Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Comunicações

Segmentação Automática e Treinamento Discriminativo Aplicados a um Sistema de Reconhecimento de Dígitos Conectados

por

Fabrício Lira Figueiredo
Eng. Eletricista (UFPE, 1995)

Orientador: Prof. Dr. Fábio Violaro
DECOM – FEEC - UNICAMP

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

Banca Examinadora:

Dr. Fábio Violaro – FEEC/UNICAMP - Presidente

Dr. Luís Geraldo Pedroso Meloni – FEEC/UNICAMP

Dr. Abraham Alcaim – CETUC/PUC-Rio

Campinas, 17 de dezembro de 1999

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

Figueiredo, Fabrício Lira
F469s Segmentação automática e treinamento discriminativo aplicados a um sistema de reconhecimento de dígitos conectados / Fabrício Lira Figueiredo.--Campinas, SP: [s.n.], 2000.

Orientador: Fábio Violaro
Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Reconhecimento automático da voz.. 2. Redes neurais (Computação). 3. Markov, Processos de. 4. Processamento de palavras. 5. Algorítmos. I. Violaro, Fábio. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Resumo

Os Modelos Ocultos de Markov constituem, atualmente, a principal abordagem para o problema de Reconhecimento de Fala, pois proporcionam bom desempenho e alto grau de flexibilidade. Infelizmente, este modelo acústico não é ideal e alguns problemas afetam sua robustez e desempenho em condições adversas.

A inconsistência do modelamento temporal implícito nos HMM's é um exemplo de um sério problema sem soluções bem definidas. De fato, o Modelo de Duração de Estados com distribuição exponencial é incompatível com o comportamento estatístico das unidades lingüísticas reais. A hipótese de independência entre observações representa outra limitação dos HMM's, já que não se verifica nos experimentos práticos. De fato, existe forte dependência contextual no caso de quadros pertencentes a regiões de transição entre unidades acústicas de uma elocução. Alguns modelos e algoritmos têm sido propostos para tentar transpor estes obstáculos, tais como Modelos Segmentais e Duração Explícita de Estados. Nesta tese, uma estratégia alternativa é proposta para atenuar estes problemas, sem acréscimos significativos no custo computacional. A informação relativa às transições entre fones, ao longo de uma elocução, é obtida através de métodos de segmentação automática. Realiza-se uma ponderação no algoritmo de Viterbi, a fim de penalizar os modelos que gerarem segmentações inconsistentes. Bons resultados são obtidos, para várias condições relacionadas a uma aplicação de Dígitos Conectados. O objetivo atual é aplicar esta técnica para o caso de vocabulários extensos.

Uma outra classe de problemas dos HMM´s consiste na estimação dos parâmetros que compõem cada modelo, a fim de minimizar as taxas de erro de reconhecimento do sistema. O algoritmo de treinamento convencional (Baum-Welch) se baseia no critério da Máxima Verossimilhança, e não pode garantir boas propriedades discriminativas para os modelos obtidos, resultando em perda de desempenho à medida que o número de modelos aumenta. Nesta tese, os aspectos teóricos da estratégia de Treinamento Discriminativo, recentemente proposta na literatura, são analisados e os algoritmos são descritos em detalhes.

Finalmente, foi proposto um algoritmo para estimar automaticamente os parâmetros que compõem o Fator de Ponderação Temporal empregado no algoritmo de Viterbi, quando utilizando informação de segmentação.

Abstract

Hidden Markov Model is actually the main approach to Speech Recognition problem, because of the good performance and high degree of flexibility that can be achieved. Unfortunately, this acoustical modeling is not optimum and some problems still affect it's robustness and performance in a more realistic condition.

The weakness of the temporal modeling embedded in HMM is an example of a serious problem without well defined solutions. In fact, the implicit state duration model with exponential distribution may not describe the real linguistic units distributions. The hypothesis of independence between observations is other difficult problem to solve and it is incompatible with practical experiments because there is strong correlation between frames in the same acoustic segment. Some models and algorithms have been proposed to overcome or, at lest, attenuate those problems, such as Stochastic Segment Models and Explicit State Duration. This thesis presents an alternative approach to alleviate these problems, with relatively low computational cost. The information on phoneme boundaries in time is obtained through an automatic segmentation algorithm and it is used in a Weighted Viterbi Algorithm in order to penalize the models that generates inconsistent segmentations. Good results were achieved for various conditions related to connected digits application. The actual objective is to expand it to continuous speech recognition.

Another class of problems in HMM is the estimation of the parameters that compose each model to obtain as higher recognition accuracy as possible. The traditional training algorithm (Baum-Welch) is based on Maximum Likelihood criterion and may not guarantee good discriminative properties to the obtained models. This results in a lost of performance when the number of models is increased. In this thesis, the theoretical aspects of the recently proposed Discriminative Training approach are analyzed and the algorithms for different cases are described in details.

Finally, an algorithm based on Discriminative Training is proposed to estimate some parameters that composes the Weighting Factor used in the Viterbi algorithm when using segmentation information (as described above).

À minha família,

Aldvan, Malena e Lucas.

Agradecimentos

Primeiramente, agradeço a Deus por ter me iluminado ao longo deste período e agradeço por todos os obstáculos que ele, sabiamente, colocou em meu caminho para que eu pudesse evoluir como pessoa e como profissional.

Agradeço imensamente à minha família, Aldvan, Malena e Lucas, por terem me apoiado em todos os momentos, me dando forças para continuar em frente nos momentos difíceis, apesar da distância que nos separou durante boa parte deste tempo. Dedico cada linha, cada pensamento, cada emoção contida neste trabalho a estas pessoas tão especiais. Agradeço à minha esposa, Aldvan, por sua imensa bondade e paciência, bem como por sua presença em minha vida.

Agradeço a meus pais, por todos os ensinamentos que me deram ao longo de toda a minha vida e pelo carinho e amor a mim dispensados. Agradeço por terem me ajudado a superar todas as dificuldades e por terem apoiado a minha família, com sua presença e suas atitudes construtivas.

Agradeço aos pais de minha esposa, Marlene e Arnóbio, por toda ajuda que nos deram durante este período e, principalmente, pela dedicação aos meus filhos, amenizando sua tristeza devido à separação de nossa família.

Agradeço ao meu orientador, Prof. Fábio Violaro, por ter acreditado em meu potencial e por ter me incentivado em todos os trabalhos que tive oportunidade de realizar ou participar, dentro da Universidade. Agradeço pela paciência em ouvir minhas propostas e esclarecer minhas dúvidas, bem como por ter participado intensamente dos momentos decisivos deste trabalho.

Agradeço aos grandes amigos Luís, Leonardo, Eduardo Massato, Flávio, Edmílson, Carlos, Antônio Marcos, Raquel e Irene, pelos diversos momentos de alegria e pelas várias discussões técnicas e sugestões que tanto me ajudaram na realização desta tese. Agradeço especialmente ao amigo Antônio e sua família, por todo apoio e carinho para comigo e minha família. Agradeço aos amigos Zander, Luís Ósis, Oséas, Hugo e Ricardo pela saudável convivência, por todos os conhecimentos que foram produzidos e pelos vários ensinamentos que me foram transmitidos em nossas reuniões. Agradeço à Noêmia por toda sua paciência e por sua competência em resolver os problemas burocráticos que tanto atormentam a nós, estudantes.

Finalmente, quero agradecer ao CNPQ por ter financiado e tornado possível a realização deste trabalho.

Sumário

1. Introdução

1.1. Reconhecimento de Fala e Modelos Acústicos

As pesquisas na Área de Reconhecimento Automático de Fala vêm se desenvolvendo ao longo das últimas três décadas e, apesar de todos os esforços, ainda existem muitos problemas a serem resolvidos.

Atualmente já estão disponíveis alguns sistemas comerciais, para determinados tipos de aplicações, que apresentam bom desempenho dentro de condições específicas. Contudo, a capacidade de reconhecimento destes sistemas ainda está muito abaixo da capacidade humana, principalmente quando se considera a presença de condições adversas, tais como ruídos, distorções de canal, variações de pronúncia, etc.

Um dos principais problemas relacionados ao Reconhecimento de Fala está no modelamento das características do sinal de voz que permitem sua inteligibilidade, ou seja, a determinação inequívoca das unidades lingüísticas que compõem uma elocução. Os Modelos Acústicos constituem a base do sistema de reconhecimento e o desempenho final obtido está intrinsecamente ligado à sua capacidade de lidar com as variações acústicas que ocorrem entre locutores distintos e até para um mesmo locutor.

Os Modelos Ocultos de Markov (Hidden Markov Models - HMM's) têm sido empregados com êxito em várias aplicações envolvendo reconhecimento de palavras isoladas ou conectadas, bem como de fala contínua. Estes modelos apresentam várias características que os tornam capazes de representar de forma consistente as variabilidades acústicas do sinal de fala. A consistência dos HMM's se deve, em grande parte, à sua forte fundamentação estatística, que permite que a definição, análise e resolução dos problemas de uma determinada aplicação possam ser tratados de maneira mais formal, através dos princípios da Teoria da Probabilidade.

Entretanto, os HMM's apresentam algumas limitações que têm sido objeto de trabalho de vários grupos de pesquisa em Reconhecimento de Fala. Um exemplo é seu modelo de duração de estados, que é implicitamente representado por uma distribuição exponencial e não descreve corretamente as reais características temporais das unidades acústicas. Outro exemplo é a necessidade de assumir a hipótese de independência entre quadros, que não se verifica na prática, principalmente entre quadros pertencentes às regiões de transição entre fones de uma elocução, que apresentam forte dependência contextual.

Neste trabalho é proposta uma estratégia para minimizar os problemas ocasionados pelo modelo de duração de estados dos HMM's. Utiliza-se a informação de variação espectral ao longo do tempo, fornecida por alguns sistemas de Segmentação Automática, para ponderar o algoritmo de Viterbi, durante a fase de reconhecimento. Define-se, então, o Fator de Ponderação Temporal, que penaliza os modelos que geram segmentações parciais que não combinam com a informação de variação espectral obtida. Com isto, aumenta-se a precisão da segmentação em fones gerada a partir dos HMM's e diminui-se a taxa de erros de reconhecimento do sistema. Adicionalmente, obtém-se um segmentador automático mais preciso, abrindo caminho para novas aplicações, tais como Síntese e Codificação de Voz, bem como Segmentação Automática de Bases de Dados.

Outro problema que atinge os Modelos Ocultos de Markov está na etapa de treinamento dos modelos, na qual são estimados os parâmetros que compõem cada um dos modelos do sistema. O algoritmo mais empregado para realizar o treinamento dos HMM's é denominado Baum-Welch e se baseia no critério da Máxima Verossimilhança. Um problema desta abordagem é que o algoritmo não apresenta propriedades discriminativas, ou seja, os parâmetros são estimados de modo a maximizar a probabilidade do modelo correto gerar a elocução de treinamento correspondente, mas não é levada em consideração a necessidade de minimizar a

probabilidade de que os demais modelos (incorretos) gerem tal elocução. Outro problema está na necessidade de bases de dados extensas, uma vez que se trata de um processo de estimação estatística. Desta forma, tornou-se necessário desenvolver algoritmos de Treinamento Discriminativo, a fim de permitir a obtenção de modelos mais adequados.

Recentemente, esta abordagem tem se firmado devido à utilização do algoritmo Segmental GPD [Chou92], que consiste em um processo de otimização de uma função de custo relacionada com a taxa de erros de reconhecimento do sistema. Neste trabalho, serão descritos em detalhes os aspectos teóricos relacionados à aplicação deste algoritmo em Reconhecimento de Palavras Isoladas e Reconhecimento de Fala Contínua.

1.2. Objetivos

Esta tese tem como objetivos principais:

- Propor e validar o emprego de técnicas de Segmentação Automática para atenuar os problemas decorrentes do modelo de duração de estados inadequado dos HMM's;
- Formalizar o problema da Segmentação Automática da Fala e descrever as principais técnicas encontradas na literatura;
- Descrever detalhadamente o algoritmo de Treinamento Discriminativo para Palavras Isoladas e para Fala Contínua;
- Descrever um procedimento baseado no algoritmo de Treinamento Discriminativo para a estimação automática dos parâmetros empíricos que compõem o Fator de Ponderação Temporal (utilizado para introduzir a informação de segmentação durante a etapa de reconhecimento).

1.3. Estrutura da Tese

No Capítulo 2, os principais algoritmos relacionados aos Modelos Ocultos de Markov são descritos com maior grau de detalhamento. Desta forma, é analisado o algoritmo de treinamento (Baum-Welch), que se baseia no critério de Máxima Verossimilhança, bem como os algoritmos de Viterbi e Level-Building.

No Capítulo 3 da tese são analisados vários aspectos relacionados ao problema de modelamento temporal dos HMM's. São relacionadas as principais abordagens descritas na literatura para atenuar estes problemas, tais como os Modelos de Duração Explícita e os Modelos Segmentais. Em seguida, propõe-se a utilização de medidas de variação espectral ao longo do tempo como uma estratégia alternativa para minimizar os efeitos do modelo exponencial de duração de estados dos HMM's. Faz-se, então, uma descrição das principais técnicas de segmentação automática descritas na literatura, bem como a devida formalização do problema da Segmentação Automática da Fala. Por fim, define-se o Fator de Ponderação Temporal, que é empregado ao longo do Level-Building para penalizar os modelos que produzirem segmentações desalinhadas com a informação de variação espectral.

No Capítulo 4, o problema do Treinamento Discriminativo de HMM's é abordado. Propõe-se a adoção do algoritmo Segmental GPD para estimar os parâmetros dos modelos, de modo a minimizar uma função objetiva que corresponde a uma estimativa suavizada da taxa de erros do sistema. Trata-se de um algoritmo do tipo Gradiente Descendente, cujas propriedades serão descritas em detalhes. Aplica-se então este algoritmo aos problemas de Reconhecimento de Palavras Isoladas (empregando HMM's Contínuos) e Reconhecimento de Fala Contínua (empregando HMM's Discretos). Por fim, utiliza-se o algoritmo Segmental GPD para estimar os parâmetros empíricos que compõem o Fator de Ponderação Temporal.

No Capítulo 5 são analisados os resultados obtidos a partir das simulações realizadas no laboratório. Inicialmente, caracteriza-se o Sistema Básico a partir de resultados obtidos com diferentes combinações de parâmetros de entrada. Os resultados se baseiam na taxa de acerto percentual de frases e palavras, nas taxas de erros de inserção e deleção de palavras, bem como nas medidas de distorção de segmentação. Em seguida, são analisados os resultados obtidos com a introdução da informação de segmentação gerada a partir da Filtragem Paramétrica e das

MLP's. Finalmente, são descritos alguns aspectos práticos dos algoritmos de treinamento discriminativo, obtidos a partir de experimentos divulgados na literatura.

No Capítulo 6, as conclusões a respeito do emprego das técnicas propostas neste trabalho são enumeradas e analisadas. São ressaltados os pontos positivos e negativos das abordagens propostas e são propostas estratégias para a continuação do trabalho. Vale salientar que não foi possível implementar o algoritmo de Treinamento Discriminativo, pois, como será mostrado no Capítulo 4, seria necessário implementar algoritmos de busca do tipo "stack", que não estão disponíveis atualmente no Laboratório de Processamento Digital da Fala (LPDF) da UNICAMP. Desta forma, esta implementação tornou-se incompatível com o tempo disponível para a realização desta tese. Fez-se, entretanto, todo o desenvolvimento teórico necessário para permitir a realização de futuros trabalhos nesta área.

2. Modelos Ocultos de Markov

2.1. Fundamentação Teórica

Os Modelos Ocultos de Markov correspondem ao modelo acústico mais empregado em Reconhecimento de Fala. Esta abordagem se baseia na Teoria dos Processos de Markov, utilizada para modelar processos estocásticos, geralmente não-estacionários e constituídos por uma seqüência de processos estacionários (estados). Neste caso, o modelo é composto por uma rede de estados finita caracterizada por uma arquitetura e pelas probabilidades de transição de estados a_{ij} (para os modelos de primeira ordem):

$$a_{ij} = P(q_t = j \mid q_{t-1} = i) = P(q_t = j \mid q_{t-1} = i, q_{t-2} = k, \ldots)$$

onde q_t designa o estado do modelo associado ao instante t.

Vale salientar que as probabilidades de transição de estados são responsáveis, em última instância, pelo modelamento das variabilidades temporais associadas aos padrões de voz.

No caso dos Modelos Ocultos de Markov, descreve-se adicionalmente um processo estocástico não observável associado à variabilidade espectral dos padrões de voz. Para tanto,

utiliza-se uma distribuição de probabilidade $b_j(o_t)$ de emissão de símbolos associada ao estado j do modelo e ao símbolo $o_t \in O$, onde O é a seqüência de observação, definida por:

$$O = \{o_1, o_2, ..., o_T\}$$

Desta forma, os elementos que definem um HMM são:

1- Número N de estados q_j no modelo, onde $1 \leq j \leq N$;

2- Alfabeto $X = \{x_1, ..., x_M\}$ dos possíveis símbolos observados, o qual é finito no caso de HMM's Discretos;

3- Matriz de probabilidades de transição entre estados $A = \{a_{ij}\}$, na qual $a_{ij} = P(q_{t+1} = j \mid q_t = i)$, $1 \leq i \leq N$ e $1 \leq j \leq N$.

4- Distribuição de probabilidade de emissão de símbolos $\{b_j(o_t)\}$, onde $1 \leq j \leq N$;

5- Distribuição de estado inicial $\Pi = \{p_i\}$, onde $p_i = P(q_1 = i)$.

Os HMM's podem ser classificados em Discretos, Semi-Contínuos ou Contínuos, dependendo do tipo de distribuição associada às probabilidades de emissão de símbolos, sendo mais utilizados os HMM's Discretos e Contínuos. No caso dos HMM's Discretos, assume-se que a variabilidade espectral do sinal de voz pode ser modelada através de uma distribuição de probabilidade do tipo discreta. Desta forma, a variável aleatória o_t é discreta, sendo dada por:

$$o_t = \{v_k\}_{k=1}^{L}$$

A distribuição de probabilidade de emissão fica, então, definida por:

$$b_j(k) = P(o_t = v_k \mid q_t = j)$$

O conjunto das distribuições $b_j(k)$ compõe, então, a Matriz de Probabilidade de Emissão B que, em conjunto com as matrizes A e Π, completa a definição de um HMM Discreto $l = (A, B, \Pi)$.

Em aplicações práticas, a discretização da variável o_t é realizada através de algoritmos de Quantização Vetorial, que consistem na construção de dicionários ("codebooks") compostos por um número finito de símbolos-código que são utilizados para representar quaisquer símbolos de entrada. De fato, seja um dicionário $C = \{v_i\}_{i=1}^{L}$, de comprimento *L*. A associação de um símbolo de entrada o_t a um símbolo código $v_j \in C$ é feita utilizando, por exemplo, uma regra de distância mínima, como mostrado na expressão abaixo:

$$o_t = v_k \Leftrightarrow v_k = \arg\min_{1<i<L} dist(o_t, v_i)$$

No caso dos HMM's Contínuos, assume-se que o_t é uma variável aleatória contínua, cuja densidade de probabilidade pode ser modelada a partir de uma mistura de *L* gaussianas, resultando:

$$b_j(o_t) = \sum_{k=1}^{L} c_{jk} \cdot N(o_t, m_{jk}, W_{jk}), \quad j = 1,...,N \tag{2.1}$$

onde *m* é o vetor de médias e *W* é a matriz de covariância inversa de dimensão *D* (normalmente assume-se que *W* é uma matriz diagonal).

Em geral, os HMM's Contínuos proporcionam melhores resultados, uma vez que modelam melhor as variabilidades espectrais que os HMM's Discretos. Outra vantagem dos HMM's Contínuos é que, neste caso, não é necessário realizar Quantização Vetorial, que introduz alguma distorção na representação das observações. Entretanto, os HMM's Discretos são mais simples de construir, uma vez que o processo de estimação de seus parâmetros (treinamento) constitui um algoritmo menos complexo que no caso dos HMM's Contínuos. Além disto, o processo de reconhecimento tende a ser mais rápido no caso dos HMM's

Discretos, uma vez que não é necessário realizar, durante o algoritmo de busca, o cálculo da densidade definida em (2.1).

Os HMM's podem também ser classificados segundo sua arquitetura, ou seja, pelo tipo de estrutura da matriz A. Um HMM é dito Ergódico quando a partir de um estado qualquer do modelo, é possível atingir todos os demais estados, resultando em uma matriz A completamente preenchida. Outro tipo de arquitetura, mais adequado ao modelamento da fala, foi proposto por Bakis [Bakis76], sendo a matriz A definida, neste caso, a partir da seguinte relação:

$$a_{ij} = 0, \quad j < i \tag{2.2}$$

Neste caso, o modelo é denominado "Left-Right", podendo ser visualizado na figura (2.1).

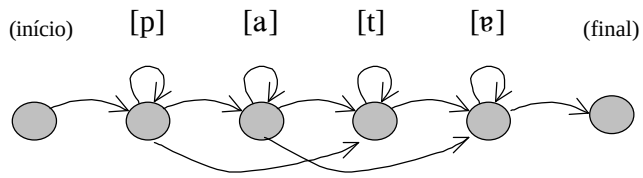

Figura (2.1) – Modelo "Left-Right" da palavra *pata*.

Uma vez que, nos modelos de Bakis, a seqüência de estados deve começar no estado 1 e terminar no estado *N*, torna-se válida a seguinte propriedade para as probabilidades dos estados iniciais, p_i:

$$p_i = \begin{cases} 0, & i \neq 1 \\ 1, & \mathrm{i} = 1 \end{cases}$$

Ao longo deste trabalho, será adotado o modelo "Left-Right".

2.2. Problemas Básicos Relacionados aos HMM's

Existem três problemas básicos relacionados aos HMM's que devem ser resolvidos a fim de permitir sua utilização em aplicações práticas [Rabiner93].

