classificação, quando iremos associar uma entrada do sistema a um dos modelos previamente treinados.

Em ambas etapas temos que extrair do objeto físico, no caso a voz transformada em sinal elétrico, as informações necessárias para o tratamento que iremos realizar. Esta fase será denominada *extração de atributos*.

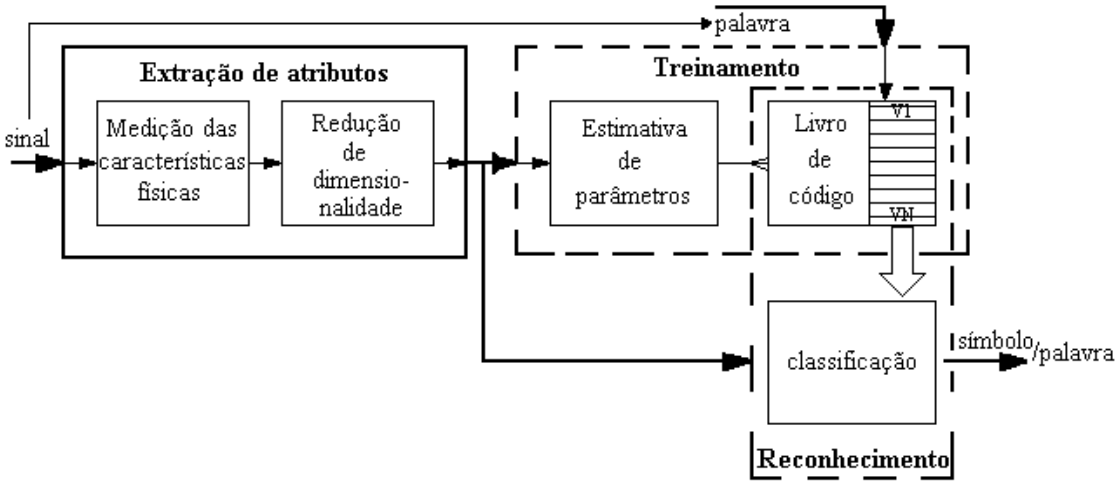


Figura 1.A Diagrama de um sistema de reconhecimento.

Do sinal são extraídas grandezas físicas diretas, tais como a energia ou taxa de cruzamentos por zero, ou indiretas, através de transformações como Fourier ou o cálculo de parâmetros de modelos de síntese, os coeficientes de predição linear.

Procuramos então um modelo que da melhor forma aprenda as características relevantes de um conjunto de dados, de tal modo que um outro conjunto de dados semelhante possa ser identificado como tal.

Até aqui, estamos realizando um sistema de reconhecimento com a informação *orientada por dados*. A partir deste nível de informação, podemos incluir um mecanismo de reconhecimento não mais baseado em comparações de medidas físicas, mas baseado em estruturas lógicas de decisão. Isto permitiria um reconhecimento que se aproximasse daquele realizado pelo ser humano.

Apresentada desta forma, a realização de um sistema de reconhecimento de voz poderia parecer um problema trivial para o qual bastaria um microfone e um computador para resolvê-lo. No entanto, os computadores tem se mostrado bem menos flexíveis que os seres humanos.

A percepção humana não apresenta maiores dificuldades com relação ao tipo de voz, ou se as palavras são ditas isoladamente ou são co-articuladas. Além disso, extrai informações “além da palavra”, como expressões de aprovação, alegria, raiva, etc. E, já mencionamos, também é capaz de reconhecer palavras que não constam no “seu vocabulário interno”. Já para os sistemas automáticos de reconhecimento de voz destacam-se, sob estes aspectos, limitações quanto a:

- dependência do locutor - sensibilidade às características particulares de cada locutor, seja em relação ao timbre de voz, ritmo ou estilo de fala, ou classes de locutores tipo vozes masculinas, femininas ou infantis;
- dimensão ou complexidade do vocabulário - número de palavras tratadas, bem como as particularidades do vocabulário, tais como a utilização de vocabulários por temas específicos;
- fluxo de emissão - palavras emitidas isoladamente ou tratamento de discurso contínuo;
- tratamento da expressão - capacidade do sistema de extrair as palavras sobre diferentes formas de entoação;
- tempo de resposta - possibilidade de reconhecimento em tempo real;
- meio de transmissão - banda de passagem do meio que transmite o sinal de voz;
- nível de ruído ambiente.

As duas últimas características, por implicarem em uma degradação da informação, também afetam o reconhecimento pelo homem.

*

Neste trabalho estudamos um sistema de reconhecimento, baseado em modelos estatísticos, bastante limitado com relação às restrições acima. Tratamos um vocabulário restrito, sem co-articulação, sem variações prosódicas expressivas, em baixo nível de ruído, com um bom meio de transmissão. A questão da dependência do locutor, por outro lado, é extensamente analisada.

A intenção inicial era desenvolver um sistema com características semelhantes às do SPHINX (LEE, [1991]). No entanto, isto logo se mostrou inviável pela complexidade de uma proposta tão abragente e pela necessidade de um banco de dados bastante extenso. Assim, optamos por um sistema simples mas viável, baseado nos primeiros trabalhos de LEVINSON [1983] e RABINER[1983] com os modelos ocultos de Markov.

Mesmo considerando-se as restrições do projeto, o seu desenvolvimento permite a compreensão da forma como estes modelos operam no reconhecimento de voz. O método apresenta interesse uma vez que permite o desenvolvimento de sistemas mais complexos pela incorporação de outras fontes de informação tais como sintaxe ou gramática.

*

No capítulo seguinte (Cap.2), apresentamos um histórico dos sistemas de reconhecimento e as idéias básicas envolvidas, destacando tanto os métodos de alinhamento dinâmico como os processos estatísticos baseados em modelos ocultos de Markov, que constituem duas formas distintas de abordar o problema.

No Cap. 3 apresentamos a teoria dos modelos ocultos de Markov, que iremos utilizar na implementação do sistema. Desenvolvemos a seguir (Cap. 4), a análise por síntese do sinal de voz através de coeficientes de predição linear que fornecerá parâmetros das grandezas físicas (atributos) para o tratamento estatístico. Estas grandezas podem ser tratadas diretamente, no modelo contínuo, ou devem ser transformadas em símbolos discretos conforme discutimos na seção 3.4.

O modelo discreto que utilizamos requer que o espaço contínuo desses parâmetros seja transformado em índices, associados a vetores aproximados, o que será feito através da quantização vetorial (Cap. 5).

Apresentamos no Cap. 6 os detalhes de implementação do sistema para reconhecimento de palavras isoladas, com vocabulário restrito, depentente ou não do locutor. Tratamos particularmente neste capítulo das dificuldades de implementação desse sistema com modelos ocultos de Markov, tais como a estrutura do modelo, problemas numéricos e questões relativas à dimensão do conjunto de treinamento.

Os resultados obtidos em diversos testes realizados sob uma variedade de condições são expostos no Cap. 7. Um exemplo das etapas do processo é apresentado no Anexo A, que analisa também dois erros no reconhecimento do sistema. Esta análise fornece indícios das possíveis falhas, apontando algumas propostas implementadas no final do Cap. 7.

2 Histórico

FLANAGAN [1976] identifica os primeiros ensaios de sistemas de reconhecimento de voz nos protótipos de "máquinas de escrever comandadas por voz" de Fry (1958) e Dreyfus-Graf (1961). O desenvolvimento dos computadores e a técnica digital modificaram significativamente as perspectivas de realizar estes sistemas. Assim, os esforços para a criação de sistemas automáticos de reconhecimento de fala (ASR-Automatic Speech Recognition) ganham fôlego na década de 70, surgindo trabalhos que ainda hoje são paradigmas da pesquisa.

Os primeiros sistemas buscavam uma semelhança entre amostras e referências baseados em conhecimentos acústicos do sinal de voz. Tais métodos tinham base na capacidade de leitura de espectrogramas, supondo a existência de elementos capazes de classificar os diversos fenômenos acústico-linguísticos do sinal. Sambur e Rabiner (apud FLANAGAN, [1976]) implementaram um sistema de reconhecimento para os 10 dígitos (0 a 9), independente do locutor, baseado na comparação direta entre atributos das amostras e referências, tendo obtido taxas de acerto de 94%.

ITAKURA [1975] desenvolve a técnica de alinhamento dinâmico entre amostra e referência (dynamic time warp - DTW), baseada na programação dinâmica de Bellman, tendo obtido 97% de acerto para reconhecimento de 200 palavras isoladas para um único locutor.

O projeto ARPA é iniciado em 1971 com um conjunto de metas bastante ambicioso para a época, como tratar o discurso contínuo e realizar a compreensão de frases montadas sobre um vocabulário de 1000 palavras. Baseando-se em uma estrutura de partilhamento de dados proposta para sistemas de inteligência artificial, conhecida como "*blackboard model*" (apud MARIANI [1989]), esse projeto produziu seus principais resultados em meados da década de 70 nos sistemas DRAGON, HEARSAY e HARPY.

BAKER [1975] introduz os modelos ocutlos de Markov no reconhecimento de voz, desenvolvendo o DRAGON System que utiliza um modelo estocástico uniforme para diferentes fontes de informação; trabalhando com 194 palavras, dependente do locutor, obteve 84% de acerto.

O HEARSAY System (LESSER et al., [1975]) reconhecia, em um sistema dependente do locutor, 87% das palavras corretamente, com informações fonéticas e linguísticas. O HARPY System (Lowerre, 1976, apud LEE [1991]) combinou vantagens do HEARSAY e do DRAGON, utilizando estrutura de rede e busca orientada, atingindo 97% de reconhecimento nas mesmas condições do HEARSAY.

Na mesma época, um grupo de pesquisadores da IBM (JELINEK, [1976]) apresenta também um sistema que utiliza modelos ocultos de Markov, obtendo taxas de 90% de acerto no reconhecimento contínuo de dígitos, dependente do locutor.

Estes trabalhos refletem os principais paradigmas que envolvem os sistemas de reconhecimento de voz: de um lado o alinhamento dinâmico direto dos padrões ou a aplicação de estruturas estatísticas, de outro os diferentes níveis de informação, sejam as medidas físicas extraídas do sinal de voz ou dados de "alto nível" como os referentes ao léxico e à sintaxe. Este último aspecto corresponde à forma de tratamento da informação *por dados* e *por conceitos* como mencionado na introdução.[1]

O sistema Tangora da IBM (apud LEE, [1991]) aparece em meados da década de 80, sendo o primeiro a trabalhar com vocabulários robustos (5000 palavras). Alcançava taxas de 97% de acerto aplicado a um locutor, com palavras isoladas; o sistema também era aplicável a discurso contínuo mas com degradação da performance.

Uma tentativa de contornar o problema da dependência do locutor surge com a técnica de agrupamento de vozes (speaker clustering) implementada no sistema dos Bell Labs (Wilpon, 1982, apud LEE [1991]) que a utilizou para gerar uma única referência a partir de diversos locutores; obteve 91% de acerto para 129 palavras, isoladas, independente do locutor. Outro sistema dos Bell Labs (Rabiner et al. 1988, apud LEE, [1991]) trabalhando com modelos ocultos de Markov contínuos, com diferentes distribuições, obteve 97% de acerto no reconhecimento de sentenças, independente do locutor. A questão da independência do locutor permance ainda hoje como uma restrição das mais difíceis de ser superada.

No reconhecimento de discurso contínuo, utilizando informações dependentes do contexto e o modelamento de fonemas, surge o BYBLOS System (Chow e Kubala, 1987 apud LEE, [1991]). O sistema, a princípio dependente do locutor, permitia a adaptação com treino relativamente rápido. Obteve reconhecimento de 93% para condições semelhantes ao HARPY, mas utilizando uma grámatica mais livre.

O SPHINX (LEE, [1991]) começa a ser desenvolvido em meados da década de 80, surgindo no início desta década como um sistema robusto de reconhecimento de palavras em discurso contínuo, independente do locutor. Integrando informações sobre a fala com modelos de Markov, atingiu 96% de reconhecimento, nas mesmas condições do BYBLOS, sujeito à

[1] ALLERHAND [1987] denomina estas duas abordagens de "Pattern-Recognition" e "Knowledge-Based" considerando-as como duas escolas distintas, divididas por motivos que vão do econômico ao ideológico! É evidente que, pela complexidade da proposta de um sistema de reconhecimento de voz, os grupos de pesquisa tenham seguido linhas diferentes. Naturalmente os progressos desses diferentes grupos irão se integrar no sentido de resolver o problema, como o autor involuntariamente acaba por sugerir.

dependência do locutor. Utilizando técnicas de adaptação ao locutor, o índice de acerto melhorou até 0,4%.

Outra abordagem atual nos sistemas de reconhecimento de voz é o uso das redes neurais, ou modelo conectivista (TATTERSALL, [1990]). Após as tentativas frustradas das máquinas de Minsky e Papert's da década de 60, esta técnica ganha fôlego com as propostas de "perceptrons multi-níveis" e a máquina de Boltzmann. Este método inspira-se na forma de funcionamento do cérebro humano. Partindo-se de unidades que representam modelos de neurônios, os "perceptrons", cria-se uma rede que, através de um conjunto de conexões, pode ser aplicada à solução de determinado problema. No treinamento, um processo iterativo determina o conjunto de parâmetros que regem essas conexões.

*

Na língua portuguesa destacamos os trabalhos de FRAGA [1991] e SANCHES [1989] que realizaram sistemas para o reconhecimento dos dez dígitos isolados, independentes do locutor. Esses sistemas baseavam-se no conhecimento das características acústícas das palavras. Nessa linha citamos também o trabalho de VIEIRA [1989] que procura de identificar os fenômenos acústicos da fala como parte do projeto de um sistema de reconhecimento.

Entre os trabalhos que utilizam os modelos de Markov aparecem os de FAGUNDES [1993] e MINAMI [1993]. O primeiro, utilizando estrutura sintática, obteve resultados modestos no reconhecimento de discurso contínuo independente do locutor. Minami obteve, com modelos de Markov discretos, índices de acerto de 100% dentro do conjunto de treinamento, trabalhando com os dez dígitos isolados para diversos locutores.

*

Os sistemas de reconhecimento de voz já atingiram o mercado, particularmente o americano, sendo cada vez mais acessíveis. Os mais simples apresentam apenas um suporte para "controle e comando", i.e., permitem que a entrada de voz se relacione com os comandos da interface gráfica Windows. Nesta classe estão os sistemas Listen da Vertex (LABRIOLA, [1995]) e Voice Assist da Creative Labs (CREATIVE TECH., [1993]). Este último utiliza uma técnica de blocos de vocabulário que variam conforme a janela ativa do Windows, reduzindo o vocabulário de busca de acordo com o aplicativo em uso. Requerem treinamento e identificação do locutor.

Os sistemas mais robustos, como o DragonDictate da Dragon Systems e o Kurzweil Applied Intelligence's Voice manipulam vocabulários mais extensos, anunciando taxas de erro de 2 a 5% após 5 a 10 horas de treino-uso e apresentam tempo de reconhecimento da ordem de 30 palavras por minuto. A IBM (WATTERSON, [1995]) apresenta o ICSS (IBM Continuous

Speech System) que, trabalhando com a tecnologia do SPHINX, permite introduzir recursos de voz nos aplicativos em linguagem contínua, para um vocabulário de 1000 palavras, independente do locutor.

2.1 Modelos Ocultos de Markov e Alinhamento Dinâmico de Tempo

No modelamento dos sistemas de reconhecimento de voz, duas abordagens se destacam: o alinhamento dinâmico de padrões, que procura associar diretamente medidas físicas das amostras com as referências previamente treinadas, e os modelos ocultos de Markov, que são processos probabilísticos, onde aplicamos um modelo estatístico, e avaliamos a probabilidade da amostra ter sido gerada pela referência treinada.

No primeiro caso, considerando-se o reconhecimento de voz, encontramos uma série de dificuldades para associar uma amostra a uma referência. Entre as primeiras tentativas aparece o trabalho citado de Sambur e Rabiner (apud FLANAGAN, [1976]) que, utilizando grandezas extraídas diretamente do sinal de voz (tais como, energia, taxa de cruzamento por zero, coeficientes de predição linear até 2ª ordem e erro residual), obtiveram certo êxito (94% de acerto).

O alinhamento dinâmico de tempo (DTW) foi desenvolvido com a intenção de contornar o problema dos diferentes ritmos de emissão de uma mesma palavra. Neste processo, representado na fig. 2.1, procura-se ajustar a amostra a uma referência por um procedimento que flexibiliza o eixo tempo. Isto permite que uma emissão seja expandida ou comprimida buscando minimizar o erro introduzido pelo fator "*ritmo de emissão*" na fase de reconhecimento. Este processo é utilizado por ITAKURA [1975] reduzindo significativamente as margens de erro.

A outra abordagem é a estatística na qual aplica-se um modelo aos atributos extraídos do sinal de voz. Criam-se referências que o representam por um conjunto de parâmetros estatísticos. No processo de classificação avaliamos a probabilidade da amostra ter sido gerada por um dos modelos de referência previamente treinados.

A aplicação de um modelo estatístico fornece uma alternativa para contornar o problema de alinhamento entre referência e amostra. Os algorítmos convencionais, como o DTW, procuram ajustar amostra e referência, repetindo ou eliminando segmentos da amostra (ou da referência) com base na semelhança entre os atributos deste segmento e dos adjacentes. Um processo estatístico irá realizar tal alinhamento de modo indireto. A utilização de processos estocásticos compostos, como os modelos ocultos de Markov (HMM), pressupõe a existência

de uma cadeia de estados, como na fig. 2.2, e gera uma função densidade de probabilidade (*fdp*) da transição entre os mesmos. Assim, o que no algoritmo convencional era uma transição de segmentos pela comparação dos seus atributos, passa a ser uma transição de estados, baseado numa avaliação estatística.

RABINER et al. [1983] desenvolveram um trabalho com o objetivo de comparar essas duas abordagens. Utilizando processo estatístico discreto que requer a quantização vetorial (VQ), observaram também o comportamento do sistema convencional associado a essa técnica de redução de dimensionalidade. Num sistema de reconhecimento para os 10 dígitos, independente do locutor, obtiveram as seguintes médias de acerto:

Tabela 2.A Resultados de testes comparativos dos métodos DTW e HMM.

	técnica utilizada		
nº de locutores/amostras por locutor	HMM/VQ	DTW	DTW/VQ
100 locutores / 10 amostras	96,3%	98,5%	96,5%
10 locutores / 200 amostras	92,8%	98,7%	95,5%

Dentre as principais conclusões desse trabalho, no que diz respeito ao que estamos analisando, podemos citar :

- o sistema utilizando HMM/VQ teve excelente performance na tarefa de reconhecimento, sendo que os resultados ligeiramente inferiores se devem ao problema de dimensionamento do conjunto de treinamento;
- o uso de modelos de Markov exige uma fase de treinamento mais complexa, mas isso é realizado apenas uma vez, quando criamos o conjunto de referências; em contrapartida, na fase de reconhecimento, o espaço de armazenamento (para o conjunto de referência) e o tempo de computação são sensivelmente menores.

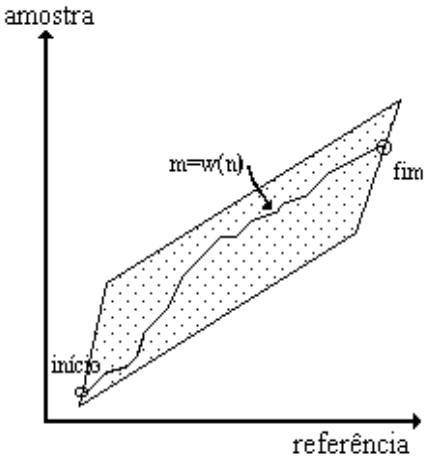


Figura 2.A Alinhamento dinâmico (DTW).

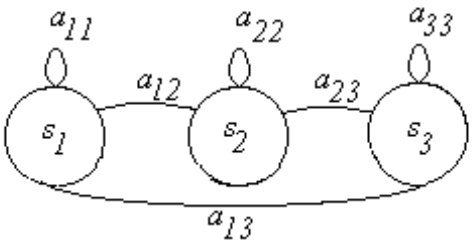


Figura 2.B Representação de um modelo oculto de Markov.

3 Modelos Ocultos de Markov

Uma seqüência de valores s_t de uma variável aleatória discreta S_t caracteriza uma cadeia de Markov se:

$$P(S_{t+1} = s_{t+1} \mid S_t = s_t) = P(S_{t+1} = s_{t+1} \mid S_t = s_t , S_{t-1} = s_{t-1} ,..., S_1 = s_1) \quad (3.1)$$

i.e., a probabilidade do valor da variável S em $t+1$ ser s_{t+1} condicionada ao valor desta variável ao longo de toda a seqüência, é a mesma da condicionada apenas ao último valor.

Podemos generalizar esta definição, igualando o lado direito de (3.1) à probabilidade condicionada aos *n* últimos valores, definindo uma cadeia de Markov de *n*-ésima ordem.

Os modelos ocultos de Markov representam um processo estocástico duplo, com estados internos e símbolos externos. Os estados, bem como as transições entre os mesmos, não são observáveis (são ocultos). Esta parte interna, corresponde a uma cadeia de Markov no sentido clássico, finita, na qual a variável aleatória discreta é o estado. Define-se para esta cadeia uma matriz na qual os elementos representam a função densidade de probabilidade (*fdp*) para as transições entre estados.

Em cada estado ocorre um símbolo de saída, sendo a seqüência desses símbolos a parte observável do processo. Esses símbolos representam outra variável aleatória (daí o processo estocástico duplo) com uma *fdp* própria para cada estado.

Os elementos que caracterizam um modelo oculto de Markov são :

N = número de estados;

M = número de símbolos;

T = número de símbolos observados ou comprimento da seqüência de observação;

S : $\{s_1,...,s_N\}$ = conjunto de estados, incluindo estados inicial e final;

A : $\{a_{ij}\}$ = matriz de *fdp* de transições, onde a_{ij} é a probabilidade de ocorrer uma transição entre os estados *i* e *j*;

B: $\{b_{jk}\}$ = matriz de *fdp* de símbolos, onde b_{jk} é a probabilidade de ocorrer um símbolo *k* quando se atingir o estado *j*.

Π: $\{\pi_j\}$ = matriz de *fdp* dos estados iniciais, onde π_j é a probabilidade de o processo iniciar-se no estado *j*;

O: $\{o_1,...,o_T\}$ = sequência de símbolos observada.

e o modelo estará definido se conhecermos as matrizes **A**, **B**, e Π.

Sendo **A**, **B** e Π matrizes de variáveis de *fdp*, devem ser observadas as seguintes condições :

$a_{ij} > 0, b_{jk} > 0, \pi_j > 0, \quad \forall\, i,j,k;$ (3.2.a)

$\Sigma\, a_{ij} = 1, \quad \forall\, i,j;$ (3.2.b)

$\Sigma\, b_{jk} = 1, \quad \forall\, j,k;$ (3.2.c)

$\Sigma\, \pi_j = 1, \quad \forall\, j;$ (3.2.d)

Definimos os modelos ocultos de Markov de 1ª ordem como aqueles nos quais as características de transição de estado, bem como as *fdp* dos símbolos, dependem apenas do estado presente, sendo independentes dos anteriores.

Desse modo temos :

$$P(S_{t+1} = s_{t+1} \mid S_{1..t} = s_{1..t}) = P(S_{t+1} = s_{t+1} \mid S_t = s_t) \quad (3.3)$$

i.e., a probabilidade de o processo passar ao estado $S=s_{t+1}$ depende apenas do estado presente (s_t), sendo independente dos outros estados anteriores a este, e

$$P(O_t = o_t \mid O_{1..t-1} = o_{1..t-1} , S_{1..t} = s_{1..t}) = P(O_t = o_t \mid S_t = s_t) \quad (3.4)$$

i.e., a probabilidade de observar-se o símbolo O no instante *t*, depende, do mesmo modo, apenas do estado no qual se encontra o sistema.

Podemos diferenciar alguns tipos de modelos conforme as características das transições entre estados. O modelo mais geral permite a transição entre quaisquer estados, sendo ergódico no sentido de que não existe um estado que seja atingido com probabilidade unitária, nem uma periodicidade na seqüência de estados. Podemos, por outro lado, considerar um modelo com restrições às liberdades de transições. Para determinadas aplicações torna-se interessante o modelo denominado *left-to-right*, representado na última seção pela fig. 2.2, onde o processo possue uma direcionalidade, partindo de um estado inicial e dirigindo-se ao final.

No estudo dos modelos ocultos de Markov três problemas se apresentam :

a avaliação - dados um modelo com **A**, **B**, e Π definidos e uma seqüência de observações, determinar qual a probabilidade de que tal modelo tenha gerado tal seqüência.

a decodificação - dados um modelo com **A**, **B**, e Π definidos e uma seqüência de observações, determinar qual a seqüência de estados mais provável.

o aprendizado - dadas uma estrutura de modelo **M** e seqüências de observações (que se supõe geradas pelo modelo), determinar quais os parâmetros de **A**, **B**, e Π que maximizam a probabilidade de que essas seqüências tenham sido geradas pelo modelo.

Vamos, a seguir, analisar esses três problemas.

3.1 A avaliação

O problema pode ser colocado da seguinte forma: temos um modelo definido por **A**, **B**, e Π e vamos calcular a probabilidade de que uma seqüência observada $o_{1..T}$ tenha sido gerada por ele.

A abordagem direta nos levaria a computar todas as possíveis seqüências de estados-símbolos que gerariam tal seqüência $o_{1..T}$ e somá-las. Desse modo teríamos :

$$P(O_{1..t} = o_{1..t}) = \sum_{S=1..t} P(S_{1..t} = s_{1..t}) . P(O_{1..t} = o_{1..t} | S_{1..t} = s_{1..t}) \quad (3.1.1)$$

Tal cálculo direto pode, no entanto, ser simplificado nos processos de Markov de 1ª ordem considerando-se que:

$$P(S_{1..t} = s_{1..t}) = \prod_{\tau=1}^{t} P(S_\tau = s_\tau | S_{\tau-1} = s_{\tau-1}) \quad (3.1.2)$$

e, sendo os símbolos de saída independentes dos estados anteriores :

$$P(O_{1..t} = o_{1..t} | S_{1..t} = s_{1..t}) = \prod_{\tau=1}^{t} P(O_\tau = o_\tau | S_\tau = s_\tau) \quad (3.1.3)$$

Substituíndo os dois últimos termos na 1ª equação :

$$P(O_{1..t} = o_{1..t}) = \sum_{S1..t} \prod_{\tau=1}^{t} P(S_\tau = s_\tau | S_{\tau-1} = s_{\tau-1}) . P(O_\tau = o_\tau | S_\tau = s_\tau) \quad (3.1.4)$$

Essa equação pode ser avaliada diretamente pelas matrizes **A**, **B**, e Π, mas isto requer ainda o cálculo de todos os trajetos possíveis, sendo que o número de cálculos envolvidos varia exponencialmente com T.