O primeiro problema está relacionado com o cálculo da probabilidade $P(O \mid l)$ de um modelo l gerar uma determinada seqüência de observação *O*. Este problema é resolvido através das recursões "Forward" ou "Backward", descritas abaixo:

- *Recursão "Forward"*

Seja a variável forward $a_t(i)$ definida como a probabilidade de uma observação parcial $o_1,o_2,...,o_t$ (t ≤ T) e de o estado no instante *t* ser o estado *i*. Tem-se, portanto:

$$a_t(i) = P(o_1, o_2, ..., o_t, q_t = i \mid l) \qquad (2.3)$$

a) *Inicialização*:

$$a_1(i) = p_i \cdot b_i(o_1) \qquad 1 \le i \le N$$

b) *Indução*:

$$a_{t+1}(j) = \left[\sum_{i=1}^{N} a_t(i) \cdot a_{ij} \right] \cdot b_j(o_{t+1}), \qquad 1 \le t \le T\text{-}1, \quad 1 \le j \le N$$

c) *Conclusão*:

$$P(O \mid l) = \sum_{i=1}^{N} a_T(i)$$

- *Recursão "Backward"*

Seja a variável backward $b_t(i)$ definida como a probabilidade de uma observação parcial $o_{t+1},o_{t+2},...,o_T$ (t ≤ T) e de o estado no instante *t* ser o estado *i*. Tem-se, portanto:

$$b_t(i) = P(o_{t+1}, o_{t+2},...,o_T \mid q_t = i, l) \tag{2.3}$$

a) Inicialização:

$$b_T(i) = 1, \qquad 1 \le i \le N$$

b) Indução:

$$b_t(i) = \sum_{j=1}^{N} a_{ij} \cdot b_j(o_{t+1}) \cdot b_{t+1}(j), \qquad t = T-1, T-2,...,1, \quad 1 \le i \le N$$

c) Conclusão:

$$P(O \mid l) = \sum_{i=1}^{N} b_1(i)$$

O segundo problema está relacionado com a busca da seqüência ótima de estados associada a uma dada seqüência de observação *O*. Este problema é resolvido através do algoritmo de Viterbi, no qual se define a seqüência ótima *q**, dentre todas as possíveis seqüências *q*, a partir do seguinte critério:

$$q^* = \arg\max_q P(q \mid O, l)$$

Note que maximizar $P(q \mid O, l)$ é equivalente a maximizar $P(q, O \mid l)$, pois:

$$P(q \mid O, l) = \frac{P(q, O \mid l)}{P(O \mid l)}$$

Desta forma, tem-se o seguinte algoritmo:

- *Algoritmo de Viterbi*

Define-se, inicialmente, a probabilidade $d_t(i)$ como:

$$d_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P(q_1, q_2, q_{t-1}, q_t = i, o_1, o_2, \ldots, o_t \mid l) \qquad (2.4)$$

Por indução, tem-se:

$$d_{t+1}(j) = \left[\max_i d_t(i) \cdot a_{ij} \right] \cdot b_j(o_{t+1}) \qquad (2.5)$$

Para recuperar o melhor caminho, é necessário armazenar os estados que maximizam a equação (2.5), para cada instante *t* e para cada estado *j*.

a) Inicialização:

$$d_1(i) = p_i \cdot b_i(o_1), \qquad 1 \le \mathrm{i} \le \mathrm{N}$$
$$y_1(\mathrm{i}) = 0.$$

b) Recursão:

$$d_t(j) = \max_{1 \le i \le N} \left[d_{t-1}(i) \cdot a_{ij} \right] \cdot b_j(o_t), \qquad 2 \le t \le T,\ 1 \le j \le N$$

$$y_t(j) = \arg\max_{1 \le i \le N} \left[d_{t-1}(i) \cdot a_{ij} \right], \qquad 2 \le t \le T,\ 1 \le j \le N$$

c) Conclusão:

$$P^* = \max_{1 \le i \le N} d_T(i)$$

$$q_T^* = \arg\max_{1 \le i \le N} d_T(i)$$

d) Rastreamento Reverso ("Backracking"):

$$q_t^* = y_{t+1}(q_{t+1}^*), \qquad t = T\text{-}1, T\text{-}2, \ldots, 1$$

Finalmente, o terceiro problema básico relacionado aos HMM's consiste no processo de estimação (treinamento) dos parâmetros que compõem os modelos. O treinamento dos HMM's é realizado através do algoritmo Baum-Welch, que se baseia no critério da Máxima Verossimilhança (vide Capítulo 4).

A fim de melhor compreender as equações de treinamento, é necessário definir algumas variáveis auxiliares. Inicialmente, a probabilidade de estar no estado *i*, no instante *t*, dada a observação *O* e o modelo l, é dada por:

$$g_t(i) = P(q_t = i \mid O, l) = \frac{a_t(i) b_t(i)}{\sum_{i=1}^{N} a_t(i) b_t(i)} \qquad (2.6)$$

Tomando-se a média ao longo do tempo da variável $g_t(i)$, obtém-se o número esperado de passagens pelo estado *i*, para a observação *O*, ou seja:

$\sum_{t=1}^{T-1} g_t(i) =$ número esperado de transições a partir do estado *i*, para a observação *O*

Outra variável importante é definida como a probabilidade de estar no estado *i*, no instante *t*, e no estado *j*, no instante *t+1*, dada a observação *O* e o modelo l:

$$x_t(i,j) = P(q_t = i, q_{t+1} = j, O \mid l) = \frac{a_t(i) \cdot a_{ij} \cdot b_j(o_{t+1}) \cdot b_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} a_t(i) \cdot a_{ij} \cdot b_j(o_{t+1}) \cdot b_{t+1}(j)} \qquad (2.7)$$

Novamente, tomando-se a média ao longo do tempo de $x_t(i,j)$, tem-se:

$\sum_{t=1}^{T-1} x_t(i,j) =$ número esperado de transições do estado *i* para o estado *j*, ao longo de *O*

Assim, é possível obter as fórmulas de reestimação dos parâmetros que compõem os HMM's. Uma vez que o Sistema de Reconhecimento de Fala empregado e simulado neste trabalho se baseia apenas em HMM Discreto, somente serão apresentadas as expressões para o treinamento deste tipo de HMM:

$\bar{p}_j =$ número esperado de passagens pelo estado *i* no instante $(t=1) = g_1(i)$

$$\bar{a}_{ij} = \frac{\text{número esperado de transições do estado } i \text{ para o estado } j}{\text{número esperado de passagens pelo estado } i} = \frac{\sum_{t=1}^{T-1} x_t(i,j)}{\sum_{t=1}^{T-1} g_t(i)}$$

$$b_j(\mathrm{k}) = \frac{\text{número esperado de passagens pelo estado } j, \text{ com emissão do símbolo } v_k}{\text{número esperado de passagens pelo estado } j} = \frac{\sum_{\substack{t=1, \\ o_t = v_k}}^{T-1} g_t(j)}{\sum_{t=1}^{T-1} g_t(j)}$$

Pode-se mostrar que este procedimento sempre atualiza os parâmetros de modo a maximizar $P(O|l)$, exceto no caso em que λ está associado a um ponto crítico da função de verossimilhança. Assim:

$$P(O \mid \bar{l}) > P(O \mid l)$$

onde $\bar{l}$ corresponde ao modelo l após uma iteração do algoritmo Baum-Welch.

As equações de reestimação podem ser formalmente deduzidas, partindo-se da definição da função de verossimilhança adotada no algoritmo EM (Expectation Maximization), na qual é incluído o conceito de espaço não-observável, a fim de permitir o correto modelamento do processo estocástico não-observável característico dos Modelos Ocultos de Markov. Adota-se, então, um procedimento de otimização a fim de encontrar as equações de treinamento que anulam o gradiente da função de verossimilhança, de modo a aproximar-se de um máximo local, a cada iteração. Vale salientar que, durante o processo de busca do ponto crítico da função de custo, as restrições sobre os parâmetros de Markov (vide Capítulo 4) devem ser respeitadas. Trata-se, portanto, de um algoritmo de Otimização Restrita, que é estabelecido através dos operadores Lagrangianos.

2.3. Algoritmo de Busca

Uma vez definido o modelo acústico, deve-se encontrar um algoritmo capaz de encontrar a seqüência ótima de palavras *W**, segundo o seguinte critério:

$$W^* = \arg\max_{\forall W} P(W \mid O) \qquad (2.8)$$

Uma primeira estratégia consistiria em realizar-se um procedimento de busca exaustiva, no qual seriam avaliadas, através do algoritmo de Viterbi, cada uma das verossimilhanças $P(W|O)$ obtidas a partir de todas as possíveis seqüências $W = \{w_i\}_{i=1}^{L}$, onde L é o número de palavras da seqüência. Seria então possível encontrar a seqüência ótima a partir da equação (2.8). Contudo, este procedimento torna-se inviável, devido ao grande número de possíveis seqüências de palavras, o que implica em um custo computacional extremamente elevado.

Foram, então, estabelecidos diversos algoritmos de busca bem mais eficientes, que podem ser classificados segundo duas estratégias principais: *depth-first* ou *breadth-first* [Pessoa99]. Na estratégia *depth-first*, as hipóteses mais promissoras são seguidas até o final da elocução ser atingido. Como exemplo, temos o algoritmo *Stack* e o algoritmo A* [Fagundes98]. Na estratégia *breadth-first*, as hipóteses são tratadas em paralelo. Algoritmos de decodificação usando *breadth-first* exploram o princípio de otimalidade de Bellman e são normalmente chamados de decodificadores de Viterbi.

Outro critério de classificação dos algoritmos de busca diz respeito à integração de fontes de conhecimento (tais como Modelos de Duração e Modelos da Língua), ao longo do processo de busca. Desta forma, a busca é dita *não-integrada* quando as fontes de conhecimento são introduzidas em uma etapa de pós-processamento, sendo necessário a obtenção de um número *N* de frases candidatas, além da frase vencedora (*N*, aqui, não é o número de estados de um HMM). Por outro lado, a busca é dita *integrada* quando as fontes de conhecimento atuam durante o algoritmo de busca, interferindo de forma mais significativa sobre o resultado final. O procedimento de busca integrada tende a proporcionar melhores resultados que a busca não-integrada, além de não necessitar das *N* frases mais prováveis. Vale salientar que as *N* frases mais prováveis são obtidas mais facilmente através de algoritmos do tipo *depth-first.*

Neste trabalho, será adotada a estratégia da busca integrada, realizada através de um algoritmo baseado no algoritmo *Level Building* [Rabiner93] que realiza a decodificação através de métodos de Programação Dinâmica, que se baseiam no princípio de otimalidade de Bellman: "um conjunto ótimo de soluções tem a propriedade de que qualquer que seja a primeira decisão, as decisões subseqüentes devem ser ótimas com relação ao resultado da primeira". A adoção deste tipo de algoritmo permitirá a introdução, durante o processo de busca, da informação de variação espectral obtida a partir de métodos de segmentação automática (Vide Capítulo 3).

O algoritmo Level-Building (LB) representa uma das mais simples estratégias possíveis para contornar o problema da busca exaustiva da seqüência ótima de palavras. Segundo esta estratégia, o procedimento de busca é dividido em níveis, nos quais realiza-se uma seleção parcial dos modelos mais prováveis, para cada instante $t \leq T$ (redução de nível).

O algoritmo se inicia no nível $l = 1$. Executa-se o algoritmo de Viterbi e determinam-se as verossimilhanças $P(o_1 o_2 ... o_t, q_1 q_2 ... q_t \mid w_1)$ e os caminhos ótimos que chegam ao último estado de cada palavra e para cada instante de tempo, como mostra a figura (2.2). Desta forma, tem-se, ao final do primeiro nível, as informações referentes às verossimilhanças acumuladas, para cada instante de tempo $t \leq T$ e para cada modelo w do vocabulário $\{w_i\}_{i=1}^{V}$.

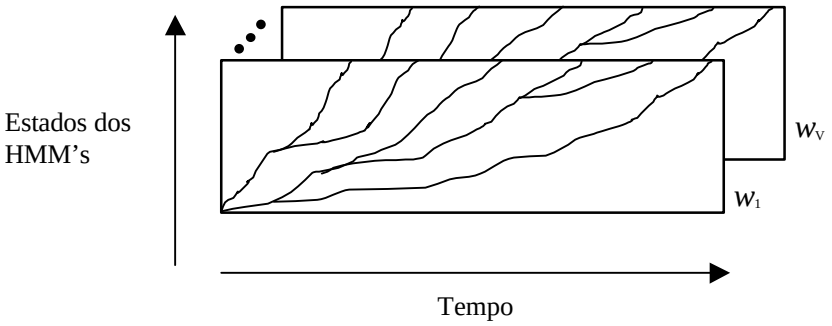

Figura (2.2) : Execução do LB para palavras na primeira posição da frase (primeiro nível)

A partir do segundo nível, inicia a busca pelas demais palavras da seqüência ótima W^*. Neste ponto realiza-se a "redução de nível" (figura (2.3)), que consiste em preservar apenas as informações relativas aos caminhos ótimos provenientes do nível anterior. De fato, não é necessário utilizar todas as informações obtidas no passo anterior, mas somente os maiores valores de $P(o_1 o_2 ... o_t, q_1 q_2 ... q_t \mid w_1)$, dentre todas as palavras testadas, para cada instante de tempo. Com isto, a partir do segundo nível serão encontrados também os caminhos ótimos e, para que o caminho seja ótimo de forma "global", é necessário partir dos "máximos" encontrados no nível anterior, já que estes também correspondem a caminhos ótimos.

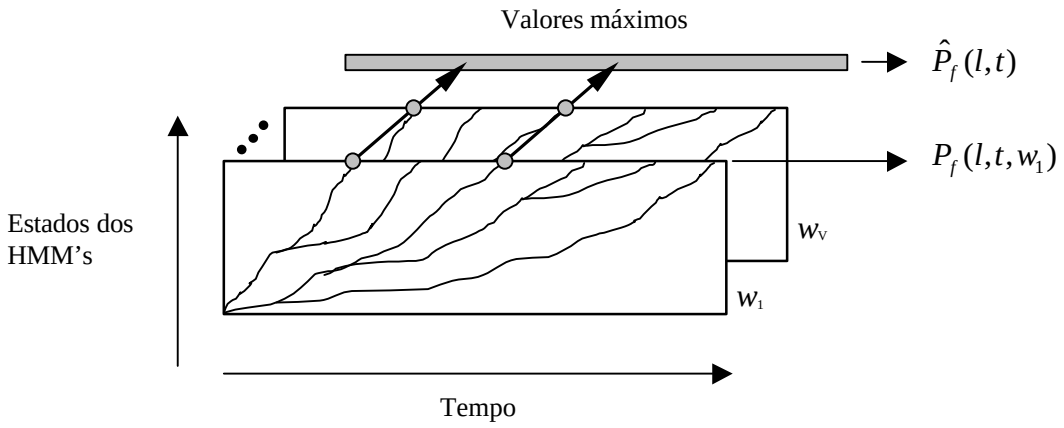

Figura (2.3): Processo de redução de nível no Level Building.

Tem-se, abaixo, uma síntese do algoritmo Level-Building [Rabiner93].

1- Nível $l = 1$

Executam-se os procedimentos a seguir para cada palavra w do vocabulário e para o primeiro nível.

1 .1– Inicialização

Inicializa-se a variável $d_t(j)$ de modo que todas as palavras possam iniciar a frase com a mesma probabilidade:

$$d_0(0) = \frac{1}{V}$$

onde V é o número de palavras do vocabulário.

O tempo $t = 0$ é usado como inicialização durante a busca. Os demais valores de $d_t(j)$ são definidos como:

$$\begin{aligned} d_t(j) &= -\infty, \quad j = 1,2,3,4...N_w \\ d_t(0) &= -\infty, \quad t = 1,2,3,...T \end{aligned} \tag{2-1}$$

onde N_w é o número de estados da palavra *w*.

Na prática, o símbolo ∞ indica que um valor *alto* deve ser definido.

1.2- Recursão (Viterbi)

Fazendo $1 \le t \le T$ e $1 \le j \le N^w$, obtém-se:

$$d_t(j) = \max_{0 \le i \le j} \left\{ d_{t-1}(i) + \bar{a}_{ij}^w \right\} + \bar{b}_j^w(o_t)$$

onde $\bar{a}_{ij}^w = \log a_{ij}^w$ e $\bar{b}_j^w(o_t) = \log b_j^w(o_t)$.

1.3- Finalização

Calcula-se a log-probabilidade final de cada palavra para $1 \le t \le T-1$:

$$\begin{aligned} P_f(l,t,w) &= d_t(N^w) + \bar{a}_{ij}^w \\ B(l,t,w) &= 0 \end{aligned}$$

A matriz $P_f(l,t,w)$ armazena a log-probabilidade do último estado relativa a cada palavra w, para cada nível l e tempo t. A matriz auxiliar $B(l,t,w)$ armazena os ponteiros de retorno que permitem encontrar a seqüência ótima de palavras.

1.4- Redução de Nível

No final do nível $l = 1$, mantém-se somente a palavra com maior probabilidade acumulada final, para cada instante *t*, descartando-se todas as demais palavras, bem como suas informações armazenadas.

Assim, para $1 \le t \le T$, faz-se:

$$\hat{P}_f(l,t) = \max_{1 \le w \le W} P_f(l,t,w)$$
$$\hat{B}(l,t) = 0$$
$$\hat{W}(l,t) = \arg\max_{1 \le w \le W} P_f(l,t,w)$$

2- Níveis $1 < l \le L_{\max}$

Deve-se executar os procedimentos seguintes para todas as palavras.

2.1- Inicialização

São realizados os seguintes procedimentos:

$$d_0(j) = -\infty, \qquad 0 \le j \le N^w$$
$$d_t(0) = P_f(l-1,t), \quad 1 \le t \le T$$

2.2- Recursão (Viterbi)

O procedimento de recursão anterior é repetido aqui:

$$d_t(j) = \max_{0 \le i \le j} \left\{ d_{t-1}(i) + \bar{a}_{ij}^{w} \right\} + \bar{b}_j^{w}(o_t)$$

Neste caso, porém, os ponteiros temporários $B(t,l,w)$ deverão ser atualizados com os instantes de início de cada caminho parcial encontrado durante o passo "Backtracking" do algoritmo de Viterbi.

2.3- Finalização

No final do nível, devem ser calculados novamente os valores das verossimilhanças, de modo análogo ao primeiro nível:

$$P_f(l,t,w) = d_t(N^w) + \overline{a}_{ij}^w, \quad 1 \le t \le T-1$$

2.4- Redução de Nível

O procedimento de redução de nível, neste caso, é definido como se segue:

$$\hat{P}_f(l,t) = \max_w P_f(l,t,w)$$
$$\hat{B}(l,t) = B\left(l,t,\arg\max_w P_f(l,t,w)\right)$$
$$\hat{W}(l,t) = \arg\max_w P_f(l,t,w)$$

Ao final da execução do passo referente ao último nível, deve-se obter o valor final *P** da verossimilhança para a seqüência ótima *W**, utilizando a seguinte equação:

$$P^* = \max_{1 \le l \le L_{\max}} P_f(l,T)$$

Finalmente, a frase reconhecida pode ser obtida através de um procedimento de "backtracking", utilizando-se a matriz $\hat{W}(l,t)$ e os ponteiros de retorno armazenados na matriz $\hat{B}(l,t)$.

Analisando o algoritmo e a figura (2.3), fica claro que os valores $P(o_1 o_2 ... o_t, q_1 q_2 ... q_t \mid w_i)$ de cada palavra são armazenados na matriz $P_f(l,t,w)$, onde l indica o nível, t indica o tempo e w_i, a i-ésima palavra do vocabulário. Os valores máximos obtidos na redução de nível são armazenados numa matriz do tipo $P_f(l,t)$, ficando a palavra vencedora, em cada tempo t, armazenada numa matriz $\hat{W}(l,t)$, de maneira a permitir a posterior recuperação de toda a seqüência de palavras reconhecidas.

3. Segmentação Automática e Modelos de Duração em HMM's

3.1. Introdução

Os Modelos Ocultos de Markov (HMM – Hidden Markov Models) têm sido amplamente utilizados em Reconhecimento Automático de Fala e representam, atualmente, um modelo acústico bastante consistente e confiável. Vários fatores tornam os HMM's especialmente indicados para aplicações de Reconhecimento de Fala, tais como:

- Existência de algoritmo eficiente e de complexidade computacional relativamente baixa para a estimação dos parâmetros do modelo (Baum-Welch);
- Existência de algoritmos eficientes baseados em Programação Dinâmica para realizar a decodificação acústica (Level-Building, One-Pass, Herman-Ney, A*, etc);
- Possibilidade de integração de diferentes fontes de informação lingüística (sintaxe, semântica, etc) e acústica (traços acústicos);

- Possibilidade de utilização de diferentes tipos de distribuição de probabilidade de emissão de símbolos (discreta, mistura de gaussianas, etc), permitindo grande flexibilidade no modelamento acústico;
- Pode-se utilizar diferentes parâmetros de entrada, combinados ou não, a fim de melhorar a precisão do modelamento acústico (Mel-Cepstrais, PLP, Energia Normalizada e derivadas destes);
- Possui grande versatilidade para o modelamento das unidades lingüísticas, através das várias arquiteturas que podem ser utilizadas.

Os Modelos Ocultos de Markov apresentam, entretanto, alguns problemas que limitam seu desempenho quando utilizados em Reconhecimento de Fala [Ostendorf97][Ostendorf89], tais como:

- *Modelo de duração deficiente:* adota-se um modelo de duração implícito de estados com distribuição exponencial, que quase nunca é compatível com as distribuições associadas às unidades lingüísticas (fones, sílabas, etc);
- *Independência entre observações:* adota-se a hipótese de independência entre observações, que não é verificada na prática, uma vez que os parâmetros estão associados aos quadros e existe forte correlação entre quadros de um mesmo segmento estacionário do sinal de voz. Além disto, existe forte dependência contextual entre quadros pertencentes a regiões de transição entre os segmentos acústicos de uma elocução;
- *Restrições sobre parâmetros:* adota-se como procedimento padrão a extração de parâmetros quadro a quadro, que impõe restrições sobre alguns tipos de parâmetros acústicos;
- *Limitações do conjunto de treinamento:* os algoritmos utilizados para a estimação dos parâmetros têm seus desempenhos bastante degradados quando a base de dados disponível é reduzida. Este problema é agravado com o aumento do número de parâmetros a serem estimados, representando um limitante para o emprego de modelos com arquiteturas mais complexas.

A primeira limitação pode ser tratada utilizando-se modelos de duração explícita de estados [Rabiner93][Russel85][Levinson86], nos quais, durante a etapa de treinamento, é estimada a distribuição de probabilidade que caracteriza a duração dos estados. O problema desta técnica, que

será descrita mais adiante, está no elevado custo computacional, podendo ser algumas ordens de grandeza superior ao do sistema sem duração explícita.

Uma segunda estratégia é descrita em [Rabiner89], onde o modelo de duração é empregado em uma etapa de pós-processamento, utilizando-se um histograma das durações dos estados para gerar um fator de ponderação para *N* frases candidatas.

Outra possível abordagem consiste em empregar-se modelos de duração de palavras [Morais97][Rabiner85], onde são estimados parâmetros relativos à duração das palavras do vocabulário. Em geral, assume-se que as durações podem ser modeladas a partir de uma distribuição gaussiana. Neste caso, deve-se obter, a partir do conjunto de treinamento, a variância e o desvio padrão das durações de cada palavra, a fim de determinar completamente cada um dos modelos.

Um mecanismo simples para suavizar a segunda limitação consiste na utilização das derivadas dos parâmetros acústicos de entrada, a fim de aumentar o espaço de observação. Pode-se ainda empregar variações do HMM, tais como "Segmental HMM's" [Russel93][Gales87] e HMM's condicionalmente gaussianos [Wellekens87][Kenny90].

A idéia de empregar parâmetros extraídos a partir de segmentos originou os trabalhos de Bush e Kopec [Bush87] e de Zue e colegas [Zue89]. Posteriormente, os Modelos Segmentais [Ostendorf97] surgiram como uma forte estratégia para contornar a limitação imposta pelo procedimento de extração de parâmetros quadro a quadro. Estes modelos correspondem a uma generalização dos HMM's e apresentam, em geral, desempenho superior, apesar da elevada complexidade computacional.

Finalmente, os problemas relacionados com a insuficiência de dados de treinamento, que normalmente ocorrem em casos práticos, podem também ser amenizados por meio de diferentes estratégias. O primeiro procedimento consiste em reavaliar os modelos definidos, a fim de tentar simplificar sua arquitetura, reduzindo o número de parâmetros. Em seguida, estratégias como Adaptação Bayesiana [Lee91] , "Deleted Interpolation" [Rabiner93] ou Treinamento Corretivo [Bahl88] podem ser empregadas para melhorar a estimação. Recentemente, uma nova estratégia baseada no critério MCE (Minimum Classification Error), denominada Treinamento Discriminativo [Chou92], vem sendo empregada com resultados bastante significativos, tornando os parâmetros mais robustos ao problema de insuficiência de dados. Esta técnica será discutida em detalhes no próximo capítulo.

3.2. Modelo de Duração Explícito

Sabe-se que os HMM's apresentam um modelo implícito de duração de estados do tipo exponencial, cuja densidade de probabilidade é dada por $p_i(d)$, que corresponde à probabilidade de d observações consecutivas no estado i:

$$p_i(d) = (a_{ii})^{d-1}(1 - a_{ii})$$

Este modelo é incompatível com a maioria dos sinais físicos (especialmente a fala) e pode ser substituído utilizando-se de forma explícita outra densidade de probabilidade, que será responsável por determinar os momentos apropriados para se realizar uma transição de estados. O modelo acústico passa, então, a ser denominado semi-Markov.

O objetivo principal desta abordagem é determinar com maior precisão o instante em que uma transição de estados deverá ocorrer. Para tanto utiliza-se uma densidade $p_q(d)$, que depende do estado atual e tem comportamento decrescente para valores elevados de d. Na prática esta densidade é truncada em uma duração máxima permitida D, ou seja, o número máximo de observações seguidas em um mesmo estado q é dado por D.

O mecanismo de transição de estados proposto implica em uma série de alterações nas fórmulas de reestimação de Baum-Welch [Rabiner93]. Inicialmente, assumindo-se que o primeiro estado começa em t = 1 e o último termina em t = T, a variável "forward" $a_t(i)$ pode ser redefinida como:

$$a_t(i) = P(o_1, o_2, ..., o_t, \textit{ permanência em i termina em } t \mid l) \qquad (3.1)$$

onde $\{o_i\}_{i=1}^{t}$ é a seqüência de observações de entrada até o instante t e l é o conjunto de parâmetros do HMM.

Assumindo-se que um total de r estados foram visitados durante as primeiras t observações e denotando-se os estados por $q_1, q_2, ..., q_r$ com durações $d_1, d_2, ..., d_r$, tem-se as seguintes restrições sobre a equação acima:

$$q_r = i$$

$$\sum_{s=1}^{r} d_s = t$$

A equação (3.1) pode ser escrita por indução como:

$$a_t(i) = \sum_{i=1}^{N} \sum_{d=1}^{D} a_{t-d}(i) a_{ij} p_j(d) \prod_{s=t-d+1}^{t} b_j(o_s)$$

onde *N* é o número de estados do modelo .

Note que a expressão acima é semelhante à expressão da variável forward sem modelo de duração, considerando-se a adição dos termos referentes a todas as possibilidades de permanência no estado *i* até o instante *t*, durante *D* observações no máximo.

A inicialização para o cálculo de $a_t(i)$ é composta pelo conjunto $\{a_t(i) : t \leq D\}$:

$$a_1(i) = p_i p_i(1) \cdot b_i(o_1)$$

$$a_2(i) = p_i p_i(2) \prod_{s=1}^{2} b_i(o_s) + \sum_{\substack{j=1 \\ j \neq i}}^{N} a_1(j) a_{ji} p_i(1) b_i(o_2)$$

$$a_3(i) = p_i p_i(3) \prod_{s=1}^{3} b_i(o_s) + \sum_{d=1}^{2} \sum_{\substack{j=1 \\ j \neq i}}^{N} a_{3-d}(j) a_{ji} p_i(d) \prod_{s=4-d}^{3} b_i(o_s)$$

$$\vdots$$

$$a_D(i) = p_i p_i(D) \prod_{s=1}^{D} b_i(o_s) + \sum_{d=1}^{D-1} \sum_{\substack{j=1 \\ j \neq i}}^{N} a_{D-d}(j) a_{ji} p_i(d) \prod_{s=D-d+1}^{D} b_i(o_s)$$

A probabilidade da observação O dado o modelo l pode ser calculada de forma análoga ao HMM sem modelo de duração explícita:

$$P(O|l) = \sum_{i=1}^{N} a_T(i)$$

Neste ponto, torna-se necessário definir outras variáveis forward-backward :

$$a_t^*(i) = P(o_1, o_2, ..., o_t, \textit{ permanência no estado i começa em } t+1 \mid l) \tag{3.2}$$

$$b_t(i) = P(o_{t+1}, ..., o_T \mid \textit{permanência no estado i termina em } t, l) \tag{3.3}$$

$$b_t^*(i) = P(o_{t+1}, ..., o_T \mid \textit{permanência no estado i começa em } t+1, l) \tag{3.4}$$

As relações entre $a_t(i)$, $a^*_t(i)$, $b_t(i)$ e $b^*_t(i)$ podem ser obtidas diretamente a partir das definições acima, sendo dadas por:

$$a^*_t(j) = \sum_{i=1}^{N} a_t(i) a_{ij} \tag{3.5}$$

$$a_t(i) = \sum_{d=1}^{D} a^*_{t-d}(i) p_i(d) \prod_{s=t-d+1}^{t} b_i(o_s) \tag{3.6}$$

$$b_t(i) = \sum_{\substack{j=1 \\ j \neq i}}^{N} a_{ij} b_t^*(j) \tag{3.7}$$

$$b_t^*(i) = \sum_{d=1}^{D} b_{t+d}(i) p_i(d) \prod_{s=t+1}^{t+d} b_i(o_s) \tag{3.8}$$

Com base nestas definições, as fórmulas de reestimação de Baum-Welch passam a ter a seguinte forma:

$$\bar{p}_i = P(q_1 = i \mid O, l) = \frac{P(O \mid q_1 = i, l) P(q_1 = i \mid l)}{P(O \mid l)} = \frac{b_0^*(i) p_i}{P(O \mid l)} \tag{3.9}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T} a_t(i) a_{ij} b_t^*(j)}{\sum_{j=1}^{N} \sum_{t=1}^{T} a_t(i) a_{ij} b_t^*(j)} \tag{3.10}$$

$$\bar{b}_i(k) = \frac{\sum_{\substack{t=1 \\ o_t=v_k}}^{T} \left[\sum_{t<t} a_t^*(i) \cdot b_t^*(i) - \sum_{t<t} a_t(i) b_t(i) \right]}{\sum_{k=1}^{M} \sum_{\substack{t=1 \\ o_t=v_k}}^{T} \left[\sum_{t<t} a_t^*(i) \cdot b_t^*(i) - \sum_{t<t} a_t(i) b_t(i) \right]} \tag{3.11}$$

$$\bar{p}_i(d) = \frac{\sum_{t=1}^{T} a_t^*(i) p_i(d) b_{t+d}(i) \prod_{s=t+1}^{t+d} b_i(o_s)}{\sum_{d=1}^{D} \sum_{t=1}^{T} a_t^*(i) p_i(d) b_{t+d}(i) \prod_{s=t+1}^{t+d} b_i(o_s)} \tag{3.12}$$

A expressão para $\bar{p}_i$ corresponde à probabilidade de o estado *i* ter sido o primeiro estado, dada a observação *O*. A expressão para $\bar{a}_{ij}$ é bastante semelhante à do HMM usual, pois combina as variáveis forward e backward para estimar o número médio de transições do estado *i* no instante *t* para o estado *j* no instante *t+1*. Vale salientar que, neste caso, $a_{ii} = 0$, de modo que a tendência de permanência no estado *i* é determinada apenas pela distribuição $p_i(d)$ (Modelo Semi-Markov). A expressão para os parâmetros $\bar{b}_i(k)$ corresponde ao número médio de ocorrências da observação v_k no estado *i*, sendo computada a partir da diferença $a_t^*(i) \cdot b_t^*(i) - a_t(i) b_t(i)$, que corresponde à probabilidade de um caminho passar pelo estado *i*, no instante *t*. Finalmente, a expressão (3.12) corresponde à razão entre número esperado de vezes no estado *i*, com duração *d*, e o número de ocorrências do estado *i* com duração qualquer (menor que *D*).

Apesar deste modelamento proporcionar, em algumas aplicações, uma melhoria significativa no desempenho, existem alguns problemas que dificultam a sua utilização em casos práticos. O primeiro problema é o elevado custo computacional, exigindo capacidade de armazenamento *D* vezes superior e $D^2/2$ vezes mais operações durante o processo de reestimação que o sistema usual [Rabiner93]. Para *D* igual a 25 (valor razoável para os problemas de processamento de fala), tem-se um aumento de cerca de 300 vezes no custo computacional. Há também um aumento no custo computacional na etapa de reconhecimento, uma vez que o mecanismo de transição de estados baseado em $p_i(d)$ deverá ser empregado. Um terceiro problema é o aumento significativo do número de parâmetros a serem estimados (*D* novos parâmetros por estado), que se torna grave principalmente nos casos de insuficiência de dados de treinamento.

O problema do aumento do número de parâmetros é atenuado com a utilização de uma densidade paramétrica, ao invés da não-paramétrica até então considerada. As densidades usualmente propostas são:

- *Famílias de Gaussianas*

$$p_i(d) = \mathrm{N}\left(d, m_i, s_i^2\right)$$

onde m_i é a duração média no estado *i* e s_i^2 é a variância associada à duração no estado *i*.

- *Família de Funções Gamma:*

$$p_i(d) = \frac{h_i^{v_i} d^{v_i - 1} e^{-h_i d}}{\Gamma(v_i)}$$

onde h_i e v_i são os parâmetros que caracterizam a distribuição Gamma.

Naturalmente, paga-se um preço por esta redução, tendo em vista a maior complexidade das fórmulas de reestimação para estes parâmetros.

Desta forma, podemos concluir que a estratégia de utilizar um modelo de duração explícita para melhorar o desempenho dos sistemas de reconhecimento apresenta sérias limitações práticas. De fato, utilizá-la poderia ajudar a atenuar os problemas causados pelo modelo de duração exponencial, mas certamente agravaria os problemas relacionados à insuficiência de dados de treinamento.

Um estratégia alternativa para auxiliar a combater os problemas do modelo de duração de estados do HMM será proposta em seguida, utilizando informações relativas às variações espectrais do sinal de fala. Estas informações podem ser obtidas através de técnicas de segmentação automática.

3.3. Segmentação Automática do Sinal de Fala

O problema da Segmentação Automática do sinal de fala tem sido objeto de estudos e pesquisas intensos, pois está intimamente relacionado com o problema de Reconhecimento de Fala, podendo ainda ser de grande utilidade em sistemas de Síntese e Codificação de Fala.

Uma das primeiras abordagens em Reconhecimento de Fala consistia em resolver o problema em dois passos [Vidal90]. Primeiramente, a elocução seria segmentada em "unidades lingüísticas", tais como fonemas, sílabas ou palavras. Em seguida, cada segmento seria reconhecido por meio de alinhamento e comparação com modelos das unidades lingüísticas, obtendo-se a frase ou palavra reconhecida através da concatenação das unidades identificadas. Entretanto, esta estratégia foi abandonada, pois estes procedimentos se mostraram excessivamente complexos e, durante algum tempo, as pesquisas na área se restringiram às palavras isoladas. A maior parte das pesquisas se concentrou no estudo de técnicas de alinhamento de padrões utilizando DTW (Dynamic Time Warping) e HMM, que representaram grande avanço na área.

Após esta evolução, voltou-se à proposta original de reconhecimento de fala contínua e o problema de Segmentação Automática foi retomado. Verificou-se então que o procedimento em dois passos antes idealizado não apresentava bom desempenho pois, no primeiro passo, referente à segmentação, não eram obtidas soluções suficientemente confiáveis.

Adotou-se, em seguida, a estratégia de integrar as etapas de segmentação e reconhecimento em um procedimento único e global. De fato, assumindo-se a existência de modelos apropriados, procedimentos adequados baseados em Programação Dinâmica podem ser empregados a fim de obter-se a seqüência ótima de unidades e respectivos segmentos, dada a seqüência de vetores acústicos de entrada. Naturalmente, ao se resolver conjuntamente os dois problemas, ocorre uma perda de desempenho na segmentação e decodificação acústica [Ostendorf89], em relação à abordagem em que eram considerados isoladamente. Contudo, o resultado global viabiliza as aplicações de reconhecimento de fala contínua. Vale salientar, no entanto, que esta degradação é bastante atenuada nos casos em que o treinamento dos modelos é mais robusto e preciso.

O desenvolvimento teórico do modelo de duração explícita de estados em HMM's, descrito na seção 3.2, deixa clara a necessidade de buscar técnicas que otimizem o mecanismo de transição de estados do HMM, tornando-o mais sensível às variações espectrais da elocução, de modo a atenuar os efeitos da distribuição exponencial. Adicionalmente, é importante procurar suavizar o problema relacionado à hipótese de independência entre quadros, assumida nos modelos de Markov usuais. Como foi dito anteriormente, uma estratégia simples a ser adotada é a utilização das derivadas dos parâmetros de entrada, a fim de aumentar o espaço de observação e, consequentemente, diminuir os efeitos, sobre o modelo, da existência de correlação entre quadros. Um problema desta técnica está relacionado com a insuficiência de dados de treinamento, uma vez que há um grande aumento do

número de parâmetros a serem estimados. Além disto, há um aumento do custo computacional no processo de reconhecimento.

A estratégia proposta neste trabalho para melhorar o desempenho de um sistema de reconhecimento de fala consiste em extrair do sinal medidas de variação espectral ao longo do tempo e incorporá-las diretamente ao algoritmo de decodificação acústica (neste caso, o Level-Building). Assim, proporciona-se meios para atenuar os problemas de modelamento temporal do HMM, com aumento da complexidade computacional relativamente pequeno e sem a necessidade de estimar uma grande quantidade de parâmetros.

A informação de variação temporal pode ser obtida através de técnicas de segmentação automática, que muitas vezes utilizam tal informação para determinar as fronteiras dos segmentos estacionários do sinal de fala. Assim, será realizada uma síntese das principais abordagens do problema de segmentação e, em seguida, uma descrição detalhada das técnicas selecionadas para este trabalho.

3.3.1. Formulação do Problema Geral

O problema de Segmentação Automática da Fala pode ser formulado no contexto de Reconhecimento de Padrões (RP) [Vidal90]. Na figura (3.1), coloca-se o problema como um mapeamento de um conjunto de entrada, que compreende a representação do sinal voz, para um conjunto de saída, composto pelas fronteiras de segmentação.

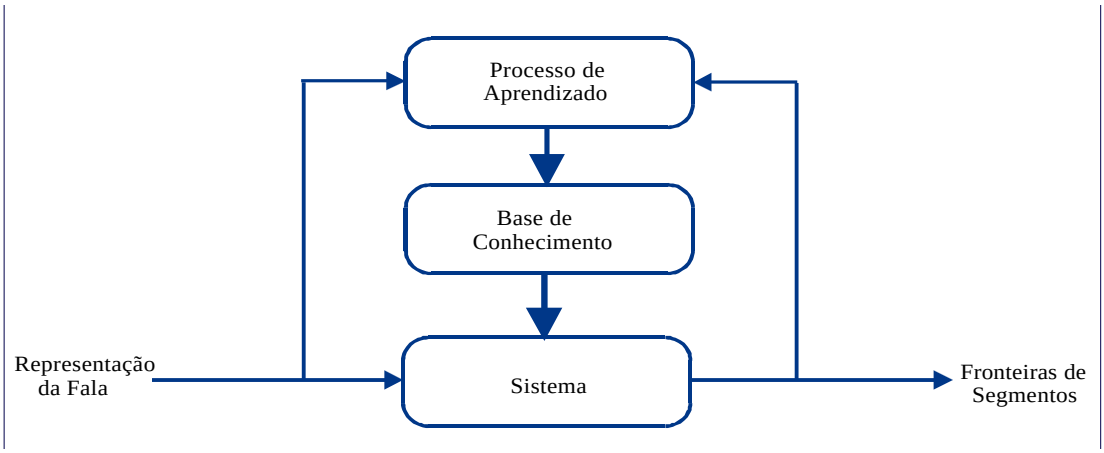

Figura 3.1- Diagrama representativo do problema de Segmentação Automática da Fala

Nesta abordagem, este mapeamento é realizado por um sistema que utiliza informações e/ou parâmetros gerados a partir de uma base de conhecimentos a priori. Esta base de conhecimento pode estar embutida nos procedimentos de interpretação do sistema ou pode ser adquirida automaticamente a partir de um conjunto de treinamento com pares entrada-saída previamente estabelecidos por um especialista. Segundo este ponto de vista, parte deste conhecimento pode se fornecido de forma dedutiva, através de resultados matemáticos, ou de forma indutiva, através de aprendizado.

Nesta generalização do problema de segmentação, as entradas podem ser compostas por pares de seqüências de entrada, sendo uma de Observações Acústicas e outra de Categorias Lingüísticas (em alguns caso, dispensável). As saídas, por sua vez, podem ser consideradas como seqüências simples de representações de Fronteiras de Segmentos. Esta definição inclui o caso em que existem várias seqüências de entrada a serem segmentadas, quando se assume a sua concatenação em uma única seqüência global.

Seja A uma seqüência de Observações Acústicas, L o conjunto de Categorias Lingüísticas e B o conjunto em que as Fronteiras de Segmentação são representadas. Então um método de segmentação é uma função $s: A^* \times L^* \rightarrow B^*$ onde, para qualquer conjunto X, X^* denota o conjunto das seqüências finitas dos elementos de X. Se $L = \varnothing$, então s é classificada como Lingüisticamente Irrestrita, e o mapeamento passa a ser $s: A^* \rightarrow B^*$. Para que a segmentação descrita por s seja válida, certas restrições devem ser respeitadas para todo $a \in A^*$, $b \in B^*$ e $l \in L^*$:

(i) $|a| > 0; |b| > 0$ (3.1)

(ii) se $b = s(a, l)$, então $|b| \leq |a|$ (3.2)

(iii) se $L \neq \varnothing$ e $b = s(a, l)$, então $|l| \leq |a|$ (3.3)

onde para toda seqüência X, $|X|$ denota o seu comprimento.

Usualmente, Fronteiras de Segmentos são representadas por marcas que indicam as posições dos segmentos nas seqüências acústicas de entrada. Neste caso, $B^* \equiv N$ e as seguintes restrições adicionais devem ser obedecidas:

(iv) $\forall a \in A^*, \forall l \in L^*, \forall b \in N,$ se $b = s(a,l),$ então $b_1 > 0; b_{j-1} \leq b_j, 1 \leq j \leq |b|; b_{|b|} = |a|$ (3.4)

onde para toda seqüência X, X_i denota o *i*-ésimo elemento de X.

Normalmente as Observações Acústicas, ou quadros, são vetores acústicos que correspondem a uma representação de curto termo do sinal de fala. Considerando-se que estes vetores pertencem a um espaço de dimensão *p*, tem-se que $A^* \equiv R^p$. Alternativamente, as observações podem corresponder aos índices da palavra-código de um "codebook", obtidos através de uma técnica de Quantização Vetorial.

3.3.2. Classificação das Metodologias

As diferentes abordagens sobre Segmentação Automática são analisadas em [Vidal90] e [Dosierre93], podendo ser classificadas de acordo com alguns critérios básicos, como mostrado na figura (3.2).

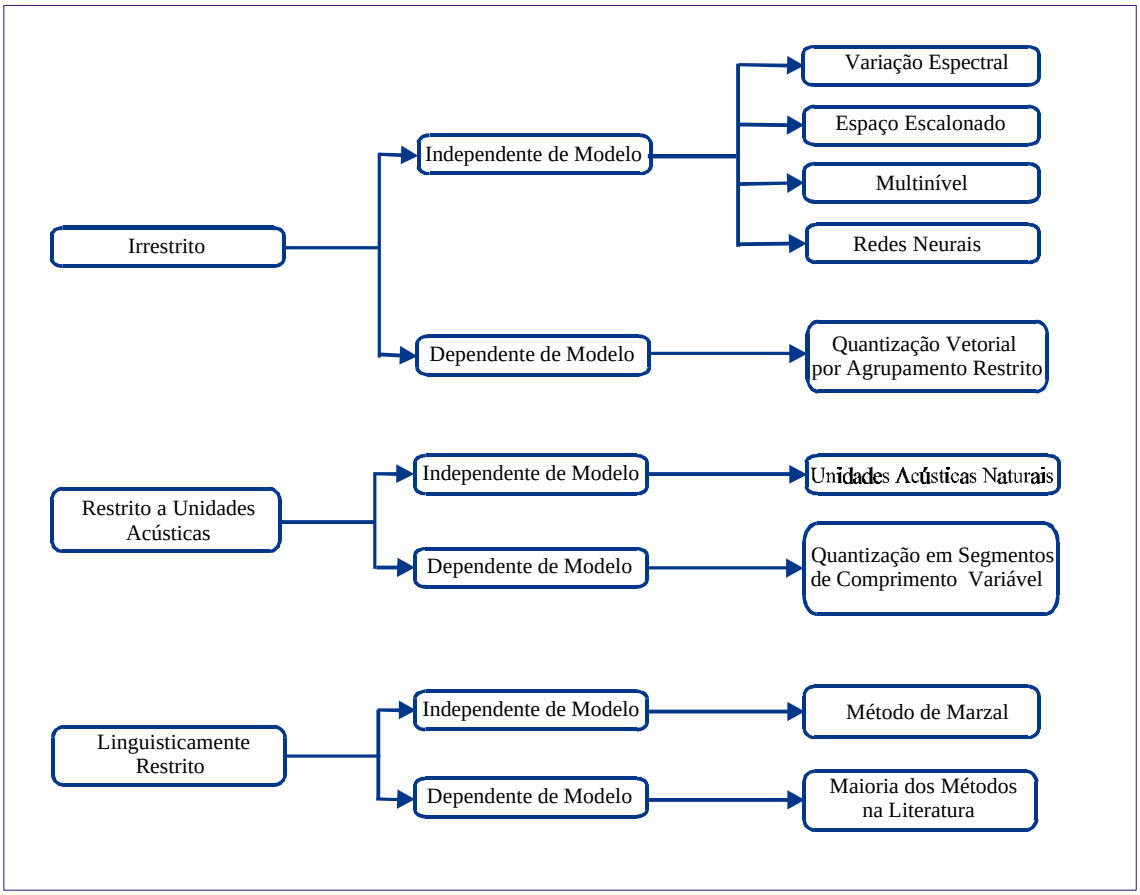

Figura 3.2- Diagrama das metodologias de Segmentação Automática

A metodologia de segmentação pode ser classificada como *Independente de Modelo*, quando não são necessários modelos para realizá-la, ou *Dependente de Modelo*, no caso em que a segmentação pode ser obtida assumindo-se hipóteses associadas a um determinado tipo de modelo para unidades acústicas ou lingüísticas.

Outro critério está relacionado com a presença do bloco de treinamento, mostrado na figura (3.1). Quando este bloco está presente, tem-se uma *Segmentação Supervisionada*, sendo necessário um conjunto de treinamento com marcas de segmentação manual para que o sistema possa ser treinado. No caso da ausência deste bloco, tem-se uma *Segmentação Não-Supervisionada*, quando admite-se que todo o conhecimento é fornecido por meio de métricas (medidas de distância espectral) ou modelos que não necessitam de treinamento.

Pode-se, ainda, classificar uma técnica de segmentação quanto à existência de restrições lingüísticas. No caso em que se admite uma seqüência de unidades lingüísticas como entrada para o segmentador, tem-se uma *Segmentação Lingüisticamente Restrita* ($L \neq \varnothing$). Caso contrário, tem-se uma *Segmentação Lingüisticamente Irrestrita* ($L = \varnothing$).