No entanto, como as parcelas envolvidas para o cálculo no instante *t* dependem apenas das parcelas de *t-1*, podemos efetuar o cálculo por recursão em t. Vamos definir a variável $\alpha_s(t)$ como a probabilidade *progressiva*, i.e., a probabilidade de o processo estar no estado s, tendo gerado a seqüência parcial $o_{1..t}$, i.e.:

$$\alpha_s(t) = P(O_{1..t} = o_{1..t} | S_t = s_t) \quad (3.1.5)$$

temos assim :

$$\alpha_s(1) = \pi_s . b_{sk|\, k=o_1}, \qquad 1 \le s \le N \quad (3.1.6)$$

e para $t = 1,...,T-1$:

$$\alpha_j(t+1) = \left[\sum_{s=1}^{N} \alpha_s(t) . a_{ij|\, i=s} \right] b_{jk|\, k=o_{t+1}}, \qquad 1 \le s \le N \quad (3.1.7)$$

sendo a probabilidade da seqüência dada por :

$$P(O_{1..T} = o_{1..T}) = \sum_{s=1}^{N} \alpha_s(T) \qquad (3.1.8)$$

A equação (3.1.7) calcula a probabilidade conjunta da primeira observação o_1 para os N possíveis estados. Na equação (3.1.8) temos o somatório das N possibilidades de se alcançar o estado *j* partindo-se do estado *s*, considerando-se o símbolo o_{t+1}.. Sendo $\alpha_s(t)$ a probabilidade de o processo estar em *s* tendo sido observada a seqüência $o_{1..t}$, o produto $\alpha_s(t)a_{ij|\,i=s}$ representa a probabilidade de o processo ter realizado a transição do estado *i* para *j* em *t+1*. A soma dos N possíveis caminhos para alcançarmos o estado *j* fornecerá a probabilidade de estarmos em *j* no instante *t+1*. Feito isto, $\alpha_s(t+1)$ será determinado pelo produto da somatória por $b_{jk|\,k=o_{t+1}}$, i.e., a probabilidade de observarmos em *j* o símbolo $k=o_{t+1}$. Finalmente, a soma das N possíveis trajetórias para o instante *T* (final), fornece a probabilidade da seqüência $o_{1..T}$.

Esse procedimento é conhecido na literatura por "*forward algorithm*" (RABINER, [1983]). Tal procedimento permite reduzir a quantidade de cálculos envolvidos para algo da ordem de N^2T, enquanto o cálculo direto envolve $2TN^T$ operações. O que obtemos é a probabilidade de que uma dada seqüência tenha sido gerada por um modelo **M**: $P(o_{1..T}|\mathbf{M})$. Para obtermos a probabilidade de um modelo a partir de uma dada seqüência utilizamos a regra de Bayes :

$$P(\mathbf{M}|o_{1..T}) = \frac{P(o_{1..T}|\mathbf{M}).P(\mathbf{M})}{P(o_{1..T})} \qquad (3.1.9)$$

Considerando-se $P(o_{1..T})$, a probabilidade de observarmos uma seqüência, como constante, a equação resume-se ao produto $P(o_{1..T}|\mathbf{M}).P(\mathbf{M})$. A primeira parcela é calculada da forma descrita acima. A segunda, $P(\mathbf{M})$, é a probabilidade de ocorrência do modelo, que será definida pela probabilidade de cada modelo dentro do conjunto do sistema.

3.2 A decodificação

Neste caso, estamos interessados em saber qual a seqüência de estados que, com maior probabilidade, gerou determinada observação $o_{1..T}$ para um dado modelo. Um método possível é escolhermos os estados s_t que possuem máxima probabilidade no instante *t*.

Utilizando o algoritmo de VITERBI [1967], vamos definir $\gamma_s(t)$ e $\Psi_s(t)$ como :

$$\gamma_s(1) = \pi_s.b_{sk|\,k=o_1} \qquad 1 \le s \le N \qquad (3.2.1)$$

$$\Psi_s(1) = 0 \qquad (3.2.2)$$

e para $t = 1,...,T-1,\ 1 \le j \le N$:

$$\gamma_j(t+1) = MAX\ [\ \gamma_s(t).a_{ij|\ i=s}\ ].b_{jk|\ k=o_{t+1}} \qquad 1 \le s \le N \qquad (3.2.3)$$

$$\Psi_j(t+1) = argMAX\ [\ \gamma_s(t).a_{ij|\ i=s}\ ] \qquad 1 \le s \le N \qquad (3.2.4)$$

Teremos então os valores finais :

$$P^* = MAX\ [\gamma_s(T)] \qquad 1 \le s \le N \qquad (3.2.5)$$

$$s^* = argMAX\ [\gamma_s\ (T)] \qquad 1 \le s \le N \qquad (3.2.6)$$

e a seqüência de estados ótima para $t = T, T-1, ..., 2$:

$$s_{t-1}^* = \Psi_t(s_t^*) \qquad (3.2.7)$$

Este algorítmo é semelhante ao definido no item anterior para o cálculo da probabilidade progressiva sendo utilizado, no lugar do somatório dos N estados, o valor do estado de máxima probabilidade.

3.3 O aprendizado

Este é o problema mais complexo, uma vez que não existe uma solução analítica. O que se procura é ajustar os parâmetros das matrizes **A**, **B**, e Π no sentido de maximizar as probabilidades das seqüências de observações que se pretende associar ao modelo. Utiliza-se o algoritmo iterativo "Baum-Welch" (ou "*forward-backward*") (Baum, 1972 apud LEVINSON et al., [1983])

De modo análogo ao realizado no ítem 3.1, vamos definir $\beta_s(t)$ a *probabilidade regressiva*, i.e., a probabilidade de o processo estar no estado s, tendo sido observada (agora a partir do instante final) $o_{t..T}$:

$$\beta_s(t) = P(\ O_{t..T} = o_{t..T}\ |\ S_t = s_t\ ) \qquad (3.3.1)$$

temos assim :

$$\beta_s(T) = 1 \qquad (3.3.2)$$

e para $t = T-1,...,1$:

$$\beta_j(t) = \left[\sum_{s=1}^{N} \beta_j(t+1).a_{ij|\ i=s}\right] b_{jk|\ k=o_{t+1}}, \qquad 1 \le j \le N \qquad (3.3.3)$$

Definimos então :

$$\xi_{ij}\ (t) = P\ (\ s_t = i, s_{t+1} = j, |\ o_{1..t}\ ) \qquad (3.3.4)$$

i.e., a probabilidade de uma transição de *i* para *j* em *t*, para uma dada seqüência.

Tal probabilidade pode ser escrita, em termos de $\alpha_s(t)$ e $\beta_s(t)$ como :

$$\xi_{ij}\ (t) = \frac{\alpha_i(t-1).a_{ij}.b_{jk|k=o_t}.\beta_j(t)}{P(O_{1..T} = o_{1..T})} \tag{3.3.5}$$

Na equação acima temos que $\alpha_s\ (t\text{-}1)$ considera a probabilidade de o processo estar no estado *i* em *t-1*, o produto $a_{ij}\ .b_{jk|\ k=o_t}$ considera a transição de *i* para *j* em *t*, observada a saída o_t, e $\beta_j(t)$ computa a probabilidade do restante da seqüência a partir de *j*. O fator do quociente normaliza os valores de ξ_{ij}.

Assim, podemos definir para uma dada seqüência $o_{1..T}$:

$\sum_{t=1}^{T} \xi_{ij}\ (t)$, o número esperado de transições de *i* para *j*, e

$\sum_{t=1}^{T} \sum_{k=1}^{j} \xi_{ij}\ (t)$, o número esperado de transições de *i* para qualquer estado.

Realizamos então um processo de aproximações sucessivas, partindo-se de quaisquer valores iniciais para $\bar{a}_{ij}$ e $\bar{b}_{jk}$ e recalculando-se seus valores pelas expressões :

$$\bar{\pi}_s = \frac{\alpha_s(1).\beta_s(1)}{P(O_{1..T} = o_{1..T})} \tag{3.3.6}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T} \xi_{ij}(t)}{\sum_{t=1}^{T} \sum_{k=1}^{j} \xi_{ik}(t)} \tag{3.3.7}$$

$$\bar{b}_{jk} = \frac{\sum_{t|k=o_t}^{T} \xi_{ij}(t)}{\sum_{t=1}^{T} \sum_{k=1}^{j} \xi_{jk}(t)} \tag{3.3.8}$$

A cada iteração das duas equações acima prova-se que é aumentada a probabilidade da seqüência, a menos que um ponto crítico tenha sido atingido. Neste ponto a re-estimação dos parâmetros fornece os mesmos valores.

3.3.1 Demonstração do algoritmo de Baum-Welch

Vamos apresentar a seguir a demonstração de Baum (apud LEVINSON et al., [1983]) para o algoritmo de re-estimação utilizado. Iniciamos pela apresentação de dois lemas.

Lema 1: sejam u_i , $i=1,...S$ reais positivos e v_i , $i=1,...S$ reais não negativos, tal que $\sum_i v_i > 0$. Da concavidade da função logaritma, temos

$$\ln\left(\frac{\sum v_i}{\sum u_i}\right) = \ln\left[\sum_i \left(\frac{u_i}{\sum_k u_k}\right).\frac{v_i}{u_i}\right]$$

$$\geq \sum_i \frac{u_i}{\sum_k u_k}.\ln\left(\frac{v_i}{u_i}\right)$$

$$= \frac{1}{\sum_k u_k}\left[\sum_i (u_i \ln v_i - v_i \ln u_i)\right] \qquad (3.3.1.1)$$

tomando-se todos os somatórios de1 a S.

Lema 2 : Se $c_i > 0$, $i = 1,..., N$, então, imposta a condição $\sum_i x_i = 1$ a função

$$F(\mathbf{x}) = \sum_i c_i \ln x_i \qquad (3.3.1.2)$$

possue um único máximo global quando

$$x_i = \frac{c_i}{\sum_i c_i} \qquad (3.3.1.3)$$

A prova disso decorre da aplicação do método de Lagrange

$$\frac{\partial}{\partial x_i}\left[F(\mathbf{x}) - \lambda\sum_i x_i\right] = \frac{c_i}{x_i} - \lambda = 0 \qquad (3.3.1.4)$$

Multiplicando por x_i e realizando o somatório sobre i, temos $\lambda = \sum_i c_i$.

Seja, no Lema 1, S o número de estados de determinada seqüência com comprimento T. Para o i-ésimo estado da seqüência, seja

$$u_i = \mathrm{P}(i, \mathrm{O}|\mathbf{M}) \qquad (3.3.1.5)$$

i.e., a probabilidade do estado i, dada a observação O e o modelo **M**, e

$$v_i = \mathrm{P}(i, \mathrm{O}|\overline{\mathbf{M}}) \qquad (3.3.1.6)$$

Temos então

$$\sum_i u_i = \mathrm{P}(\mathrm{O}|\mathbf{M}) = \mathrm{P}(\mathbf{M}) \tag{3.3.1.7}$$

$$\sum_i v_i = \mathrm{P}(\mathrm{O}|\overline{\mathbf{M}}) = \mathrm{P}(\overline{\mathbf{M}}) \tag{3.3.1.8}$$

e, aplicando o Lema 1,

$$\ln \frac{\mathrm{P}(\mathbf{M})}{\mathrm{P}(\overline{\mathbf{M}})} \geq \frac{1}{\mathrm{P}(\mathbf{M})}.\left[\mathrm{Q}(\mathbf{M},\overline{\mathbf{M}}) - \mathrm{Q}(\mathbf{M},\mathbf{M})\right] \tag{3.3.1.9}$$

onde

$$\mathrm{Q}(\mathbf{M},\overline{\mathbf{M}}) = \sum_i u_i \ln v_i \tag{3.3.1.10}$$

Assim, se conseguirmos encontrar um modelo $\overline{\mathbf{M}}$ que faça o lado direito da equação (3.3.1.9) positivo, estaremos aprimorando o modelo **M**. O máximo desse processo está em encontrar um $\overline{\mathbf{M}}$ que torne máximo Q (**M**, $\overline{\mathbf{M}}$).

Sejam então as seqüências de estados $s_1,...,s_T$ e de símbolos observados o_1 , ..., o_T. Então

$$\ln v_s = \ln \mathrm{P}(\mathrm{s},\mathrm{O}|\overline{\mathbf{M}}) = \ln \overline{\pi}_{s_0} + \sum_{t=0}^{T-1} \ln \overline{\mathrm{a}}_{s_t s_{t+1}} + \sum_{t=0}^{T-1} \ln \overline{\mathrm{b}}_{s_{t+1}k|k=o_{t+1}} \tag{3.3.1.11}$$

Substituindo esta equação em (3.3.1.10) e re-agrupando os termos da equação

$$\mathrm{Q}(\mathbf{M},\overline{\mathbf{M}}) = \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij} \ln \overline{\mathrm{a}}_{ij} + \sum_{j=1}^{N}\sum_{k=1}^{M} d_{jk} \ln \overline{\mathrm{b}}_{jk} + \sum_{i=!}^{N} e_i \ln \overline{\pi}_i \tag{3.3.1.12}$$

onde

$$c_{ij} = \sum_{\mathrm{s}=1}^{\mathrm{S}} \mathrm{P}(\mathrm{s},\mathrm{O}|\mathbf{M}).n_{ij}(\mathrm{s}) \tag{3.3.1.13a}$$

$$d_{ik} = \sum_{\mathrm{s}=1}^{\mathrm{S}} \mathrm{P}(\mathrm{s},\mathrm{O}|\mathbf{M}).m_{jk}(\mathrm{s}) \tag{3.3.1.13b}$$

$$e_i = \sum_{\mathrm{s}=1}^{\mathrm{S}} \mathrm{P}(\mathrm{s},\mathrm{O}|\mathbf{M}).r_i(\mathrm{s}) \tag{3.3.1.13c}$$

e para o s-ésimo estado da seqüência temos

n_{ij} (s) = número de transições de q_i para q_j,

m_{jk}(s) = número de vezes que o símbolo o_k foi gerado em q_j, e

r_i (s) = 1 se estivermos no estado inicial, ou

0, caso contrário.

Assim c_{ij}, d_{jk}, e e_i são os valores esperados para n_{ij}, m_{jk}, e r_i respectivamente.

A expressão (3.3.1.12) corresponde à soma de 2N+1 expressões independentes, do tipo das que o Lema 2 maximiza. Assim Q ($\mathbf{M}, \overline{\mathbf{M}}$) pode ser maximizado por

$$\bar{a}_{ij} = \frac{c_{ij}}{\sum_j c_{ij}} \qquad (3.3.1.14a)$$

$$\bar{b}_{jk} = \frac{d_{jk}}{\sum_k d_{jk}} \qquad (3.3.1.14b)$$

$$\bar{\pi}_i = \frac{e_i}{\sum_i e_i} \qquad (3.3.1.14c)$$

que correspondem as fórmulas de re-estimação do algorítmo Baum-Welch.

3.4 Modelos discretos e contínuos

A solução dos três problemas propostos foi desenvolvida considerando os símbolos observados como uma variável discreta, daí derivando a denominação de modelos ocultos de Markov *discretos*. No sistema que implementamos, esses símbolos correspondem a um índice que representa um vetor multi-dimensional. Este vetor, nomeado centróide, é obtido pelo processo de quantização vetorial descrito no capítulo 5. A centróide corresponde ao vetor mais próximo daquele que representa as grandezas físicas do sinal obtidas pela análise por predição linear que apresentamos no próximo capítulo.

A formulação apresentada pode ser estendida ao caso de vetores multi-dimensionais contínuos. Podemos assumir para $P(O_t = o_t \mid S_t = s_t)$ alguma forma de distribuição conhecida, p.ex., gaussiana, estimando diretamente, a partir das medidas físicas observadas, o vetor médio e a martriz de covariância que descrevem esta *fdp*.

Entre as vantagens do modelo contínuo se destacam a possibilidade de tratamento direto dos vetores, sem erros decorrentes da quantização e uma pequena redução do número de parâmetros. Em contrapartida, os modelos contínuos demandam uma carga computacional mais elevada tanto no treinamento como no reconhecimento. Tal demanda pode ser contornada pelo uso, no caso de distribuições gaussianas, de matriz de covariância diagonal. LEE [1991] destaca, no entanto, que estudos de Rabiner et al. demonstraram que o uso dessas matrizes não representa adequadamente certos parâmetros da voz, enquanto Brown demonstrou que a matriz

de covariância completa reduz as taxas de erro em 50% quando comparadas à diagonal mas, neste caso, voltamos ao problema da complexidade computacional.

Diversos trabalhos utilizam um e outro modelo - o contínuo ou o discreto - não havendo consenso sobre qual o mais adequado. LEE [1991] aponta esta falta de consenso nos resultados obtidos por Bahl, Brown e Rabiner et al.

Bahl, comparando os dois sistemas para o reconhecimento de 1000 palavras, obteve taxas de erro de 10,5% para o discreto e 21,9% para o contínuo, assumindo distribuição gaussiana com matriz de covariância diagonal. Brown obteve, do mesmo modo, para o reconhecimento do "E-set" (conjunto de palavras em inglês de difícil reconhecimento em razão da semelhança de sons /e/) taxas de erro menores para o modelo discreto.

Por outro lado, Rabiner et al. observaram redução na taxa de erro de 2,9% para 2,4% a 0,7% com diversos modelos contínuos no reconhecimento de dígitos isolados independente do locutor. Gupta et al. obtiveram, para um vocabulário de 60.000 palavras isoladas, dependente do locutor, uma redução do erro de 31% para 24% ao passar de modelos discretos para contínuos com matriz de covariância completa.

Uma interpretação desses resultados é proposta por Brown (apud LEE[1991]). Nos processos de estimativa de parâmetros por máxima verossimilhança ("maximum likelihood estimation") assumem-se três hipóteses: a) o modelo da distribuição é correto; b) a distribuição é bem comportada; c) o conjunto de amostras é suficientemente grande. Se essas hipóteses não são verificadas, o comportamento do modelo não é previsível. Os modelos discretos, não assumindo uma forma de distribuição a priori, não enfrentam as limitações de a) e b). Sugere ainda que o uso de modelos contínuos exige adoção de distribuições representativas, como a gaussiana com matriz de covariância completa.

Considerando estas observações e a maior simplicidade computacional, optamos em nosso trabalho pelo modelo discreto.

4 Análise do sinal por coeficientes de predição linear

O sinal de voz pode ser modelado por um sistema de tempo discreto com duas fontes: uma tipo trem de pulsos com freqüência variável e outra de ruído aleatório representando, respectivamente, as componentes vocálicas e não vocálicas. Os efeitos de radiação, variações no trato vocálico e da glótis sobre o espectro são aplicados por um filtro digital variante no tempo.

A idéia básica da análise por predição linear é aproximar cada amostra do sinal de voz por uma combinação linear das anteriores. Os parâmetros que minimizam a soma dos quadrados das diferenças entre a amostra real e aquela predita, através da combinação linear das amostras anteriores por eles ponderadas, são os coeficientes de predição linear.

Seja s(t) um sinal contínuo, amostrado numa freqüência $f_s = 1/T$, gerando as amostras $s(nT)$, ou s_n. O sinal s_n será considerado como a saída de um sistema com entrada desconhecida u_n, trem de pulsos ou ruído, de forma que se mantenha a relação

$$s_n = -\sum_{k=1}^{p} a_k s_{n-k} + G\sum_{l=0}^{q} b_l u_{n-l} \tag{4.1}$$

onde a_k, $1 \le k \le p$, b_l , $1 \le l \le q$ e o ganho G são os parâmetros do modelo. Sem perda de generalidade, consideramos b_0=1.

Transformando a equação de diferenças acima para o domínio de freqüências temos a função de transferência

$$H(z) = \frac{S(z)}{U(z)} = G\frac{1+\sum_{l=1}^{q} b_l z^{-l}}{1+\sum_{k=1}^{p} a_k z^{-k}} \tag{4.2}$$

onde

$$S(z) = \sum_{n=-\infty}^{\infty} s_n z^{-n} \tag{4.3}$$

é a transformada z de s_n e $U(z)$, analogamente, é a transformada z de u_n.

O uso deste modelo geral, com pólos e zeros na função de transferência, torna a determinação dos coeficientes a_k e b_l bastante complexa do ponto de vista computacional, implicando na resolução de um sistema com muitas variáveis. Alguns métodos heurísticos sugerem modelos mais simplificados. A função de transferência do trato vocálico, para sons

produzidos com trem de pulsos e sem componentes nasais, não apresenta zeros, o que sugere o uso do modelo só com pólos. Para sons nasais e sons gerados por excitação por ruído, no entanto, podem aparecer zeros. O efeito de um zero numa função de transferência pode, porém, ser obtido aumentando-se o número de pólos, uma vez que

$$\left(1 - az^{-1}\right) \approx \frac{1}{1 + az^{-1} + a^2 z^{-2} + \ldots} \tag{4.4}$$

se |a| < 1, o que é o caso para zeros no círculo unitário.

O modelo apenas com os polos, representado na figura 4.1, terá como função de transferência

$$H(z) = \frac{G}{A(z)} \tag{4.5}$$

onde

$$A(z) = 1 + \sum_{k=1}^{p} a_k z^{-k} \tag{4.6}$$

é chamado *filtro inverso* do sistema.

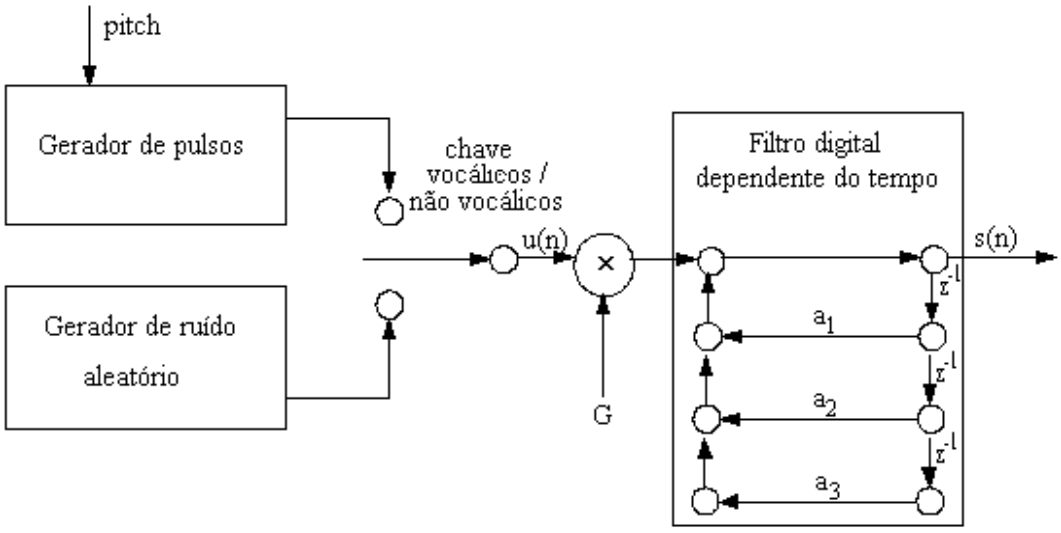


Figura 4.A Modelo para a análise por predição linear, apenas com pólos.

No domínio do tempo, as equações acima significam que o sinal s_n pode ser obtido pela combinação linear de valores passados desse sinal e alguma entrada u_n.

$$s_n = -\sum_{k=1}^{p} a_k s_{n-k} + G u_n \tag{4.7}$$

desde que se conheçam os *p* parâmetros a_k e o ganho *G*.

Podemos assumir que o sinal de voz é quase-estacionário em pequenos intervalos. Se a entrada u_n é desconhecida, o sinal s_n só pode ser estimado pela soma ponderada das amostras passadas, i.e.,

$$\hat{s}_n = -\sum_{k=1}^{p} a_k s_{n-k} \tag{4.8}$$

Os parâmetros do preditor linear serão calculados sobre esses intervalos, minimizando o erro quadrático médio do preditor.

$$E = \sum_m e_n^2 = \sum_m \left(s_n - \hat{s}_n\right)^2 = \sum_m \left(s_n + \sum_{k=1}^{p} a_k s_{n-k} \right)^2 \tag{4.9}$$

O somatório pode ser realizado sobre um intervalo finito, conduzindo ao método de covariâncias, ou sobre um intervalo infinito, conhecido como método das autocorrelações. Nesse último, que iremos analisar, o sinal é considerado nulo fora do intervalo de análise.

Assumir essa condição implica em elevado erro no início do intervalo, quando tentaremos predizer valores não nulos a partir de entradas que foram fixadas nulas, e no final do intervalo, quando tentaremos prever saídas fixadas nulas por amostras não nulas. Por essa razão, aplicamos ao sínal uma envoltória, ou "janela", que aproxime de zero os valores das extremidades do intervalo.

Tipicamente, utilizamos a função de Hamming:

$$w(n) = 0{,}54 + 0{,}46\cos\left(\frac{2n}{N-1}\right) \qquad \text{se } 0 \le n \le N\text{-}1, \text{ e}$$

$$= 0 \qquad \text{caso contrário.} \tag{4.10}$$

transformando o sinal s_n em

$$s'_n = s_n \,.\, w_n \tag{4.11}$$

Apresentamos a seguir o método das autocorrelações e os algorítmos de Durbin e de LeRoux e Gueguen para a solução das equações envolvidas.

4.1 Cálculo dos parâmetros - método das autocorrelações

Seja o sistema da fig. 4.1. com a entrada u_n totalmente desconhecida. Neste caso, o sinal s_n só pode ser estimado pela soma ponderada de amostras passadas.

Seja $\hat{s}_n$ essa aproximação, isto é

$$\hat{s}_n = -\sum_{k=1}^{p} a_k s_{n-k} \tag{4.1.1}$$

Assim, o erro entre o valor real s_n e o preditor $\hat{s}_n$ será

$$e_n = s_n - \hat{s}_n = s_n + \sum_{k=1}^{p} a_k s_{n-k} \tag{4.1.2}$$

Realizando-se o somatório sobre um intervalo infinito temos o erro quadrático médio total

$$E = \sum_{n=-\infty}^{\infty} e_n^2 = \sum_{n=-\infty}^{\infty} \left(s_n + \sum_{k=1}^{p} a_k s_{n-k} \right)^2 \tag{4.1.3}$$

A minimização será obtida fazendo-se as derivadas parciais

$$\frac{\partial E}{\partial a_i} = 0, \qquad 1 \le i \le p \tag{4.1.4}$$

A partir das duas equações acima obtém-se

$$\sum_{k=1}^{p} a_k \sum_{n=-\infty}^{\infty} s_{n-k} s_{n-i} = -\sum_{n=-\infty}^{\infty} s_n s_{n-i}, \quad 1 \le i \le p \tag{4.1.5}$$

A resolução do sistema de equações (4.1.5) fornece os valores a_k que minimizam *E* em (4.1.3).

Considerando-se essas duas equações, obtemos para o cálculo do erro quadrático médio total mínimo a expressão

$$E_p = \sum_{n=-\infty}^{\infty} s_n^2 + \sum_{k=1}^{p} a_k \sum_{n=-\infty}^{\infty} s_n s_{n-k} \tag{4.1.6}$$

Definindo-se a função de autocorrelação do sinal s_n

$$R(i) = \sum_{n=-\infty}^{\infty} s_n s_{n+i} \tag{4.1.7}$$

podemos reduzir as expressões (4.1.5) e (4.1.6) a

$$\sum_{k=1}^{p} a_k R(i-k) = -R(i), \qquad 1 \le i \le p \tag{4.1.8}$$

$$E_p = R(0) + \sum_{k=1}^{p} a_k R(k) \tag{4.1.9}$$

Note-se que a função *R(i)* é par, isto é

$$R(-i) = R(i) \tag{4.1.10}$$

Considerando-se isto, a expansão de (4.1.8) fornece

$$\begin{bmatrix} R(0) & R(1) & R(2) & \cdots & R(p-1) \\ R(1) & R(0) & R(1) & \cdots & R(p-2) \\ R(2) & R(1) & R(0) & \cdots & R(p-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R(p-1) & R(p-2) & R(p-3) & \cdots & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{bmatrix} = - \begin{bmatrix} R(1) \\ R(2) \\ R(3) \\ \vdots \\ R(p) \end{bmatrix} \tag{4.1.11}$$

que é uma matriz de Toeplitz, simétrica e com os elementos de cada diagonal iguais. A divisão dos elementos dessa matriz por uma constante não altera a igualdade em (4.1.11). Escolhendo-se a constante $R(0)$ podemos trabalhar com o coeficiente de autocorrelação normalizado

$$r(i) = \frac{R(i)}{R(0)} \tag{4.1.12}$$

Determinamos desse modo os coeficientes $\hat{a}_k$ da equação (4.7). Resta determinar o ganho G que pode ser obtido igualando-se a energia das amostras do sinal analisado com a energia das amostras preditas.

Mostra-se (MAKHOUL, [1975]) que o ganho é dado por

$$G^2 = E_p = R(0) + \sum_{k=1}^{p} a_k R(k) \tag{4.1.13}$$

4.1.1 Algoritmo de Durbin

O cálculo dos coeficientes a_k, $1 \le k \le p$, envolve a solução do sistema de p equações lineares a p incógnitas descrito por (4.1.11). O fato de a matriz resultante ser de Toeplitz permite utilizar métodos eficientes para a solução. O algoritmo de Durbin resolve de forma recursiva através do seguinte conjunto de equações :

$E_0 = R(0)$ (4.1.1.1)

$$k_i = -\frac{R(i) + \sum_{j=1}^{i-1} a_j^{(i-1)} R(i-j)}{E_{i-1}} \tag{4.1.1.2}$$

$$a_i^{(i)} = k_i \tag{4.1.13}$$

$$a_j^{(i)} = a_j^{(i-1)} + k_i a_{(i-j)}^{(i-1)} \qquad 1 \le j \le i-1 \tag{4.1.1.4}$$

$$E_i = (1 - k_i^2) E_{i-1} \tag{4.1.1.5}$$

As equações acima são resolvidas recursivamente para *i = 1, 2,..., p* e a solução final é dada por:

$$a_j = a_j^{(p)}, \qquad 1 \le j \le p \qquad (4.1.1.6)$$

Observe-se que, para se calcular os coeficientes de predição de ordem *p*, é necessário calcularmos os coeficientes de predição de todos os preditores de ordem menor que *p*.

As variáveis intermediárias k_i são chamadas de coeficientes de reflexão, ou PARCOR (coeficientes de correlação parcial). Esses parâmetros caracterizam, como os a_k, de forma única o preditor, apresentando a vantagem de oferecer melhores características de quantização e de interpolação. O termo "*coeficientes de reflexão*" deve-se a uma interpretação física que surge quando se modela o trato vocálico por uma sucessão infinita de tubos de diâmetros variáveis.

4.1.2 Algoritmo de Le Roux e Gueguen

LeRoux e Gueguen (LE ROUX et al., [1977]) desenvolveram o seguinte conjunto de equações utilizando os coeficientes autocorrelação normalizados $r(i) = e_i^0$:

$$e_i^{h+1} = e_i^h + k_{h+1} + e_{h+1-i}^h \qquad (4.2.1.1)$$

$$k_{h+1} = -\frac{e_{h+1}^h}{e_0^h} \qquad (4.2.1.2)$$

$$e_0^{h+1} = e_0^h\left(1 - k_{h+1}^2\right) \qquad (4.2.1.3)$$

que fornecem a solução recursiva para as variáveis k_m.

Os coeficientes de predição a_k podem ser obtidos a partir dos coeficientes de reflexão através das equações recursivas:

$$a_i^i = k_i \qquad (4.2.1.4)$$

$$a_j^i = a_j^{i-1} + k_i a_{i-j}^{i-1} \qquad 1 \le j \le i\text{-}1 \qquad (4.2.1.5)$$

$$a_j = a_j^p \qquad 1 \le j \le p \qquad (4.2.1.6)$$

Pode-se mostrar que o erro normalizado para o preditor de ordem *h* será

$$V_h = \prod_h^{i-1}\left(1 - k_i^2\right) \qquad (4.2.1.7)$$

MARKEL e GRAY [1973] demonstraram que

$$|k_m| < 1 \qquad (4.2.1.8)$$

o que garante a estabilidade do filtro modelado. Em função desse resultado, vê-se também que o erro normalizado V_h decresce com a ordem *h* do filtro. No método introduzido por LE ROUX, et al. [1977], todas as variáveis intermediárias e_h tem valores entre -1 e 1 desde que se utilize os coeficientes de autocorrelação normalizados, o que torna este método bastante conveniente para implementação em processadores de ponto fixo.

5 Quantização vetorial

Nos modelos ocultos de Markov discretos, os parâmetros que os individualizam são gerados a partir de seqüências de símbolos. A análise do sinal por coeficientes de predição linear descreve o sinal num espaço multi-dimensional contínuo, sendo necessária uma redução desse espaço a valores discretos. A quantização vetorial (Vector Quantization - VQ) (LINDE, [1980], GRAY, [1984] e MAKHOUL, [1985]) realiza essa redução de dados, criando partições neste espaço contínuo e associando a cada um desses vetores um símbolo que o represente à custa de alguma distorção.

Um sistema de quantização completo é definido por um codificador que associa a cada vetor $\mathbf{x}$ de entrada um símbolo $S(\mathbf{x})$ e por um decodificador que recupera o sinal codificado. Para nossa aplicação iremos nos ater apenas à codificação, uma vez que não nos interessa a qualidade do sinal decodificado, mas apenas a qualidade da informação codificada, i.e., até que ponto a redução de dados preserva a informação necessária ao reconhecimento do sinal original. Nesse sentido, o projeto do quantizador deve considerar as características do sistema de reconhecimento como um todo pois, se estivermos implementando, por exemplo, um sistema de vocabulário muito restrito, podemos utilizar um sistema de quantização relativamente simples, que seja apenas suficiente para discernir as palavras desse vocabulário.

O codificador é um sistema que possui um conjunto de V vetores protótipos e utiliza uma medida de distorção. Dada uma distribuição de vetores, iremos agrupar os vetores em sub-espaços vetoriais procurando minimizar a média das distorções entre esses vetores e os centróides, ou centros de gravidade $\hat{\mathbf{x}}$ dos sub-espaços. Um bom sistema de quantização deverá apresentar uma distorção média pequena. Em 1957, S. Lloyd (apud LINDE, [1980]) propôs dois métodos para a implementação de um sistema ótimo de quantização. Em um dos métodos o sistema se desenvolvia de modo formal, com base em derivadas, supondo distribuições conhecidas e contínuas, e utilizava um espaço de até duas dimensões. O outro método proposto, que o conduziu aos mesmos resultados para os casos que foram estudados, superava as dificuldades da abordagem variacional, conduzindo a um algoritmo eficiente para o desenvolvimento de sistemas de quantização que é, com algumas adaptações, o apresentado na seção 5.2.

Quando não se conhece a distribuição, podem ser utilizadas seqüências infinitas de treinamento, tomando-se como média das distorções o limite definido por

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} d(\mathbf{x}_i, \hat{\mathbf{x}}_i), \tag{5.1}$$

desde que tal limite exista. Se o processo for ergódico e estacionário esse limite existe com probabilidade igual a um e representa a expectativa $\mathbf{E}(d(\mathbf{x}_i, \hat{\mathbf{x}}_i))$ Assim, para um processo cuja distribuição não é conhecida, é razoável utilizarmos uma seqüência de treinamento suficientemente longa e admitirmos, se o processo for ergódico e estacionário, que tal seqüência representa da forma adequada as amostras futuras do processo.

Assim, do que foi exposto, concluímos que um projeto de quantizador para um sistema de reconhecimento de voz, deve considerar:

- o número de particões (sub-espaços) suficiente para preservar as informações do sinal necessárias ao reconhecimento;
- a dimensão do conjunto de treinamento, que deve ser representativo do universo de amostras do processo;
- a medida de distorção a ser utilizada.

A quantização vetorial pode ser aplicada a vetores com componentes que não tenham a mesma natureza como, por exemplo, a energia e os coeficientes de reflexão.[2] Neste caso, deve-se tomar cuidado no projeto, para evitar que a medida de distorção global seja afetada demasiadamente por uma das grandezas em detrimento das demais. A adequação da variabilidade dos diversos componentes pode ser feita por transformações ou pela utilização de medidas de distorção diferenciadas.

5.1 Medidas de distorção

A distorção, ou distância, é o erro causado por representarmos o vetor **x** pelo vetor **y,** centróide da partição à qual o queremos associar. Podemos defini-la como um operador d(**x**,**y**), **x** e **y** $\in \mathbf{R}^n$ que deve possuir as seguintes propriedades:

a) $d(\mathbf{x},\mathbf{y}) = d(\mathbf{y},\mathbf{x})$ (5.1.1)

b) $d(\mathbf{x},\mathbf{y}) > 0$, se $\mathbf{x} \neq \mathbf{y}$ (5.1.2)

c) $d(\mathbf{x},\mathbf{x}) = 0$ (5.1.3)

[2] Em sistemas de transmissão a energia pode ser quantizada e transmitida separadamente. Para o uso em sistemas baseados em HMM discretos é necessário que cada segmento da fala seja representado por um único símbolo.

A primeira impõe a simetria, de modo que a a distorção seja a mesma se tomada de um ou outro ponto, e as duas outras impõem que seja positiva e nula quando **y**=**x**. Além disso, se valer a propriedade:

d) $d(\mathbf{x},\mathbf{y}) \leq d(\mathbf{x},\mathbf{z}) + d(\mathbf{z},\mathbf{y})$ (5.1.4)

então $d(\mathbf{x},\mathbf{y})$ é métrica.

A medida de distorção mais usual é a euclidiana, ou erro quadrático médio, definido por:

$$d_2(\mathbf{x},\mathbf{y}) = \frac{1}{N}(\mathbf{x}-\mathbf{y})(\mathbf{x}-\mathbf{y})' = \frac{1}{N}\sum_{k=1}^{N}(x_k - y_k)^2 \quad (5.1.5)$$

Mais genericamente define-se a medida L_r - norma, ou de Minkowski ordem *r*:

$$d_r(\mathbf{x},\mathbf{y}) = \frac{1}{N}\sum_{k=1}^{N}(x_k - y_k)^r \quad (5.1.6)$$

A expressão acima corresponde à euclidiana para *r*=2. Outros valores utilizados tipicamente para *r* são 1 (“distância módulo”) e ∞.

Tais distâncias tratam do mesmo modo todos os componentes dos vetores. A distância ponderada, ou de Pearson, procura adequar a variabilidade de cada componente introduzindo um fator $1/s_j^2$, onde s_j é a variância da j-ésima variável:

$$d_p(\mathbf{x},\mathbf{y}) = \frac{1}{N}\sum_{k=1}^{N}\left(\frac{1}{s_j^2}\right)(x_k - y_k)^2 \quad (5.1.7)$$

A distância de Mahalanobis pode ser considerada uma generalização da ponderada:

$$d_M(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})'Q^{-1}(\mathbf{x}-\mathbf{y}) \quad (5.1.8)$$

onde *Q* é a matriz de covariância dos vetores **x** e **y**. Desse modo, considera não só as variações intra-componentes mas também as correlações inter-componentes. A desvantagem do uso dessas medidas é a sobrecarga computacional envolvida.

Para o caso particular de coeficientes de predição linear, a distorção entre **x** e **y** pode ser vista como a distorção entre os parâmetros de dois modelos de filtros inversos. Esta medida, desenvolvida a partir da proposta inicial de ITAKURA [1975] tem a forma:

$$d_I(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})'R_x(\mathbf{x})(\mathbf{x}-\mathbf{y}) \quad (5.1.9)$$

onde $R_x(\mathbf{x})$ é uma matriz de autocorrelação normalizada (NxN, simétrica, definida positiva) cujos coeficientes são os obtidos no cálculo dos coeficientes de predição linear. Essa matriz pondera as diferenças por valores dependentes do vetor **x**. A aparente complexidade computacional é contornada tanto pelo fato de que os coeficientes são subproduto da análise

por LPC, como por algoritmos que reduzem o produto matricial ao escalar. Tal medida, nomeada como razão de verossimilhança, não é métrica nem mesmo distância nos termos da definição apresentada anteriormente ($d_I(\mathbf{x},\mathbf{y}) \neq d_I(\mathbf{y},\mathbf{x})$) e a dependência de **x** com **y** é bastante complexa.

*

Os sistemas de reconhecimento vêm utilizando uma grande variedade de medidas de distorção, não havendo uma conclusão definitiva sobre qual a mais adequada. Um estudo especialmente voltado para esta questão (RABINER et al., [1985]) demonstra bastante semelhança no desempenho de sistemas de reconhecimento utilizando diversas distâncias. Observa-se ainda nesse trabalho que as taxas de erro dependem de forma muito mais acentuada do número de classes do quantizador do que das diversas medidas de distância utilizadas.

Em nosso sistema utilizamos o erro quadrático médio, calculado sobre os coeficientes de correlação. A simplicidade matemática e computacional viabilizaram a realização de diversos testes relacionados a outros aspectos mais relevantes no estudo dos sistemas baseados em HMM, tal como o número de repetições das palavras no treinamento, ou mais significativos na formação da taxa de erro, como o número de classes do quantizador. Realizamos também um teste incorporando a energia ao vetor dos coeficientes de correlação, utilizando uma distorção ponderada (seção 7.3.2.2.2- “a”).

5.2 Algoritmo

No processo de quantização vetorial devemos encontrar um conjunto de vetores protótipos, denominados centróides, que, a partir de um conjunto de vetores de treinamento, minimize a distorção média. Partimos então da premissa de que esses vetores de treinamento representam a forma da distribuição das amostras no espaço considerado.

O algoritmo que apresentamos é baseado nos trabalhos de LINDE [1980], tendo como idéia básica a realização de sucessivas bi-divisões do espaço multidimensional, determinando-se dois novos sub-espaços. Para cada sub-espaço assim gerado, arbitramos centróides eqüidistantes e reclassificamos os vetores do conjunto de treinamento.

As etapas desse algoritmo, representado no diagrama da figura 5.1, são :

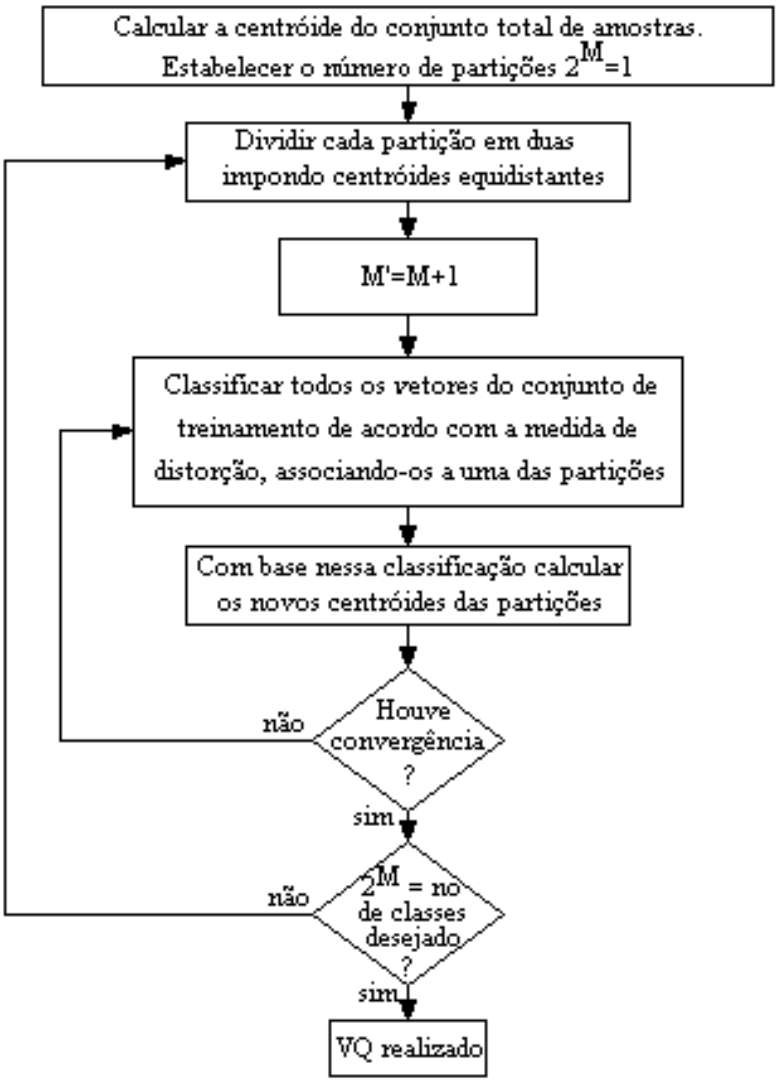


Figura 5.A Diagrama de blocos do quantizador.

1. Calcula-se o centróide do conjunto de vetores de treinamento. Estabelece-se o número de partições M=1;

2. Divide-se cada partição em dois sub-espaços, impondo-se, para os sub-espaços gerados, centróides eqüidistantes àquele do espaço original;

3. Classificam-se todos os vetores de treinamento de acordo com os novos centróides, de acordo com a mínima distorção (“nearest neighbour”);

4. Calcula-se, a partir dessa reclassificação, o novo centróide de cada sub-espaço;

5. Se não for verificado algum critério de convergência entre o cálculo dessa nova centróide e o da anterior retoma-se 3.

6. Se não alcançado o número de classes desejado, retoma-se 2.

Os $V=2^M$ centróides determinados pelo algoritmo serão os vetores-protótipos que classificarão os vetores, associando-se a cada vetor um símbolo discreto.

5.3 Medidas de qualidade do quantizador

Um bom sistema de quantização deve apresentar, conforme já dito, uma baixa distorção média entre os vetores pertencentes ao agrupamento e seu centróide. Essa distorção interna do agrupamento é denominada distorção intra-grupo, e podemos defini-la como

$$D = \frac{1}{2^{m_j}} \sum_{2^{m_j}} d(\mathbf{x}_i, \mathbf{x}_{C_j}) \tag{5.3.1}$$

Além disso, é interessante que os agrupamentos estejam relativamente bem separados. Definimos, com base na mesma medida de distorção, uma distorção inter-grupos

$$s = \frac{1}{2^M} \sum_{i=1}^{M} \left(\frac{1}{2^{M-1}} \right) \sum_{j=1}^{M} d(\mathbf{x}_i, \mathbf{x}_j) \tag{5.3.2}$$

e vamos considerar como meta de otimização do quantizador obter uma alta razão s/D.

Devemos observar também a distribuição dos vetores pelos agrupamentos (cardinalidade). Uma distribuição muito irregular dos vetores concentrando, por exemplo, grande número de vetores num único agrupamento, indica um sistema de quantização fraco, que deve estar aglutinando informações significativas em uma única classe (partição).

Temos assim medidas de qualidade que podem ser monitoradas em função do número de classes, da dimensão do conjunto de treinamento, e da medida de distorção utilizada.

No entanto, é importante lembrar que o quantizador é um processo intermediário, que serve ao sistema de reconhecimento. Assim, deve-se reconsiderar os resultados fornecidos pela avaliação isolada do quantizador em vista do projeto do sistema como um todo.

6 Reconhecimento de voz

O sistema de reconhecimento de voz independente do locutor que vamos apresentar trabalha com palavras isoladas e vocabulário restrito. Este sistema foi implementado em microcomputador IBM-PC, com placa de processamento digital de sinais, com processador M56001. Os algoritmos foram implementados em linguagem PASCAL.

O sinal de voz foi obtido em ambiente de baixo ruído e gravado em fita cassete. Este sinal foi amostrado em uma freqüência de 8 KHz, segmentado em períodos de 20 ms com aplicação da janela de Hamming e a análise por coeficientes de predição linear (de 10ª ordem) foi realizada na própria placa DSP com processador M56001, utilizando o método LeRoux e Gueguen, extraindo-se os coeficientes de correlação parcial. A cada segmento da amostra corresponde então um vetor decadimensional. Reduzimos a dimensão dos dados através da quantização vetorial. Esta fase do processo corresponde ao que denominamos na fig. 1.1 de *extração de atributos*.

O sistema de reconhecimento envolve, como já dissemos, uma etapa anterior de treinamento, quando criamos as referências para o sistema, e o reconhecimento, quando identificaremos uma entrada do sistema com uma dessas referências. Tanto a fase de extração de atributos como a de treinamento/reconhecimento apresentam particularidades que iremos apresentar a seguir.

6.1 Determinação do início e fim das palavras

Um dos problemas que apresenta a análise do sinal extraído da voz é a identificação do ponto preciso de início e término de uma palavra de uma dada elocução. Em princípio, poderíamos pensar em utilizar a variação da energia do sinal como uma indicação de limites. No entanto, a dificuldade de determinar esses limites está justamente no fato de que certas palavras, particularmente aquelas com fricativas (f,s,v) iniciais ou finais, plosivas iniciais (p,t), e nasais finais apresentam uma energia muito fraca nesses pontos, sendo necessária outra fonte de informação para detectá-las.

O algoritmo desenvolvido por RABINER et al. [1975] utiliza, como fonte adicional de informação, a taxa de cruzamentos por zero (ZCR) do sinal. Essa taxa é definida como o número de vezes que o sinal torna-se positivo ou negativo em um dado intervalo, em nosso caso

o corresponde ao período do segmento. Isso permite a percepção de informações oriundas de sons não vocálicos de alta freqüência e baixa energia.

Apesar de esta medida ser bastante sensível a offset e ruído de linha (60 Hz), fornece uma boa indicação da presença de não vocálicos. Além disso, é uma grandeza fácil e rápida de ser medida.

No algoritmo proposto, assume-se que durante os primeiros 100 ms do sinal não exista a presença de voz. Nesse intervalo são calculadas as estatísticas de silêncio (relativas ao ambiente). Define-se uma banda passante, no caso de 100 a 4000 Hz. Essas estatísticas incluem a medida de energia média, ZCR médio, e desvio padrão de ambas. Fixa-se então como limiar de ZCR para não vocálicos, o valor mínimo entre um limiar (determinado experimentalmente, ZCR_{min}=50, p.ex.) e o valor médio acrescido de duas vezes o desvio padrão.

$$IZCT = \min(ZCR_{min} , IZC+ 2\sigma IZC) \quad (6.1.1)$$

Calcula-se a energia para os intervalos das amostras. A energia de pico IMX, e a de silêncio IMN, são usadas para estabelecer os limiares ITL e ITU de acordo com as seguintes expressões :

$$I_1 = 0,03 . (IMX-IMN) + IMN \quad (6.1.2)$$

$$I_2 = 4 . IMN \quad (6.1.3)$$

$$ITL = \min (I_1 , I_2) \quad (6.1.4)$$

$$ITU = 5 . ITL \quad (6.1.5)$$

isto é, I_1 é 3% do valor de pico da energia (IMX), ajustado ao valor da energia de silêncio (IMN), enquanto I_2, é 4 vezes a energia de silêncio. O limiar inferior (ITL) é o mínimo entre esses valores e o superior (ITU) cinco vezes o limiar inferior.

O algoritmo procura o primeiro ponto da amostra onde foi ultrapassado o limiar inferior. Esse ponto é considerado temporariamente como o início da palavra. Se nos intervalos seguintes o nível de energia cair abaixo de ITL antes de atingir ITU desconsideramos esse ponto e passamos ao próximo ponto onde a energia ultrapassar ITL. Para o final da palavra o procedimento é semelhante. O primeiro ponto onde o nível cai abaixo de ITL é considerado provisoriamente o final da palavra. Se nos intervalos seguintes o sinal ultrapassar novamente ITU desconsideramos este ponto e prosseguimos a busca até encontrar um ponto onde estas condições sejam satisfeitas.

Até aqui utilizamos apenas a energia do sinal. Sendo as hipóteses desse procedimento, para determinação de início e fim de palavra com base apenas na energia, bastante conservadoras, podemos supor que os pontos que realmente correspondem ao início e fim da palavra estejam fora desse intervalo determinado.

Observamos então o comportamento da taxa de cruzamentos por zero (ZCR) em um intervalo imediatamente anterior (para o início) ou posterior (para o término). No caso, analisamos os 260 ms adjacentes aos intervalos. Se o número de vezes que ZCR excedeu IZCT for superior a três, consideramos que existe a presença de sons não vocálicos nos limites da palavra e transferimos o ponto limite para o primeiro (para o início) ou o último (para o término) ponto onde esse limiar foi ultrapassado.

Tal procedimento não permite determinar com absoluta certeza os pontos de início e término de uma palavra (uma fricativa fraca, por exemplo, pode ser desprezada). No entanto, fornece uma informação confiável da presença de sons não vocálicos (de baixa energia) nos limites. Para a nossa aplicação específica, aquilo que interessa é selecionar da amostra bruta a parte que contém informação relevante para a caracterização da palavra. Neste sentido, este algoritmo se apresenta como simples e eficaz.

6.2 Uso dos modelos ocultos de Markov no reconhecimento de palavras isoladas

Na utilização dos modelos ocultos de Markov para o reconhecimento de palavras isoladas vamos assumir que cada palavra é representada por uma seqüência de símbolos $\mathbf{O}=o_1$, o_2 , ..., o_T. Esses símbolos, extraídos de um livro de código finito, correspondem aos coeficientes de correlação de cada segmento da amostra obtidos pelo processo de quantização vetorial.

Algumas particularidades devem ser examinadas tanto no que diz respeito à estrutura do modelo como a problemas práticos de implementação dos algoritmos.

6.2.1 Estrutura dos modelos ocultos de Markov

Nos sistemas de reconhecimento de palavras encontrados na literatura, tem sido utilizados os modelos uni-direcionais (ou "*left-to-right*") que apresentam as seguintes características:

- a primeira observação é produzida quando o sistema está em um estado determinado denominado inicial;
- a última observação é produzida quando o sistema está em um estado determinado denominado final;
- uma vez que o processo deixa um estado, este não pode mais ser alcançado.