Alternativamente, Van Hermert [Hermert91] propôs a classificação das técnicas de segmentação de acordo com um critério um pouco mais geral. Neste caso, uma *Segmentação Implícita* se baseia apenas em informações de natureza acústica e uma *Segmentação Explícita* utiliza outros tipos de informação, tais como as restrições lingüísticas. Ainda neste trabalho, Van Hermert relata uma característica dos métodos de segmentação quando analisados sob este aspecto. Verifica-se na prática que os métodos de Segmentação Implícita geram fronteiras mais precisas mas não garantem o número correto de segmentos, enquanto que métodos de Segmentação Explícita fornecem o número correto de segmentos e suas classificações em unidades lingüísticas, perdendo, porém, precisão na determinação das fronteiras.

3.3.2.1. Segmentação Acústica Irrestrita

Inicialmente serão consideradas as técnicas de segmentação que empregam apenas informação acústica, não sendo restritas por informações lingüísticas. Para este tipo de segmentação, são utilizados mecanismos baseados em Medidas de Distorção (métrica) entre observações acústicas. Serão analisadas quatro classes de *Segmentação Acústica Irrestrita*.

A- Variação Espectral

Seja $\mathsf{A} \equiv \mathrm{R}^p$ o conjunto de Observações Acústicas e seja $\vec{a}(t) \in \mathsf{A}$ um vetor de parâmetros correspondentes a determinadas características acústicas do sinal, no instante *t*. A Variação Espectral de $\vec{a}$ no tempo é dada por:

$$\vec{a}'(t) = \frac{\partial \vec{a}(t)}{\partial t}$$

A magnitude da derivada $\vec{a}'(t)$ representa a taxa de variação espectral do sinal, podendo ser utilizada para identificar as fronteiras dos segmentos, onde normalmente ocorrem picos de variação espectral.

Em aplicações práticas as observações são discretas no tempo e a derivada passa a ser aproximada por uma equação de diferenças. Como as diferenças de primeira e segunda ordem são aproximações excessivamente ruidosas da derivada, utiliza-se uma aproximação polinomial obtida por uma estimativa de mínimos quadrados [Rabiner93]. Estas equações levam a definições de Medidas de Distorção Espectral, normalmente envolvendo várias observações adjacentes ao vetor no instante *t*. Na abordagem convencional, utiliza-se uma janela de ponderação h_k, $-K \pounds k \pounds K$, que compõe a definição de Magnitude de Variação Espectral, dada por:

$$\left\|\vec{a}'(t)\right\| = \frac{\left\| \sum_{-K \le k \le K} k \cdot h_k \cdot \vec{a}(t+k) \right\|}{\sum_{-K \le k \le K} h_k}, \qquad t = 1+K, \ldots, |a|-K \qquad (3.4)$$

Uma opção normalmente adotada para h_k é a janela retangular, definida por:

$$h_k = \begin{cases} 1, & \text{se } -K \le k \le K \\ 0, & \text{caso contrário} \end{cases}$$

Neste caso:

$$\left\|\vec{a}'(t)\right\| = \frac{\left\| \sum_{-K \le k \le K} k \cdot \vec{a}(t+k) \right\|}{(2 \cdot K + 1)}, \qquad t = 1+K, ..., |a|-K \qquad (3.5)$$

O parâmetro *K* deve ser escolhido adequadamente, uma vez que valores pequenos podem levar ao problema de sobre-segmentação (excesso de fronteiras de segmentos) e valores elevados podem levar a dificuldades na detecção de segmentos curtos e, normalmente, importantes (consoantes, por exemplo).

Avaliando-se o perfil desta medida ao longo do tempo e capturando os instantes de ocorrência de máximos locais (picos), é possível determinar a segmentação desejada. Uma estratégia possível consiste em determinar os instantes em que os picos de $\left\|\vec{a}'(t)\right\|$ superam um determinado limiar. O problema desta técnica está na comum ocorrência de sobre-segmentação, decorrente da dificuldade (às vezes, impossibilidade) de encontrar-se o limiar adequado e da característica ruidosa da aproximação (3.4). Uma forma de atenuar este problema consiste em impor restrições sobre os intervalos de tempo onde um pico pode ser detectado, impedindo-se que dois máximos locais muito próximos sejam associados a fronteiras de segmentos. Entretanto, devido ao caráter estritamente empírico destas abordagens, observa-se um sério comprometimento da robustez dos resultados.

Uma técnica alternativa e mais robusta, empregada para segmentar bases de dados de elocuções, consiste em realizar uma comparação entre as segmentações de várias elocuções da mesma sentença, a fim de reduzir os eventuais efeitos de variações inconsistentes na medida $\left\|\vec{a}'(t)\right\|$. Neste caso, realiza-se uma alinhamento prévio das elocuções através do algoritmo DTW (Dynamic Time Warping) e computa-se a Variação Espectral Mútua, que é obtida a partir da variação espectral no tempo de uma elocução em relação a outras elocuções.

Em situações práticas verifica-se que a Segmentação por Variação Espectral não apresenta bom desempenho na determinação das fronteiras de todas as unidades lingüísticas de uma elocução. Entretanto, pode ser bastante útil na determinação de segmentações mais precisas, porém parciais (constituídas por um número de marcas inferior ao correto). Adicionalmente, as segmentações obtidas por este método podem ser utilizadas para inicializar procedimentos de segmentação manual,

normalmente bastante lentos e tediosos. Alguns exemplos de métodos de Segmentação Automática baseados em variação espectral são encontrados em [Wilpon87], [Algazi88], [Gibson96] e [Hermert91], destacando-se o método da Filtragem Paramétrica, proposto por Gibson, o qual será descrito em detalhes mais adiante.

B- Espaço Escalonado e Segmentação Multinível

Uma forma de atenuar os problemas encontrados nas técnicas baseadas em Variação Espectral consiste em avaliar-se, de forma sistemática, várias opções de segmentação relacionadas com a medida de distorção espectral adotada.

A primeira abordagem com resultados significativos neste sentido foi denominada Espaço Escalonado e foi proposta em [Witkin84] , correspondendo a uma tentativa de se obter uma descrição estruturada do sinal de fala, em diferentes escalas de resolução.

No contexto de Segmentação Automática, esta propriedade de multi-resolução é implementada variando-se o parâmetro *K*, de modo a se obter diferentes graus de suavização da informação de variação espectral. Pode-se mostrar [Altosaar88] que ao aumentar-se o valor de *K* não ocorrerão novos picos no contorno da função de variação espectral, desde que o processo de suavização seja adequado. Esta propriedade resulta em um conjunto de segmentações que serão organizadas em uma árvore, com o grau de suavização crescendo a cada nível. Em seguida, um processo de busca deve ser realizado para determinar a melhor segmentação.

Outra abordagem é denominada Segmentação Multinível e foi proposta por Glass e Zue [Glass88]. Esta técnica consiste em um procedimento de agrupamento hierárquico das observações acústicas em segmentos, de acordo com uma estruturação em níveis apropriada. Inicialmente, considera-se cada observação acústica como um segmento. No primeiro nível, associa-se cada vetor acústico ao seu antecessor ou sucessor (de acordo com um critério de distância mínima), determinando um conjunto inicial de segmentos que serão representados pelos seus centróides. No segundo nível, repete-se este procedimento utilizando-se, entretanto, os centróides dos segmentos do primeiro nível como observações acústicas, de modo a obter-se uma segmentação a partir do agrupamento dos segmentos do nível anterior. Repete-se então, para os demais níveis, o procedimento correspondente ao segundo nível até que se obtenha, no último nível, um único segmento e um único

centróide. Por fim, através da análise da distorção de cada segmentação gerada, seleciona-se a que apresenta distorção mínima.

As técnicas de Espaço Escalonado e Segmentação Multinível se baseiam em princípios diferentes mas apresentam a característica comum de procurar selecionar, dentre várias opções possíveis, a segmentação final. Apesar das diferenças, verifica-se que os desempenhos de ambas as técnicas são semelhantes e suficientemente apurados para permitir sua utilização como pré-processamento para procedimentos de segmentação manual.

C- Quantização Vetorial por Agrupamento Restrito

Uma estratégia alternativa para o problema de segmentação foi proposta por Svendsen [Svendsen87] e consiste em agrupar os vetores de modo a maximizar o grau de homogeneidade de cada segmento. Para tanto, propõe-se uma medida de distorção que é utilizada para encontrar a segmentação ótima, através de um algoritmo de programação dinâmica.

Seja a seqüência de entrada $\{a_i\}_{i=1}^{m}$, agrupada em uma seqüência de segmentos delimitados pelas fronteiras $\{b_i\}_{i=1}^{l}$, onde $b_l=m$ e $l<m$. Considerando-se a seqüência dos centróides dos segmentos, $\{c_i\}_{i=1}^{l}$, pode-se avaliar a distorção global de uma segmentação somando-se as distorções intra-segmentos, resultando na definição abaixo:

$$D_G = \sum_{i=1}^{l} \sum_{j=b_{i-1}+1}^{b_i} \left\| a_j - c_i \right\|$$

onde a distorção intra-segmento para o segmento *i* é dada por:

$$D_{IS} = \sum_{j=b_{i-1}+1}^{b_i} \left\| a_j - c_i \right\|$$

Para resolver o problema de busca da segmentação ótima é necessário definir a distância $d(i,j)$ como sendo a medida de distorção de um possível segmento composto pelos vetores acústicos

$\{a_k\}_{k=i}^{j}$. Desta forma, o problema de Segmentação por Agrupamento Restrito pode ser formulado como o problema de obter-se a distorção *D*(*m*,*n*), onde, para 1£*j*£*m*, 1£*k*£*n*, *D*(*j*,*k*) é definido como a distorção mínima de uma segmentação dos vetores $\{a_i\}_{i=1}^{j}$ em *k* segmentos. Esta minimização pode ser realizada através de um algoritmo de Programação Dinâmica Multi-Estágios, onde os estágios (ou níveis) correspondem ao número de segmentos. Resulta, então, a seguinte formulação recursiva:

$$D(j,k) = \min_{j-t_2 \le i \le j-t_1} \left[D(i,k-1) + d(i+1,j) \right] \qquad (3.6)$$

onde t_1 e t_2 são o menor e o maior número de vetores permitidos em um segmento.

A segmentação ótima é encontrada realizando o procedimento "backtracking", sendo possível obter todas as possíveis segmentações de $\{a_i\}_{i=1}^{m}$ em k segmentos, com *k*£*n*. Vale ainda salientar que é possível especificar o número de segmentos desejado, pois sempre é possível encontrar a distorção *D(m,m)*, que implica na obtenção de todas as possíveis segmentações de $\{a_i\}_{i=1}^{m}$.

Este método também pode ser aplicado para obter-se uma segmentação Dependente de Modelo, originando a Segmentação de Máxima Verossimilhança. Neste caso, normalmente são utilizados modelos auto-regressivos para os segmentos e avalia-se sua verossimilhança ao invés da medida de distorção intra-segmento. O procedimento de busca, no entanto, é idêntico ao descrito pela equação (3.6).

D- Redes Neurais

Recentemente, novos métodos de segmentação automática irrestrita, que utilizam diretamente técnicas de Reconhecimento de Padrões baseadas em Redes Neurais, têm sido propostos na literatura. Em [Suh96], utiliza-se um método supervisionado baseado em uma rede do tipo MLP, treinada com um algoritmo Back-Propagation modificado, a fim de obter uma estimativa das fronteiras de segmentos acústicos, em fala contínua. Em [Rubio95] e [Fukada97], são propostos métodos que utilizam Redes Recorrentes para determinar as fronteiras dos segmentos em aplicações envolvendo fala contínua, uma vez que tais estruturas são mais apropriadas para modelar a variabilidade temporal dos padrões da voz humana.

Apesar de não ser a abordagem predominante, algumas técnicas baseadas em treinamento não supervisionado têm sido propostas. No Apêndice A desta tese, tem-se um artigo que descreve um algoritmo de segmentação automática de palavras isoladas utilizando Redes de Kohonen para obter os segmentos. Adicionalmente, são geradas entradas de comprimento fixo para um classificador de padrões (MLP, por exemplo), independentemente da duração da elocução. Este conjunto de entrada é formado a partir dos centróides dos segmentos, que são aproximados pelos pesos dos neurônios da Rede de Kohonen.

3.3.2.2. Segmentação em Unidades Acústicas

As Segmentações Acústicas Irrestritas apresentam a característica comum de considerar apenas propriedades acústicas locais, ou seja, não consideram as relações acústicas ou lingüísticas entre segmentos. As técnicas de Segmentação em Unidades Acústicas, por sua vez, consideram aspectos mais globais, permitindo o agrupamento de segmentos em Unidades Acústicas.

Para esta classe de algoritmos, praticamente não são adotados procedimentos Independentes de Modelo, que consistem na utilização de unidades acústicas naturais, obtidas por métodos de reconhecimento de padrões a partir de um conjunto de dados de treinamento. O motivo deste fato está na grande dificuldade de obtenção de unidades consistentes a partir de tais procedimentos. Desta forma, destacam-se as técnicas Dependentes de Modelo, tal como a descrita a seguir.

A- Segmentação Restrita a Unidades Acústicas ou Quantização em Segmentos de Comprimento Variável

Esta abordagem foi proposta por Shiraki e Honda [Shiraki88] e pode ser classificada como Dependente de Modelo. Representa uma extensão da técnica de Quantização Vetorial por Agrupamento Restrito, adicionando-se a restrição de correspondência entre cada segmento e uma determinada Unidade Acústica.

Seja a seqüência acústica $\{a_i\}_{i=1}^{m}$ a ser segmentada em *n* segmentos. Seja *N* o número de Unidades Acústicas desejadas e *Q* um conjunto de modelos destas unidades. Seja $d_Q(i,j)$ a distorção do segmento $\{a_k\}_{k=i}^{j}$ com respeito a *Q*, definida por:

$$d_Q(i,j) = \min_{q \in Q} d(a_i ... a_j, q)$$

onde $d(x,z)$ é distorção ou dissimilaridade entre o segmento *x* e o Modelo de Unidade *z*.

O problema de segmentação pode então ser formulado de forma mais consistente como sendo a tarefa de encontrar conjuntamente a segmentação $\{b_i\}_{i=1}^{l}$ e o conjunto $Q = (q_i)_{i=1}^{N}$ que minimiza a distorção global, ou seja:

$$D_Q(m,n) = \min_{Q,b} \sum_{i=1}^{n} d_Q(b_{i-1}+1, b_i)$$

Uma solução para encontrar um mínimo local é obtida a partir da repetição de dois passos básicos. O algoritmo é iniciado com a obtenção de uma segmentação arbitrária da seqüência $\{a_i\}_{i=1}^{m}$ em *n* segmentos e, em seguida, os dois passos básicos são executados repetitivamente. No primeiro passo são estimados os modelos do conjunto Q através de um procedimento que minimiza a distorção em relação aos segmentos atuais por meio de um algoritmo de agrupamento do tipo "K-means". No segundo passo, obtém-se uma segmentação de distorção mínima utilizando-se uma algoritmo de Programação Dinâmica Multi-Estágios, definido pela expressão:

$$D_Q(j,k) = \min_{j-t_2 \le i \le j-t_1} \{D_Q(i,k-1) + d_Q(i+1,j)\}$$

onde t_1 e t_2 são o menor e o maior número de vetores permitidos em um segmento e $D_Q(j,k)$ é a distorção mínima com respeito a *Q* de uma segmentação de $\{a_i\}_{i=1}^{j}$ em *k* segmentos. Assume-se, ainda, que a segmentação $\{b_i\}_{i=1}^{l}$ é obtida através do procedimento "backtracking".

Existe uma grande variedade de medidas de distorção e modelos que podem ser empregados no algoritmo acima. Shiraki e Honda propuseram a utilização de padrões de comprimento fixo como modelo, bem como alinhamentos lineares como meio para calcular a distorção. Um método mais genérico consiste em empregar-se padrões de comprimento variável como modelos e o algoritmo DTW para obter-se a medida de distorção. Neste caso, o passo referente à Programação Dinâmica Multi-Estágios se reduz ao algortimo "Level-Building". Finalmente, pode-se utilizar, por meio de procedimentos de agrupamento adequados, Modelos Ocultos de Markov e os Modelos Segmentais Estocásticos como alternativas para o modelamento das Unidades Acústicas.

3.3.2.3. Segmentação com Restrições Lingüísticas

Embora os algoritmos de segmentação automática baseados apenas em informações acústicas possam apresentar bom desempenho, a introdução de restrições lingüísticas pode ser adotada como uma abordagem mais ampla para este problema, reduzindo a ocorrência de sobre-segmentação.

As restrições de natureza lingüística são obtidas a partir de uma seqüência l de elementos do conjunto de categorias lingüísticas L, que podem ser fonemas, fonemas dependentes de contexto, ditongos, palavras, etc. Desta forma, a seqüência l é um elemento do conjunto L^*, formado por todas as seqüências finitas de unidades lingüísticas definidas em L . Deve-se, então, impor que os segmentos obtidos, assim como a seqüência acústica $a \in \mathsf{A}^*$, sejam consistentes com a seqüência l. De fato, uma segmentação pode ser definida de forma genérica como uma função $s : \mathsf{A}^* \times \mathsf{L}^* \rightarrow \mathsf{B}^*$, como foi descrito na seção 3.3.1. Vale salientar que, neste caso, as fronteiras de segmentação que compõem $b \in \mathsf{B}^*$ serão sempre números naturais, ou seja, $\mathsf{B}^* \equiv \mathrm{N}$.

Formalmente, serão consideradas segmentações do tipo $s : \mathsf{A}^* \times \mathsf{L}^* \rightarrow \mathrm{N}$, onde duas restrições adicionais deverão ser introduzidas.

$\forall\, a \in \mathsf{A}^*,\ l \in \mathsf{L}^*$ e $b = s(a,l) \in \mathrm{N}$, tem-se:

(1) $|b| = |l|$

(2) Uma partição Π_l em $|\mathsf{L}|$ classes equivalentes é gerada a partir de l sobre os segmentos especificados por b, de modo que:

$$a_{b_{i-1}+1} \ldots a_{b_i} \cong a_{b_{j-1}+1} \ldots a_{b_j} \Leftrightarrow l_i = l_j$$

Pode-se agrupar as técnicas envolvendo restrições lingüísticas em duas abordagens principais:

(1) *Independente de Modelo*: Este tipo de abordagem é ainda pouco explorado, tendo sido proposto em [Marzal90]. Em geral, consiste em mapear o problema de segmentação automática em um problema de otimização. Define-se uma medida de distorção D(*a*,*l*,*b*) entre segmentos acústicos, que depende das seqüências *a*, *l* e *b*. Esta medida normalmente segue o mesmo padrão das distorções empregadas em algoritmos DTW. Em seguida, realiza-se um processo de otimização especial, denominado "Greedy Hill Climbing Algorithm" [Vidal90], que verifica, a cada iteração, todos os possíveis movimentos (unitários) das fronteiras definidas em *b*, de modo a reduzir a medida de distorção D(*a*,*l*,*b*). Esta técnica tem proporcionado bons resultados [Vidal90], porém o custo computacional ainda é relativamente elevado.

(2) *Dependente de Modelo*: Esta abordagem tem sido a mais utilizada em aplicações recentes de segmentação automática e foi proposta pela primeira vez em [Bourland85] e [Schwartz85]. São definidos modelos para as unidades lingüísticas a fim de construir uma medida de distorção que depende da restrição *l* considerada. Empregando algoritmos de programação dinâmica, realiza-se um processo de busca da segmentação que minimiza esta distorção. Este procedimento é bastante dependente do tipo de modelo adotado. No caso de modelos do tipo "template", este procedimento se reduz ao algoritmo DTW aplicado à seqüência acústica *a* e às restrições *l*. No caso dos HMM's, este procedimento é realizado por meio do algoritmo de Viterbi. Apesar de proporcionar bons resultados e consistência com as informações lingüísticas, esta abordagem é dificultada pela necessidade de definir e treinar os modelos das unidades lingüísticas.

3.3.3. Técnicas Implementadas

Nesta seção iremos descrever as técnicas de Segmentação Automática implementadas neste trabalho. Vale salientar que selecionou-se apenas técnicas de Segmentação Implícita e Independentes de Modelo, tendo em vista que a informação de segmentação gerada será utilizada em um sistema de reconhecimento de fala baseado em Programação Dinâmica, que já realiza uma Segmentação Explícita e Dependente de Modelo.

O objetivo de empregar um método de segmentação automática está relacionado apenas com a obtenção de uma medida de distorção espectral, inserida no sistema de reconhecimento como uma fonte de informação sobre as variações espectrais do sinal de fala no tempo. Obter as marcas de segmentação a partir desta medida não é o objetivo principal, e sim melhorar o desempenho da segmentação explícita gerada pelo algoritmo Level-Building, bem como aumentar a taxa de reconhecimento. Sendo assim, a discussão sobre algoritmos de determinação das marcas a partir da medida de distorção está fora do escopo deste trabalho. Entretanto, várias técnicas de Segmentação Implícita são propostas na literatura, como, por exemplo, em [Zelinski83], [Luna90], [Gibson96] e [Hermert91], onde se utilizam diferentes algoritmos para extrair as seqüências de fronteiras de segmentos.

3.3.3.1. Filtragem Paramétrica

O método de Filtragem Paramétrica, proposto em [Gibson96], pode ser classificado como um método de Segmentação Implícita, Independente de Modelo e Não-Supervisionado. Baseia-se na caracterização da estrutura de correlação de um sinal estacionário através de determinados parâmetros estatísticos obtidos a partir das saídas de um banco de filtros criteriosamente projetado.

Inicialmente, serão estabelecidas as definições e propriedades relativas ao método, bem como descrição sobre as medidas de distorção espectral utilizadas e, por fim, será descrito o sistema de banco de filtros implementado.

A- Conceitos e Propriedades

Seja X_t um sinal real, caracterizado por um processo estacionário de média nula. Seja a função de autocorrelação (normalizada) de X_t:

$$r_k = \frac{E\{X_{t+k} X_t\}}{E\{X_t^2\}} \tag{3.7}$$

Seja um filtro IIR só-pólos $H(z^{-1}; a)$ definido por:

$$Y_t(a) = \sum_{l=0}^{\infty} \overline{a}^{l} X_{t-l} = \overline{a} Y_{t-1}(a) + X_t \tag{3.8}$$

onde $a = h \cdot e^{-jq}$ e $0 < h < 1$.

Este filtro é do tipo passa-faixa, centrado em θ e com a seguinte função de transferência :

$$H(z) = \frac{1}{1 - \overline{a} \cdot z^{-1}} \Rightarrow H\left(e^{jw}\right) = \frac{1 - h\cos(w-q) - jh\,\mathrm{sen}(w-q)}{1 + h^2 - 2h\cos(w-q)}$$

O módulo de $H(e^{jw})$ é dado por:

$$\left|H\left(e^{jw}\right)\right| = \frac{1}{\sqrt{1 + h^2 - 2h\cos(w-q)}}$$

Fica claro que o módulo da resposta em freqüência apresenta um pico em $w = q$. Além disto, a largura de banda de –3 dB deste filtro é dada por:

$$Bwidth = \frac{1-h}{\sqrt{h}}$$

Na figura abaixo é mostrado o comportamento, para diferentes valores de a, da função $|H(e^{jw})|$ associada ao filtro $H(z^{-1};a)$.