Isso faz com que a matriz de transição **A** tenha os elementos $a_{ij} = 0$ se $i<j$. Além disso, podemos restringir ainda mais o modelo não permitindo que ocorram transições entre mais de um ou dois estados consecutivos. Nesse caso temos $a_{ij} = 0$ se $i>j+1$ ou $i>j+2$. A figura 6.1 apresenta um processo de Markov tipo "*left-to-right*" e as respectivas matrizes.

A escolha do modelo unidirecional parte do pressuposto de que uma palavra é uma emissão com começo e fim, percorrendo estados intermediários que eventualmente podem ser "saltados".

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{k1} \\ b_{21} & b_{22} & \cdots & b_{k2} \\ b_{31} & b_{32} & \cdots & b_{k3} \\ b_{41} & b_{42} & \cdots & b_{k4} \end{bmatrix}, \Pi = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

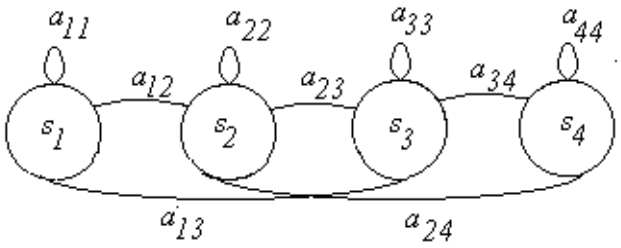


Figura 6.A Modelo left-to-right e respectivas matrizes.

No que diz respeito às restrições do modelo, destacamos uma observação de RABINER et al. [1983] com relação ao número de estados da estrutura que diz "parecer não haver nenhum método teórico para escolhermos o número de estados para modelarmos uma palavra, *uma vez que os estados não precisam estar relacionados diretamente com algum fenômeno físico observável*" (grifamos). Partindo da constatação de que estados e fenômenos físicos não estão relacionados, poderia parecer um tanto arbitrária a escolha de um modelo direcional.

Em vista de tais questões, Rabiner et al. realizaram no trabalho citado uma série de testes em um sistema com características semelhantes a este que estamos implementando, variando tanto a estrutura do modelo como o número de estados. Dos resultados obtidos concluiram que :

- estruturas com restrições na matriz de transições apresentaram resultados melhores que aquelas sem restrições;
- o número de estados do modelo deve ser da ordem de cinco; estados adicionais não contribuem significativamente para a performance do sistema, elevando por outro lado a carga computacional. Um número menor de estados compromete a performance do sistema;

Desse modo, tomamos como base para a implementação de nosso sistema o modelo "*left-to-right*" com cinco estados permitindo a transição apenas se $i=j$ ou $i>j+1$. Tal estrutura apresenta também a vantagem de redução no espaço de armazenamento.[3]

Os sistemas que trabalham com vocabulário extenso utilizam estruturas nas quais os HMMs representam fonemas ou formações de fonemas, devido à evidente impraticabilidade de tratar o vocabulário palavra por palavra (LEE [1991]). Em nosso caso, de vocabulário restrito, utilizamos como unidade a palavra, eliminando uma complexidade desnecessária.

6.2.2 Inicialização das matrizes A e B

No procedimento de treinamento, utilizando a algoritmo "Baum-Welch", temos que atribuir valores iniciais aos parâmetros a_{ij} e b_{jk}. Desse modo, atribuímos valores aleatórios entre zero e um a esses parâmetros obedecendo as condições impostas em (3.1.b) e (3.1.c), i.e., que a soma das linhas da matriz seja unitária. Isto se realiza normalizando cada linha das matrizes, por um fator que corresponde a soma dos valores da linha,

$$a'_{ij} = \frac{a_{ij}}{\sum_{i=1}^{N} a_{ij}} \qquad 1 \le j \le N \qquad (6.2.2.1)$$

$$b'_{ij} = \frac{b_{ij}}{\sum_{i=1}^{N} b_{ij}} \qquad 1 \le j \le N \qquad (6.2.2.2)$$

[3] Considerando-se que os elementos aij são nulos para $i < j$ e $i > j+1$ temos que a matriz $\mathbf{A}_{NxN}$ pode ser representada, uma vez feitas as alterações necessárias nos algoritmos, por uma matriz $\mathbf{A}_{nx2}$. O elemento a_{55} tem sempre o valor 1. No caso de N=5, temos uma redução de dados melhor que 3:1.

Apesar de o algoritmo garantir que o valor da probabilidade P da seqüência (ou conjunto de seqüências) tende a um ponto crítico, tal ponto é apenas um máximo local. Desse modo, diferentes valores de inicialização podem conduzir a diferentes valores para P.

RABINER et al. [1983] realizaram também uma série de testes para verificar a influência dos pontos de partida do algoritmo. No que diz respeito aos valores de inicialização das matrizes, os resultados mostraram que a performance do sistema de reconhecimento para 10 diferentes pontos de partida apresentou taxas de erro médias de 4% com variações de +/- 1%. Além disso, a média dos parâmetros desses 10 modelos apresentou taxa de erro de 5%, enquanto que a utilização de todos os modelos combinados, apresentou erro de cerca de 3%, comparável à performance do melhor dos 10 modelos.

Com base nessas observações, podemos considerar como desprezível o efeito dos pontos de partida das iterações do algoritmo.

6.2.3 Problemas numéricos: underflow

Observando-se as definições de $\alpha(t)$ e $\beta(t)$ pelas equações (3.1.5) e (3.3.1) verificamos facilmente um problema numérico que ocorre na implementação do algoritmo. Esses valores, que são sempre positivos e menores que um, causariam rapidamente underflow na maioria dos processadores quando calculados recursivamente em *t*. Para contornar esse problema utilizamos um *fator de escala*.

O método mais utilizado consiste em dividir todos os valores pela soma dos N valores $\alpha_i(t)$ imediatamente após estes terem sido calculados. Outros métodos tem sido utilizados como o Log Compression (LEE [1991]). Neste trabalho utilizamos o primeiro, que passamos a expor, com base em RABINER et al. [1983].

Vamos definir um fator de escala c_t da seguinte forma:

$$c_t = \left(\sum_{i=1}^{N} \alpha_i(t) \right)^{-1} \qquad (6.2.3.1)$$

Assim

$$\sum_{i=1}^{N} c_t \alpha_i(t) = 1 \qquad 1 \le t \le T \qquad (6.2.3.2)$$

No cálculo de β_t por (3.3.1) realizamos o produto $c_t\beta_t$ para $1 \le t \le T$, $1 \le i \le N$.

Aplicando esses fatores, a equação (3.3.7) pode ser rescrita como

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} C_t \alpha_i(t) a_{ij} . b_{jk} \beta_j(t+1) D_{t+1}}{\sum_{t=1}^{T-1} \sum_{\iota=1}^{N} C_t \alpha_i(t) a_{ij} . b_{\iota k} \beta_\iota(t+1) D_{t+1}} \tag{6.2.3.3}$$

onde

$$C_t = \prod_{\tau=1}^{t} c_\tau \text{ e } D_t = \prod_{\tau=t}^{T} c_\tau \tag{6.2.3.4}$$

Vemos que tanto o numerador como o denominador estão multiplicados por

$$C_t D_{t+1} = \prod_{\tau=1}^{T} c_\tau \tag{6.2.3.5}$$

e esses termos podem ser cancelados, de modo que o uso desse fator de escala não altera o valor de $\bar{a}_{ij}$. Podemos facilmente verificar que o mesmo ocorre para a equação (3.3.8) no cálculo de $\bar{b}_{ij}$.

Apesar do uso de fatores de escala não alterar as fórmulas do processo de Baum-Welch é necessário levá-los em conta na classificação. Suponhamos que c_t tenha sido calculado conforme (6.2.3.1) para *t= 1, 2,..., T*; temos de (6.2.3.2) que :

$$C_T = \prod_{i=1}^{N} \alpha_i(T) = 1 \tag{6.2.3.6}$$

e, portanto C_T=1/P. De (6.2.3.4) temos que

$$\prod_{t=1}^{T} c_t = \frac{1}{\mathrm{P}} \tag{6.2.3.7}$$

O produto dos fatores de escala individuais não pode ser calculado, mas podemos obter

$$\log \mathrm{P} = -\sum_{t=1}^{T} \log c_t \tag{6.2.3.8}$$

6.2.4 Técnicas de aproximação para conjuntos finitos de treinamento

Na apresentação do algoritmo de treinamento, assumimos que a estrutura de um modelo com distribuição não conhecida poderia ser absorvida por uma seqüência de treinamento, longa o suficiente para informar sobre esta distribuição.

Na prática, no entanto, trabalhamos com seqüências finitas, que geralmente constituem um conjunto inadequado de treinamento. Pode ocorrer que um conjunto de seqüências de observações não apresente determinado símbolo *k*, de modo que o treinamento pelas fórmulas

de estimação de Baum-Welch fornecerão o parâmetro b_{jk} =0. No entanto, este símbolo pode ser possível mas pouco provável, podendo ocorrer nas seqüências de teste. Neste caso essa seqüência, mesmo correspondendo ao modelo, pode apresentar $\alpha_i(t).a_{ij}$ diferente de zero apenas para um valor de $j=\iota$, e o símbolo $o_t = k$. Se b_{ik}=0, a probabilidade calculada para esta seqüência será nula, causando fatalmente um erro. Esse fato, mais freqüente quanto menores as seqüências de treinamento, destrói o sistema de reconhecimento.

Para contornar esse problema, podemos impor restrições à matriz de símbolos **B** (LEVINSON, et al. [1983]), de tal modo que seus elementos nunca sejam nulos. Assim, se algum parâmetro b_{jk} for menor que um determinado limite substituímos esse valor por ε. Após tal substituição, realizamos a normalização de cada linha da matriz impondo a condição (3.1.c), o que equivale a dividir todos os elementos da linha por $(1 + \lambda\varepsilon)$, onde λ é o número de parâmetros da linha que foram substituídos. Esta técnica é conhecida na literatura como "floor method" (LEE [1991]).

Isso corresponde, reportando-nos à demonstração do algoritmo Baum-Welch (seção 3.3.1), a maximizar a equação (3.3.1.2)

$$F(\mathbf{x}) = \sum_i c_i \ln x_i \qquad (6.2.4.1)$$

observadas as restrições

$$\sum_i x_i = 1 \qquad (6.2.4.2)$$

$$x_i \geq \varepsilon \qquad i=1,...,N \qquad (6.2.4.3)$$

Desconsiderando-se as restrições acima, verificamos no Lema 2 apresentado naquela seção, que F(**x**) atinge o único máximo local quando $x_i = c_i \Big/ \sum_i c_i$. Supondo agora que o máximo global ocorre fora da região especificada pelas restrições (6.2.4.2) e (6.2.4.3) teremos:

$$\bar{x}_i = \frac{c_i}{\sum_{j=1}^{N} c_j} \geq \varepsilon \qquad i=1..N-\lambda \qquad (6.2.4.4a)$$

$$> \varepsilon \qquad i=N-\lambda+1,...,N \qquad (6.2.4.4b)$$

Da concavidade de F(**x**), temos que o máximo, sujeito às restrições mencionadas, deve ocorrer nos limites especificados em (6.2.4.4b). Assim, é facilmente demonstrável que, se $\bar{x}_i$ é substituído por ε para alguns valores de $i >> N-\lambda$, o máximo global das outras variáveis irá ocorrer em valores menores do que os apresentados acima. Assim, precisamos impor

$$\bar{x}_i = \varepsilon \qquad i>N-\lambda \tag{6.2.4.5}$$

e maximizar

$$\tilde{F}(\mathbf{x}) = \sum_{i=1}^{N-\lambda} c_i \ln x_i \tag{6.2.4.6}$$

observando a condição $\sum_{i=1}^{N-\lambda} x_i = 1-\lambda\varepsilon$.

De modo análogo ao apresentado no Lema 2, isto ocorre quando

$$\bar{x}_i = (1-\lambda\varepsilon)\frac{c_i}{\sum_{j=1}^{N-\lambda} c_j} \qquad i \le N-\lambda \tag{6.2.4.7}$$

Se alguns desses novos valores de $\bar{x}_i$ não satisfazem as restrições, subsitituímo-los por ε e incrementamos λ.

Então, no algoritmo modificado, vamos impor os parâmetros $b_{jk} \ge \varepsilon$, $1 \le j \le N$, $1 \le k \le M$. Inicialmente avaliamos os parâmetros utilizando as equações (3.3.7) e (3.3.8). Aqueles λ parâmetros que violarem as restrições impostas serão substituídos por ε. Os demais parâmetros serão ajustados de acordo com:

$$\tilde{b}_{jk} = (1-\lambda\varepsilon)\frac{b_{jk}}{\sum_{i=1}^{N-\lambda} b_{ij}} \tag{6.2.4.8}$$

Podemos realizar essa transformação a cada iteração do algoritmo ou no final, quando já tivermos alcançado a convergência.

A técnica exposta, que utilizamos em nosso trabalho, resolve o problema exposto no início desta seção, i.e., o fato de a probabilidade de uma seqüência correspondente a um modelo ter valor nulo pela presença de zeros na matriz **B**, decorrentes de um conjunto de treinamento inadequado. É um método simples e bastante eficiente para conjuntos razoáveis de treinamento. Além disso, permite um procedimento de compactação de dados, uma vez que uma grande maioria dos parâmetros b_{jk} da matriz **B** terá o valor ε.[4]

[4] Já verificamos como as restrições impostas a matriz **A** permitem uma redução do espaço de memória necessário para armazená-la. Aqui a redução é mais notável. Um sistema com 5 estados e 128 símbolos, por exemplo, possui uma matriz **B** que tem, em princípio, 5x128=640 elementos. Considerando-se os resultados que temos observado, os modelos de uma palavra apresentam, por estado, poucos parâmetros b_{jk} diferentes de ε. Assumindo, numa hipótese conservadora, esse valor médio como 24, verificamos que podemos obter uma redução do espaço de armazenamento da ordem de 640:(5x24), ou 6:1. É evidente que isso envolve tempo de processamento e exige uma indexação dos elementos da matriz, o que exige

No entanto, esta técnica apresenta o inconveniente de representar os símbolos pouco prováveis do mesmo modo que os impossíveis. Um exemplo de como isto pode ocorrer é analisado no Anexo A, para a seqüência de uma amostra da palavra “três”. Outras abordagens tem sido dadas ao problema de conjunto finitos de treinamento, tais como o “smoothing” pelo método das distâncias ou das co-ocorrências (SCHWARTZ, [1989]).

O método das distâncias trabalha com a hipótese de que se dois símbolos possuem centróides próximas então devem apresentar também uma densidade de probabilidade semelhante. Desse modo, a probabilidade de um símbolo não observado no treinamento pode ser corrigida pelo valor de um símbolo com centróide próxima suficientemente observado. CRAVERO [1984] utiliza um estimador de Parzen para determinar uma transformação dos parâmetros em função da distância. Utilizamos uma variante desta técnica no teste da seção 7.3.2.2.2, “d”, onde a apresentamos mais detalhadamente.

No método das co-ocorrências procura-se mapear para cada símbolo quais outros símbolos ocorrem freqüentemente em seu lugar. Ou, de modo inverso, quando um símbolo é observado, verifica-se quão provável é a observação de outro símbolo naquele contexto.

A utilização de técnicas de “smoothing” evidencia o compromisso que existe na utilização de um número elevado de níveis na quantização. Se, por um lado, um livro código amplo melhora a representação do sinal, por outro, exige uma fase de treinamento mais complexa.

6.2.5 Múltiplas observações independentes

O algoritmo Baum-Welch foi apresentado utilizando uma seqüência de observação, mas é facilmente generalizado para múltiplas seqüências independentes. No caso de reconhecimento de palavras iremos treinar os modelos com base em diversas seqüências geradas a partir de diferentes elocuções da mesma palavra.

O treinamento de um conjunto de parâmetros para o modelo de Markov a partir de um conjunto de múltiplas seqüências pode ser feito computando-se as transições de acordo com os somatórios das equações (3.3.7) e (3.3.8). Estes valores podem ser somados e então todos os parâmetros são re-estimados. Isso corresponde a uma iteração do algoritmo.

Este procedimento visa maximizar o produto das probabilidades

---

também certo espaço de memória. Ainda assim, os resultados nesse sentido são significativos (particularmente para a implementação em sistemas com severas restrições de memória).

$$P = \prod_{k=1}^{N_{seq}} P(O^k | \mathbf{M}) \qquad (6.2.5.1)$$

onde $P(O^k | \mathbf{M})$ é a probabilidade da seqüência *k* dado o conjunto de parâmetros do modelo **M**.

6.3 Independência do locutor

Como já destacamos no histórico, a independência do locutor é uma das restrições mais difíceis de ser superada nos sistemas de reconhecimento. Algumas técnicas tem buscado contornar o problema da variabilidade entre as diversas vozes, tais como, o agrupamento de locutores (speaker clustering) (NAKAMURA e SHIKANO, [1989], Schwartz, apud LEE, [1991]; e FENG, [1989]), e a re-estimação interpolada, utilizada pelo SPHINX (LEE, [1991]).

O primeiro método procura estimar grupos de locutores com características semelhantes. Para cada grupo são gerados os HMMs correspondentes na fase de treinamento. Na fase de reconhecimento identifica-se inicialmente qual o grupo mais próximo para aquela voz, (p.ex., através de um algoritmo tipo DTW) e, em seguida, realiza-se a busca entre os modelos de Markov daquele grupo. Outra possibilidade de se gerar os modelos específicos para cada grupo é realizar o treinamento com todos os locutores, criando um modelo geral, e proceder-se a um mapeamento probabilístico para determinar os parâmetros dos específicos. Tal método exige um número significativo de vozes para tornar os agrupamentos estatisticamente válidos. Com poucos grupos apenas as diferenças mais grosseiras podem ser separadas.

O outro método, baseado na “deleted interpolation” (Jelinek, apud LEE, [1991]), consiste em realizar um adaptação partindo-se dos parâmetros de um modelo geral extensivamente treinado. A adaptação se processa interativamente com o locutor, através de uma re-estimação pelo algoritmo de Baum-Welch, inferindo os parâmetros para determinado locutor. Desse modo, junta-se as informações de um modelo geral bem treinado, com as informações particulares de cada locutor com reduzido treinamento. Este método apresenta ainda a vantagem de a adaptação processar-se durante o uso.

A utilização de qualquer um desses métodos exigia um banco de dados bastante extenso de que não dispúnhamos. Neste trabalho pudemos apenas verificar a pertinência do primeiro método, na seção 7.3.2.1, “a”.

7 Resultados

O trabalho experimental foi desenvolvido no sentido de explorar o potencial do método conforme o grau de complexidade do sistema. Desse modo, iniciamos trabalhando apenas com um locutor, usando vocabulário de onze palavras, os dígitos de zero a nove e o algarismo dez. Nesta fase verificamos os problemas relativos à captação do sinal, que apresentamos na seção 7.1. Fizemos ainda uma análise do comportamento do processo de quantização vetorial (seção 7.2) com relação aos parâmetros apresentados na parte teórica. Os resultados do sistema dependente do locutor são apresentados na seção 7.3.1.

A seguir, mantido o mesmo vocabulário, tomamos um conjunto de dez locutores, com três amostras de cada palavra. Realizamos testes, apresentados em 7.3.2, com diversas combinações desses locutores, verificando a variação da performance do sistema conforme o número e tipo de locutores.

Considerando os resultados obtidos não suficientemente satisfatórios face à proposta de um sistema de reconhecimento de palavras independente do locutor, procuramos analisar as causas dos erros e implementar alternativas que poderiam apresentar resultados mais aceitáveis.

7.1 Observações preliminares

Os primeiros testes realizados mostraram grande sensibilidade do sistema às condições físicas de captação do sinal de voz entre as quais destacamos as características do microfone e seu posicionamento.

Nestes testes preliminares com o vocabulário de onze palavras e um locutor, obtivemos taxas de erro da ordem de 25 % quando não nos preocupamos com o tipo de microfone ou sua posição. Com as mesmas características do sistema, mas mantendo fixa a posição do microfone durante o treinamento e reconhecimento essas taxas caíram para um valor médio de 1,8 %.

A questão das características de resposta do microfone não apresenta em princípio maior dificuldade. Se o sistema deve reconhecer diferentes tipos de vozes, e portanto deve ser capaz de absorver durante o treinamento, as diferenças das características das vozes, também deveria absorver as diferenças das características dos microfones (desde que com um mínimo de qualidade) que participassem do treinamento.

Quanto ao posicionamento, parece claro que o problema esteja ligado a dois fatores : a distância e a direcionalidade. A distância influi a princípio na intensidade do sinal, o que

podemos solucionar com algum tipo de normalização do sinal, transportando para um determinado valor médio, desde que o sinal esteja dentro de certos limites. Isto, aliás, deverá ser feito de qualquer modo, independentemente de mantermos ou não a distância constante, para compensarmos a diferença de intensidade natural entre as vozes dos locutores. Deve-se considerar que o aumento excessivo da distância provoca degradação do sinal reduzindo a relação sinal-ruido.

A questão da resposta direcional pode parecer em princípio semelhante à questão do tipo de microfone. O problema é, no entanto, um pouco mais complexo, uma vez que deveríamos considerar todas as direções possíveis (ou necessárias) em relação ao diagrama de resposta polar do microfone utilizado no treinamento.

As mesmas considerações podem ser feitas com relação às características acústicas do ambiente e ao nível de ruído.

Considerando o objetivo desse trabalho, decidimos eliminar esse tipo de dificuldade padronizando o sistema com relação ao microfone e seu posicionamento, e ainda com relação ao ambiente e ao ruído, isto é, em cada etapa realizaremos tanto o treinamento como o reconhecimento mantendo todos essas características constantes. Tais aspectos devem ser estudados à parte, buscando-se soluções específicas para cada um desses problemas.

7.2 Aspectos da quantização vetorial

Utilizamos inicialmente um locutor para produzir um conjunto de dez amostras dos onze números de “zero” a “dez”, sendo cinco amostras de cada palavra utilizadas ora para treino ora para teste. Em paralelo com a análise que segue na próxima seção, do desempenho do sistema com um locutor com vocabulário de onze palavras, realizamos um estudo do comportamento do processo de quantização vetorial nesse sistema.

Em nosso sistema, os coeficientes de correlação parcial derivados dos LPC até 10ª ordem foram calculados para cada segmento de 20 ms da amostra e são representados por uma classe. O vetor decadimensional é inicialmente mapeado no intervalo -1000 a 1000 para possibilitar a representação por números inteiros. O espaço é então quantizado em até 128 classes, podendo ser representado por um byte.

O algoritmo utilizado é o de Linde, descrito na seção 5.2. O princípio deste algoritmo, com poucas variantes, vem sendo amplamente usado para os sistemas de reconhecimento.

O comportamento do parâmetro de afastamento s/D, descrito em 5.3, em função do número de classes apresentou-se conforme o gráfico a seguir:

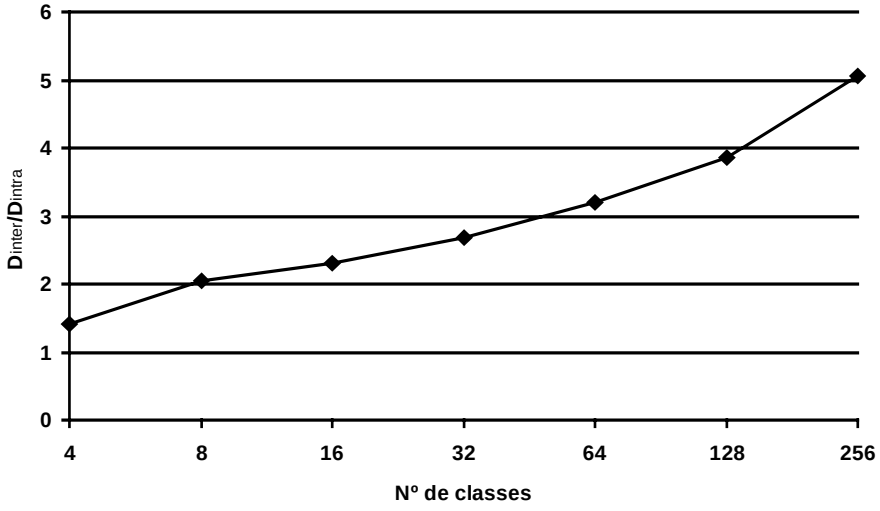


Figura 7.A Afastamento relativo inter-centróides.

conforme o esperado, i.e., crescente com o aumento do número de classes.

Para 64 classes, obtivemos a seguinte distribuição de vetores por classes:

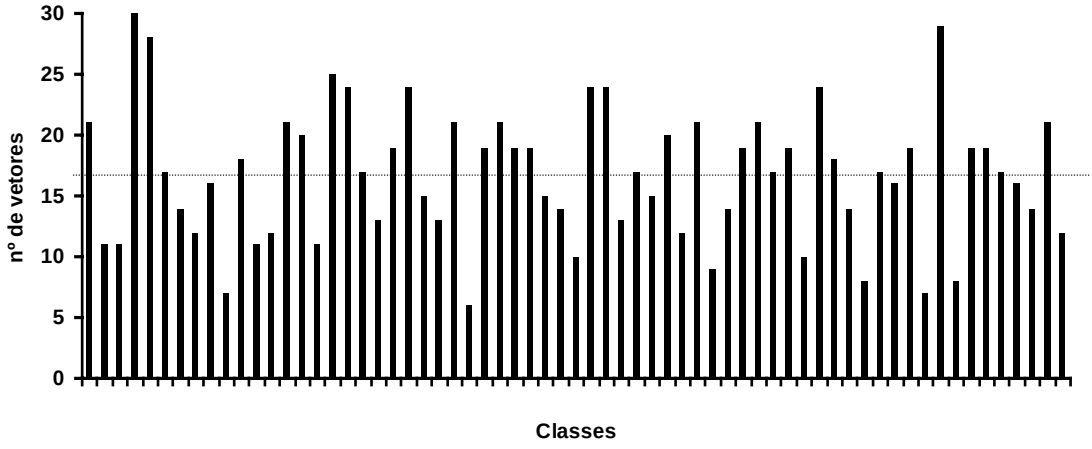


Figura 7.B Distribuição dos vetores por classe.

o que representa uma boa distribuição dos vetores pelas diversas classes, não ocorrendo grandes concentrações de amostras em determinadas classes, nem classes com número excessivamente reduzido. Verificamos que a maior parte das classes tem cardinalidade abaixo do valor médio (c_{med} = 17) representado pela linha pontilhada. A razão entre máxima e mínima cardinalidade foi de 6 para 1.

Podemos observar também que, para este conjunto de amostras, o aumento do número de classes certamente levará ao surgimento de classes com pouco elementos que, como veremos, induzirá o sistema a erro.

7.3 Reconhecimento com vocabulário de onze palavras

Os primeiros testes foram realizados com apenas um locutor, verificando a priori a consistência do programa, bem como os já mencionados problemas de captação do sinal. Desta fase saíram também os resultados da seção anterior. Estudamos então o efeito do número de classes do quantizador e do número de repetições da mesma palavra durante o treino (seção 7.3.1)

Expandimos em seguida o universo de locutores, utilizando 10 vozes diferentes (seção 7.3.2), realizando testes com diversas combinações de número e tipo de vozes.