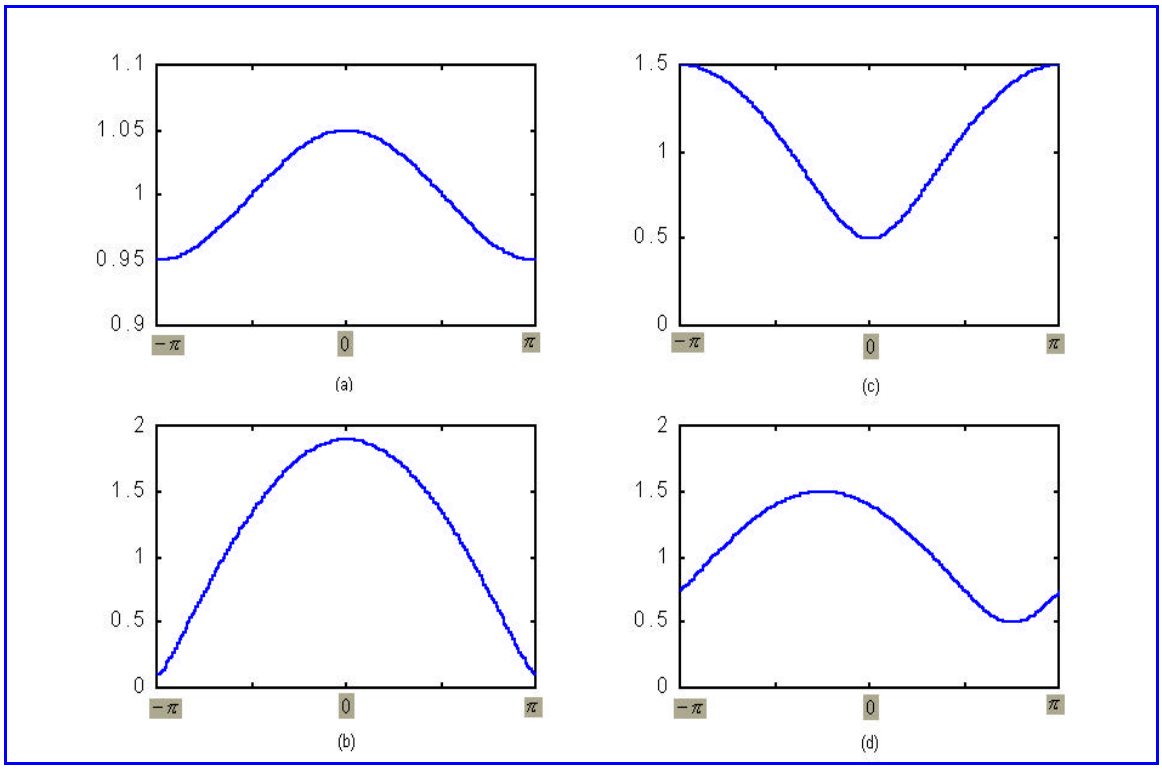

Figura 3.4- Módulo da Função de Transferência correspondente ao filtro $H(z^{-1};a)$ para os seguintes casos: (a) a=0.01 (b) a = 0.9 (c) a = 0.5.e$^{j\pi}$ (d) a = 0.5.e$^{j\pi/4}$

A autocorrelação do sinal de saída $Y_t(a)$, com distância de correlação (lag) unitária é dada por:

$$r(a) = \frac{E\left\{Y_{t+1}(a)\overline{Y}_t(a)\right\}}{E\left\{|Y_t(a)|^2\right\}} \qquad (3.9)$$

O parâmetro de interesse, denominado Função de Caracterização, é obtido demodulando-se a função $r(a)$:

$$g_q(h) = \mathrm{Re}\left\{e^{-jq}\, r(a)\right\} \qquad (3.10)$$

A justificativa para o método proposto se baseia em algumas propriedades:

- *Propriedade de Caracterização*

Primeiramente, para qualquer q fixo, pode ser mostrado que $g_q(h)$ determina unicamente a estrutura de correlação do sinal, ou seja, a seqüência das correlações com atraso k ($r_k(a)$). Por causa desta *Propriedade de Caracterização,* nenhuma informação será perdida se $g_q(h)$ for empregada para representar a estrutura de correlação de X_t. Uma relação entre $g_q(h)$ e o espectro de X_t é dada por:

$$\lim_{h\to 1-} \frac{dg_q(h)}{dh} = \frac{1}{f(q)}$$

onde $f(q) = \sum_k r_k e^{-jkq}$ é a densidade espectral normalizada de X_t.

Como extensão a esta propriedade, tem-se que quando X_t é ruído branco, $g_q(h) = h$ para qualquer q. Este característica pode ser deduzida facilmente, como se segue. Da definição de $r(a)$ em (3.7), tem-se que:

$$r(a) = \frac{E\left\{\left(\overline{a}Y_t(a) + X_{t+1}\right)\cdot \overline{Y}_t(a)\right\}}{E\left\{\left|Y_t(a)\right|^2\right\}} = \overline{a} + \frac{E\left\{X_{t+1}\overline{Y_t}(a)\right\}}{E\left\{\left|Y_t(a)\right|^2\right\}}$$

Entretanto:

$$Y_t(a) = \sum_{l=0}^{\infty} \overline{a}^l X_{t-l} \Rightarrow \overline{Y_t}(a) = \sum_{l=0}^{\infty} a^l X_{t-l}$$

Logo, tem-se que:

$$r(a) = \overline{a} + \frac{E\left\{X_{t+1}\sum_{l=0}^{\infty} a^l X_{t-l}\right\}}{E\left\{\left|Y_t(a)\right|^2\right\}} = \overline{a} + \frac{\sum_{l=0}^{\infty} a^l E\left\{X_{t+1}X_{t-l}\right\}}{E\left\{\left|Y_t(a)\right|^2\right\}} \qquad (3.11)$$

Assim, da definição de $g_q(h)$ em (3.8), tem-se:

$$g_q(h) = \mathrm{Re}\{e^{-jq}\, r(a)\} = \mathrm{Re}\left\{e^{-jq}\left[\overline{a} + \frac{\sum_{l=0}^{\infty} a^l E\{X_{t+1}X_{t-l}\}}{E\{|Y_t(a)|^2\}}\right]\right\} = h + \frac{\sum_{l=0}^{\infty} h^l \cos[q(l+1)]E\{X_{t+1}X_{t-l}\}}{E\{|Y_t(a)|^2\}}$$

Considerando-se X_t um ruído branco, de densidade espectral de potência *No/2*, tem-se que:

$$E\{X_{t+1}X_{t-l}\} = \begin{cases} 0, & l \neq -1 \\ \dfrac{N_0}{2}, & l = -1 \end{cases}$$

Porém, no somatório, l varia no intervalo $[0, \infty)$, podendo-se concluir que:

$$E\{X_{t+1}X_{t-l}\} = 0 \Rightarrow g_q(h) = h$$

Este resultado mostra a imunidade dos parâmetros de autocorrelação demodulada quando a entrada está infectada com ruído branco.

- *Robustez Estatística*

Uma propriedade importante das funções de caracterização da Filtragem Paramétrica é a relativa insensibilidade às variações entre diferentes realizações do mesmo processo aleatório. De fato, a estimação de $g_q(h)$ é bastante robusta a estas variações, ao contrário de outros mecanismos de caracterização de sinais, tais como o Periodograma, que são extremamente sensíveis a variações estatísticas.

- *Monotonicidade*

Para qualquer q fixo, pode-se mostrar que $g_q(h)$ é estritamente crescente em h, para qualquer sinal estacionário. Esta propriedade, conjuntamente com as anteriores, proporciona uma representação da evolução da estrutura de correlação de sinais não-estacionários.

B- Medidas de Distorção

Uma vez determinada a função de caracterização $g_q(h)$, torna-se necessário definir medidas de distorção espectral, a fim de detectar as regiões não-estacionárias do sinal, através das quais serão determinadas as fronteiras dos segmentos.

Dados dois quadros X_{t1} e X_{t2}, a medida de distorção deverá quantificar as diferenças nas suas estruturas de correlação. Algumas medidas têm se destacado quando utilizadas no contexto de Filtragem Paramétrica [Li94], tais como:

- *Distância Lp*

$$g_{\Omega}^{p} = \left\{ \int_{\Omega} \left| g_q^{(1)}(h) - g_q^{(2)}(h) \right|^p dqdh \right\}^{1/p}$$

onde $p \in (0, \infty)$, $g_q^{(1)}(\cdot)$ e $g_q^{(2)}(\cdot)$ são as funções de caracterização obtidas de X_{t1} e X_{t2} e W é um subconjunto de $(-\pi,\pi]$ x $[h_a, h_b]$.

- *Medidas de Divergência Simétricas tipo KL*

$$k_{1\Omega} = \int_{\Omega} \left\{ K\left[\frac{p_q^{(1)}(h)}{p_q^{(2)}(h)} \right] + K\left[\frac{p_q^{(2)}(h)}{p_q^{(1)}(h)} \right] \right\} dqdh$$

$$k_{2\Omega} = \int_{\Omega} \left\{ p_q^{(2)}(h) K\left[\frac{p_q^{(1)}(h)}{p_q^{(2)}(h)} \right] + p_q^{(1)}(h) K\left[\frac{p_q^{(2)}(h)}{p_q^{(1)}(h)} \right] \right\} dqdh$$

onde W é um subconjunto de (-π,π] x $[h_a,h_b]$, $p_q^{(1)}(\cdot)$ e $p_q^{(2)}(\cdot)$, são as funções de densidade de probabilidade normalizadas em $[h_a,h_b] \subset$ (-1,1), definidas por:

$$p_q(h) = \frac{1}{2}\left\{\frac{dg_q(h)}{dh} + [g_q(h_a)+1]d(h-h_a) + [1-g_q(h_b)]d(h-h_b)\right\}$$

Além disto:

$$K(u) = u - \log u - 1$$

Uma medida de distorção espectral ideal teria sua magnitude proporcional ao grau de variação espectral do sinal, em um determinado intervalo de tempo. No entanto, esta propriedade é extremamente difícil de se obter, senão impossível, devido a enorme variabilidade das trajetórias espectrais da fala humana, principalmente nos casos de sistemas de múltiplos locutores. De fato, algumas medidas apresentam problemas neste sentido, uma vez que apresentam acentuada sensibilidade a pequenas flutuações de picos espectrais dominantes [Li95] e insensibilidade a grandes variações do envelope espectral na presença de picos espectrais acentuados, com características similares. Pode-se mostrar, no entanto, que as medidas descritas acima são robustas a estes tipos de problemas.

C- Implementação do Método

O método da Filtragem Paramétrica pode ser implementado através de um banco de filtros (figura (3.3)) que realiza seqüencialmente as operações descritas no item A desta seção, admitindo-se a discretização da variável h.

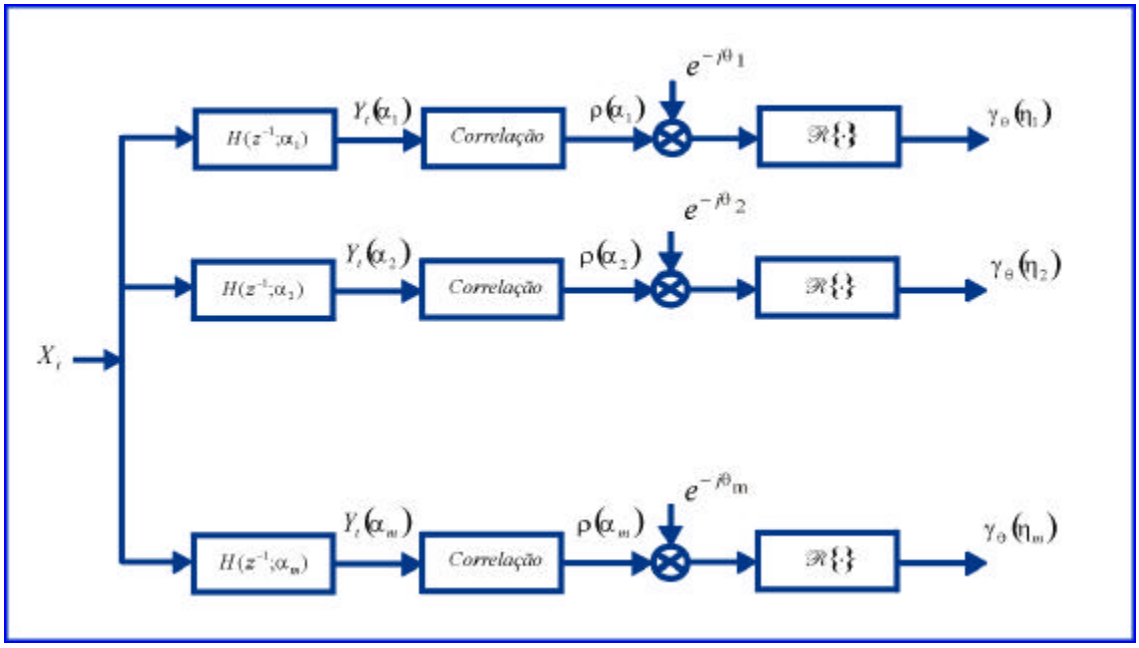

Figura 3.3- Banco de Filtros associado ao método da Filtragem Paramétrica

A medida de distorção $g^2_{q,t}$ foi utilizada por apresentar desempenho semelhante às medidas KL, porém com menor complexidade computacional. Esta medida corresponde uma discretização de g^p_W, para $p = 2$:

$$g^2_{q,t} = \sqrt{N}\frac{1}{m}\sum_{k=1}^{m}\left|g^{(1)}_{q,t}(h_k) - g^{(2)}_{q,t}(h_k)\right|^2$$

onde m é o número de filtros do banco e N é o número de quadros da representação do sinal.

Assumiu-se as seguintes especificações para os parâmetros q e h:

- $q \in \{q_i\}_{i=1}^{m}$
- $h_k \in [h_a, h_b] : h_k = h_a + (k-1)\dfrac{(h_a - h_b)}{(m-1)}$, para $k = 1, \ldots, m$

Neste caso, adotou-se vários valores para os parâmetros h_a , h_b e m. Foi verificado [Gibson96], entretanto, um comportamento mais preciso da função de variação espectral $g^2_{q,t}$ para h_a=0.1, h_b=0.85 e m=4. Vale salientar que o quadro X_t deve passar por um filtro de pré-ênfase, antes de passar pelo banco de filtros paramétricos. Adotou-se o coeficiente de pré-ênfase Cp = 0.95.

Verificou-se, na prática, que a medida de distorção $g^2_{q,t}$ apresentou, em alguns casos, pouca sensibilidade a variações abruptas (plosivas, por exemplo). Decidiu-se, então, combinar esta medida com os parâmetros delta-energia, de modo a compensar este efeito, resultando:

$$s(t) = x \cdot g^{\,2}_{q,t} + (1-x) \cdot \Delta E_t^{(K)}$$

onde $\Delta E_t^{(K)}$ corresponde aos parâmetros delta-energia para $t \in [0,T]$, T é o número de quadros da elocução e *K* é a janela utilizada para a estimação da derivada (vide 3.5).

Finalmente, é realizado um procedimento de normalização sobre *s*(*t*), de modo que esta função fique compreendida no intervalo [0,1].

Desta forma, ficou estabelecida uma primeira abordagem para gerar uma medida de distorção relativamente robusta, que possa ser utilizada pelo Sistema de Reconhecimento de Fala.

3.3.3.2. Redes Multi-Layer Perceptron

Nesta abordagem empregou-se um método de Segmentação Implícita, Independente de Modelo e Supervisionada, baseado em uma Rede Neural Multi-Layer Perceptron (MLP) treinada com o algoritmo Error Back-Propagation.

Trata-se de uma estratégia para a inserção do bloco de treinamento (figura (3.1)) no sistema de segmentação, permitindo a utilização de conhecimento extraído diretamente de dados que compõem o conjunto de treinamento. Em outros sistemas de segmentação do gênero, utiliza-se um conjunto de regras como base de conhecimento, implicando em uma maior complexidade e um comprometimento da robustez e da capacidade de generalização.

Recentemente, várias arquiteturas têm sido propostas como solução para o problema da segmentação automática do sinal de fala em fonemas [Buniet95][Rubio95]. Entretanto, será utilizada a rede MLP por apresentar bom desempenho na maioria das aplicações de Reconhecimento de Padrões. De fato, em [Suh96], tem-se um exemplo de segmentação realizada com uma MLP, onde se obteve resultados significativos.

Em seguida, serão descritos o algoritmo de treinamento e a implementação do sistema de segmentação.

A- Algoritmo de Treinamento

O algoritmo Back-Propagation pode se dividido em dois passos principais. Primeiramente, para cada amostra do conjunto de treinamento, executa-se o passo "forward" e obtém-se o erro quadrático na saída da rede. Em seguida, executa-se o passo "backward" a fim de obter-se as estimativas dos gradientes que serão utilizadas para atualizar as matrizes de pesos da camada intermediária e da camada de saída.

O treinamento pode ser do tipo *instantâneo*, quando as matrizes são atualizadas a cada amostra apresentada à entrada da rede, ou do tipo *lote*, quando o gradiente da função erro é obtido ao final de cada época (apresentação de todo o conjunto de treinamento) como uma média dos gradientes obtidos para cada amostra do conjunto de treinamento. No presente trabalho, utilizou-se treinamento em lote, considerando-se uma rede com N_i entradas, M neurônios na camada intermediária e N neurônios na camada de saída.

Inicialmente, tem-se a descrição das expressões dos gradientes parciais e locais para as camadas de saída e intermediária, considerando, por simplicidade, uma rede de três camadas.

- *Camada de Saída*

O gradiente parcial da função de erro quadrático em relação à matriz de pesos da camada de saída é dado por:

$$\Delta w_{ji}^{Ps}(n) = h \cdot d_j(n) \cdot y_i(n)$$

onde:

j = Índice referente ao neurônio j da camada de saída;

i = Índice referente ao neurônio i da camada intermediária;

h = Taxa de aprendizagem;

$\Delta w_{ji}^{Ps}(n)$ = Estimativa parcial do gradiente da camada de saída**;**

$d_j(n)$ = Gradiente local;

$y_i(n)$ = Saída do neurônio *i* da camada intermediária.

Neste caso, o gradiente local é dado por:

$$d_j(n) = e_j(n) \cdot j'_j[v_j(n)]$$

onde:

$e_j(n)$ = Erro na saída do neurônio *j* da camada de saída;

$j_j(\cdot)$= Função de ativação do neurônio *j* da camada de saída;

$j'_j(\cdot)$= Derivada da função de ativação do neurônio *j* da camada de saída;

$v_j(n)$ =Nível de ativação na entrada do neurônio *j* da camada de saída;

- *Camada Intermediária*

$$\Delta w_{ji}^{Pi}(n) = h \cdot d_j(n) \cdot x_i(n)$$

onde:

j = Índice referente ao neurônio *j* da camada intermediária;

i = Índice referente à entrada *i* da Rede Neural;

h = Taxa de aprendizagem;

$Dw^{Pi}_{ji}(n)$ = Estimativa parcial do gradiente da camada intermediária;

$d_j(n)$ = Gradiente local;

$x_i(n)$ = entrada *i* da Rede Neural.

Neste caso, o gradiente local é dado por:

$$d_j(n) = j'_j[v_j(n)] \cdot \sum_{k=1}^{N} d_k(n) w_{kj}(n)$$

onde:

k = Índice referente ao neurônio *k* da camada de saída;

$j_j(\cdot)$= Função de ativação do neurônio *j* da camada intermediária;

$v_j(n)$ = Nível de ativação na entrada do neurônio *j* da camada intermediária;

w_{kj} = Peso que liga a saída do neurônio *k* da camada intermediária à entrada do neurônio *j* da camada de saída.

O algoritmo foi implementado utilizando função de ativação do tipo tangente hiperbólica, a fim de permitir excursões negativas nas saídas dos neurônios. Esta função é dada por:

$$j(v) = a \cdot \tanh(bv)$$

E sua derivada pode ser escrita na seguinte forma:

$$j'(v) = \frac{b}{2a}\left[a^2 - j^2(v)\right]$$

O parâmetro *a* serve para ajustar os limites de excursão da função e o parâmetro *b* ajusta a declividade da função na região de transição.

Finalmente, o algoritmo em lote é dado por:

1- Inicialização de constantes, vetores e matrizes de pesos;

2- Enquanto o critério de parada não for satisfeito, faça:

- *2.1- Apresentar o conjunto de amostras de entrada à rede, com taxa de aprendizagem fixa ao longo de toda a época e acumular os gradientes a cada iteração;*
- *2.2- Ao final da época, calcular o gradiente médio;*
- *2.3- Atualizar as matrizes de pesos da iteração anterior utilizando o gradiente médio;*
- *2.4- Calcular o erro atual, utilizando as matrizes de pesos atualizadas;*
- *2.5- Volta ao passo 2.*

No passo 2.2 do algoritmo, calcula-se os gradientes médios das camadas de saída e intermediária ao final da época, para *L* amostras de treinamento:

$$grad_{ji}^{i} = \frac{1}{L}\sum_{n=1}^{L}\Delta w_{ji}^{Pi}(n)$$

$$grad_{ji}^{s} = \frac{1}{L}\sum_{n=1}^{L}\Delta w_{ji}^{Ps}(n)$$

Desta forma, tem-se as seguintes equações de atualização dos pesos:

$$w_{ji}^{i}(n_e+1) = w_{ji}^{i}(n_e) + \eta \cdot grad_{ji}^{i}$$

$$w_{ji}^{s}(n_e+1) = w_{ji}^{s}(n_e) + \eta \cdot grad_{ji}^{s}$$

B- Implementação do Método

A rede MLP é utilizada neste trabalho como um estimador para as fronteiras dos segmentos, gerando uma função de variação espectral no tempo do sinal de voz, de forma semelhante às medidas de distorção espectral discutidas no item 3.3.3.1.

Os dados de entrada (amostras) da rede são obtidos a partir da representação $X=\lfloor \bar{x}_1,..,\bar{x}_N \rfloor$ de cada uma das elocuções de treinamento, onde N é o número de quadros da elocução e $\bar{x}_i$ é o i-ésimo quadro da elocução. Constrói-se então as amostras de entrada da rede utilizando uma janela retangular de largura pré-determinada, formando o conjunto $X_{in} = [\bar{x}_{in1},..,\bar{x}_{inN}]$. Como exemplo, tem-se a definição de X_{in} para uma janela de largura igual a três:

$$\begin{bmatrix} \bar{x}_{in1} = [\bar{x}_0, \bar{x}_1, \bar{x}_2]; \\ \bar{x}_{in2} = [\bar{x}_1, \bar{x}_2, \bar{x}_3]; \\ \vdots \\ \bar{x}_{inl} = [\bar{x}_{l-1}, \bar{x}_l, \bar{x}_{l+1}] \\ \vdots \\ \bar{x}_{inN} = [\bar{x}_{N-1}, \bar{x}_N, \bar{x}_0] \end{bmatrix}$$

onde $\bar{x}_0$ é o vetor nulo.

Ao conjunto X_{in} associa-se o conjunto de saídas desejadas $D=[d_1,...,d_N]$, construído a partir das marcas de segmentação manual previamente determinadas por um especialista. Neste trabalho, adotou-se dois tipos de regras para a determinação de D:

- *Alvos Abruptos*

$$d_l = \begin{cases} 1, \text{ se existe marca de segmentação associada a } x_{inl} \\ -1, \text{ caso contrário} \end{cases}$$

Alvos Suaves

$$d_l = \begin{cases} -0.01, \text{se existe uma marca de segmentação associada a } x_{inl+1} \\ 1 \quad , \text{ se existe marca de segmentação associada a } x_{inl} \\ -0.01, \text{se existe uma marca ssociada a } x_{inl-1} \\ -1 \quad , \text{ caso contrário} \end{cases}$$

A arquitetura da rede (vide figura (3.4)) compreendeu uma camada intermediária e uma camada de saída com um único neurônio. Foram adotados diferentes números de neurônios da camada intermediária, a fim de avaliar seu impacto no desempenho do sistema.

Os vetores $x_l \in X$ são formados por 12 coeficientes Mel-Cepstrais, obtidos com Freqüência de Amostragem Fs=11,025 kHz. Como se adotou o tamanho da janela igual a 3, os vetores $x_{inl} \in X_{in}$ são formados a partir da concatenação de 3 vetores acústicos de dimensão 12. Têm, portanto, dimensão 36, que corresponde ao número de entradas da rede.

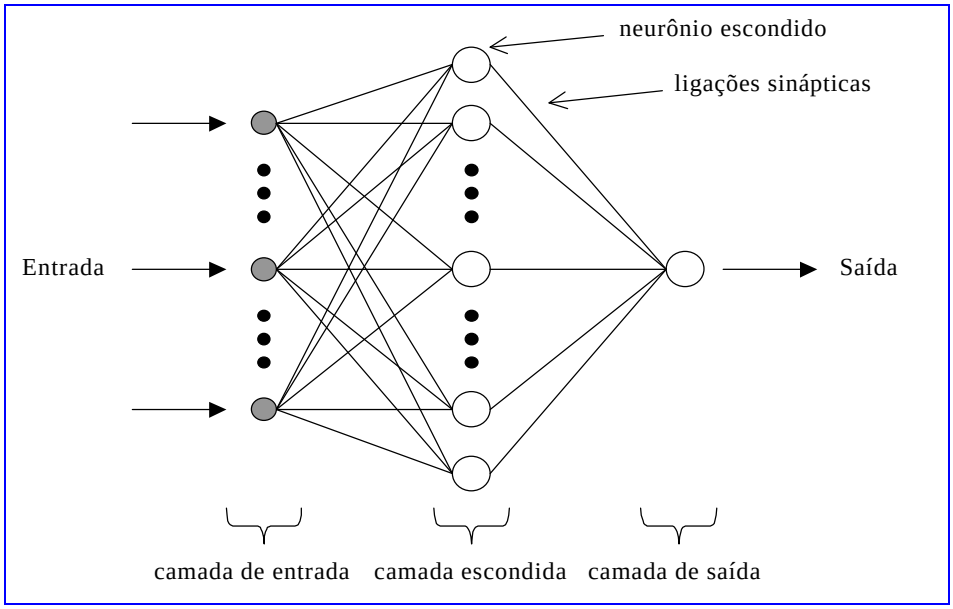

Figura (3.4)- Arquitetura da Rede MLP empregada como segmentador.

3.3.4. Introdução da Informação de Segmentação no Sistema de Reconhecimento

Como foi explicado anteriormente, propomos empregar de forma alternativa a informação de variação espectral contida no sinal de voz. Utilizando-se uma técnica de segmentação automática, é possível estimar com boa precisão as fronteiras dos fonemas de uma elocução. Esta informação pode ser inserida diretamente no processo de reconhecimento, sem a necessidade de se estimar novos parâmetros. Pode-se, então, ponderar os caminhos de Viterbi ao final de cada nível (no caso do Level-Building), favorecendo os modelos que gerarem a melhor segmentação.

Os algoritmos de reconhecimento mais utilizados procuram encontrar conjuntamente a seqüência ótima de fones (ou palavras) e a segmentação ótima da elocução de entrada. De fato, em [Ostendorf89] esta propriedade é descrita com bastante clareza, pois são analisados os problemas de reconhecimento e segmentação separadamente e, em seguida, analisa-se o problema da busca conjunta, sempre resolvidos através de programação dinâmica. É verificado um aumento de 4 a 5% na taxa de erro de reconhecimento de fones ao migrar-se de um sistema que realiza apenas o reconhecimento (segmentação conhecida) para um sistema que realiza conjuntamente o reconhecimento e a segmentação. Por outro lado, na abordagem conjunta, verica-se uma degradação na segmentação obtida, que pode ser agravada pelo modelo exponencial de duração de estados e pela estimação deficiente dos parâmetros dos modelos. Pode-se, portanto, inferir que uma segmentação muito distorcida em relação à ideal é um indicativo da existência de problemas no modelo acústico, os quais provocaram o desalinhamento do caminho de Viterbi em relação à seqüência correta de estados.

Na abordagem proposta, avaliam-se as segmentações parciais ao longo do processo de reconhecimento, as quais são comparadas com as segmentações parciais obtidas por um processo de Segmentação Implícita, gerando-se um fator de ponderação que penaliza os modelos que geraram segmentações desalinhadas.

Este procedimento atenua os problemas relacionados à inconsistência do modelo de duração exponencial, uma vez que, assim como o modelo de duração explícito, permite a determinação de forma mais precisa dos instantes de transição de estados. Adicionalmente, diminui-se os efeitos decorrentes da correlação entre quadros, uma vez que a informação de segmentação é obtida a partir da variação temporal de parâmetros espectrais, facilitando a detecção da redução do nível de correlação entre quadros de segmentos diferentes.

Uma vantagem desta abordagem está em não aumentar o número de parâmetros dos HMM's, diminuindo os problemas de custo computacional no treinamento e evitando a influência negativa da escassez de dados para a estimação dos mesmos. Outra vantagem significativa é que a informação de variação espectral pode ser utilizada de forma mais direta, sem sofrer os efeitos de suavização decorrentes do processo de estimação das densidades de emissão dos símbolos relacionados às derivadas dos parâmetros de entrada.

O Fator de Ponderação Temporal é calculado ao final de cada nível, para todos os modelos e, para cada instante de tempo *t*, é definido por:

$$Fp[w,t,l] = \left\{ \prod_{j=1}^{N_f(w)} s(t_{j,l}) \right\}^{g_g \cdot g_w} \tag{3.12}$$

onde:

$s(t)$ = Medida de Variação Espectral Normalizada (Segmentação Implícita)

$N_f(w)$ = Número de fonemas da palavra *w*;

$\{t_{j,l}\}_{j=1}^{N_f(w)}$ = Seqüência dos instantes das transições entre fonemas, para o modelo *w* e para o caminho ótimo no nível *l*, terminando no instante *t*.

g_g = Fator de atenuação global;

g_w = Fator de atenuação dependente da palavra

Para facilitar a compreensão desta definição, tem-se um exemplo ilustrativo na figura (3.5). Na figura (3.5-a), o Fator de Ponderação Temporal é obtido para o instante *t*, para o modelo w_1 e para o nível $l = 1$. Observa-se que, neste caso, o modelo originou uma segmentação parcial $\{t_i\}_{i=1}^{5}$ alinhada com a informação de variação espectral $s(t)$ e, consequentemente, o modelo w_1 não será muito penalizado. Na figura (3.5-b), mostra-se exatamente o caso oposto, em que o modelo w_2 gerou um segmentação parcial $\{t_i\}_{i=1}^{5}$ desalinhada com relação a $s(t)$, resultando em uma maior penalização para as verossimilhanças obtidas através deste modelo ($F_p(w_2,t,l) < F_p(w_1,t,l)$).

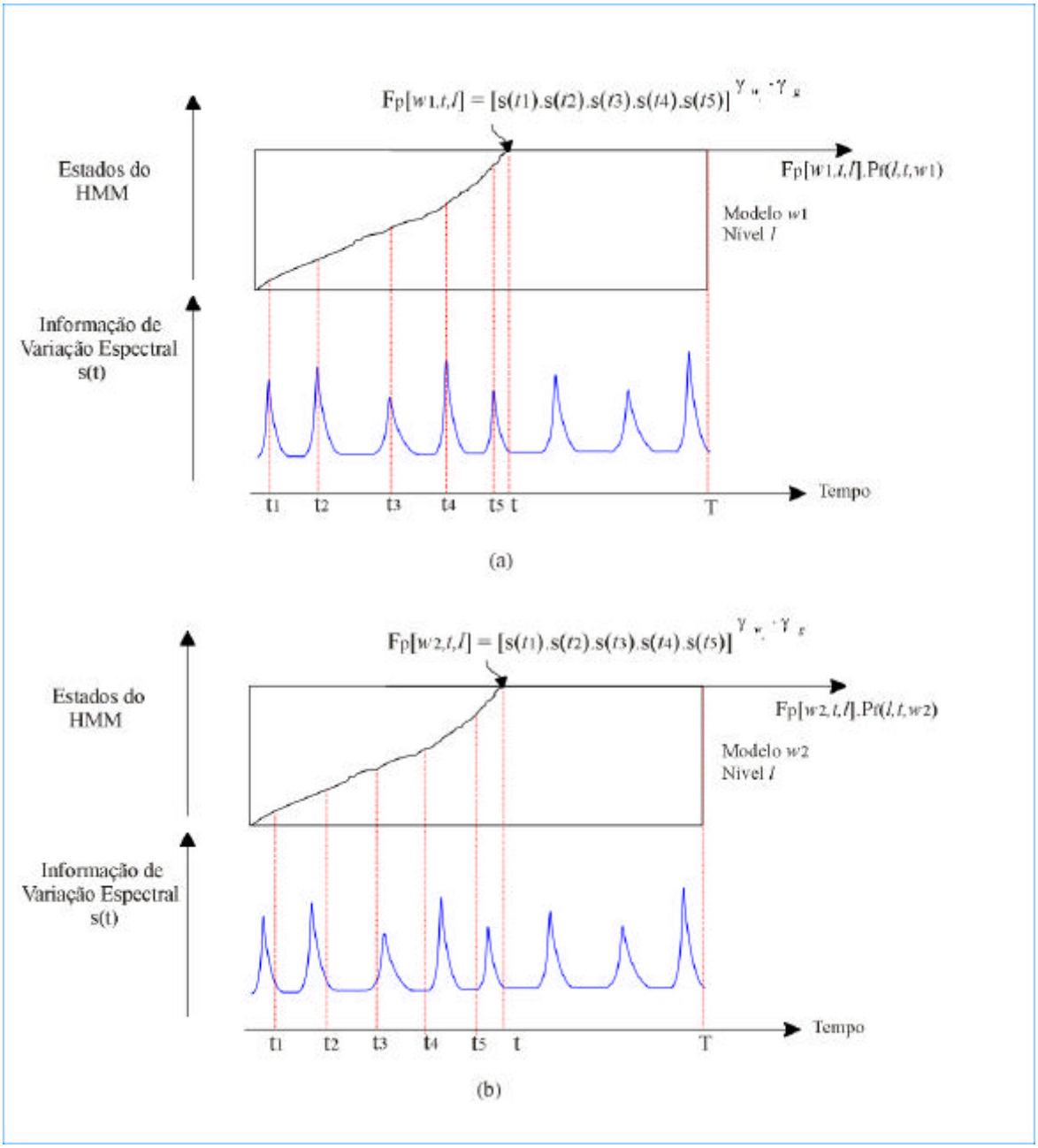

Figura (3.5)- (a) Exemplo da obtenção do Fator de Ponderação Temporal para um modelo w_1 que gerou segmentação alinhada com a informação de variação espectral $s(t)$ (b) Exemplo da obtenção do Fator de Ponderação Temporal para um modelo w_2 que gerou segmentação desalinhada com a informação de variação espectral $s(t)$.

Assim, o valor das verossimilhanças parciais $P_f(w,t,l)$ obtidas durante o Level-Building (vide Capítulo 2), para cada instante ***t***, palavra ***w*** e nível ***l***, passam a ser dadas por:

$$P_{s,w} = P_f(w,t,l) * Fp[w,t,l] = P(w,t,l) * \left\{ \prod_{j=1}^{N_f(w)} s(t_{j,l}) \right\}^{g_g \cdot g_w} \tag{3.13}$$

No entanto, o algoritmo de Viterbi normalmente é implementado aplicando-se logaritmo às probabilidades, de modo que:

$$\overline{P}^{s}_{wtl} = \log\left\{ P_f(w,t,l) * \left[\prod_{j=1}^{N_f(w)} s(t_{j,l}) \right]^{g_g \cdot g_w} \right\} = \overline{P_f}(w,t,l) + g_w \cdot g_g \sum_{j=1}^{N_f(w)} \log s\left(t_{j,l}\right) \tag{3.14}$$

onde $\overline{P_f}(w,t,l) = \log P_f(w,t,l)$

Como se pode observar, o fator $F_p[w,t,l]$ é obtido a partir do produto das magnitudes da função de variação espectral $s(t)$ amostrada nos instantes das transições entre fonemas, correspondentes à seqüência $\{t_{j,l}\}_{j=1}^{N_f(w)}$. Esta seqüência corresponde à segmentação parcial do sinal em fonemas fornecida pelo caminho de Viterbi terminando no instante t, para o nível l e para a palavra w. Desta forma, se as marcas de segmentação representadas por $\{t_{j,l}\}$ não coincidirem com os picos da seqüência $s(t)$, o fator $F_p[w,t,l]$ poderá ser bastante atenuado, penalizando a verossimilhança $P_f(w,t,l)$ e reduzindo as chances do modelo w ser incorporado na seqüência final de palavras reconhecidas. Ou seja, quanto maior o desalinhamento da segmentação gerada pelo modelo w em relação aos picos da seqüência $s(t)$, maior a penalização imposta a este modelo.

O fator de atenuação global g_g é utilizado para ponderar a influência do fator $F_p[w,t,l]$ na verossimilhança $\overline{P}^{s}_{wtl}$, evitando uma penalização excessiva dos modelos. Os fatores g_w servem para ponderar a influência de $F_p[w,t,l]$ de forma diferenciada para cada modelo w, de modo reduzir o grau de confusão entre modelos acusticamente semelhantes (ex., três e seis).

Ao final do algoritmo Level-Building, a frase reconhecida apresentará um valor de verossimilhança modificado pelos fatores de ponderação obtidos ao longo de cada nível. O valor final da verossimilhança pode ser escrito como:

$$P^* = \log\left[P\left(X, q_{\hat{F}}, \hat{F}; \Lambda \right) \cdot S\left(\hat{F} \right) \right] \tag{3.15}$$

onde $\hat{F}$ é a frase reconhecida, $q_{\hat{F}}$ é a seqüência de estados ótima associada a frase $\hat{F}$, $P\left(X, q_{\hat{F}}, \hat{F}; \Lambda\right)$ é a verossimilhança final da frase reconhecida $\hat{F}$ e $S\left(\hat{F}\right)$ corresponde à influência do Fator de Ponderação Temporal, sendo dado por:

$$S\left(\hat{F}\right) = \left[\prod_{l=1}^{L} \left(\prod_{j=1}^{N_{fl}} s\left(t_{lj}\right) \right)^{g_{wl}} \right]^{g_g}$$

onde L é o número de níveis e N_{fl} é o número de fones da l-ésima palavra da frase $\hat{F}$.

Desenvolvendo o logaritmo, obtém-se:

$$P^* = \log P\left(X, q_{\hat{F}}, \hat{F}; \Lambda\right) + g_g \cdot \sum_{l=1}^{L} g_{wl} \cdot \sum_{j=1}^{N_{fl}} \log s\left(t_{lj}\right) \qquad (3.16)$$

Os fatores γ_g e γ_w podem ser determinados empiricamente através do seguinte procedimento:

1. ***Estima-se o fator g_g:***
   - *1.1. Faz-se $g_w = 1$, para todo w;*
   - *1.2. Inicializa-se g_g com um valor pequeno (0.01, por exemplo);*
   - *1.3. Observando os resultados do sistema de reconhecimento, avalia-se a direção em que g_g deve ser ajustado. Caso se verifique pouca influência nos resultados, deve-se aumentar g_g.*
   - *1.4. Volta a 1.3, até que se verifique a estabilização da taxa de erro.*
2. ***Estima-se os parâmetros g_w:***
   - *2.1. Utilizando-se os resultados do sistema de reconhecimento, avalia-se a direção em que g_w deve ser ajustado, para cada w. Caso se verifique que um determinado erro ocorre com certa freqüência (principalmente erros de substituição), procura-se identificar, quando possível, o modelo w_i que está sendo afetado e, dependendo do caso, diminui-se ou aumenta-se o fator g_{wi}.*
   - *2.2. Retorna-se ao passo 2.1. até que o melhor desempenho possível seja atingido.*

Em síntese, esta abordagem apresenta algumas características que permitem sua utilização como uma alternativa para melhorar o desempenho de um sistema de reconhecimento de fala, dentre as quais podemos citar:

1- Não aumenta o número de parâmetros de entrada e nem dos HMM's;
2- Permite a obtenção e utilização da informação de variação espectral de forma direta a partir do sinal a ser reconhecido, evitando-se a utilização de estimativas estatísticas;
3- Não aumenta excessivamente a complexidade computacional do reconhecimento, quando comparado aos algoritmos clássicos de reconhecimento, que não empregam a informação de segmentação diretamente, ao longo do processo de busca.

Para o caso de aplicações com vocabulários restritos, o procedimento acima tende a gerar os fatores g_w e g_g de forma relativamente rápida. No caso de grandes vocabulários, torna-se necessário um procedimento de estimação automática para estes parâmetros, o qual pode ser realizado (como será mostrado mais tarde) utilizando-se algoritmos de treinamento discriminativo.

4. Treinamento Discriminativo de HMM's

4.1. Introdução

A predominância dos HMM's nas pesquisas e aplicações envolvendo Reconhecimento de Fala tem aumentado a necessidade de novos algoritmos voltados à suavização dos problemas inerentes a este modelamento.

A qualidade da estimação dos parâmetros dos HMM's constitui um dos principais fatores determinantes no desempenho deste modelo acústico. Apesar da existência de um algoritmo muito bem definido para a realização da tarefa de treinamento dos parâmetros, denominado algoritmo de Baum-Welch, nem sempre é possível satisfazer todas as condições necessárias para a obtenção de um desempenho ótimo.

O algoritmo de Baum-Welch consiste em um método estatístico de estimação de parâmetros baseado, em última instância, na Teoria da Decisão de Bayes (ou Teoria Bayesiana), que é empregada em Reconhecimento de Padrões para permitir a classificação de observações de entrada, de acordo com um conjunto de probabilidades a posteriori previamente estimadas. Segundo esta abordagem, o problema de Reconhecimento de Padrões pode ser modelado como um problema de

estimação de distribuições de probabilidade que representem adequadamente um conjunto de dados de treinamento.

Entretanto, existem diferenças entre estes dois problemas. Sabe-se que o melhor desempenho em sistemas de reconhecimento é obtido quando são empregadas distribuições (abordagem paramétrica) para representar os parâmetros de emissão (HMM Contínuo). Porém, para definir as distribuições a serem utilizadas é necessário assumir hipóteses sobre os dados disponíveis, que normalmente resultam em um modelamento, no mínimo, limitado. Desta forma, obter a distribuição ótima relativa aos dados de treinamento não implica na minimização da taxa de erro de reconhecimento. Mais detalhes sobre as limitações da abordagem estatística são discutidos na próxima seção.

4.2. Teoria da Decisão de Bayes

Na abordagem baseada na Teoria Bayesiana, propõe-se a utilização de uma função de custo para determinar a classe C_i de uma observação de entrada X. Esta função corresponde a uma estimativa da perda condicional $R(C_i|X)$, obtida a partir dos custos e_{ji} de classificar um elemento da classe i como um elemento da classe j:

$$R(C_i \mid X) = \sum_{j=1}^{M} e_{ji} P(C_j \mid X)$$

Os custos e_{ji} podem ser definidos como se segue:

$$e_{ji} = \begin{cases} 0, & \text{i} = \text{j} \\ 1, & \text{i} \neq \text{j} \end{cases} \qquad i,j \in [1, M]$$

Deste modo, a perda $R(C_i|X)$ é dada por:

$$R(C_i \mid X) = \sum_{j \neq i} P(C_j \mid X) = 1 - P(C_i \mid X)$$

O classificador ótimo pode, portanto, ser obtido através do procedimento denominado "Decisão pelo Máximo a Posteriori", definido pela seguinte expressão:

$$C(X) = C_i \text{ se } P(C_i \mid X) = \max_{1 \le j \le M} P(C_j \mid X)$$

Normalmente as probabilidades a posteriori são estimadas de forma indireta, utilizando-se a Regra de Bayes:

$$P(C_i \mid X) = \frac{P(X \mid C_i) \cdot P(C_i)}{P(X)}$$

Neste caso, observa-se que $P(X)$ não afeta a decisão, já que independe da classe C_i. Pode-se, então, redefinir a função de custo $R(C_i|X)$:

$$R(C_i \mid X) = 1 - P(X \mid C_i) \cdot P(C_i)$$

O critério de decisão também pode ser redefinido de modo a minimizar o Risco de Bayes, como se segue:

$$C(X) = C_i \text{ se } P(C_i \mid X) = \max_{1 \le j \le M} P(X \mid C_j) \cdot P(C_j)$$

As probabilidades $P(C_j)$ dependem de conhecimentos a priori sobre a distribuição das classes. Nos sistemas de Reconhecimento de Fala, estas probabilidades podem, por exemplo, ser fornecidas pelo Modelo da Língua. As probabilidades condicionais $P(X|C_j)$ são obtidas a partir do modelo acústico empregado. Vale ainda salientar que o critério acima descrito é denominado de Máxima Verossimilhança (ML).

Este procedimento estatístico clássico apresenta limitações práticas significativas. Primeiramente, existe uma perda intrínseca no modelamento da distribuição das probabilidades de emissão, mesmo no caso das distribuições paramétricas, tais como Misturas de Gaussianas, empregadas em HMM's contínuos. De fato, esta perda está relacionada à necessidade de assumir que

as variabilidades espectrais do padrão de voz podem ser modeladas exatamente por uma Mistura de Gaussianas, por exemplo. Trata-se, portanto, de uma aproximação, implicando necessariamente em uma distorção em relação à a um hipotético modelo exato. Devido a esta distorção, o critério de Bayes não é satisfeito exatamente, permanecendo o "Risco de Bayes" um limite inferior para o erro de classificação em sistemas práticos. Em outras palavras, como conseqüência da imprecisão dos modelos de distribuição de probabilidades para representar os dados reais, maximizar a verossimilhança não implica em minimizar o erro de classificação. Adicionalmente, a natureza estatística dos algoritmos de estimação de parâmetros empregados nesta abordagem exige a utilização de conjuntos de treinamento bastante extensos, a fim de preservar a confiabilidade dos resultados. Este fato pode representar uma séria limitação prática, já que, em geral, dispõe-se apenas de conjuntos de treinamento restritos.

4.3. Treinamento Discriminativo

Uma abordagem alternativa para o problema da estimação dos parâmetros dos HMM's consiste em utilizar-se o critério do Erro Mínimo de Classificação (MCE - Minimum Classification Error), que consiste em empregar técnicas de otimização baseadas no algoritmo Gradiente Descendente a fim de obter um conjunto de parâmetros que minimize diretamente o erro de reconhecimento do sistema.

Em seguida, serão determinados os algoritmos de Treinamento Discriminativo para diversos casos onde se utilizam diferentes combinações entre:

- Reconhecimento de Fala Contínua e Reconhecimento de Palavras Isoladas;
- HMM's Discreto e HMM's Contínuos;
- Modelos de Palavras e Modelos de Sub-unidades (fonemas);

4.3.1. Caso A: Reconhecimento de Palavras Isoladas, HMM's Contínuos e Modelos de Palavras

A fim de introduzir os principais conceitos que caracterizam um algoritmo de treinamento discriminativo, será descrito um procedimento que pode ser empregado no caso do reconhecimento de Palavras Isoladas, com HMM's Contínuos e Modelos de Palavras [Juang97].