7.3.1 Resultados com um único locutor

7.3.1.1 Efeito do número de classes do quantizador

Nesta fase estudamos o comportamento do erro em função do número de classes que utilizamos na quantização vetorial.

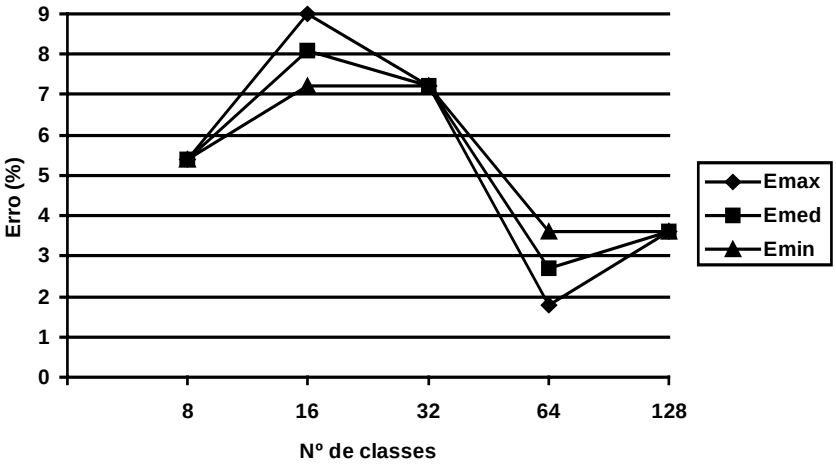


Figura 7.C Taxa de erro x nº de classes, dependente do locutor.

O que naturalmente deveríamos ter observado no gráfico da figura 7.3 seria um decréscimo do erro quando aumentássemos o número de classes. No entanto, outro fator concorre na formação dessa taxa de erro. A utilização de um conjunto de amostras pequeno

face ao número de classes do quantizador faz com que a quantização vetorial torne-se uma fonte de erro.

De fato, considerando-se que utilizamos 5 observações de 11 palavras tanto no conjunto de treinamento como no de teste e, tendo cada amostra (dígito) média de 30 segmentos, nossos conjuntos de treinamento ou teste tem no total cerca de 1600 segmentos cada. Classificando essas amostras em 128 classes teremos uma cardinalidade média de cerca de 12 (1600/128). Essa média é bastante baixa, podendo ocorrer classes com poucos elementos, ou mesmo apenas um.

Desse modo, o aumento excessivo do número de classes para um conjunto de dados pequeno resulta na degradação da performance do quantizador, no sentido em que este tende a separar, ao invés de agrupar, os fenômenos acústico-vocálicos semelhantes. Por outro lado, a redução do número de níveis faz com que o mesmo conjunto de amostras seja mais representativo. Assim, quando procedemos à quantização com apenas 8 níveis, obtemos uma representação mais adequada do ponto de vista estatístico.

Da concorrência desses dois fatores (aumento do número de classes x limitação do conjunto de dados) podemos explicar o comportamento sinuoso da taxa de erro. Os resultados obtidos poderiam sugerir a conclusão errônea de que 8 classes, ou até menos, seriam suficientes, apresentando grandes vantagens do ponto de vista computacional. No entanto, esse é um número muito baixo para representar a diversidade de fenômenos presentes no vocabulário utilizado.

A menor taxa de erro foi obtida para 64 classes. RABINER [1983] observa que a redução da distorção média quando se passa de 64 para 128 níveis não justifica o aumento da carga computacional, optando por 64 classes no desenvolvimento de sistema com características semelhantes.

A figura 7.4 mostra o erro como número de ocorrências em função do número de classes para as diversas palavras do vocabulário.

Verificamos que o erro se concentra em determinadas palavras, particularmente no “três”, “seis”, e “dez”. Além disso, como podemos extrair da tabela 7.1, tais erros ocorrem com muita freqüência em virtude da confusão entre essas palavras. Por exemplo, “dez” foi reconhecido como “três” por nove vezes, representando 40% do total dos erros observados.

Por outro lado, o erro que corresponde ao dígito “zero” no gráfico citado foi seu reconhecimento como “cinco” justamente quando utilizamos apenas 8 classes, o que reforça a idéia que esse número de classes não é suficiente para representar a diversidade de fenômenos desse vocabulário.

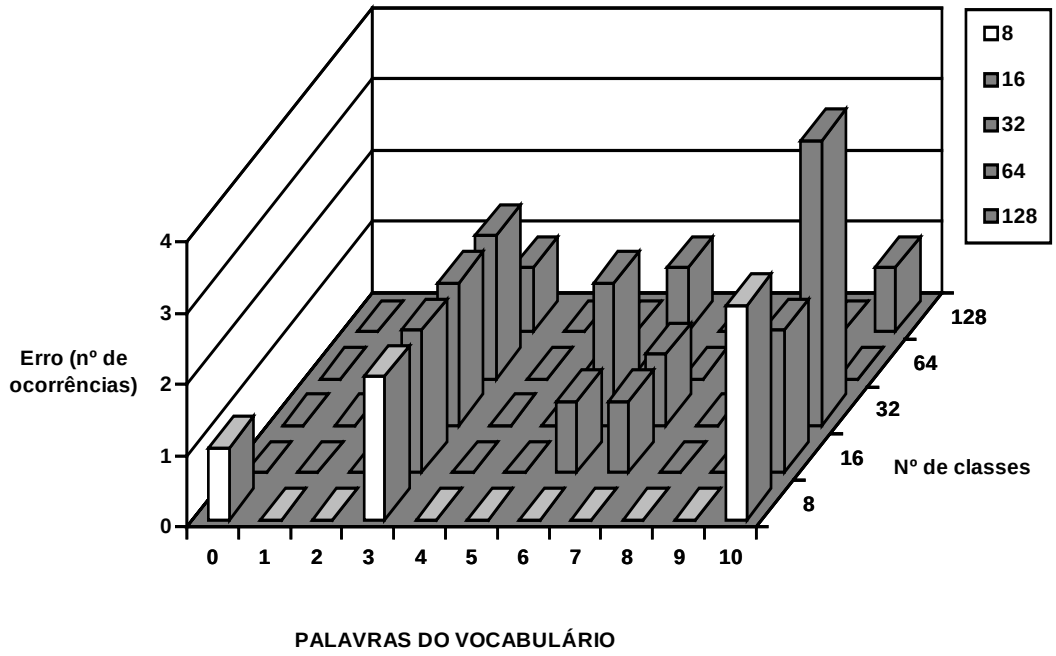


Figura 7.D Erros por palavras e nº de classes, dependente do locutor.

Tabela 7.A Matriz de confusão, teste de nº de classes, dependente do locutor.

palavra do vocabulário	palavra errada reconhecida (nº de ocorrências)											% do total de erros
	0	1	2	3	4	5	6	7	8	9	10	
0						1						4,0
1												
2												
3							3				5	32,0
4												
5												
6				3				1				16,0
7				1							1	8,0
8												
9												
10			1	9								40,0

7.3.1.2 Efeito do número de observações no treinamento

Verificamos o comportamento da taxa de erro do sistema quando aumentamos o número de repetições da mesma palavra durante o treinamento. Utilizamos a quantização vetorial com 64 níveis, por esta ter apresentado os melhores resultados na seção precedente.

Os resultados obtidos mostraram, conforme esperado, que a taxa de erro diminui sensivelmente quando aumentamos as repetições, conforme o gráfico 7.5.

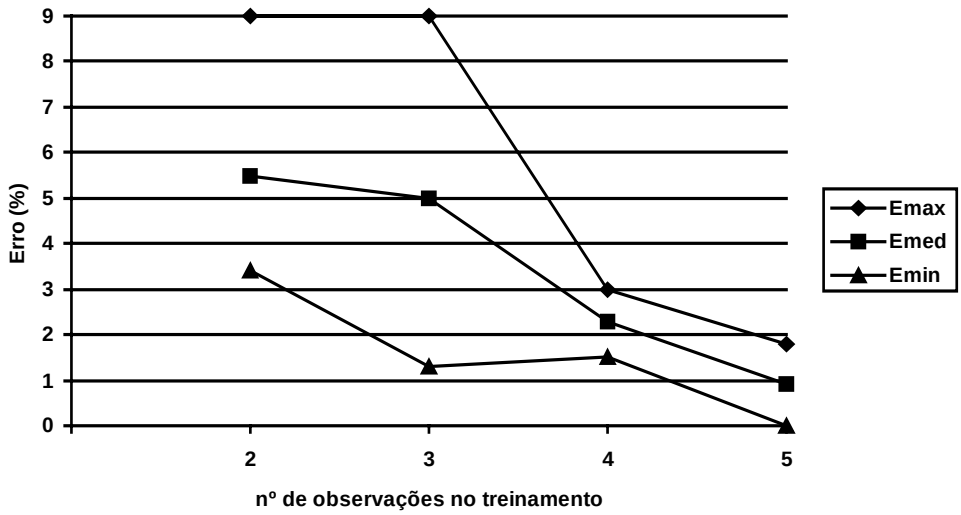


Figura 7.E Taxa de erro x nº de repetições, dependente do locutor.

Chegamos a obter erro nulo com 5 repetições, indicando ser este um bom número para sistemas com essas características (um locutor, onze palavras, quantização com 64 classes). É importante destacar que essas repetições foram realizadas procurando dar ritmos e entoações diferentes às palavras, pois o que se pretendia era determinar em que medida que tais variações passariam a não induzir erro, tendo o modelo de Markov extraído as características essenciais da palavra, uma vez que a proposta final é a de um sistema que não dependa do locutor.

Desse modo, é preciso certa cautela na análise desse resultado. Como exemplo, imaginemos um sistema que se propusesse a trabalhar com 10 locutores determinados. Poderíamos inferir dos resultados obtidos, que tal sistema deveria ser treinado por 5 repetições de cada palavra por cada locutor para obtermos uma performance semelhante. Tal conclusão é no mínimo precipitada, pois as características de cada palavra emitida pelos diversos locutores poderiam ser absorvidas pelo modelo com um número menor de repetições pela presença de elementos comuns.

Com relação à distribuição dos erros no vocabulário observamos novamente, e mais acentuadamente, a sua concentração no par “três” - “seis” que responde por 61% do total dos erros observados. Lembrando que “três” é freqüentemente pronunciado como “*treis*”, começamos a verificar a dificuldade que os modelos de Markov apresentam para tratar com os fenômenos mais rápidos (não vocálicos) em relação aos lentos (vocálicos). Neste sentido, como veremos mais a frente, têm sido propostos modelos dinâmicos, que procuram dar mais atenção a esses segmentos das palavras. Reforçando esta idéia, observamos que não houve nessa fase nem na anterior nenhum erro no reconhecimento do dígito “dois”, que em princípio imaginávamos que poderia ser confundido com o “dez”. Ao contrário, os erros que observamos para o dígito “dez” sempre estiveram relacionados a confusão com o mesmo “três” e o “seis”, justamente pela semelhança na parte vocálica.

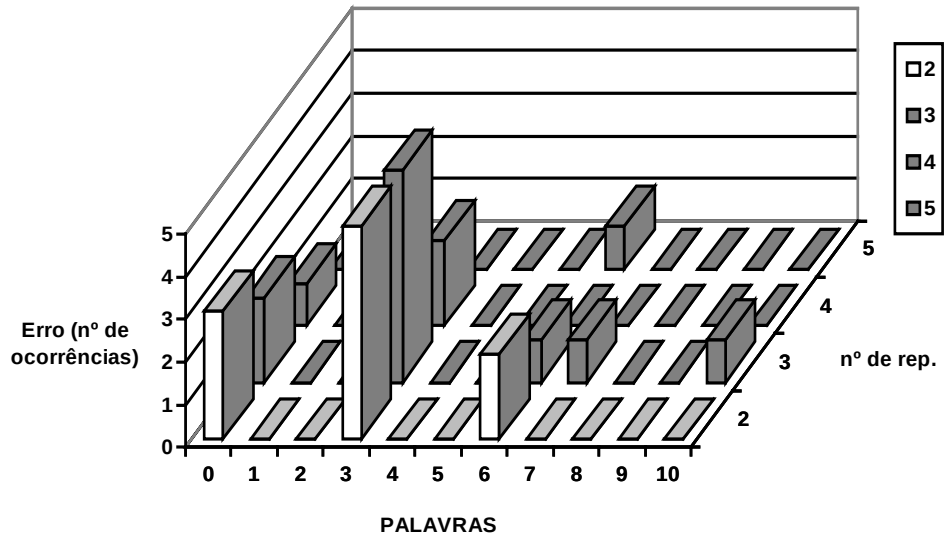


Figura 7.F Erros por palavras e nº de repetições, dependente do locutor.

Tabela 7.B Matriz de confusão, teste de nº de repetições, dependente do locutor.

palavra do vocabulário	palavra errada reconhecida (nº de ocorrências)											% do total de erros
	0	1	2	3	4	5	6	7	8	9	10	
0				3			1		1		1	27,2
1												
2												
3							8				2	45,4
4								1				4,5
5												
6				3								13,6
7												
8												
9												
10				1		1						9,1

7.3.1.3 Resumo dos resultados com um locutor

Podemos extrair dos testes realizados algumas conclusões:

- o número de classes deve ser suficiente para representar as diversidades de fenômenos físicos presentes nas palavras do vocabulário e, ao mesmo tempo, ser compatível com a dimensão do conjunto de amostras disponível;
- o desempenho do sistema melhora quando acrescentamos informação ao treinamento, aumentando o número de observações de cada palavra.

Chegamos a resultados bastante satisfatórios, tendo atingido 100 % de reconhecimento sobre o conjunto de teste, utilizando 64 classes e 5 observações por palavra, condições estas razoáveis em face da reduzida dimensão do vocabulário e estarmos trabalhando apenas com um locutor. O aumento do número de locutores não deve, em princípio, influir muito no comportamento no que toca ao número de classes, mas devemos considerá-lo com cuidado no que diz respeito ao número de observações de cada palavra para cada locutor utilizado no treino. Isto será realizado na próxima seção.

Em resumo, neste sistema simples, implementado apenas com onze palavras e um locutor, podemos verificar a viabilidade da aplicação dos modelos de Markov para o reconhecimento de palavras.

7.3.2 Resultados com dez locutores

Nesta fase realizamos testes com o mesmo vocabulário da seção anterior (as palavras de zero a dez) com um universo de 10 locutores. O banco de dados se consistiu em 3 amostras de cada palavra falada por cada um dos locutores, em condições acústicas semelhantes.

O conjunto de locutores constituiu-se por 5 vozes musculinas adultas, 4 vozes femininas adultas e uma voz masculina infantil. Todos possuíam uma mesma identidade de pronúncia regional, sem fortes distinções de sotaque.

Os testes foram divididos em dois grupos:

- na seção 7.3.2.1 procuramos a taxa de erro no reconhecimento das vozes que não participaram do treinamento; realizamos o treinamento sempre com 3 amostras de cada voz, fazendo-o com uma voz masculina, ou com uma feminina, ou com duas masculinas, ou duas femininas, ou duas vozes uma de cada, ou quatro masculinas, ou femininas, ou duas de cada tipo; o reconhecimento foi realizado sempre sobre todas as amostras das vozes que não participaram do treino;
- na seção 7.3.2.2 verificamos a taxa de erro para as amostras das vozes que não participaram do treinamento; realizamos o treino com 1 ou 2 das amostras de 2, ou 4, ou 10 vozes, fazendo o reconhecimento sempre sobre as amostras restantes das mesmas vozes que participaram do treino.

Esta divisão se refere diretamente ao tipo de independência do locutor. Esta *independência* comporta pelo menos dois significados. Podemos entendê-la de modo absoluto, o que corresponde ao primeiro grupo, de tal modo que o sistema, se treinado por um suficiente conjunto de dados, seria capaz de reconhecer o mesmo vocabulário para qualquer voz. De outro modo (segundo grupo), podemos entendê-la no sentido de grupos de locutores, i.e., o sistema é capaz de reconhecer qualquer voz entre as pertencentes ao grupo de treinamento.

Nesta seção, demos ênfase ao aspecto número e tipo de locutores utilizados no treinamento, mantendo fixas as características da quantização vetorial, utilizando sempre 64 classes, calculadas as centróides sobre todos os dados incluídos no treinamento, no sentido de trabalhar com um conjunto ótimo de centróides em cada caso. É evidente que tal conjunto é ótimo apenas para os dados incluídos no treino, consistindo em fonte adicional de erro na etapa de reconhecimento, particularmente quando esta é realizada com vozes que não participaram do treino. Mesmo assim, permanecem válidas as considerações feitas sobre o cálculo do limite da equação (5.1), se consideramos o processo ergódico e estacionário.

7.3.2.1 Reconhecimento com vozes que não participaram do treino

a) Treino com uma voz adulta masculina

Realizado o treinamento com as 3 amostras de uma das vozes adultas masculinas, (Loc=A, Tipo M-A) obtivemos o seguinte desempenho:

Tabela 7.C Taxas de erro, independente do locutor, treino com uma voz tipo M-A.

Loc	Tipo	Erro (%)
F	M-A	6,1
B	M-A	21,2
G	M-I	48,5
N	M-A	54,5
S	M-A	54,5
T	F-A	66,7
E	F-A	69,6
L	F-A	69,6
M	F-A	87,8
Erro Médio		**53,2**

VQ = 64 cl. - 1047 amostras
s/D = 2,98
1 v.m., Loc=A, 3 am/pal
Tipo: M= masculina, F= feminina,
A= adulto, I= infantil

Dentro do conjunto de treinamento o acerto foi de 100%. No reconhecimento verificamos na tabela 7.3 que a taxa de erro foi: a) relativamente baixa para os locutores F e B (ambos irmãos de A, com características de timbre de voz semelhantes, embora não o estilo de fala); b) situou-se em torno de 50% para as demais vozes masculinas; e c) situou-se acima de 66% para todas as vozes femininas. As baixas taxas de erro observadas para F mostram o fundamento do método de agrupamento de locutores (speaker clustering) mencionado na seção 6.3.

Sendo tais taxas demasiadamente elevadas, a distribuição dos erros pelas diferentes palavras do vocabulário não faz qualquer sentido. Destacamos apenas que para o locutor F houve apenas dois erros, um “três” reconhecido como “seis”, e um “oito” como “quatro”. O primeiro, já analisamos na etapa anterior, referindo-se a semelhança na parte vocálica da palavra. Já o segundo, faz parte do grande volume de erros obtido no sistema, devendo-se basicamente a um treinamento inadequado.

A grande fonte de erro começa já na exclusão das amostras no processo de quantização vetorial. Os conjuntos de coeficientes PARCOR dos vetores LPC, que representam as diversas palavras nas vozes femininas, encontram-se em regiões bastante diversas daquelas que os representam nas vozes masculinas. A quantização feita com apenas uma voz masculina cria regiões bem determinadas para esse conjunto de dados, mas que pode ser completamente confusa para o outro. Desse modo, o símbolo pelo qual um vetor de coeficientes de uma voz feminina é representado pode não apresentar significado algum. A partir daí, estaria comprometido o treinamento, mesmo se o conjunto de vozes femininas fosse incluído na fase de treinamento do modelo.

b) Treino com uma voz adulta feminina

A taxa de erro, utilizando-se apenas uma voz feminina (Loc=E, Tipo F-A) no treinamento, apresentou para as demais vozes os seguintes resultados:

Tabela 7.D Taxas de erro, independente do locutor, treino com uma voz tipo F-A.

Loc	Tipo	Erro(%)
M	F-A	45,4
L	F-A	54,5
B	M-A	57,5
N	M-A	66,6
T	F-A	69,6
G	M-I	69,6
F	M-A	69,6
S	M-A	72,7
A	M-A	75,7
Erro Médio		**64,6**

VQ = 64 cl. - 1098 amostras
s/D = 3,16
1 v.m., Loc=E, 3 am/pal
Tipo: M= masculina, F= feminina, A= adulto, I= infantil

O erro foi bastante elevado de um modo geral, já não se apresentando tão nítida a distinção entre vozes masculinas ou femininas. Considerando tal teste mais genérico que o anterior, uma vez que não há qualquer grau de parentesco entre as vozes femininas, podemos verificar que o sistema é extremamente sensível ao tipo de voz, não sendo possível o reconhecimento independente do locutor nesse caso.

Observamos igualmente acerto de 100% para o conjunto treinado, indicando suficiente definição do sistema, seja quanto ao número de classes da quantização, ou quanto à estimativa dos parâmetros do modelo.

c) Treino com 2 e 4 vozes adultas femininas

Utilizamos 3 amostras de duas e quatro das vozes adultas femininas (Loc=E e L no 1º caso, e Loc=E, L, M, e T no 2º, todas do tipo F-A) obtendo as seguintes taxas de erro :

Tabela 7.E Taxas de erro, independente do locutor, treino com 2 e 4 vozes tipo F-A.

Loc	Tipo	Erro 2 (%)	Erro 4 (%)
B	M-A	30,3	9,1
G	M-I	33,3	12,0
M	F-A	39,4	-
F	M-A	42,4	33,3
T	F-A	45,4	-
N	M-A	51,5	36,3
A	M-A	66,6	48,5
S	M-A	51,5	63,6
Erro Médio		**45,0**	**33,8**

VQ = 64 cl. - 2550(2)-5070(4) amostras
s/D = 2,69(2) - 2,49(4)
2 v.f., Locs=E, L, 3 am/pal
4.v.f., Locs=E, L, M, T; 3 am/pal
Tipo: M= masculina, F= feminina, A= adulto, I= infantil

As colunas Erro 2 e Erro 4 referem-se às taxas de erro quando treinamos com duas e quatro vozes respectivamente.

Podemos observar que o erro médio total do sistema foi um pouco inferior em comparação aos testes com apenas uma voz no treino. Por outro lado, não houve uma distribuição característica da taxa de erro em função do tipo de voz.

No próprio conjunto de treinamento observamos o reconhecimento correto de todas as amostras. Estes resultados estão de acordo com os obtidos por MINAMI [1993] que obteve 100% de acerto no reconhecimento dentro do conjunto de treinamento com 5 locutores, 10 palavras, usando 64 classes.

d) Treino com 2 e 4 vozes adultas masculinas

Realizamos então o treinamento apenas com vozes masculinas utilizando todas as 3 amostras de duas e quatro das vozes masculinas. (Loc=A e S, no 1º caso e Loc=A, S, N e B, no 2º, todas tipo M-A).

Neste caso, obtivemos no próprio conjunto de treinamento erro de 6% no reconhecimento do locutor S, quando utilizamos quatro vozes. Tal fato indica que não se chegou a um modelo apropriado que representasse as características desse locutor face aos demais dentro do próprio treino.

Os resultados deste teste são apresentadas na tabela 7.6. As colunas Erro 2 e Erro 4 referem-se às taxas de erro quando treinamos com duas e quatro vozes respectivamente.

Tabela 7.F Taxas de erro, independente do locutor, treino com 2 e 4 vozes tipo M-A.

Loc	Tipo	Erro 2 (%)	Erro 4 (%)
F	M-A	26,9	15,0
B	M-A	30,3	-
G	M-I	63,6	33,3
E	F-A	48,4	39,3
T	F-A	60,6	42,0
L	F-A	60,6	48,4
N	M-A	66,6	-
M	F-A	78,7	60,0
Erro Médio		**54,5**	**39,7**

VQ = 64 cl. - 1368(2) - 2885(4) amostras
s/D = 2,60(2) - 2,55(4)
2 v.m.., Locs=A, S , 3 am/pal
4.v.m., Locs=A, S, N, B; 3 am/pal
Tipo: M= masculina, F= feminina,
A= adulto, I= infantil

Lembrando as semelhanças entre os locutores A (que participou do treino) e F (que não participou) observamos que neste caso, apesar de ser a taxa de erro do locutor F a mais baixa, ainda assim foi elevada, sendo superior à menor taxa que obtivemos no ítem “c”. Ainda, com

relação ao ítem anterior, de um modo geral, o erro situou-se na mesma média de 40% atingindo até 60% para o pior reconhecimento.

Apesar de termos observado uma redução do erro quando aumentamos o número de vozes no conjunto de treinamento, constatamos novamente que o sistema é extremamente sensível ao tipo de voz, o que nos conduz ao teste seguinte.

e) Treino com uma e duas vozes de cada tipo

Estudamos então o comportamento do processo incluindo vozes de diferentes tipos no treinamento. Realizamos testes utilizando duas e quatro vozes, a fim de comparar os resultados com os dos ítens anteriores, sendo que, das duas vozes, uma era masculina a outra feminina (Loc=A, tipo A-M e Loc=M, tipo F-A), e das quatro vozes, duas masculinas (Loc=A e S, tipo M-A) e duas femininas (Loc=M e T, tipo F-A).

Constatamos novamente erro no reconhecimento do próprio conjunto de treinamento quando utilizamos 4 vozes, ocorrendo uma vez a confusão de *dez* com *três* para o locutor S, e duas trocas do dígito *um* (por *dois* e *oito*) para a locutora T.

As taxas de erro obtidas para os demais locutores que não participaram do treino foram:

Tabela 7.G Taxas de erro, independente do locutor, treino com 2 e 4 vozes M-A e F-A.

Loc	Tipo	Erro 2 (%)	Erro 4 (%)
B	M-A	12,1	27,2
F	M-A	24,2	21,2
L	F-A	33,3	30,3
G	M-I	42,2	24,2
S	M-A	48,4	-
T	F-A	48,4	-
E	F-A	51,1	36,3
N	M-A	54,5	33,3
Erro Médio		**39,4**	**28,8**

VQ = 64 cl. - 1548(2)-3090(4) amostras
s/D = 2,90(2) - 2,46(4)
1vf+1vm., Locs=A, M; 3 am/pal
2vf+2vm, Locs=A, S, M, T; 3 am/pal
Tipo: M= masculina, F= feminina,
A= adulto, I= infantil

Observamos que a inclusão de vozes de tipos diferentes reduziu a taxa de erro média, sendo os valores obtidos mais baixos que os verificados para o treinamento com 2 e 4 vozes do mesmo tipo. Tal fato corresponde à expectativa que tínhamos de que o sistema deve possuir informação da diversidade do universo de vozes. Desse modo, para um sistema de reconhecimento absolutamente independente do locutor, podemos inferir que o treinamento deve ser feito com um número elevado de amostras incluindo o máximo da diversidade dos tipos de vozes.

7.3.2.2 Reconhecimento com vozes que participaram do treino

Nesta fase realizamos os testes tomando para cada voz participante do teste, parte das amostras para o treino e parte para o reconhecimento. Os testes foram realizados com 2, 4 e 10 vozes, sendo o treino com 1 ou 2 das amostras e o reconhecimento com as restantes.

A seguir, apresentamos as taxas de erro obtidas para os diversos testes:

Tabela 7.H Taxas de erro, múltiplos locutores, treino com uma voz M-A e uma F-A.

Loc	Tipo	Erro1 (%)	Erro2 (%)
A	M-A	4,5	0
M	F-A	31,8	9,1
Erro Médio		**18,1**	**4,6**

1vf+1vm., Locs = A, M; 1 e 2 am/pal
Tipo: M= masculina, F= feminina, A= adulto, I= infantil

Tabela 7.I Taxas de erro, múltiplos locutores, treino com duas vozes M-A e duas F-A.

Loc	Tipo	Erro1 (%)	Erro2 (%)
A	M-A	22,7	9,1
M	F-A	27,2	9,1
S	M-A	18,1	9,1
T	F-A	22,7	18,2
Erro Médio		**22,7**	**11,37**

2vf+2vm, Locs = A, M, S, T; 1 e 2 am/pal
Tipo: M= masculina, F= feminina, A= adulto, I= infantil

Tabela 7.J Taxas de erro, múltiplos locutores, com 4 vozes F-A.

Loc	Tipo	Erro1 (%)	Erro2 (%)
E	F-A	13,6	18,2
L	F-A	13,6	0
M	F-A	40,9	9,1
T	F-A	22,7	18,2
Erro Médio		**22,7**	**11,37**

4vf, Locs = E,L,M,T; 1 e 2 am/pal
Tipo: M= masculina, F= fem.inina, A= adulto, I= infantil

Tabela 7.K Taxas de erro, múltiplos locutores, com 4 vozes M-A.

Loc	Tipo	Erro1 (%)	Erro2 (%)
A	M-A	9,1	9,1
B	M-A	22,7	9,1
N	M-A	27,2	18,2
S	M-A	22,7	9,1
Erro Médio		**20,4**	**11,37**

4vm, Locs = A,B,N,S; 1 e 2 am/pal
Tipo: M= masculina, F= feminina, A= adulto, I= infantil

Tabela 7.L Taxas de erro, múltiplos locutores, com dez vozes.

Loc	Tipo	Erro 1 (%)	Erro 2 (%)
A	M-A	18,1	0
B	M-A	4,5	9,1
E	F-A	13,6	9,1
F	M-A	18,1	0
G	M-I	27,2	9,1
L	F-A	18,1	18,2
M	F-A	13,6	18,2
N	M-A	36,3	9,1
S	M-A	31,8	18,2
T	F-A	18,1	18,2
Erro Médio		**19,9**	**10,9**

10v., Locs = A, B, E, F, G, L, M, N, S, T;
1 e 2 am/pal
Tipo: M= masculina, F= feminina,
A= adulto, I= infantil

As colunas Erro 1, Erro 2 referem-se às taxas de erro obtidas individualmente por cada locutor na fase de reconhecimento com uma e duas amostras no treinamento, respectivamente.

Os resultados obtidos nos permitem estabelecer uma relação direta entre o número de amostras que participaram do treino e a taxa de erro. Podemos, grosso modo, dizer que esta taxa situou-se em torno de 20% para 1 amostra e em 10% para 2 amostras, destacando que no caso de 2 vozes essa taxa caiu para 4,5% em função do próprio número reduzido de vozes.

Os resultados obtidos, se comparados aos da seção anterior, particularmente ítens “c”,”d”, e “e”, confirmam a observação de LEE [1991] na introdução de seu trabalho: consideradas as mesmas características, os sistemas independentes do locutor apresentam taxas de erro três a cinco vezes maiores que os dependentes. Para 4 vozes obtivemos taxas de erro da ordem de 30% a 40 % no reconhecimento de vozes que não participaram do treino, caindo essa taxa para cerca de 11% com vozes que participaram do treino.

A distribuição do erro pelas diversas palavras, para 10 locutores, duas amostras no treino apresentou-se da seguinte forma:

Tabela 7.M Matriz de confusão, dez vozes, treino com 2 amostras.

palavra do vocabulário	palavra errada reconhecida (nº de ocorrências)											% do total de erros
	0	1	2	3	4	5	6	7	8	9	10	
0												
1	1								1			16,7
2												
3			1				2				1	33,3
4												
5												
6				2								16,7
7												
8			1									8,3
9			1								1	16,7
10				1								8,3

Nota-se novamente a concentração do erro nos números “três”, “seis” e “dez”, e algumas trocas entre “oito” e “dois”. Nesses casos encontramos também em comum a semelhança da parte vocálica. No caso “dois”-”oito”, p.ex., a baixa energia do final da palavra “oito” deve ter apresentado classes que não permitiram a correta distinção. Apresentamos no Anexo A uma exposição detalhada dos modelos de “três” e “seis” gerados nesta fase (com duas amostras no treino) e a análise de dois erros decorrentes da troca no reconhecimento dessas palavras, o “três” pelo “seis” e vice-e-versa.

Parte desses erros, no entanto, pode ser resolvida no treinamento. Como exemplo, as duas amostras do dígito “um”, da locutora L, foram na fase de treinamento classificadas de modo errado. Isso conduziu ao reconhecimento errado também da terceira amostra. Realizando-se um treinamento interativo, o que não foi o nosso caso uma vez que as amostras haviam sido previamente gravadas, o sistema poderia ter identificado tal erro durante o treino e solicitado a repetição. Isto é possível, pois nessa fase se impõe qual é a palavra falada. Desse modo, se o sistema identifica uma elocução da palavra como confusa com algum outro modelo do sistema, a amostra pode ser rejeitada.

7.3.2.1.1 Efeitos do número de classes do quantizador e “floor method”

Do mesmo modo que no ítem 7.3.1.1., realizamos os testes do sistema com 10 locutores com duas amostras de cada locutor. Inicialmente, com relação ao número de classes, os resultados obtidos, apresentados nas tabelas do Anexo B - “a”, estão sintetizadas no gráfico da figura 7.7.

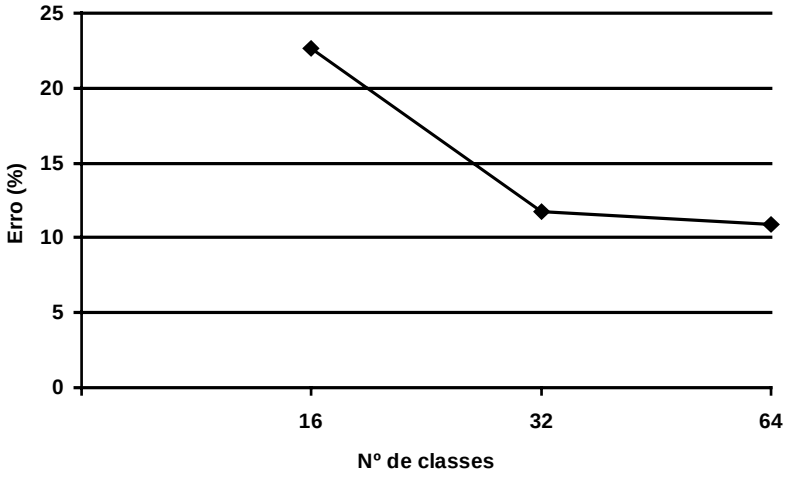


Figura 7.G Taxa de erro por número de classes, múltiplos locutores.

Observamos que o comportamento do sistema praticamente não se alterou quando reduzimos o número de classes para 32. Isto indica que outros fatores estão formando esse erro, de tal modo que, se o número de classes fosse aumentado acima desse valor o erro praticamente não iria se alterar. A questão é saber quais seriam estes fatores. A partir da análise dos erros realizada no Anexo A, levantamos algumas hipóteses que serviram de base para as propostas alternativas da próxima seção.

Com relação ao "floor method" descrito na seção 6.2.4, os resultados obtidos, apresentados nas tabelas do Anexo B - "b", confirmam os testes de RABINER [1983], i.e., a taxa de erro é praticamente insensível a esse parâmetro na faixa de 10^{-3} a 10^{-10}.

7.3.2.2.2 Propostas alternativas

Face aos resultados obtidos, consideramos mais pertinente desenvolver o trabalho através de uma análise das falhas e busca de aprimoramento do desempenho do sistema do que aumentarmos o grau de complexidade expandindo, por exemplo, o vocabulário. Nesse sentido realizamos algumas "variações sobre o mesmo tema", cujos resultados apresentamos a seguir.

a) incorporação da energia

Os sistemas convencionais baseados em HMM utilizam apenas os coeficientes de predição linear (ou de corrrelação parcial, ou ainda, o cepstrum[5]) no processo de quantização. A inclusão da energia no processo como uma das dimensões não pode ser implementada diretamente, por tratar-se de uma variável com ordem de grandeza diversa.

RABINER et al. [1984] propõem um método para incorporar a energia, que realiza o cálculo da distância do vetor dos coeficientes de predição e da energia separadamente, utilizando uma distância total que é a soma das duas, sendo a energia ponderada e transformada por uma função não linear. O trabalho utiliza como medida de distância a razão de máxima verossimilhança, mas a adaptação para a distância que utilizamos é direta.

A medida de distorção total para dois vetores é dada por:

$d_{2T} = d_2(\mathbf{x},\mathbf{y}) + \alpha f\,(d_2(E_x,E_y))$

onde $d_2(\mathbf{x},\mathbf{y})$ é o erro quadrático médio entre os coeficientes de predição linear, α um fator multiplicativo que pondera a medida de energia na distância total, f uma função não linear para reduzir a distorção para valores baixos de energia, e $d_2(E_x,E_y)$ o erro quadrático entre os valores de energia.

E_x representa o logaritmo da energia normalizado pelo valor máximo observado na amostra:

$E_x = 10 \log_{10} (E(0)) - 10 \log_{10} (E_{max})$

A função não linear tem a forma:

$f(x) = 0, \quad |x| \leq CLIP$

$\qquad x, \quad |x| > CLIP$

onde CLIP é um valor determinado experimentalmente.

Os resultados obtidos por Rabiner no reconhecimento de dez dígitos, utilizando uma amostra de cada dígito para treino e uma para teste, gravadas por 100 locutores através de linha telefônica, não apresentaram diferenças significativas. Os melhores resultados obtidos, com HMM de 10 estados e quantização de 128 e 256 classes, apresentaram taxas de erro de 3,0 % sem o uso da energia e 2,2 % incorporando-a, com os parâmetros acima α=0,1, e CLIP=6 dB. Utilizando os HMM de 5 estados e 64 classes (que são as características do sistema que

[5] Os coeficientes cepstrais, deduzidos a partir de uma transformação homomórfica do sinal, podem ser obtidos diretamente dos coeficientes LPC por recurssão. (RABINER et al., [1978]) . Estes coeficientes apresentam melhores características de quantização que os coeficientes PARCOR, particularmente se estes últimos tiverem valores próximos de um (MAKHOUL et al., [1985]).

estamos utilizando nesta fase) o sistema sem energia apresentou melhor desempenho, com taxas de 3,9 % contra 4,5 % quando incluída a energia.

Incorporamos a energia com base neste algoritmo utilizando, como um dos componentes de cada vetor, o valor de energia normalizado e ponderado pelo fator α. Verificamos que o valor de α que equilibrou a distorção da energia com a dos coeficientes de correlação parcial, considerados individualmente, foi 0,3. Nesta situação, constatamos que o valor de CLIP não apresentava influência significativa quando mantido na região de baixas energias. Mantivemos esse valor de corte em 1% da energia máxima. Os resultados obtidos, apresentados na tabela 7.14, indicam que não houve alteração significativa no erro quando incorporamos a energia (as coluna Erro s/E e Erro c/E indicam as taxas de erro por locutor sem e com a energia respectivamente).

Tabela 7.N Resultados comparativos incorporando a energia.

Loc	Tipo	Erro s/E (%)	Erro c/E (%)
A	M-A	0	9,1
B	M-A	9,1	9,1
E	F-A	9,1	9,1
F	M-A	0	9,1
G	M-I	9,1	9,1
L	F-A	18,2	9,1
M	F-A	18,2	9,1
N	M-A	9,1	18,2
S	M-A	18,2	18,2
T	F-A	18,2	9,1
Erro Médio		**10,9**	**10,9**

Tabela 7.O Matriz de confusão, incorporando a energia.

palavra do vocabulário	palavra errada reconhecida (nº de ocorrências)											% do total de erros
		0	1	2	3	4	5	6	7	8	9	10	
	0		1										8,3
	1				1						1		16,7
	2												
	3							1				4	41,7
	4												
	5												
	6								1				8,3
	7												
	8												
	9		1	1									16,7
	10				1								8,3

b) perda de início e fim das palavras

A análise visual da segmentação, feita por gráficos como os apresentados no Anexo A, levantou dúvidas quanto à influência do algoritmo de determinação de início e fim das palavras na taxa de erro.

Verificamos assim se a eventual presença de silêncio nas extremidades estaria prejudicando o sistema. As variações nos pontos de corte não apresentaram, no entanto, nenhuma alteração significativa. As pequenas variações nos segmentos iniciais ou finais das palavras são absorvidas pelo modelo. Utilizando-se o mesmo algoritmo no treino e no teste a presença ou não desses segmentos se manifesta nas *fdp* dos símbolos.

Deve-se, no entanto, procurar determinar o ponto de corte da forma mais confiável possível para evitar que o modelo absorva informações não relevantes.

c) busca extensiva pelos modelos individuais

A dificuldade do sistema no trato com as palavras "três", "seis", e "dez" nos levou à busca de uma outra forma de treinamento e reconhecimento. Identificamos o problema como uma dificuldade do modelo em tratar essas palavras que apresentam uma parte vocálica muito semelhante, dominando todo a parte central da palavra.

Decidimos então desenvolver um sistema de reconhecimento mais elaborado, realizando uma busca extensiva pelo universo de locutores. A partir dessa idéia, alguns procedimentos foram implementados.

Realizamos o treinamento de cada palavra para cada locutor isoladamente, obtendo assim um conjunto de 110 (11x10) modelos. O reconhecimento foi realizado examinando-se a amostra de teste diante do conjunto de modelos das palavras de cada locutor. Inicialmente se verificou qual o modelo (palavra) mais provável para aquele locutor, e a partir do conjunto das dez palavras mais prováveis (uma para cada locutor) decidimos, por uma regra de maioria, qual a palavra reconhecida. Em caso de empate, decidimos pelo conjunto que apresentava o modelo com a máxima probabilidade dentre todos.

Como uma seqüência deste método, buscamos um modelo misto, onde o modelo geral foi utilizado como desempate, i.e., em caso de empate acrescentamos a amostra do modelo geral, e decidíamos novamente por regra de maioria; permanecendo o empate utilizamos o critério anterior.

Os resultados obtidos estão apresentados nas tabelas 7.16 e 7.17, correspondendo ao treino realizado com 1 e 2 amostras respectivamente. A coluna Erro I apresenta os resultados no caso do desempate por máxima probabilidade e Erro II, para o desempate com modelo geral.

Tabela 7.P Taxas de erro, busca extensiva, treino com 1 amostra.

Loc	Tipo	Erro I	Erro II
A	M-A	18,2	13,6
B	M-A	22,7	22,7
E	F-A	40,9	40,9
F	M-A	27,2	27,2
G	M-I	18,2	22,7
L	F-A	18,2	27,2
M	F-A	40,9	20,8
N	M-A	45,4	50,0
S	M-A	54,5	54,5
T	F-A	40,8	36,3
Erro Médio		**32,9**	**31,8**

Tabela 7.Q Taxas de erro, busca extensiva, treino com 2 amostras.

Loc	Tipo	Erro I	Erro II
A	M-A	27,2	27,2
B	M-A	27,2	27,2
E	F-A	27,2	27,2
F	M-A	9,1	9,1
G	M-I	27,2	36,3
L	F-A	27,2	27,2
M	F-A	27,2	27,2
N	M-A	36,3	36,3
S	M-A	36,3	36,3
T	F-A	36,3	36,3
Erro Médio		**28,3**	**29,3**

A técnica não se mostrou satisfatória, uma vez que as taxas de erro foram bem mais elevadas, chegando a ser três vezes maiores do que as anteriormente obtidas apenas com o modelo geral. Tal fato pode ser explicado pelas seguintes razões:

- os modelos individuais foram treinados com um conjunto muito reduzido de amostras (1 ou 2 amostra por palavra), enquanto o modelo geral, por incluir todos os locutores, tinha 10 vezes mais amostras;
- a busca extensiva é realizada sobre modelos completamente estranhos à amostra; desse modo as amostras teste do locutor “X” são testadas contra modelos que não contém nenhum tipo de informação a seu respeito, decorrendo daí resultados imprevisíveis. Destaque-se que, para 10 locutores, 90% da busca é feito sobre esse conjunto “estranho”.

d) busca extensiva em caso de probabilidades próximas

Outro método foi implementado invertendo-se o princípio da proposta acima. Utilizou-se basicamente o modelo geral, entretanto, no caso de a probabilidade do segundo modelo mais provável ser muito próxima à do primeiro classificado, realizamos a busca extensiva. A motivação desta proposta baseia-se no fato de que cerca de 50% dos erros ocorriam nesses casos de probabilidades próximas.

As duas amostras reconhecidas erroneamente que analisamos no Anexo A exemplificam o problema. Para a seqüência da amostra de teste "três__B3" obtivemos os valores de probabilidade $\ln P(\mathbf{M_6}|\text{três\_\_B3}) = -121$ e $\ln P(\mathbf{M_3}|\text{três\_\_B3}) = -127$. Ou seja, esta amostra foi reconhecida como "seis" ($\mathbf{M_6}$), mas o valor (logaritmo) da probabilidade do modelo correto foi apenas 5% menor. Para a seqüência da amostra "seis_L3" esses valores foram $\ln P(\mathbf{M_3}|\text{seis\_L3}) = -156$ e $\ln P(\mathbf{M_6}|\text{seis\_L3}) = -159$, o que representa uma variação menor que 2%.

A proposta foi utilizar os modelos individuais como uma regra auxiliar de decisão nesses casos. Assim, quando os valores das probabilidades dos dois primeiros classificados não diferissem por mais que um determinado limiar (p.ex. 5%), efetuávamos a busca extensiva, sendo que:

a) se na busca extensiva encontrássemos como modelo mais provável o correspondente ao primeiro ou ao segundo classificado no geral, decidíamos por regra de maioria, escolhendo como dígito reconhecido o primeiro ou segundo, respectivamente.

b) se na busca extensiva não encontrássemos nenhum dos dois mais prováveis no modelo geral, escolhíamos o mais provável do modelo geral.

A busca extensiva era realizada como na seção anterior, i.e., calculadas todas as probabilidades, sendo a decisão feita por regra de maioria e, em caso de empate, decidia-se pela máxima probabilidade.

Os resultados são apresentados na tabela 7.18 (Erro 1 e Erro 2 correspondem ao treino com 1 e 2 amostras respectivamente).

Tabela 7.R Taxas de erro, busca extensiva para probabilidades próximas.

Loc	Tipo	Erro 1	Erro 2
A	M-A	9,1	9,1
B	M-A	9,1	0
E	F-A	13,6	9,1
F	M-A	18,2	18,2
G	M-I	13,6	9,1
L	F-A	13,6	9,1
M	F-A	9,1	18,2
N	M-A	22,7	9,1
S	M-A	13,2	27,2
T	F-A	27,2	9,1
Erro Médio		**15,3**	**12,2**

Estes resultados apresentaram alguma melhora apenas para o treino realizado com 1 amostra. O que constatamos foi que alguns erros foram corrigidos enquanto alguns acertos se perderam, i.e., no caso de dúvida o modelo individual não forneceu sempre a indicação correta.

e) "smoothing" pelo método das distâncias

Esta técnica, apresentada brevemente na seção 6.2.4, procura contornar o problema do conjunto finito de treinamento.[6] A idéia básica é aproximar o valor dos parâmetros b_{jk} da matriz **B** no caso de os vetores j e k serem próximos.

Para isso, aplicamos uma transformação na matriz **B**, da seguinte forma:

B' = **B** x **T**

onde **T** é uma matriz MxM, sendo

$$t_{i,j} = \begin{cases} 0, & \text{se } d_{ij} > P_1 \\ (P_1-d_{ij})/(P_1-P_2), & \text{se } P_2 \le d_{ij} < P_1 \\ 1, & \text{se } d_{ij} \le P_2 \end{cases}$$

Os valores de P_1 e P_2 foram tomados com base nas distâncias inter (s) e intra-centróides (D) definidas na seção 5.3, e d_{ij} é a distância entre as centróides i e j. Utilizamos inicialmente P_1=s-D e P_2=D, mas constatamos que os valores sofriam alteração excessiva. Verificamos experimentalmente que P_1=s/2 e P_2=D/2 forneciam um "smoothing" mais conveniente.

Desse modo, os vetores relativamente próximos tem seus valores de probabilidade b_{jk} ajustados, procurando-se dessa forma compensar a limitação no conjunto de treinamento.

[6] SCHWARTZ [1989] propõe uma matriz de transformação baseado numa função de Parzen, i.e., os parâmetros são alterados de acordo com uma matriz de transformação T, com $t_{ij}=\exp(-d^2/\sigma^2)$, onde d é a distância entre os centróides i e j. Optamos por uma formulação mais simples que permitisse manipular mais facilmente os valores de T.

A etapa de reconhecimento foi então realizada tomando-se apenas o modelo geral, substituindo-se nos modelos a matriz **B** por **B'**.

Os resultados apresentados na tabela 7.19 apresentam, como no teste anterior, alguma melhora para o treino com uma amostra e, novamente, enquanto alguns erros foram corrigidos, novos surgiram (Erro 1 e Erro 2 correspondem ao treino com 1 e 2 amostras respectivamente).

Tabela 7.S Taxas de erro, "smoothing" pelo método das distâncias.

Loc	Tipo	Erro 1	Erro 2
A	M-A	9,1	0
B	M-A	4,5	18,2
E	F-A	13,6	9,1
F	M-A	18,2	18,2
G	M-I	13,6	9,1
L	F-A	13,6	9,1
M	F-A	9,1	9,1
N	M-A	22,7	9,1
S	M-A	13,2	18,2
T	F-A	27,2	9,1
		144,8	
Erro Médio		**14,9**	**11,4**

f) limiar de rejeição

A verificação de que algumas amostras no reconhecimento apresentavam valores muito baixos de probabilidade nos levou a tentativa de estabelecer um limiar de rejeição. Desse modo eram descartadas as amostras que apresentavam valores abaixo de um determinado limite.

Utilizamos como limite o valor lnP < -170. Um valor menor praticamente não excluía qualquer amostra, enquanto um maior excluía muitas amostras reconhecidas corretamente.

Não obtivemos alteração sensível no desempenho, tendo observado inclusive a rejeição de amostras que teriam sido interpretadas corretamente. A questão é um pouco confusa pois, algumas vezes amostras que apresentam pequena probabilidade (ou seja, amostras "ruins") ainda tem informação suficiente para serem identificadas corretamente, enquanto outras possuem informação que traduzem para o sistema informação errada (de uma outra palavra), apresentando uma alta taxa de probabilidade, que estaria acima do limite proposto, não sendo portanto rejeitadas.

Isto indica que o modelo desta classe de rejeição, para ser adequado, deve ter uma formulação bem mais complexa.

7.3.3 Discussão

Elaboramos um quadro com os resultados obtidos, organizando em ordem decrescente a taxa de erro:

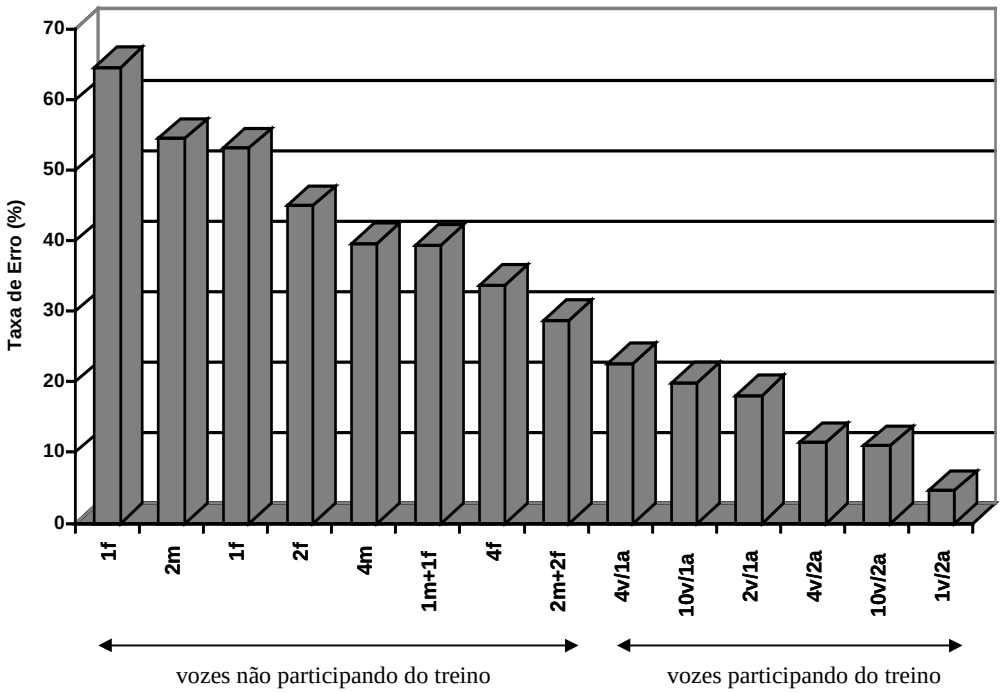


Figura 7.H Erro médio para os diversos testes.

Podemos dividir esse gráfico nas duas regiões indicadas, ou seja, a taxa de erro no reconhecimento de vozes que participaram ou não do treino. Percebe-se que tal sistema para ser absolutamente independente do locutor necessitaria uma quantidade muito significativa de vozes incluídas no universo do treino.

Já na parte que considera o reconhecimento das vozes que participaram do treino os resultados foram significativamente melhores. Verificamos que com apenas duas amostras de cada palavra obtivemos uma taxa de erro razoável, em torno de 10%. RABINER [1983] obteve taxas de erro de 4% em um sistema semelhante (10 locutores, 64 classes) para o reconhecimento dos dez dígitos na língua inglesa.

Devemos considerar que os resultados obtidos utilizaram apenas duas amostras de cada palavra no máximo. Retomando os resultados obtidos com apenas um locutor (7.3.1.2), verificamos que o sistema era bastante sensível ao número de repetições utilizados no treino. Trabalhamos com a hipótese de que, com um número maior de locutores, seriam necessárias menos repetições de cada palavra, uma vez que as características essenciais da palavra estariam presentes em todas as vozes, o que não ocorreu.

A utilização de um maior número de repetições no treino seguramente conduziria a resultados melhores. Com um vocabulário reduzido isso não apresenta maiores problemas. No entanto, se considerarmos um vocabulário um pouco mais extenso, um número excessivo de repetições transforma o treinamento em uma tarefa bastante trabalhosa. Um processo adaptativo parece ser mais razoável.

Esses resultados mostram a inviabilidade do sistema implementado, baseado em modelos ocultos de Markov discretos, na aplicação de reconhecimento de voz independente do locutor. As elevadas taxas de erro obtidas para um vocabulário e universo de locutores assim restrito deixam a desejar, indicando a necessidade de uma sofisticação no sistema, pela aplicação de modelos estatísticos ou de quantização mais elaborados.