O elemento básico em um algoritmo de treinamento discriminativo é a Função Discriminante $g_i(X;L)$, associada ao *i*-ésimo modelo de um vocabulário, que permite a construção da função de erro a ser utilizada no procedimento de otimização. Pode-se destacar três formas básicas de Função Discriminante:

- $g_i(X;\Lambda) = \sum_{\forall q} g_i(X,q;\Lambda)$ (4.1)
- $g_i(X;\Lambda) = \max_q g_i(X,q;\Lambda)$ (4.2)
- $g_i(X;\Lambda) = \left[\frac{1}{Q}\sum_{q=1}^{Q} g_i^a(X,q;\Lambda)\right]^{\frac{1}{a}}$ (4.3)

onde X é elocução de entrada, q é uma seqüência de estados genérica, Q é o número total das seqüências de estado q e L é o conjunto dos HMM's $\{l_j\}_{j=1}^{W}$ associados às W palavras do vocabulário. Adicionalmente, as funções $g_i(X,q;\Lambda)$ são funções auxiliares avaliadas sobre o caminho q, para a elocução X.

Estas definições são adequadas aos HMM's, uma vez que o processo de decodificação acústica se baseia nos valores de verossimilhança obtidos a partir de seqüências de estados q. Desta forma, pode-se definir a função discriminante a partir dos HMM's, para o modelo de palavra i e para uma elocução composta por T quadros, como:

$$g_i(X,q;\Lambda) = p^{\,i}_{q_0} \prod_{t=1}^{T} a^{i}_{q_{t-1}q_t} \cdot b^{i}_{q_t}(x_t)$$

Neste trabalho, será adotada a forma segmental de função discriminante definida em (4.2), utilizando uma definição para a função $g_i(X,q;L)$ baseada na verossimilhança fornecida pelo HMM. Aplicando a equação (4.2) e tomando o logaritmo, tem-se a seguinte definição para a função discriminante:

$$g_i(X;\Lambda) = \log\left[\max_q g_i(X,q;\Lambda)\right] = \log[g_i(X,\bar{q};\Lambda)]$$

onde $\bar{q}$ é a seqüência de estados ótima ou caminho de máxima verossimilhança.

Uma conseqüência imediata da utilização deste tipo de função discriminante é a necessidade de obter-se, a cada época, as segmentações de cada elocução de treinamento. Em geral este processo é realizado através do algoritmo de Viterbi que, em função dos parâmetros dos modelos HMM, decodifica a seqüência ótima $\overline{q}$.

Desta forma a definição de $g_i(X;L)$ pode ser desenvolvida, resultando na seguinte expressão:

$$g_i(X;\Lambda) = \log p_{\overline{q}_0}^{(i)} + \sum_{t=1}^{T} \log\left(a_{\overline{q}_{t-1}\overline{q}_t}^{(i)}\right) + \sum_{t=1}^{T} \log\left[b_{\overline{q}_t}^{(i)} x(t)\right] \qquad (4.4)$$

A partir da função $g_i(X;L)$, pode-se obter uma função de custo referente ao erro de classificação do sistema. Idealmente, esta função pode ser construída a partir da seguinte propriedade das funções $g_i(X;L)$:

$$k = \arg\max_j g_j\left(X; l_j\right)$$

onde k corresponde à classe C_l reconhecida e $l_j \subset L$.

Desta forma, é possível definir uma função própria para a contagem dos erros de classificação:

$$\hat{l}_i\left(X; l_i\right) = \begin{cases} 1, & X \in C_i \text{ e } i \neq \arg\max_j g_j\left(X; l_j\right) \\ 0, & \text{caso contrário} \end{cases}$$

O processo de estimação dos parâmetros pode ser realizado por meio da minimização do valor esperado do erro de classificação ao longo de todo o conjunto de treinamento, resultando na função de custo $L(\Lambda)$, definida por:

$$L(\Lambda) = E\left[\sum_{k=1}^{W} \hat{l}_k\left(X; l_k\right)\right] \qquad (4.5)$$

Outra possibilidade é a utilização de um estimador para o valor esperado, resultando na definição de uma taxa de erro empírica:

$$L_0(\Lambda) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{W}\hat{l}_k(X_i; l_k) \qquad (4.6)$$

onde N é o tamanho do conjunto de treinamento.

As funções de custo $L(\Lambda)$ e $L_0(\Lambda)$, entretanto, apresentam alguns problemas significativos. Primeiramente, o processo de otimização é numericamente complicado, já que a função não é contínua. Além disto, a característica abrupta da função $l_i(X; l_k)$ implica na incapacidade de discriminar condições de proximidade entre modelos, resultando em perda de informação e na conseqüente degradação do desempenho do sistema.

Uma estratégia para contornar os problemas das funções $L(\Lambda)$ e $L_0(\Lambda)$ foi proposta por Chou, Juang e Lee [Chou92], por meio do algoritmo de treinamento discriminativo de HMM's denominado "Segmental GPD" (Generalised Probabilistic Descent). Este algoritmo apresenta as seguintes características principais:

- Baseia-se no critério MCE (Minimum Classification Error), onde os processos de estimação de parâmetros e segmentação são conjuntamente otimizados;
- A inicialização pode ser feita a partir de outro HMM, obtido a partir de outros critérios, tais como ML e MMI;
- O algoritmo utiliza tanto os erros quanto os acertos de reconhecimento para ajustar os parâmetros dos HMM's de forma teoricamente consistente, maximizando a separabilidade entre as classes concorrentes

No algoritmo "Segmental GPD", propõe-se uma função de custo que se caracteriza pela consistência com a taxa de erro de classificação do sistema e por se adaptar bem aos métodos de otimização mais utilizados. Esta função fundamenta-se em medidas da distância entre a classe correta e as demais classes concorrentes e é obtida seguindo-se um procedimento composto de três etapas.

- *Etapa 1:*

Considerando-se a função discriminante $g_j(X;l_j)$ como sendo o logaritmo da verossimilhança para a entrada X e para o modelo l_j da j-ésima palavra do vocabulário, define-se a função de erro de classificação para classe i como se segue:

$$d_i(X;\Lambda) = -g_i(X;l_i) + \log\left[\frac{1}{W-1}\sum_{j\neq i} e^{g_j(X;l_j)h}\right]^{\frac{1}{h}} \tag{4.7}$$

onde h é um número positivo, W é o número total de classes ou palavras do vocabulário.

Vale salientar que a amostra de treinamento X é uma elocução da palavra i e que os demais modelos do vocabulário geram as funções discriminantes concorrentes. De fato, a contribuição dos modelos concorrentes é introduzida na medida $d_i(X;L)$ com sinal invertido em relação à função discriminante $g_i(X;l_i)$.

A função $d_i(X;L)$ apresenta algumas propriedades e características peculiares. Seja $m_i(j)$ uma medida de distância discreta definida no conjunto de inteiros $\{j \mid j \neq i \text{ e } 1 \leq j \leq W\}$ com peso uniformemente distribuído igual a $\frac{1}{W-1}$, para cada um dos elementos j. Tem-se, então, a seguinte relação:

$$\left[\frac{1}{W-1}\sum_{j\neq i} e^{g_j(X;l_j)h}\right]^{\frac{1}{h}} = \left\| e^{g_j(X;l_j)} \right\|_h$$

onde $\left\| e^{g_j(X;l_j)} \right\|_h$ representa uma norma L^h.

Pode-se mostrar que a norma L^h apresenta a seguinte propriedade assintótica para grandes valores de h:

$$\lim_{h\to\infty} \left\| e^{g_j(X;l_j)} \right\|_h = \max_{j\neq i} e^{g_j(X;l_j)}$$

Esta relação pode ser demonstrada de forma mais direta como se segue:

$$\lim_{h\to\infty}\left[\frac{1}{W-1}\sum_{j\neq i}e^{g_j(X;l_j)h}\right]^{\frac{1}{h}}=\lim_{h\to\infty}\left(\frac{1}{W-1}\right)^{\frac{1}{h}}\cdot e^{g_{j\max}(X;l_{j\max})}\cdot\left\{1+\sum_{i\neq j\max}e^{\left[g_j(X;l_j)-g_{j\max}(X;l_{j\max})\right]h}\right\}^{\frac{1}{h}}=e^{g_{j\max}(X;l_{j\max})}$$

onde $j_{\max}=\arg\max_{j\neq i}g_j(X;l_j)$

Para uma elocução *X* pertencente à classe *i*, valem as seguintes relações:

- Se $d_i(X;\Lambda)>>0$, *houve erro de classificação;*
- Se $d_i(X;\Lambda)<<0$, *a classificação foi correta;*

Estas relações podem ser facilmente deduzidas a partir da equação (4.7).

- *Etapa 2:*

Aproveitando as propriedades da medida $d_i(X)$, pode-se definir uma função de custo suavizada por uma função do tipo sigmóide:

$$l_i(X;l)=l_i[d_i(X;l)]=\frac{1}{1+e^{-g\cdot d_i(X;\Lambda)}} \qquad (4.8)$$

A função $l_i(X;L)$ apresenta, portanto, boas características de diferenciabilidade, sendo limitada ao intervalo (0,1).

O parâmetro *g* pode ser utilizado, em conjunto com o parâmetro *h*, para melhorar a aproximação da função de contagem de erro.

- *Etapa 3:*

Pode-se generalizar a função de custo para todo o conjunto de treinamento, utilizando a função $l_i(X;L)$ e a função indicadora $I(X;W_k)$, resultando na seguinte relação:

$$l(X;\Lambda) = \sum_{k=1}^{W} l_k(X;\Lambda) \cdot I(X;W_k) \qquad (4.9)$$

onde W_k é a porção do conjunto de treinamento correspondente à palavra *k.*

A função indicadora $I(X;W_k)$ é definida por:

$$I(X;W_k) = \begin{cases} 1, & X \in W_k \\ 0, & X \notin W_k \end{cases}$$

4.3.1.1. Método de Otimização

No algoritmo "Segmental GPD", o problema da estimação dos parâmetros dos modelos HMM é mapeado em um problema de otimização baseado em duas possíveis funções de custo.

A- Custo Médio

Neste caso utiliza-se a média ao longo do conjunto de treinamento da medida definida em (4.9), obtendo-se a seguinte definição para o Custo Médio:

$$L(\Lambda) = E_X[l(X;\Lambda)] = \sum_{i=1}^{M} \int_{X \in C_i} l_i(X;\Lambda) p(X) dX$$

O problema de minimização do Custo $L(L)$ pode ser resolvido utilizando-se métodos tradicionais de otimização, tal como o Método do Gradiente Descendente, resultando no seguinte processo iterativo:

$$\Lambda_{k+1} = \Lambda_k - e_k \cdot \nabla l(X;\Lambda_k) \tag{4.10}$$

onde L_k é o conjunto dos modelos HMM na iteração *k*.

Pode-se mostrar que, para garantir a convergência deste algoritmo, são necessárias as seguintes condições:

C1: $\sum_{k=1}^{\infty} e_k = \infty, \quad \sum_{k=1}^{\infty} e_k^2 < \infty, \quad e_k \geq 0$

C2: $\begin{bmatrix} \exists\, 0 \leq V < \infty, \text{ tal que } \forall k\text{:} \\ R_k(e_k) =< \nabla l(X;\Lambda_k), H[X,\Lambda_k + e_k \cdot \nabla l(X,\Lambda)_k] \cdot \nabla l(X;\Lambda_k) >\leq V \end{bmatrix}$

onde $H(\cdot)$ é a matriz Hessiana obtida a partir das derivadas parciais de segunda ordem.

C3: $\begin{bmatrix} \Lambda^* = \arg\min l(X;\Lambda) \text{ é o único } \Lambda \text{ tal que :} \\ \nabla l(X;\Lambda)|_{\Lambda=\Lambda^*} = 0 \end{bmatrix}$

O algoritmo pode, ainda, ser modificado introduzindo-se uma matriz positiva definida U_k de modo a melhorar a estimativa instantânea do gradiente, resultando:

$$\Lambda_{k+1} = \Lambda_k - e_k \cdot U_k \cdot \nabla l(X;\Lambda_k) \tag{4.11}$$

B- Custo Empírico

Seja um conjunto de treinamento formado por N amostras $\{X_i\}_{i=1}^{N}$. A medida empírica de probabilidade P_N definida neste conjunto é uma distribuição de probabilidade discreta uniforme. De forma análoga à expressão (4.6), o Custo Empírico é dado por:

$$L_0(\Lambda) = \frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{M} l_i(X_j;\Lambda)\cdot I(X_j \in C_i) = \int l(X;\Lambda) dP_N$$

Pode-se mostrar que, se os elementos X_i do conjunto de treinamento são obtidas por um processo de amostragem independente com uma distribuição de probabilidades P, a distribuição empírica P_N converge para a P, para grandes valores de N. Ou seja:

$$\lim_{N\to\infty} \int f \cdot dP_N = \int f \cdot dP$$

A partir desta propriedade, conclui-se que o Custo Empírico tende a aproximar o Custo Médio. Entretanto, a qualidade desta estimativa depende do tamanho do conjunto de treinamento e da taxa de convergência da distribuição empírica P_N para a distribuição limite P.

4.3.1.2. Transformação de Parâmetros

O algoritmo de Baum-Welch é o mais empregado para estimar os parâmetros do modelo HMM e se baseia no critério ML. Através deste algoritmo obtém-se fórmulas de reestimação que representam um elegante método para encontrar um máximo local da função objetiva (função de verossimilhança). Este processo de otimização está submetido às restrições lineares que caracterizam o modelamento HMM, a saber:

(1) Para o conjunto $\{p_i\}_{i=1}^{Q}$, tem-se:

$$\sum_{i=1}^{Q} p_i = 1 \quad \text{e} \quad p_i \geq 0, \quad i = 1,...,Q \tag{4.12}$$

onde Q é o número de estados dos modelos das palavras do vocabulário.

(2) Para o conjunto $\{a_{ij}\}_{i,j=1}^{Q}$, tem-se:

$$\sum_{i=1}^{Q} a_{ij} = 1 \quad \text{e} \quad a_{ij} \geq 0, \quad i, j = 1,...,Q \tag{4.13}$$

(3) No caso do HMM discreto com *L* símbolos no codebook, o conjunto $\{b_{jk}\}$ está submetido às seguintes restrições:

$$\sum_{k=1}^{L} b_{jk} = 1 \text{ e } b_{jk} \geq e_1, \quad j = 1,...,Q \text{ e } k = 1,...,L \tag{4.14}$$

onde e_1 é um número positivo próximo de zero.

(4) No caso do HMM contínuo, o conjunto B é associado a uma função densidade de probabilidade paramétrica $b_j(x)$, normalmente formada por uma mistura de gaussianas:

$$b_j(x) = \sum_{k=1}^{L} c_{jk} \cdot N(x, m_{jk}, W_{jk}), \quad j = 1,...,Q$$

onde *m* é o vetor de médias e *W* é a matriz de covariância inversa de dimensão *D* e *x* é um vetor acústico de entrada.

Neste caso, tem-se as seguintes restrições:

$$\sum_{k=1}^{L} c_{jk} = 1 \text{ e } c_{jk} \geq 0, \quad j = 1,...,Q \text{ e } k = 1,...,L \tag{4.15}$$

$$s^{2}_{jmd} \geq e_2, \quad d = 1,...,D \text{ e } m = 1,..,M \tag{4.16}$$

onde e_2 é um número positivo próximo de zero, s^{2}_{jmd} é o *d*-ésimo elemento da diagonal da matriz W_{jk} e *M* é o número de modelos do sistema.

O algoritmo Segmental GPD é um processo de busca irrestrito e, portanto, não obedece necessariamente as restrições acima. Uma possível solução para este problema seria a utilização de métodos de otimização restritos, que incluem em sua concepção a imposição de limites à região a que podem pertencer os parâmetros estimados.

Um exemplo deste tipo de abordagem é o Método da Projeção do Gradiente [Huo93][Huo95], segundo o qual o processo de busca do máximo local é realizado por meio da projeção do gradiente no sub-espaço formado a partir das restrições descritas acima. Em outras palavras, trata-se de um método de gradiente ascendente no sub-espaço definido pelas restrições ativas dos parâmetros do HMM. Este algoritmo é linearmente convergente e a taxa de convergência é determinada pelos autovalores da Hessiana e da Lagrangiana no espaço das restrições ativas.

Uma outra abordagem, proposta em [Chou92], consiste em realizar transformações sobre os parâmetros a fim de manter todas as restrições ao longo do processo de treinamento. A transformação e normalização exponencial, originalmente proposta em [Bridle90], pode ser empregada para satisfazer as restrições (4.12)-(4.15). Para estes casos, tem-se:

(1) Para os parâmetros $\{p_i\}_{i=1}^{Q}$:

$$p_i = \frac{\exp(\bar{p}_i)}{\sum_{j=1}^{Q} \exp(\bar{p}_i)}, \quad i = 1,...,Q \tag{4.17}$$

onde $\{\bar{p}_i\}_{i=1}^{Q}$ são os parâmetros irrestritos estimados no treinamento discriminativo.

(2) Para os parâmetros $\{a_{ij}\}_{i,j=1}^{Q}$:

$$a_{ij} = \frac{\exp(\bar{a}_{ij})}{\sum_{j=1}^{Q}\exp(\bar{a}_{ij})}, \qquad i = 1,...,Q \tag{4.18}$$

onde $\{\bar{a}_{ij}\}_{i,j=1}^{Q}$ são os parâmetros irrestritos.

(3) Para os parâmetros $\{b_{jk}\}$:

$$b_{jk} = \frac{\exp(\bar{b}_{jk})}{\sum_{k=1}^{L}\exp(\bar{b}_{jk})}, \qquad j = 1,...,Q \tag{4.19}$$

onde $\{\bar{b}_{jk}\}$ são os parâmetros irrestritos.

(4) Para os parâmetros $\{c_{jk}\}$:

$$c_{jk} = \frac{\exp(\bar{c}_{jk})}{\sum_{k=1}^{L}\exp(\bar{c}_{jk})}, \qquad j = 1,...,Q \tag{4.20}$$

onde $\{\bar{c}_{jk}\}$ são os parâmetros irrestritos.

Para os parâmetros $\{s_{jm}^{2}\}$, entretanto, emprega-se uma transformação logarítmica mais simples, cuja função é apenas garantir a propriedade (4.16):

$$\bar{s}_{jmd}^{2} = \log s_{jmd}^{2} \tag{4.21}$$

onde $\{\bar{s}_{jmd}^{2}\}$ são os parâmetros irrestritos.

Por fim, para os parâmetros $\{m_{jm}\}$, aplica-se uma transformação linear a partir da variância, bastante útil para compensar os efeitos numéricos associados a valores pequenos de s_{jm}:

$$m_{jmd} = \overline{m}_{jmd} \cdot s_{jmd} \tag{4.22}$$

onde $\{\overline{m}_{jmd}\}$ são os parâmetros irrestritos.

O método da Transformação de Parâmetros tem sido mais utilizado nos trabalhos envolvendo treinamento discriminativo, principalmente pela baixa complexidade computacional. Esta abordagem apresenta, no entanto, algumas desvantagens.

Primeiramente, os procedimento de normalização da variância tendem a aumentar a dispersão dos parâmetros dos modelos HMM's e podem afetar o desempenho do algoritmo Segmental GPD. Verificou-se em experimentos práticos [Juang97], que a variância pode apresentar diferenças na ordem de 10^4 a 10^6 vezes entre HMM's. Devido a este fato, a utilização de um parâmetro e_k (equação 4.10) constante para todos os modelos não é um procedimento adequado para o problema. De fato, um e_k pode ser um passo muito pequeno para o ajuste de um conjunto de parâmetros e, ao mesmo tempo, muito grande para outro conjunto. As chances deste fato ocorrer aumentam quando se considera a grande quantidade de parâmetros que compõem os modelos de um sistema de reconhecimento (tipicamente entre 10^4 e 10^5, para modelos de palavras e HMM's Contínuos). A utilização da matriz U_k é uma alternativa para compensar estas diferenças de sensibilidade tão acentuadas. Normalmente trata-se de uma matriz diagonal positiva definida, dada por:

$$U_k = diag(s_1^2(k),...,s_D^2(k))$$

onde $s_d^2(k)$ é a variância na iteração k para a dimensão d dos parâmetros de um HMM.

Outro problema é o aumento do grau de não-linearidade na função objetiva devido a natureza exponencial das transformações. Consequentemente, há um aumento no número de mínimos locais, dificultando o processo de otimização de primeira ordem em que se baseia o Segmental GPD. Apesar de não haver solução teórica para este problema, os experimentos têm mostrado que a

transformação de parâmetros é uma técnica viável, pois apresenta baixa complexidade computacional e geralmente proporciona bons resultados.

O Método da Projeção do Gradiente é bastante robusto aos problemas que afetam o método de Transformação de Parâmetros. O principal problema desta abordagem está no elevado custo computacional que é inerente a algoritmos de otimização restrita. Entretanto, com a evolução destas técnicas e com o aumento da velocidade dos computadores, este tipo de solução tende a se tornar mais atrativa no futuro.

Neste trabalho, por questões de simplicidade, será adotado o método de Transformação de Parâmetros

4.3.1.3. Estimação dos Parâmetros

O procedimento de treinamento utilizando o algoritmo Segmental GPD, para o caso dos HMM contínuos, inclui as expressões para o ajuste dos parâmetros $\left\{a_{ij}^{(i)}\right\}_{i,j=1}^{Q}$ e $\left\{b_j^{(i)}(x)\right\}_{j=1}^{Q}$, que compõem o modelo *i*. Os parâmetros $b_j^{(i)}(x)$ são, no caso dos HMM Contínuos, funções densidade de probabilidade formadas por misturas de Gaussianas:

$$b_j^{(i)}(x) = \sum_{k=1}^{K} c_{jk}^{(i)} \cdot N\left(x; m_{jk}^{(i)}, W_{jk}^{(i)}\right), \quad j = 1,...,Q \tag{4.23}$$

onde:

$$\begin{aligned} N\left(x; m_{jk}^{(i)}, W_{jk}^{(i)}\right) &= \frac{1}{(2p)^{\frac{d}{2}} \left|W_{jk}^{(i)}\right|^{\frac{1}{2}}} \cdot \exp\left\{-\frac{1}{2}\left(x - m_{jk}^{(i)}\right)^t \cdot \left|W_{jk}^{(i)}\right|^{-1} \cdot \left(x - m_{jk}^{(i)}\right)\right\} \\ N\left(x; m_{jk}^{(i)}, W_{jk}^{(i)}\right) &= \frac{1}{(2p)^{\frac{d}{2}} \left|W_{jk}^{(i)}\right|^{\frac{1}{2}}} \cdot \exp\left\{-\frac{1}{2}\sum_{l=1}^{D}\left(\frac{x_{tl} - m_{jkl}^{(i)}}{s_{jkl}^{(i)}}\right)^2\right\} \end{aligned} \tag{4.24}$$

é a mistura de Gaussianas, composta pelos vetores médios $\left\{m_{jk}^{(i)}\right\}$, pela matriz de covariância $W_{jk}^{(i)}$ e pelos fatores $\left\{c_{jk}^{(i)}\right\}$.

Em geral, assume-se, por simplicidade, que a matriz de covariância é do tipo diagonal, de dimensão *D*, sendo definida por:

$$W_{jk}^{(i)} = \begin{bmatrix} s_{jk1}^{2\,(i)} & 0 & \dots & 0 \\ 0 & s_{jk2}^{2\,(i)} & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & s_{jkD}^{2\,(i)} \end{bmatrix}$$

onde $\left\{s_{jkl}^{2\,(i)}\right\}_{l=1}^{D}$ é o conjunto das variâncias que caracterizam cada modelo.