Nas propostas alternativas realizamos, como dissemos, “variações sobre o mesmo tema”, i.e., não abandonamos em nenhum momento o príncipio do modelo oculto de Markov. O fato de não termos obtido nenhuma melhora indica que se deve considerar uma proposta que acrescente informação de outra natureza ou que modifique o modelo.

Essa é a tendência dos trabalhos mais recentes dentre os quais destacamos:

- a taxa variável de segmentação (variable frame rate - VFR), procura descartar segmentos muito semelhantes dando ênfase a fenômenos transientes; tal método tenderia a reduzir os erros que observamos em “três”, “seis” e “dez”; dentre os trabalhos mais recentes, LE CERF et al. [1994] propõe um método baseado no cálculo da norma das derivadas dos parâmetros, em oposição ao método do cálculo direto das distâncias inter-frames;
- os modelos ocultos de Markov com dependência intersegmentos (interframe dependent HMM), consideram, ao contrário do modelo convencional, uma dependência entre os segmentos sucessivos da palavra; nesse sentido citamos o trabalho de MING et al.[1994];
- o treinamento adaptativo dos parâmetros do HMM; um conjunto de informações a priori é utilizado para inferir, combinado com as observações, os parâmetros do modelo; HUO et al. [1995] apresentam um trabalho nesse sentido.

DI MARTINO et al. [1994] destacam a possibilidade de treinamento incremental possibilitado pela proposta de autômatos de estados finitos, similar às técnicas de inferência dos modelos ocultos de Markov de LOCKWOOD et al. [1993], em oposição ao modelo convencional.

Além destes, muitos trabalhos tem sido desenvolvidos utilizando como base os modelos ocultos de Markov. Mesmo os dois últimos citados ainda são, em essência, inspirados neste paradigma. A idéia dos modelos estatísticos duplos se mantém, até o presente momento, como a que tem apresentado mais resultados.

7.4 Sobre os tempos de processamento e memória

Uma das principais motivações do uso dos modelos ocultos de Markov para o reconhecimento de voz está na eficiência quanto ao tempo de reconhecimento e demanda de memória para armazenamento de modelos. No sistema que implementamos, não examinamos detalhadamente estes aspectos. Apenas quanto à memória, desenvolvemos alternativas para contornar as limitações impostas pelo PASCAL e pelo MS-DOS, evitando o processamento sobre memória não volátil.

Assim, os programas não foram implementados no sentido de otimizar o desempenho quanto ao tempo de treinamento e reconhecimento, mas de permitir a visualização de resultados intermediários para análise. No entanto, apresentamos alguns valores estimados para os tempos e memória requerida pelo sistema para ilustrar as ordens de grandeza envolvidas.

Cada palavra do vocabulário está representada pelos parâmetros das matrizes **A**, diferentes de zero apenas para $i=j$ e $i=j+1$, para $1 \leq i,j \leq 4$, (a_{55} sempre tem o valor 1), e da matriz **B**, 5xM, onde M é o número de classes da quantização vetorial. A matriz Π, de inicialização, tem sempre a mesma forma ($\pi_1=1$, $\pi_i=0$, $2 \leq i \leq 5$). Para 64 classes temos, então, 320 parâmetros da matriz **B**, além dos 8 da matriz **A**, totalizando 328 valores reais de 4 bytes, ou 1328 bytes. Esse valor pode ainda ser reduzido consideravelmente se armazenarmos apenas os valores de **B** maiores que o limiar imposto pelo "floor method". Utilizando um par de índices (dois bytes) correspondentes à linha e coluna dos valores de **B** acima deste limiar, e o valor do parâmetro correspondente, obtivemos uma redução da ordem de 60% no espaço de memória requerido, à custa de um tempo de montagem da matriz para aplicarmos o algoritmo de avaliação descrito em 3.1. Tal artifício foi necessário em alguns testes, p. ex., ao realizarmos a busca extensiva, quando havia 110 modelos na memória.

Quanto aos tempos de processamento, vamos dividir as fases de extração de atributos, geração do livro código, treinamento e reconhecimento. Todos os tempos correspondem a estimativas para micro-computadores com processadores 486-DX4/100. A amostragem e a análise por LPC são as únicas fases realizadas na placa DSP. A amostragem é realizada sobre 1,5 s. de sinal de voz compreendendo uma palavra delimitada por silêncios no intervalo. Para a análise LPC de toda a amostra estimamos um tempo de 2 s. Todas as outras fases são realizadas por programas em PASCAL. A determinação dos extremos da palavra é realizada pelo algoritmo descrito em 6.1 e demora cerca de 4 s. por amostra.

A geração do livro código é realizada sobre todo o conjunto de teste, já excluídos os silêncios. Para 64 níveis, com cerca de 1000 segmentos das amostras de teste, temos um tempo médio de 6 min. para a determinação dos centróides.

A classificação pode já ser tomada como subproduto da geração do livro-código na fase de treino. Para a fase de reconhecimento estimamos um tempo de 100 ms. por amostra.

O tempo de treinamento para o teste com 10 locutores e 2 repetições de cada uma das 11 palavras situa-se em torno de 45 min. Tal tempo tem dependência linear com o produto nº de amostras X nº de palavras. Desse modo, podemos estimar um tempo de 12 s. por amostra X palavra. O número máximo de iterações do procedimento de Baum-Welch que utilizamos é 50.

Para o reconhecimento, o tempo médio de busca, já realizada a análise LPC, segmentação e classificação é cerca de 10 ms. por modelo de palavra do vocabulário. No caso, com 11 palavras, o tempo de reconhecimento fica ao redor de 110 ms.

Estes valores servem para ilustrar que:

a) a eficiência do uso dos modelos ocultos de Markov na fase de reconhecimento se perde se a fase de extração de atributos não for otimizada, uma vez que a relação entre os tempos de processamento destas duas fases é da ordem de 1:500;

b) na fase de treinamento, freqüentemente atingimos o limite máximo de iterações, o que talvez não fosse necessário; as propostas de estimação por processos adaptativos, que partem de um modelo a priori, fornecem parâmetros bastante confiáveis com apenas duas re-iterações do algorítmo Baum-Welch (LEE, [1991]).

Estas questões devem ser consideradas na implementação de um sistema, particularmente o item “a”, no que diz respeito ao reconhecimento em tempo real. O seu estudo detalhado, no entanto, não estava no escopo deste trabalho.

8 Considerações finais

A idéia básica que fundamenta o uso de dois processos estatísticos inter-relacionados, como utilizamos neste trabalho, é a de que enquanto um modela a variabilidade dos ritmos de emissão da fala, o outro representa a diversidade das medidas físicas extraídas do sinal.

A validade de tal hipótese é demonstrada pelos resultados obtidos com um único locutor. Neste caso, atingimos o reconhecimento correto de todo o conjunto de teste utilizando quantização em 64 níveis e 5 repetições no treino de cada palavra.

Das limitações existentes nos sistemas de reconhecimento de voz mencionadas na introdução, trabalhamos apenas uma: a independência do locutor. O potencial do uso dos modelos ocultos de Markov nesta questão residiria no fato de que as variabilidades entre as vozes poderiam ser integradas em um único modelo através das funções de densidade de probabilidade de observação dos símbolos (a matriz **B**).

Em princípio, o uso desses modelos permitiria, com um número representativo do universo de vozes na fase de treino, o reconhecimento para qualquer locutor. Mas isto não é tão simples. Primeiro, pela determinação de qual seria tal conjunto significativo. Além disso, o aumento excesssivo do número de locutores traz também o problema da difusão dos parâmetros dos modelos. A representação da diversidade dos fenômenos acústicos para um grande número de locutores tende a dispersar os parâmetros da matriz **B** e, desse modo, pode criar modelos que se confundam na avaliação estatística. Finalmente, relembramos as considerações feitas nas observações preliminares (7.1), quando verificamos uma grande sensibilidade do sistema ao tipo de microfone e seu posicionamento, ou seja, a questão não é apenas o “tipo de voz”.

Assim, a idéia de criar com este sistema modelos robustos que pemitiriam o reconhecimento para qualquer voz em qualquer condição é inviável. As soluções para o problema da independência do locutor apontam para métodos adaptativos, onde um modelo básico é refinado para a voz que está utilizando o sistema. Tais métodos envolvem um integração do processo de quantização com uma re-estimação dos parâmetros.

Os trabalhos mais recentes tem procurado realizar esta adaptação durante o uso. Este é um aspecto importante, pois o grande diferencial apontado para o uso dos modelos ocultos de Markov - a vantagem computacional no reconhecimento - diminui se ponderada com a necessidade de repetição do treinamento.

Enfim, os sistemas de reconhecimento baseados em modelos ocultos de Markov devem procurar alternativas que compatibilizem a solução da independência do locutor com as

vantagens desses modelos. Do mesmo modo, a viabilidade de sistemas complexos, mais flexíveis quanto às restrições de discurso contínuo e vocabulário, dependem de soluções eficientes que permitam explorar o grande potencial deste método.

Anexo A

Nesta seção vamos apresentar de modo mais detalhado uma descrição das etapas do teste da seção 7.3.2.2. realizado com dez locutores e duas amostras no treinamento (Tabela 7.12, Erro2). Escolhemos este teste por ter sido o que envolveu maior diversidade de dados. Vamos nos limitar às palavras “três” e “seis”, sistematicamente reconhecidas erroneamente pelo sistema ao longo dos diversos testes.

As tabelas A.1 e A.2 apresentam os Vetores-Índices (VI) para as 20 amostras de cada uma dessas palavras. Essas seqüências, obtidas pelo processo de quantização, foram utilizadas para gerar os parâmetros dos modelos pelo algoritmo de Baum-Welch.

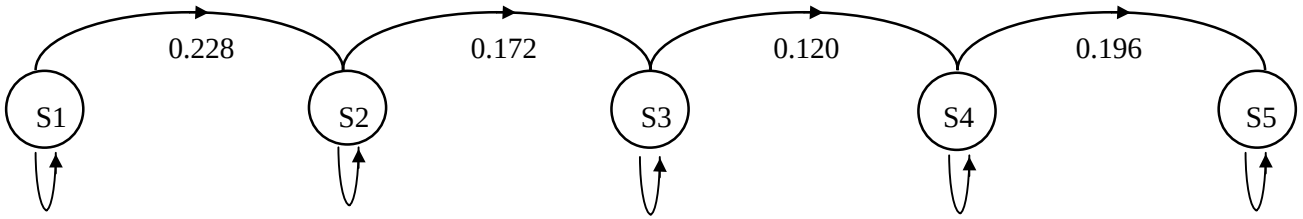
Os diagramas das figuras A.1 e A.2 representam os modelos correspondentes a essas duas palavras, mostrando como as seqüências se refletem nas probabilidades de observação dos símbolos nos estados. Apresentamos apenas as probabilidades maiores que 10^{-3} , valor utilizado como limiar para o “smoothing - floor method”, que impõe um valor mínimo para os símbolos não observados.

Em seguida, apresentamos análise de duas ocorrências de erros no reconhecimento. Em um dos casos a amostra de reconhecimento da palavra “três” do locutor B (três__B3) foi reconhecida como “seis”. No outro, o sistema reconheceu como “três” a palavra “seis” da locutora L (seis__L3).

São consideradas hipóteses com relação à causa de tais erros, as quais motivaram algumas das propostas alternativas realizadas na seção 7.3.2.2.2.

Tabela A.A Palavra “três” - Vetores-Índices.

A-1	9 57 43 43 53 13 45 43 43 43 43 43 13 29 57 57 57 49 25 25 25 25 25 7 57 39 23 47 47 55 15
A-2	47 55 47 47 61 31 7 53 43 43 13 43 43 59 52 36 38 52 20 43 61 61 11 61 57 57 57 49 57 25 57 23 31 31 55 31 47 55 23
B-1	53 43 53 13 11 11 20 20 29 29 29 29 61 29 29 29 13 29 27 29 29 29 21 57 49 41 49 57 49 49 39
B-2	55 55 47 23 15 15 55 39 43 43 53 43 20 20 11 27 29 29 29 29 29 13 29 29 29 29 13 53 53 21 57 49 57 57 49 49 49 49 47 15 55
E-1	45 52 28 28 28 28 60 60 60 60 35 35 35 35 45 21 49 5 49 1 1 17 17 15 33 57 49 45 55 34 53 15 23 15 23
E-2	39 28 28 44 28 28 60 60 60 60 60 60 35 35 12 27 45 21 17 17 9 49 49 9 1 17 49 53 47 47
F-1	53 17 39 31 31 61 47 7 13 20 20 13 27 29 29 59 27 27 43 43 13 13 57 11 49 57 57 25 25 57 25 57 23 47 47 47 47 39
F-2	53 57 20 20 13 29 29 29 29 29 29 43 20 43 57 29 57 57 57 9 9 33 25 25 25 49 39 23
G-1	23 15 55 47 55 39 44 27 11 36 36 36 36 12 12 36 12 8 8 12 35 35 35 12 3 21 21 21 21 21 21 9 37 39 23 23 47 47 7
G-2	23 23 55 55 41 7 17 44 36 44 36 36 61 3 19 12 19 12 8 8 8 35 35 12 60 35 35 21 21 21 21 5 5 37 39 39 23 55
L-1	15 63 55 47 23 23 15 39 44 45 45 3 3 3 3 3 3 19 3 3 3 3 3 3 19 3 35 8 19 3 35 35 35 35 12 60 27 45 39 23 39 7 41 41 7 15 7 39 15 23 47 23
L-2	55 63 15 15 47 7 28 27 55 45 3 3 3 3 3 3 3 45 3 3 3 35 27 12 8 3 3 3 3 35 35 44 45 61 61 41 41 7 41 9 9 9 53 39 55 39 23 47
M-1	55 23 39 15 23 47 23 15 3 36 44 27 12 19 19 19 19 12 12 60 19 51 21 37 37 37 37 37 37 37 37 37 5 9 7 7 41 39 39 55 55 23 23
M-2	23 39 23 15 23 23 15 31 44 44 3 35 19 19 19 3 19 12 12 60 19 51 37 5 37 21 37 37 5 5 5 5 37 5 37 37 41 55 41 41 51 51 41 41 41 41
N-1	45 54 20 20 52 20 20 20 20 20 20 20 44 61 41 23 9 57 57 1 1 1 1 1 33 33 49 23 23 57 55 55 55 47
N-2	15 15 47 23 55 39 15 39 52 20 22 20 38 52 20 20 20 20 20 27 57 57 25 41 25 1 33 1 33 25 25 25 23 23 41 41 7 41 41
S-1	55 47 47 55 47 51 37 7 1 43 31 31 13 13 13 45 61 45 45 45 45 59 13 49 33 33 33 25 33 33 25 25 25 25 17 49 2 49 15 15 15 23 47 15 55
S-2	23 17 43 31 13 13 13 53 55 45 45 27 3 45 27 59 57 33 33 33 33 1 1 25 7 23 41 15 7 7 23 23 7 41 41 41
T-1	39 53 36 36 8 8 8 8 52 8 8 8 8 19 12 12 29 29 21 21 5 21 5 5 9 5 5 5 9 5 5 5 5 53 39 37 37 53 39 39 47
T-2	23 23 47 55 39 39 23 36 8 8 36 8 40 19 8 8 19 35 35 12 35 29 21 21 37 5 9 9 9 9 5 17 9 5 9 9 23 23 39 55 25 9 37

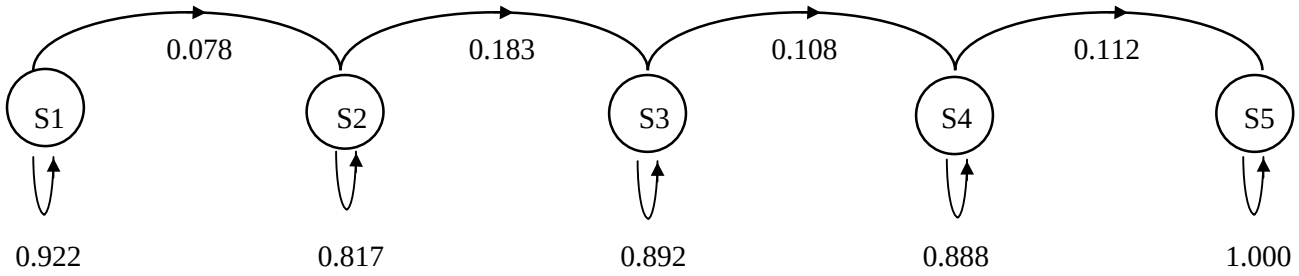


S1	S2	S3	S4	S5
P(03)=0.008	P(01)=0.008	P(03)=0.188	P(05)=0.010	P(01)=0.038
P(09)=0.011	P(03)=0.002	P(08)=0.114	P(11)=0.015	P(02)=0.003
P(15)=0.152	P(07)=0.041	P(12)=0.108	P(13)=0.070	P(05)=0.069
P(17)=0.018	P(11)=0.034	P(13)=0.004	P(20)=0.010	P(07)=0.041
P(23)=0.196	P(13)=0.089	P(19)=0.103	P(21)=0.183	P(09)=0.063
P(37)=0.003	P(17)=0.011	P(20)=0.010	P(27)=0.039	P(15)=0.038
P(39)=0.130	P(20)=0.191	P(27)=0.049	P(29)=0.308	P(17)=0.022
P(41)=0.010	P(22)=0.008	P(35)=0.143	P(43)=0.084	P(21)=0.004
P(45)=0.022	P(27)=0.011	P(36)=0.014	P(45)=0.015	P(23)=0.085
P(47)=0.140	P(28)=0.074	P(40)=0.006	P(49)=0.002	P(25)=0.076
P(51)=0.010	P(31)=0.057	P(44)=0.014	P(51)=0.018	P(31)=0.009
P(53)=0.037	P(36)=0.095	P(45)=0.087	P(53)=0.018	P(33)=0.047
P(54)=0.010	P(37)=0.006	P(52)=0.006	P(57)=0.106	P(34)=0.003
P(55)=0.163	P(38)=0.016	P(55)=0.011	P(59)=0.028	P(37)=0.069
P(57)=0.019	P(43)=0.106	P(60)=0.080	P(61)=0.046	P(39)=0.057
P(63)=0.022	P(44)=0.062	P(61)=0.015		P(41)=0.076
	P(47)=0.009			P(45)=0.004
	P(52)=0.049			P(47)=0.054
	P(53)=0.046			P(49)=0.072
	P(57)=0.002			P(51)=0.006
	P(59)=0.008			P(53)=0.016
	P(61)=0.028			P(55)=0.047
				P(57)=0.059

Figura A.A Representação do HMM da palavra “três” (M_3), com valores de $P(O_k|S) > 0.001$.

Tabela A.B Palavra “seis” - Vetores-Índices.

A-1	55 23 55 25 25 33 7 33 17 33 49 43 43 43 59 61 61 29 21 57 61 61 57 21 21 17 25 33 33 33 33 25 33 25 25 9 57 57 23 55 23
A-2	15 49 49 49 33 17 17 33 13 43 43 13 29 29 21 21 29 29 27 27 29 53 21 53 49 49 49 33 9 49 25 49 7 25 25 47 15 39 31 47
B-1	39 23 15 15 47 39 49 17 9 9 1 33 9 1 43 43 43 13 29 29 27 59 59 59 61 45 61 61 27 59 59 59 27 29 57 53 53 49 49 9 49 49 23 39 9 23 23 23 23
B-2	47 47 39 37 5 7 17 33 33 33 17 7 9 49 11 13 13 13 29 29 29 29 29 29 29 29 61 61 27 27 59 27 53 53 57 13 57 57 25 1 49 2 39 39 57 7
E-1	41 7 7 7 7 7 41 28 28 28 28 28 60 60 12 60 60 60 60 27 53 53 17 49 33 17 17 33 17 41 17 5 53 15 47 47 55 55 47 55
E-2	57 9 7 7 9 5 28 28 44 28 60 60 35 60 60 60 60 60 44 29 21 21 5 17 17 17 17 53 5 47 55 55 39 57
F-1	53 53 13 53 23 55 53 33 33 1 33 49 49 9 9 57 20 59 52 4 59 27 27 27 27 29 27 27 21 49 49 49 17 33 17 33 49 33 57 57 57 23
F-2	57 57 33 33 33 25 33 9 17 33 49 9 33 33 57 20 13 29 27 6 27 27 27 27 27 27 61 17 17 17 33 17 33 9 17 25 25 25 57 23 57
G-1	7 7 5 49 41 55 53 13 17 17 9 17 21 21 36 36 36 36 36 35 27 8 32 32 8 12 12 12 12 27 21 29 21 21 21 21 21 21 37 39 23 47 57 23 39
G-2	9 49 49 37 49 21 5 21 44 36 36 4 36 36 36 36 12 8 35 35 3 12 12 29 29 21 21 5 17 9 37 57 23 23
L-1	15 47 55 23 7 7 23 23 23 23 23 41 23 23 9 44 3 35 35 12 35 12 35 12 19 12 3 3 3 3 19 19 40 64 35 27 61 23 51 41 7 23 7 7 7 41 7 7 49 39
L-2	49 7 7 7 7 7 33 49 53 44 44 3 3 3 3 35 35 35 35 35 35 19 12 12 19 19 35 3 3 35 19 60 60 13 53 53 45 41 21 41 41 7 7 17 49 39 39 55 55 23
M-1	39 39 55 47 47 47 37 9 5 39 25 7 41 7 7 9 7 7 36 44 27 3 35 3 3 35 12 12 12 35 21 37 37 37 37 37 37 37 5 5 37 37 5 37 37 37 37 39 39 55 55 55 55 55 39
M-2	55 55 23 39 39 39 41 41 7 7 7 7 41 7 7 41 9 44 3 35 19 35 3 19 35 19 35 12 51 41 37 37 37 37 5 37 37 37 37 37 44 27 37 39 23 15 55 23
N-1	39 55 55 47 55 57 53 1 1 1 1 1 1 1 33 1 1 9 1 1 33 1 9 1 1 61 20 54 54 20 20 64 60 64 40 45 41 17 1 1 9 1 33 33 17 57 57 61 39 39 5 42
N-2	23 47 9 1 1 1 1 33 1 49 9 41 38 54 52 4 20 4 28 60 12 61 41 9 1 9 41 9 1 1 41 49 49 57 7 55 23 55 15 47
S-1	39 39 39 39 53 25 33 33 33 33 33 43 31 45 13 45 45 45 45 45 27 45 53 17 33 33 33 33 33 33 33 33 33 17 49 49 49 49 47 47
S-2	15 55 47 55 17 7 7 17 1 1 7 1 17 40 45 45 19 45 45 45 45 45 27 13 53 49 17 33 33 25 25 33 33 33 33 49 47 15 53 2 15 53 7 53 53 2 15 2 15 55 15 57 39 39
T-1	23 55 23 47 47 23 41 9 9 9 25 9 41 7 61 36 8 8 8 8 8 19 8 8 19 19 35 12 41 21 21 5 21 5 5 5 5 5 5 9 5 9 5 49 53 53 39 39 9 47 23
T-2	39 55 25 47 15 39 41 9 9 7 7 7 7 9 7 61 36 8 8 52 36 38 8 40 40 8 24 60 60 35 29 21 29 9 9 9 9 9 9 9 9 5 21 10 39 47 15 47 39



S1	S2	S3	S4	S5
P(01)=0.093	P(04)=0.035	P(03)=0.087	P(01)=0.033	P(01)=0.005
P(05)=0.013	P(05)=0.012	P(06)=0.005	P(05)=0.072	P(02)=0.023
P(07)=0.145	P(08)=0.077	P(08)=0.027	P(09)=0.100	P(05)=0.031
P(09)=0.100	P(11)=0.008	P(12)=0.096	P(17)=0.123	P(07)=0.067
P(13)=0.008	P(12)=0.012	P(13)=0.030	P(21)=0.104	P(09)=0.013
P(15)=0.022	P(13)=0.033	P(19)=0.072	P(23)=0.004	P(10)=0.005
P(17)=0.048	P(20)=0.052	P(21)=0.028	P(25)=0.050	P(13)=0.005
P(23)=0.060	P(21)=0.034	P(27)=0.143	P(33)=0.161	P(15)=0.056
P(25)=0.026	P(24)=0.009	P(29)=0.128	P(41)=0.056	P(17)=0.006
P(33)=0.104	P(28)=0.079	P(32)=0.010	P(45)=0.006	P(23)=0.107
P(37)=0.011	P(31)=0.008	P(35)=0.128	P(49)=0.117	P(25)=0.020
P(39)=0.060	P(35)=0.010	P(40)=0.006	P(51)=0.010	P(27)=0.006
P(41)=0.048	P(36)=0.132	P(45)=0.082	P(53)=0.078	P(31)=0.006
P(47)=0.048	P(38)=0.017	P(59)=0.043	P(57)=0.023	P(37)=0.141
P(49)=0.059	P(40)=0.033	P(60)=0.012	P(61)=0.014	P(39)=0.124
P(53)=0.030	P(43)=0.078	P(61)=0.048		P(41)=0.009
P(55)=0.056	P(44)=0.070	P(64)=0.006		P(42)=0.006
P(57)=0.022	P(52)=0.026			P(44)=0.006
	P(54)=0.026			P(47)=0.079
	P(59)=0.014			P(49)=0.024
	P(60)=0.147			P(53)=0.030
	P(61)=0.027			P(55)=0.096
				P(61)=0.005

Figura A.B Representação do HMM da palavra “seis” (M_6), com valores de $P(O_k|S) > 0.001$.

A amostra de reconhecimento três__B3 corresponde à palavra “três” do locutor B que utilizamos no reconhecimento. A figura A.3 apresenta os contornos de energia, taxa de cruzamento por zero e quantização vetorial (a altura das barras verticais corresponde ao Vetor-Índice). As duas linhas verticais indicam os pontos de início e fim da palavra determinados pelo algoritmo descrito na seção 6.1.

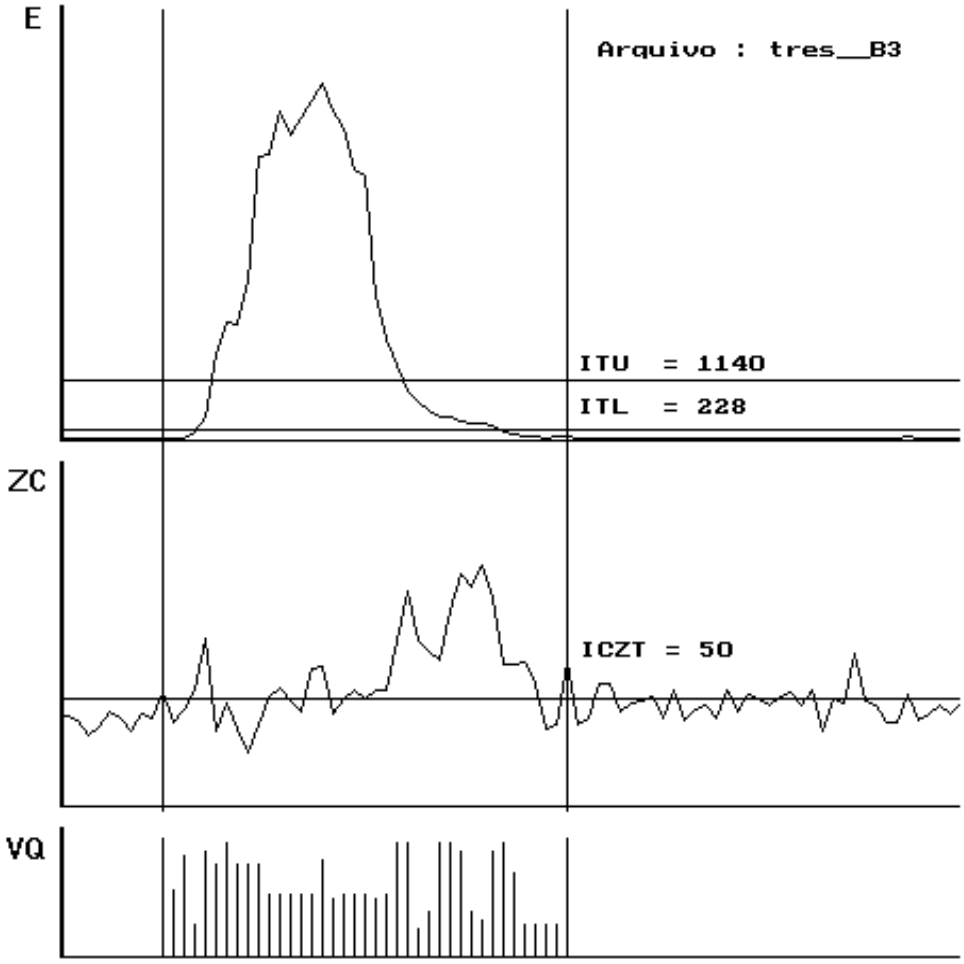


Figura A.C Contornos de energia, taxa de cruzamentos por zero, e VI de "três__B3".

Valores de VI:

55 31 47 15 49 43 53 43 43 43 29 29 29 29 29 45 27 29 29 29 27 29 53 53 13 21 53 53 49 21 17 49 53 39 15 15 15 15 55

ln P($\mathbf{M_6}$|três__B3)= -121

ln P($\mathbf{M_3}$|três__B3)= -127

A figura A.4 representa a energia, cruzamentos por zero e Vetores-Índices para a amostra seis_L3:

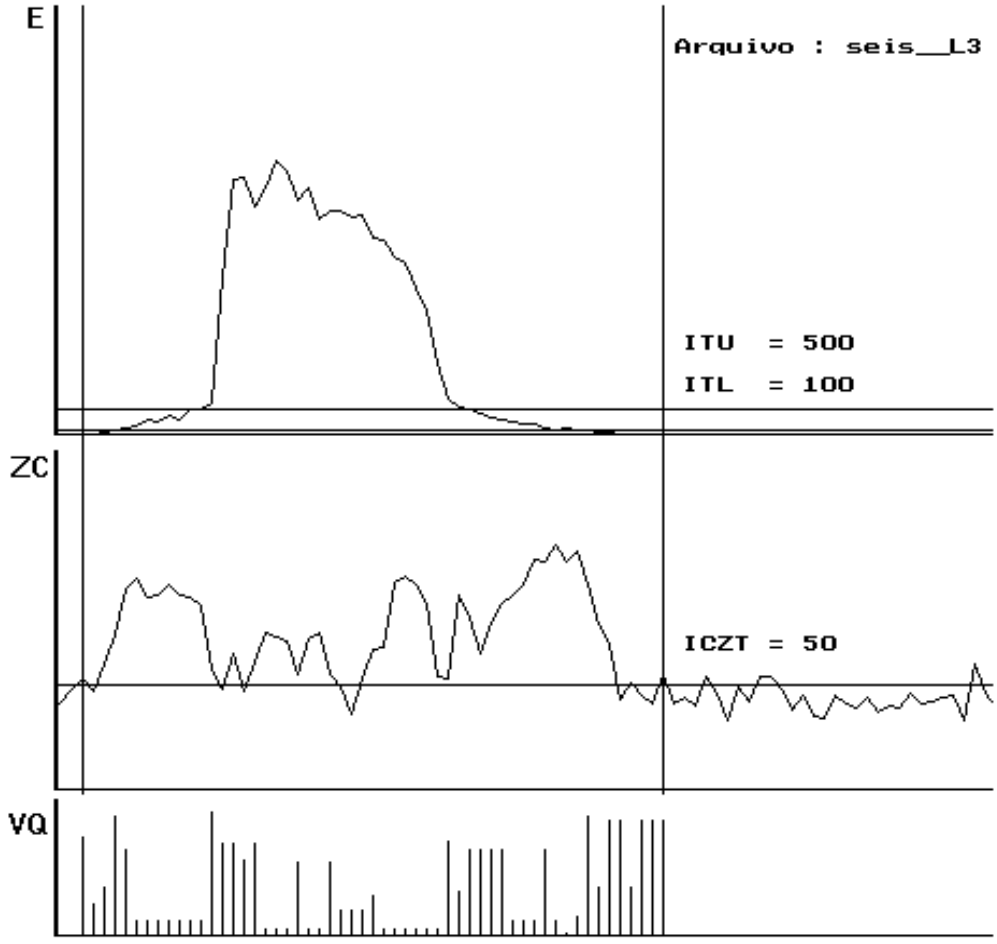


Figura A.D Contornos de energia, taxa de cruzamentos por zero, e VI de "seis_L3".

Valores de VI:

47 15 23 57 41 7 7 7 7 7 7 7 59 44 44 36 44 3 3 3 35 3 3 35 12 12 12 19 3 3 3 3 3 3 45 21 41 41 41 41 7 7 7 41 7 1 9 57 23 55 55 23 55 55 55

ln P($\mathbf{M_3}$|seis_L3)= -156

ln P($\mathbf{M_6}$|seis_L3)= -159

A partir destes dados podemos questionar inicialmente se o algoritmo de determinação dos extremos das palavras incluiu segmentos de silêncio no início e no final das mesmas. Esses segmentos de "provável silêncio" seriam os classificados como VI=55 e notamos que, em ambos os modelos (figuras A.1 e A.2), a probabilidade de observação desses símbolos é significativa, sendo até mesmo bastante elevada para o 1º estado de $\mathbf{M_3}$. Isto indica que o modelo incorporou esses símbolos, ainda que silêncios, a partir das amostras de treinamento que, como se observa nas tabelas A.1 e A.2, os apresentam nas extremidades de várias amostras.

Essas observações nos levaram aos testes realizados na seção 7.3.2.2.2, item "b", onde variamos os parâmetros do algoritmo de segmentação e notamos que a taxa de erro praticamente não se alterou. Em particular, para essas duas amostras, impusemos outros pontos de segmentação (tomados com base apenas na energia) mas o erro se manteve, i.e., as duas amostras continuaram a ser reconhecidas inversamente, o três como seis e vice-e-versa.

Desse modo, a possível falha na determinação do ponto de corte das palavras não parece ser a principal fonte de erro.

*

Observando as probabilidades - P(**M**|amostra) - verificamos que, em ambos os casos, a diferença entre o valor para o modelo correto e o reconhecido erroneamente foi muito pequena. Isto indica que essas amostras são confusas para o sistema ou, dito de outro modo, que os modelos dessas palavras não são capazes de extrair informações capazes de discriminá-las de forma clara. Tendo observado que essa proximidade dos valores de probabilidade ocorria freqüentemente em casos de erro, desenvolvemos a proposta da seção 7.3.2.2.2, ítem "d", procurando uma fonte alternativa de informação que, associada a um modelo de decisão mais complexo, produzisse resultados mais seguros.

*

O início da seqüência três__B3 (55 31 47 15 49 43 53 43...) apresenta o símbolo "49" que não foi observado no início de nenhuma das amostras do treinamento. Desse modo, o valor da probabilidade de observação desse símbolo nos primeiros estados do modelo $\mathbf{M_3}$ foi mantido em ε. Por outro lado, no modelo $\mathbf{M_6}$ a probabilidade desse símbolo é 0,059, o que pode ter induzido o sistema a erro. Retomando o que foi exposto em 6.2.4, temos um exemplo em que o "floor method" atribuiu a um símbolo possível mas pouco provável, a mesma probabilidade de um impossível.

Procuramos no livro código quais os centróides mais próximos do correspondente a esse símbolo, obtendo os VIs 9 e 17, com as distâncias inter-centróides $d_2(\mathbf{c_{17},c_{49}}) = 355$ e, $d_2(\mathbf{c_9,c_{49}}) = 379$. As medidas de qualidade desse quantizador foram s=948, D=406, s/D=2,33.

Verificamos que as duas distâncias são bem menores que a distorção média inter-grupos s, estando abaixo até mesmo da distorção média intra-grupos D. Esses dois símbolos (VI=9 e VI=17) estão presentes no 1º estado de $\mathbf{M_3}$, ainda que com valores baixos de probabilidade.

Quando tomamos, por outro lado, os símbolos mais prováveis nesse estado de $\mathbf{M_3}$, (VIs = 23, 39, 47 e 55), temos as distâncias inter-centróides $d_2(\mathbf{c_{23},c_{49}}) = 552$, $d_2(\mathbf{c_{39},c_{49}}) = 569$, $d_2(\mathbf{c_{47},c_{49}}) = 588$, e $d_2(\mathbf{c_{55},c_{49}}) = 630$. São valores também significativamente menores que a distorção s.

Estas constatações nos levaram a implementação do “smoothing” pelo método das distâncias realizado na seção 7.3.2.2.2, ítem “e”, que consiste em alterar os valores das probabilidades de observação dos símbolos de acordo com as distâncias inter-grupos. Neste exemplo, a aplicação deste método elevaria o valor da probabilidade para o VI=49 no 1º estado de $\mathbf{M_3}$, contornando o problema de este símbolo não ter sido observado no treino.

Anexo B

Apresentamos a seguir os resultados obtidos na seção 7.3.2.1, na qual examinamos o efeito do número de classes e do valor limite (ε) utilizado para o "smoothing" no desempenho do sistema. Nestes testes realizamos o treinamento com 10 locutores, utilizando duas amostras no treino e uma no reconhecimento. Nas tabelas, a coluna "Erro T" indica os erros (em nº de ocorrências) obtidos no próprio conjunto de treinamento enquanto a coluna "Erro R" corresponde aos verificados no reconhecimento. Os resultados do ítem "a" abaixo estão sintetizados na forma de taxa de erro no gráfico da figura 7.7. No ítem "b" a taxa de erro manteve-se praticamente constante, em torno de 10%.

a) variação do erro X nº de classes do quantizador.

a.1) 16 classes.

Tabela B.A Erro (nº de ocorrências) , 16 classes.

IdLoc	Tipo	Erro T	Erro R
A	M-A	1	2
B	M-A	1	0
E	F-A	4	1
F	M-A	1	0
G	M-I	4	3
L	F-A	5	3
M	F-A	7	6
N	M-A	1	4
S	M-A	5	3
T	F-A	5	3
Erro Médio		**34 (15%)**	**25(23%)**

10v., IdLoc = A, B, E, F, G, L, M, N, S, T; 2 am/pal
Tipo: M= masculina, F= feminina, A= adulto, I= infantil

Tabela B.B Matriz de confusão, 16 classes.

palavra do vocabulário	palavra errada reconhecida (nº de ocorrências)											total de erros
	0	1	2	3	4	5	6	7	8	9	10	
0								1				1
1			1			1			2			4
2									1			1
3							1				2	3
4												
5								1				1
6				2								2
7				1		1	2				1	5
8			1									1
9		1										1
10				3			3					6

a.2) 32 classes.

Tabela B.C Erro (nº de ocorrências), 32 classes.

IdLoc	Tipo	Erro T	Erro R
A	M-A	1	1
B	M-A	0	1
E	F-A	0	1
F	M-A	0	1
G	M-I	2	1
L	F-A	2	2
M	F-A	2	0
N	M-A	0	3
S	M-A	1	1
T	F-A	0	2
Erro Médio		**8(3,6%)**	**13(11,8%)**

10v., IdLoc = A, B, E, F, G, L, M, N, S, T;
2 am/pal
Tipo: M= masculina, F= feminina, A= adulto, I= infantil

Tabela B.D Matriz de confusão, 32 classes.

palavra do vocabulário	palavra errada reconhecida (nº de ocorrências)											total de erros
	0	1	2	3	4	5	6	7	8	9	10	
0				1								1
1					1				1			2
2			1									1
3			1				1				2	4
4												
5												
6												
7												
8												
9					1	1						2
10				3								3

a.3) 64 classes.

Tabela B.E Erro (nº de ocorrências), 64 classes.

IdLoc	Tipo	Erro T	Erro R
A	M-A		0
B	M-A		1
E	F-A		1
F	M-A		0
G	M-I		1
L	F-A	2	2
M	F-A		2
N	M-A		1
S	M-A	1	2
T	F-A		2
Erro Médio		**3(1,3%)**	**11(10,9%)**

10v., IdLoc = A, B, E, F, G, L, M, N, S, T;
2 am/pal
Tipo: M= masculina, F= feminina, A= adulto, I= infantil

Tabela B.F Matriz de confusão, 64 classes.

palavra do vocabulário	palavra errada reconhecida (nº de ocorrências)											total de erros
	0	1	2	3	4	5	6	7	8	9	10	
0												
1										1		1
2												
3			1				2				1	4
4												
5												
6				2								2
7												
8			1									1
9			1							1		2
10				1								1

b) taxas de erro X ε.

b.1) $\varepsilon = 10^{-3}$

Tabela B.G Erro (nº de observações), $\varepsilon = 10^{-3}$.

IdLoc	Tipo	Erro T	Erro R
A	M-A	0	1
B	M-A	0	2
E	F-A	0	2
F	M-A	0	1
G	M-I	1	0
L	F-A	0	0
M	F-A	0	1
N	M-A	0	1
S	M-A	0	1
T	F-A	1	1
Erro Médio		**2(1%)**	**10(9,1%)**

10v., IdLoc = A, B, E, F, G, L, M, N, S, T;
2 am/pal
Tipo: M= masculina, F= feminina, A= adulto, I= infantil

Tabela B.H Matriz de confusão, $\varepsilon = 10^{-3}$.

palavra do vocabulário	palavra errada reconhecida (nº de ocorrências)											total de erros
	0	1	2	3	4	5	6	7	8	9	10	
0												
1						1			1	1		3
2				1								1
3	1		1				1				1	4
4												
5												
6												
7												
8												
9												
10				2								2

b.2) $\varepsilon = 10^{-5}$

Tabela B.I Erro (nº de ocorrências), $\varepsilon = 10^{-5}$.

IdLoc	Tipo	Erro T	Erro R
A	M-A	0	0
B	M-A	0	1
E	F-A	0	1
F	M-A	0	1
G	M-I	0	1
L	F-A	0	2
M	F-A	0	1
N	M-A	1	2
S	M-A	2	2
T	F-A	1	2
Erro Médio		**4(2%)**	**13(11%)**

10v., IdLoc = A, B, E, F, G, L, M, N, S, T;
1 e 2 am/pal
Tipo: M= masculina, F= feminina, A= adulto, I= infantil

Tabela B.J Matriz de confusão, $\varepsilon = 10^{-5}$.

palavra do vocabulário	palavra errada reconhecida (nº de ocorrências)											total de erros
	0	1	2	3	4	5	6	7	8	9	10	
0												
1									2	1		3
2										1		1
3											2	2
4												
5												
6				2				2				4
7												
8		1	1									2
9												
10				1								1

b.3) $\varepsilon = 10^{-10}$

Tabela B.K Erros (nº de ocorrências), $\varepsilon = 10^{-10}$.

IdLoc	Tipo	Erro T	Erro R
A	M-A	0	2
B	M-A	1	2
E	F-A	0	1
F	M-A	0	1
G	M-I	0	1
L	F-A	0	1
M	F-A	0	1
N	M-A	0	1
S	M-A	0	1
T	F-A	0	1
Erro Médio		**1(0,5%)**	**12(10,9%)**

10v., IdLoc = A, B, E, F, G, L, M, N, S, T;
1 e 2 am/pal
Tipo: M= masculina, F= feminina, A= adulto, I= infantil

Tabela B.L Matriz de confusão, $\varepsilon = 10^{-10}$.

palavra do vocabulário	palavra errada reconhecida (nº de ocorrências)											total de erros
	0	1	2	3	4	5	6	7	8	9	10	
0		1										
1			1							2		3
2												
3								1			1	2
4												
5												
6				1				3				4
7												
8												
9												
10				1				1				2

Referências Bibliográficas

ALLERHAND, M. **Knowledge-based speech pattern recognition.** London, Kogan Page, 1987.

ARIEL CORPORATION. **DSP developer's toolkit for the Motorola DSP56001**: system overview. Highland Park, 1990.

BAKER, J.K. The DRAGON system: an overview. **IEEE Transactions on Acoustics, Speech and Signal Processing**, v.23, n.1, p.24-30, Feb. 1975.

BARUCHA-REID, A.T. **Elements of the theory of Markov processes and their applications.** New York, McGraw-Hill, 1960. Cap.1, p.9-56.

BORLAND INTERNATIONAL. **Turbo Pascal**: user's guide, version 5.0. Scotts Valley, 1989.

BORLAND INTERNATIONAL. **Turbo Pascal**: reference guide, version 5.0. Scotts Valley, 1989.

CRAVERO, M. et al. Syntax driven recognition of connected words by Markov models. In: IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, San Diego, 1984. **Proceedings.** New York, IEEE, 1984. v.3, p.35.5.1-4.

CREATIVE TECHNOLOGY LTD. **Voice Assist**. 3.ed. Singapore, 1993.

DAI, J. Hybrid approach to speech recognition using hidden Markov models and Markov chains. **IEE Proceedings Vision Image Signal Processing**, v.141, n.5, p.273-9, Oct. 1994.

DI MARTINO, J. et. al. Which model for future speech recognition systems: hidden Markov models or finite-state automata? In: IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Adelaide, 1994. **Proceedings.** New York, IEEE, 1994. v.1, p.633-5.

FAGUNDES, R.D.R. **Reconhecimento de voz, linguagem contínua, usando modelos de Markov.** São Paulo, 1993. 158p. Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo.

FENG, M.W. et al. Iterative normalization for speaker-adaptative training in continuous speech recognition. In: IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Glasgow, 1989. **Proceedings.** New York, IEEE, 1989. v.1, p.612-5.

FLAHERTY, M.J.; ROE, D.B. Orthogonal transformations of stacked feature vectors applied to HMM speech recognition. **IEE Proceedings**, Part I, v.140, n.2, p.121-6, Apr. 1993.

FLANAGAN, J.L. Computers that talk and listen: man-machine communication. **Proceedings of the IEEE**, v.64, n.4, p.405-15, Apr. 1976.

FRAGA, F.J. Implementação em tempo real de um reconhecedor de dígitos isolados. In: SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES, 9º, São Paulo, 1991. **Anais.** São Paulo, Escola Politécnica da Universidade de São Paulo, 1991. p.7.2.1-5.

FUKUNAGA, K. **Introduction to statistical pattern recognition.** 2ed. Boston, Academic Press, 1990.

GRAY, A.H.; MARKEL, J.D. Distance measures for speech processing. **IEEE Transactions on Acoustics, Speech and Signal Processing**, v.24, n.5, p.380-91, Oct. 1976.

GRAY, R.M. Vector Quantization. **IEEE ASSP Magazine**, p.4-29, Apr. 1984.

HUO, G. et al. Baysean adaptative learning of parameters of hidden Markov model for speech recognition. **IEEE Transactions on Speech and Audio Processing**, v.3, n.5, p.334-45, Sept. 1995.

IEEE ACOUSTICS, SPEECH, AND SIGNAL PROCESSING SOCIETY. **Programs for digital signal processing.** New York, IEEE, 1979. Cap.4.

ITAKURA, F. Minimum prediction residual principle applied to speech recognition. **IEEE Transactions on Acoustics, Speech and Signal Processing**, v.23, n.1, p.67-72, Feb. 1975.

JELINEK, F. Continous speech recognition by statistical methods. **Proceedings of the IEEE**, v.64, n.4, p.532-56, Apr. 1976.

JUNQUA, J.; WAKITA, H. A comparative study of cepstral lifters and distance measures for all poles models of speech in noise. In: IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Glasgow, 1989. **Proceedings.** New York, IEEE, 1989. v.1, p.476-9.

LABRIOLA, D. Straight talk. **Windows Sources**, v.3, n.2, p.144-60, Feb. 1995.

LE CERF, P.; VAN COMPERNOLLE, D. A new variable frame rate analysis method for speech recognition. **IEEE Signal Processing Letters**, v.1, n.12, p. 185-7, Dec. 1994.

LE ROUX, J.; GUEGUEN, C.A. A fixed point computation of partial correlation coefficients. **IEEE Transactions on Acoustics, Speech and Signal Processing**, v.25, n.3, p.257-9, June 1977.

LEE, K.F. **Automatic speech recognition: the development of the SPHINX system**. 2.ed. Norwell, Kluwer Academic, 1992.

LESSER, V.R. et al. Organization of Hearsay II understanding system. **IEEE Transactions on Acoustics, Speech and Signal Processing**, v.23, n.1, p.11-24, Feb. 1975.

LEVINSON, S.E.; RABINER, L.R.; SONDHI, M.M. An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. **Bell System Technical Journal**, v.62, n.4, p.1035-74, Apr. 1983.

LINDE, Y.; BUZO, A.; GRAY, R.M. An algorithm for vector quantizer design. **IEEE Transactions on Communications**, v.28, n.1, p.84-95, Jan. 1980.

LINDSAY, P.H.; NORMAN, D.A. **Traitement de l'information et comportement humain:** une introduction à la psychologie. Trad. de Réjean Jobin et al. Montreal, Études Vivantes, 1980.

LOCKWOOD, P.; BLANCHET, M. An algorithm for the inference of HMM (DHMM). In: IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Minneapolis, 1993. **Proceedings.** New York, IEEE, 1993. v.2, p.251-4.

MAKHOUL, J. Linear prediction: a tutorial review. **Proceedings of the IEEE**, v.63, n.4, p.561-80, Apr. 1975.

MAKHOUL, J.; ROUCOS, S.; GISH, H. Vector quantization in speech coding. **Proceedings of the IEEE,** v.73, n.11, p.1551-88, Nov. 1985.

MARKEL, J.D.; GRAY Jr., A.H. On autocorrelation equations as applied to speech analysis. **IEEE Transactions on Audio and Electroacoustics,** v.21, n.2, p.69-79, Apr. 1973.

MARTIN, T.B. Practical applications of voice input to machine. **Proceedings of the IEEE**, v.64, n. 4, p. 487-501, Apr. 1976.

MARIANI, J. Recent advances in speech processings. In: IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Glasgow, 1989. **Proceedings.** New York, IEEE, v.1, p.429-40, May 1989.

MINAMI, M. **Reconhecedor de palavras isoladas, independente do falante, usando HMM discreto.** São Paulo, 1993. 124p. Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo.

MING, J.; SMITH, F.J. Isolated word recognition using interframe dependent hidden Markov models. **IEEE Signal Processing Letters**, v.1, n.12, p. 188-90, Dec. 1994.

NAKAMURA, S.; SHIKANO, K. Speaker adaptation applied to HMM and neural networks. In: IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Glasgow, 1989. **Proceedings.** New York, IEEE, 1989. v.1, p.89-92.

O'SHAUGNESSY, D. **Speech communication:** human and machine. Reading, Addison-Wesley, 1987.

PAPOULIS, A. **Probability, random variables, and stochastic processes.** Tokyo, McGraw-Hill, 1965.

RABINER, L.R. et al. An algorithn for determining the endpoints of isolated utterances. **Bell System Technical Journal**, v.54, n.2, p.297-315, Feb. 1975.

RABINER, L.R.; SCHAFER, R.W. **Digital processing of speech signals.** Englewood Cliffs, Prentice-Hall, 1978.

RABINER, L.R.; LEVINSON, S.E.; SONDHI, M.M. On the application of vector quantization and hidden Markov models to speaker independent isolated word recognition. **Bell System Technical Journal,** v.62, n.4, p.1075-105, Apr. 1983.

RABINER, L.R. et al. A vector quantizer incorporating both LPC shape and energy In: IEEE International Conference on Acoustics, Speech and Signal Processing, San Diego, 1984. **Proceedings.** New York, IEEE, 1984. v.2, p.17.1.1-4.

RABINER, L.R.; SOONG, F.K. Single-frame vowel recognition using vector quantization with several distance measures. **AT&T Technical Journal,** v.64, n.10, p.2319-30, Dec. 1985.

RABINER, L.R.; JUANG, B.H. An introduction to hidden Markov models **IEEE ASSP Magazine,** p. 4-16, Jan. 1986.

RABINER, L.R. A tutorial on hidden Markov models and selected applications in speech recognition **Proceedings of the IEEE**, v.77, n.2, p. 257-86, 1989.

REDDY, D.R. Speech recognition by machine: a review. **Proceedings of the IEEE**, v.64, n.4, p.501-31, Apr. 1976.

SANCHES, I. **Reconhecedor de dígitos isolados independente do locutor.** São Paulo, 1989. 77p. Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo.

SCHWARTZ, R. et al. Improved hidden Markov modeling of phonemes for continuous speech recognition. In: IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, San Diego, 1984. **Proceedings.** New York, IEEE, 1984. v.3, p.35.6.1-4.

SCHWARTZ, R. et al. Robust smoothing methods for discrete hidden Markov models. In: IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, Glasgow, 1989. **Proceedings.** New York, IEEE, 1989. v.1, p.548-51.

TATTERSALL, G.D. et al. Neural arrays for speech recognition. In: WHEDDON, C.; LINGGARD, R. **Speech and language processing.** 1ed. London, Chapman and Hall, 1990. p. 245-90.

VIEIRA, M.N. **Módulo frontal para um sistema de reconhecimento automático de voz.** Campinas, 1989. 122p. Dissertação (Mestrado) - Faculdade de Engenharia Elétrica - Universidade Estadual de Campinas.

VITERBI, A.J. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. **IEEE Transaction on Information Theory**, v.13, n.2, p.260-9, Apr. 1967.

WATTERSON, K. Voice input today. **Windows Sources**, v.3, n.2, p.192-94, Feb. 1995.

WILPON, J.G. et al. An improved word-detection algorithm for telephone quality speech incorporating both syntatic and semantic constraints. **AT&T Bell Labs Technical Journal**, v.63, n.3, p.479-98, Mar. 1984.

WILPON, J.G. et al. A modified K-means clustering algorithm for use in isolated word recognition. **IEEE Transactions on Acoustics, Speech and Signal Processing**, v.33, n.3, p.587-94, June 1985.

WILPON, J.G. et al. Automatic recognition of keywords in unconstrained speech using hidden Markov models. **IEEE Transactions on Acoustics, Speech and Signal Processing**, v.38, n.11, p.1870-8, Nov. 1990.