Uma vez estabelecidos os parâmetros que compõem o sistema, podem ser determinadas as equações de ajuste de cada tipo de parâmetro. As equações são aplicadas aos parâmetros irrestritos e, a cada iteração, são aplicadas as transformações descritas no item 4.3.1.2.

(1) Parâmetros $\left\{m_{jk}^{(i)}\right\}$*:*

A equação de ajuste para o *l*-ésimo elemento do vetor irrestrito $\overline{m}_{jk}^{(i)}$ é obtida a partir da definição (4.10), resultando:

$$\overline{m}_{jkl}^{(i)}(n+1) = \overline{m}_{jkl}^{(i)}(n) - e \cdot \frac{\partial l_i(X;\Lambda)}{\partial \overline{m}_{jkl}^{(i)}}$$

onde,

$$\frac{\partial l_i(X;\Lambda)}{\partial \overline{m}_{jkl}^{(i)}} = \frac{\partial l_i(X;\Lambda)}{\partial d_i(X;\Lambda)} \cdot \frac{\partial d_i(X;\Lambda)}{\partial \overline{m}_{jkl}^{(i)}}$$

A partir da equação (4.8), obtém-se a seguinte relação:

$$\frac{\partial l_i(X;\Lambda)}{\partial d_i(X;\Lambda)} = g \cdot l_i(X;\Lambda)\cdot[1 - l_i(X;\Lambda)] \tag{4.25}$$

O termo $\frac{\partial d_i(X;\Lambda)}{\partial \overline{m}_{jkl}^{(i)}}$ pode ser determinado por meio das equações (4.4) e (4.7):

$$\frac{\partial d_i(X;\Lambda)}{\partial \overline{m}_{jkl}^{(i)}} = -\sum_{t=1}^{T} d(\overline{q}_t, j)\frac{\partial \log b_j^{(i)}(x_t)}{\partial \overline{m}_{jkl}^{(i)}} \tag{4.26}$$

onde $d(x, y)$ é a Função Indicadora, definida por:

$$d(x, y) = \begin{cases} 0, \text{ se } x \neq y \\ 1, \text{ se } x = y \end{cases}$$

Finalmente, o termo $\frac{\partial \log b_j^{(i)}(x_t)}{\partial \overline{m}_{jkl}^{(i)}}$ pode ser calculado utilizando as equações (4.23) e (4.24):

$$\frac{\partial \log b_j^{(i)}(x_t)}{\partial \overline{m}_{jkl}^{(i)}} = c_{jk}^{(i)} \cdot (2p)^{-\frac{d}{2}} \cdot \left|W_{jk}^{(i)}\right|^{-\frac{1}{2}} \cdot \left[b_j^{(i)}(x_t)\right]^{-1} \cdot \left(\frac{x_{tl} - m_{jkl}^{(i)}}{s_{jkl}^{(i)}}\right) \cdot \exp\left\{-\frac{1}{2}\sum_{l=1}^{D}\left(\frac{x_{tl} - m_{jkl}^{(i)}}{s_{jkl}^{(i)}}\right)^2\right\} \tag{4.27}$$

onde x_{tl} é o l-ésimo elemento do vetor acústico $x_t \in X$ e X é uma elocução de treinamento pertencente à classe C_i.

Aplicando a transformação descrita em (4.22), são obtidos os parâmetros restritos $\left\{m_{jk}^{(i)}\right\}$:

$$m_{jkl}^{(i)}(n+1) = \overline{m}_{jkl}^{(i)}(n+1)\cdot s_{jkl}^{(i)}(n)$$

(2) Parâmetros $\left\{s_{jk}^{(i)}\right\}$:

Para efetuar o ajuste dos parâmetros $\left\{s_{jk}^{(i)}\right\}$, correspondente ao *l*-ésimo elemento da diagonal da matriz $W_{jk}^{(i)}$, deve-se realizar um procedimento análogo ao cálculo dos parâmetros $\left\{m_{jk}^{(i)}\right\}$. Novamente, o processo de estimação é realizado sobre os parâmetros irrestritos.

$$\overline{s}_{jkl}^{(i)}(n+1) = \overline{s}_{jkl}^{(i)}(n) - e \cdot \frac{\partial l_i(X;\Lambda)}{\partial \overline{s}_{jkl}^{(i)}}$$

O gradiente é calculado a partir das equações (4.4), (4.7) e (4.8), resultando:

$$\frac{\partial l_i(X_n;\Lambda)}{\partial \overline{s}_{jkl}^{(i)}} = \frac{\partial l_i(X;\Lambda)}{\partial d_i(X;\Lambda)} \cdot \frac{\partial d_i(X;\Lambda)}{\partial \overline{s}_{jkl}^{(i)}} = -g \cdot l_i(X;\Lambda) \cdot \left[1 - l_i(X;\Lambda)\right] \cdot \sum_{t=1}^{T} d(\overline{q}_t, j) \frac{\partial \log b_j^{(i)}(x_t)}{\partial \overline{s}_{jkl}^{(i)}} \quad (4.28)$$

O termo $\dfrac{\partial \log b_j^{(i)}(x_t)}{\partial \overline{s}_{jkl}^{(i)}}$ é encontrado a partir das equações (4.23) e (4.24):

$$\frac{\partial \log b_j^{(i)}(x_t)}{\partial \overline{s}_{jkl}^{(i)}} = c_{jk}^{(i)} \cdot (2p)^{-\frac{d}{2}} \cdot \left|W_{jk}^{(i)}\right|^{-\frac{1}{2}} \cdot \left[b_j^{(i)}(x_t)\right]^{-1} \cdot \left[\left(\frac{x_{tl} - m_{jkl}^{(i)}}{s_{jkl}^{(i)}}\right)^2 - 1\right] \cdot \exp\left\{-\frac{1}{2}\sum_{l=1}^{D}\left(\frac{x_{tl} - m_{jkl}^{(i)}}{s_{jkl}^{(i)}}\right)^2\right\} \quad (4.29)$$

Aplica-se, então, a transformação definida em (4.21), obtendo-se:

$$s_{jkl}^{(i)}(n+1) = \exp\left[\overline{s}_{jkl}(n+1)\right]$$

(3) Parâmetros $\left\{c_{jk}^{(i)}\right\}$:

Para efetuar o ajuste dos parâmetros de ponderação $\left\{c_{jk}^{(i)}\right\}$ realiza-se um procedimento análogo aos descritos nas equações (4.23) a (4.26). Inicialmente, a equação de ajuste dos pesos é dada por:

$$\bar{c}_{jk}^{(i)}(n+1) = \bar{c}_{jk}^{(i)}(n) - e \cdot \frac{\partial l_i(X;\Lambda)}{\partial \bar{c}_{jk}^{(i)}}$$

A partir das equações (4.4), (4.7) e (4.8), são obtidos os gradientes de acordo com a seguinte expressão:

$$\frac{\partial l_i(X;\Lambda)}{\partial \bar{c}_{jk}^{(i)}} = \frac{\partial l_i(X;\Lambda)}{\partial d_i(X;\Lambda)} \cdot \frac{\partial d_i(X;\Lambda)}{\partial \bar{c}_{jk}^{(i)}} = -g \cdot l_i(X;\Lambda) \cdot [1 - l_i(X;\Lambda)] \cdot \sum_{t=1}^{T} d(\bar{q}_t, j) \frac{\partial \log b_j^{(i)}(x_t)}{\partial \bar{c}_{jk}^{(i)}} \quad (4.30)$$

O termo $\frac{\partial \log b_j^{(i)}(x_t)}{\partial \bar{c}_{jk}^{(i)}}$ pode ser calculado utilizando a equação (4.20):

$$\frac{\partial \log b_j^{(i)}(x_t)}{\partial \bar{c}_{jk}^{(i)}} = \frac{1}{b_j^{(i)}(x_t) \cdot (2p)^{\frac{d}{2}} \cdot \left|W_{jk}^{(i)}\right|^{\frac{1}{2}}} \cdot \exp\left[-\frac{1}{2}\sum_{l=1}^{D}\left(\frac{x_{tl} - m_{jkl}^{(i)}}{s_{jkl}^{(i)}}\right)^2\right] \cdot \frac{\partial c_{jk}^{(i)}}{\partial \bar{c}_{jk}^{(i)}} \quad (4.31)$$

O termo $\frac{\partial c_{jk}^{(i)}}{\partial \bar{c}_{jk}^{(i)}}$ pode ser calculado a partir da equação referente à transformação dos parâmetros $\left\{c_{jk}^{(i)}\right\}$:

$$\frac{\partial c_{jk}^{(i)}}{\partial \bar{c}_{jk}^{(i)}} = c_{jk}^{(i)}\left(1 - c_{jk}^{(i)}\right)$$

Desta forma, a equação resultante para o gradiente da função discriminante é dada por:

$$\frac{\partial \log b_j^{(i)}(x_t)}{\partial \bar{c}_{jk}^{(i)}} = \frac{c_{jk}^{(i)}\left(1 - c_{jk}^{(i)}\right)}{b_j^{(i)}(x_t) \cdot (2p)^{-\frac{d}{2}} \cdot \left|W_{jk}^{(i)}\right|^{-\frac{1}{2}}} \cdot \exp\left[-\frac{1}{2}\sum_{l=1}^{D}\left(\frac{x_{tl} - m_{jkl}^{(i)}}{s_{jkl}^{(i)}}\right)^2\right] \quad (4.32)$$

Finalmente, é aplicada a transformação (4.20) sobre os parâmetros $\left\{c_{jk}^{(i)}\right\}$, resultando:

$$c_{jk}^{(i)}\left(n+1\right)=\frac{\exp\left[\bar{c}_{jk}^{(i)}\left(n+1\right)\right]}{\sum_{k=1}^{K}\exp\left[\bar{c}_{jk}^{(i)}\left(n+1\right)\right]},\qquad j=1,\ldots,Q$$

(4) Parâmetros $\left\{a_{ij}\right\}$:

Para efetuar o ajuste dos parâmetros de ponderação $\left\{a_{ij}\right\}$ realiza-se um procedimento análogo aos descritos nas equações (4.22) a (4.25). Para evitar ambigüidades de notação, considera-se, neste caso, o processo de estimação referente ao modelo *v*.

$$\bar{a}_{ij}^{(v)}\left(n+1\right)=\bar{a}_{ij}^{(v)}\left(n\right)-e\cdot\frac{\partial l_i\left(X;\Lambda\right)}{\partial\bar{a}_{ij}^{(v)}}$$

A partir das equações (4.4), (4.7) e (4.8), o gradiente pode ser obtido como se segue:

$$\frac{\partial l_i\left(X;\Lambda\right)}{\partial\bar{a}_{ij}^{(v)}}=\frac{\partial l_i\left(X;\Lambda\right)}{\partial d_i\left(X;\Lambda\right)}\cdot\frac{\partial d_i\left(X;\Lambda\right)}{\partial\bar{a}_{ij}^{(v)}}$$

O termo $\dfrac{\partial l_i\left(X;\Lambda\right)}{\partial d_i\left(X;\Lambda\right)}$ é dado por:

$$\frac{\partial l_i\left(X;\Lambda\right)}{\partial d_i\left(X;\Lambda\right)}=g\cdot l_i\left(X;\Lambda\right)\cdot\left[1-l_i\left(X;\Lambda\right)\right]$$

O termo $\dfrac{\partial d_i\left(X;\Lambda\right)}{\partial\bar{a}_{ij}^{(v)}}$ é calculado a partir da definição (4.4). Neste caso, porém, deverão ser consideradas dois estados adjacentes do caminho de Viterbi, a fim de detectar as transições de estados desejadas. Resulta, então, a seguinte expressão:

$$\frac{\partial d_i(X;\Lambda)}{\partial \bar{a}_{ij}^{(v)}} = -\sum_{t=2}^{T} d(\bar{q}_{t-1}, i) \cdot d(\bar{q}_t, j) \cdot \frac{\partial \log a_{ij}^{(v)}}{\partial \bar{a}_{ij}^{(v)}}$$

O termo $\frac{\partial \log a_{ij}^{(v)}}{\partial \bar{a}_{ij}^{(v)}}$ pode ser facilmente obtido como se segue:

$$\frac{\partial \log a_{ij}^{(v)}}{\partial \bar{a}_{ij}^{(v)}} = \frac{1}{a_{ij}^{(v)}} \cdot \frac{\partial a_{ij}^{(v)}}{\partial \bar{a}_{ij}^{(v)}}$$

A partir da equação de transformação (4.18), tem-se:

$$\frac{\partial a_{ij}^{(v)}}{\partial \bar{a}_{ij}^{(v)}} = a_{ij}^{(v)}\left(1 - a_{ij}^{(v)}\right) \Rightarrow \frac{\partial \log a_{ij}^{(v)}}{\partial \bar{a}_{ij}^{(v)}} = \left(1 - a_{ij}^{(v)}\right) \tag{4.33}$$

Aplica-se, então, a transformação (4.18) aos parâmetros $\{a_{ij}(n+1)\}$, sendo obtida a seguinte relação:

$$a_{ij}^{(v)}(n+1) = \frac{\exp\left[\bar{a}_{ij}^{(v)}(n+1)\right]}{\sum_{j=1}^{Q} \exp\left[\bar{a}_{ij}^{(v)}(n+1)\right]}$$

Com estes procedimentos, ficam definidos os passos necessários para estimar os parâmetros dos HMM's através do treinamento discriminativo, para o caso de palavras isoladas.

4.3.1.4. Síntese do Algoritmo de Treinamento Discriminativo para Palavras Isoladas

O algoritmo de Treinamento Discriminativo para o caso dos HMM Contínuos, com palavras isoladas, pode ser sintetizado nas seguintes etapas:

1- ***Inicialização*:** Os parâmetros dos HMM são inicializados a partir de outro critério de treinamento (ML, MMI ou MDI). Normalmente, são utilizados os modelos gerados através do algoritmo de Baum-Welch (critério ML);

2- ***Segmentação do Conjunto de Treinamento*:** Através do algoritmo de Viterbi, encontram-se as segmentações de cada elocução de treinamento, a fim de compor a função de custo do algoritmo Segmental GPD;

3- ***Cálculo dos Gradientes das Funções de Custo:*** A partir dos parâmetros atuais, são calculados os gradientes das funções de custo associadas a cada parâmetro.

4- ***Ajuste dos Parâmetros*:** Uma vez calculados os gradientes, aplica-se as equações de ajuste provenientes da equação (4.10) a fim de obter os novos parâmetros;

5- ***Transformação de Parâmetros:*** Finalmente, realizam-se as operações de transformação de parâmetros definidas nas equações (4.17) à (4.22);

6- ***Parada*:** Caso o critério de parada não seja satisfeito, voltar ao passo 2. Normalmente, o critério de parada adotado está associado a um número máximo de épocas de treinamento.

Vale ainda salientar que o método de treinamento pode ser classificado de acordo com o esquema de apresentação das amostras de treinamento:

- *Apresentação por Lote*: Os gradientes utilizados para realizar o ajuste dos parâmetros são calculados a partir das médias dos gradientes parciais, os quais são obtidos para cada elocução do conjunto de treinamento;
- *Apresentação Instantânea*: Neste caso, os parâmetros são ajustados utilizando gradientes instantâneos obtidos para cada elocução de entrada.

Também é possível adotar soluções híbridas, em que o conjunto de treinamento é particionado em vários subconjuntos utilizados para atualizações parciais dos parâmetros ao longo de uma época.

4.3.2. Caso B: Reconhecimento de Fala Contínua, HMM's Discretos, Modelos de Fones

Nesta seção será descrito o algoritmo de treinamento discriminativo aplicado no contexto de Reconhecimento de Fala Contínua, utilizando HMM's Discretos e Modelos de Subunidades (fones). O procedimento a ser utilizado é, em linhas gerais, análogo ao empregado no Caso A, pois a função de custo adotada é bastante semelhante.

O problema de Reconhecimento de Fala Contínua, como se sabe, é bem mais complexo que o Reconhecimento de Palavras Isoladas, por diversos motivos, tais como:

- As fronteiras entre as palavras são desconhecidas e nem sempre podem ser determinadas com precisão, devido aos efeitos de coarticulação;
- O número de palavras na frase é desconhecido;
- O vocabulário tende a crescer bastante em aplicações práticas, implicando na necessidade de modelamento de subunidades (fones dependentes ou independentes de contexto, trifones, etc), bem como na integração de várias fontes de conhecimento, dentre os quais destacam-se os modelos da língua;

Uma das conseqüências deste aumento de complexidade é a degradação da performance dos algoritmos de treinamento tradicionais, principalmente devido à necessidade de bases de dados cada vez mais extensas. A estratégia de modelamento de subunidades atenua um pouco este problema, pois há uma grande redução no número de modelos a serem treinados. O problema desta abordagem está na dificuldade de generalização, uma vez que o modelo de uma determinada subunidade deve representar de forma consistente suas características acústicas, que apresentam grande variabilidade devido aos diferentes contextos em que podem ser encontradas.

Os algoritmos de treinamento discriminativo podem ser utilizados também para o treinamento dos modelos das subunidades, como descrito nos trabalhos de Chen e Soong [Chen94],

Reichl e Ruske [Reichl95] e Chou, Juang e Lee [Chou93]. Neste caso, porém, algumas modificações devem ser introduzidas, uma vez que as elocuções de treinamento não são mais palavras isoladas e sim frases contínuas compostas pelas palavras do vocabulário.

Primeiramente, as funções discriminantes continuarão sendo de natureza segmental e, portanto, um procedimento de segmentação automática deve ser empregado. Entretanto, este procedimento deve ser apropriado para a fala contínua e deve permitir a obtenção de uma verossimilhança que deve ser associada à função discriminante. Em outras palavras, necessita-se de um algoritmo de reconhecimento e segmentação conjuntos. Neste trabalho, será utilizado o algoritmo Level-Building para este fim, mas outros algoritmos de busca mais eficientes poderiam também ser empregados, tais como o One-Step, Herman-Ney ou A* [Fagundes98].

Outro fato a ser considerado diz respeito à necessidade de um algoritmo capaz de identificar as *N* frases mais prováveis fornecidas por um sistema de reconhecimento de Fala Contínua. Alguns aspectos destes algoritmos são discutidos a seguir.

4.3.2.1. Algoritmo de Busca das N Frases Candidatas

O problema de busca das *N* frases mais prováveis em um sistema de reconhecimento de fala contínua vem sendo bastante explorado recentemente, pois sua resolução permite a utilização de informações adicionais em etapas de pós-processamento. Estas informações podem ser de natureza lingüística, temporal ou acústica e permitem recuperar a frase correta dentre as *N* frases candidatas.

O algoritmo de treinamento discriminativo aplicado no contexto de Fala Contínua deve ser realizado a nível de frases. Para tanto, é necessário definir as frases que deverão concorrer com a correta, a fim de viabilizar a composição de uma função de custo com propriedades discriminativas. Neste caso, ao contrário do reconhecimento de palavras isoladas, nem sempre é possível a construção de um subconjunto de frases de treinamento que sejam concorrentes a frase correta. Além disto, para garantir as propriedades discriminativas a nível de frases, é imprescindível que apenas as sentenças que realmente causam confusão no processo de reconhecimento sejam consideradas como concorrentes. Para obter estas frases, deve-se utilizar um algoritmo de reconhecimento de *N* frases candidatas, que geram os *N* maiores valores de verossimilhança ao longo do processo de reconhecimento.

A utilização de algoritmos de busca de *N* candidatas em treinamentos discriminativos foi proposta inicialmente por Chow [Chow90]. Em seguida, vários outros trabalhos foram realizados nesta área, destacando-se as técnicas propostas por Chou, Lee e Juang [Chou93], Chen e Soong [Chen94] e Reichl e Ruske [Reichl95]. Como resultado destas pesquisas, surgiu uma abordagem predominante que se baseia em um algoritmo de busca em árvore, denominado "Tree-Trellis Fast Search", tal como descrito por Soong e Huang [Soong90], Jiménez [Jiménez95] e Schwartz [Schwartz97]. Trata-se de um procedimento de busca otimizado, em uma árvore léxica, cujo mecanismo fundamenta-se no algoritmo A* [Paul91]. Estes algoritmos são exatos, uma vez que garantem a obtenção das *N* frases candidatas corretas. Adicionalmente, este tipo de abordagem proporciona algoritmos mais eficientes para realizar o reconhecimento de fala contínua [Fagundes98].

Entretanto, no caso de sistemas de reconhecimento que utilizam algoritmos de busca menos eficientes, propõe-se o emprego de uma estratégia sub-ótima, descrita por Lee [Lee89] e Rabiner [Rabiner93], que se adapta ao algoritmo Level-Building, amplamente empregado ao longo de toda esta tese. Apesar de não ser exato, este tipo de algoritmo tende a aproximar bem o resultado dos algoritmos ótimos, sofrendo, porém, deterioração em seu desempenho a medida que *N* aumenta.

A determinação exata das *N* candidatas é importante para permitir que o procedimento de treinamento discriminativo seja realizado em função das frases que realmente tendem a provocar erros de reconhecimento. Contudo, não é necessário um número muito grande de frases candidatas para realizar o treinamento discriminativo, de modo que é viável a utilização de um algoritmo que apresenta bom comportamento para valores pequenos de *N*. Este algoritmo é descrito abaixo:

1- Level Building: Executa-se o algoritmo Level-Building, armazenando ao final de cada nível, para cada instante de tempo *t*, as *V* maiores verossimilhanças, bem como os respectivos modelos e instantes de início ("back-pointers"). O número *V* de modelos a serem considerados depende no número *N* desejado de frases candidatas. Na tabela abaixo, tem-se alguns exemplos de valores de *V* que devem ser adotados para se obter *N* candidatas:

V	N
1	1
2	3
3	6
4	10

***2- Busca reversa*:** Realiza-se o procedimento de busca reversa representado na figura 4.1. O instante final *T* da elocução, no nível *L*, corresponde ao nó inicial NI da busca, enquanto o instante inicial, no nível 1, está associado ao nó final NF. A partir do nó NI, partem *V* caminhos, originando, por sua vez, *V* nós no nível *L-1*. A partir do nível *L-1*, o nó correspondente ao *m*-ésimo melhor caminho é expandido em (*V-m + 1*) novos caminhos (onde $1 \le m \le V$), de modo a originar um total de $V(V+1)/2$ nós a cada nível. Dentre estes nós, seleciona-se os *V* melhores, de acordo com a verossimilhança $P_{v,m}^{l}$ dos caminhos parciais passando em cada um dos nós. A verossimilhança $P_{v,m}^{l}$ do *v*-ésimo caminho que passa pelo *m*-ésimo nó, para o nível *l*, é dada por:

$$P_{v,m}^{l} = P_{v,m}^{b} + P_{m}^{f}$$

onde $P_{v,m}^{b}$ é a verossimilhança backward, associada ao *v*-ésimo caminho parcial que se inicia no nó NF (nível 1) e termina no nó *m*. Esta verossimilhança é armazenada durante o passo 1 do algoritmo. A verossimilhança forward P_{m}^{f} está associada ao caminho ótimo começando no nó *m* e terminando no nó NI (nível *L*), sendo dada por:

$$P_{m}^{f} = P_{v}^{L} - P_{v^*,m}^{l+1}$$

onde P_{v}^{L} é a verossimilhança do *v*-ésimo melhor caminho no nível *L* que originou o caminho *m* no nível *l*. $P_{v^*,m}^{l+1}$ é a verossimilhança backward associada nó do nível *l* pertencente ao melhor caminho proveniente do nó do nível *l*+1 que originou o nó *m*.

***3- Finalização*:** Caso não tenha chegado ao nível 1, repetir o passo 2. Caso contrário, os nós associados às *N* maiores verossimilhanças definem as *N* frases candidatas.

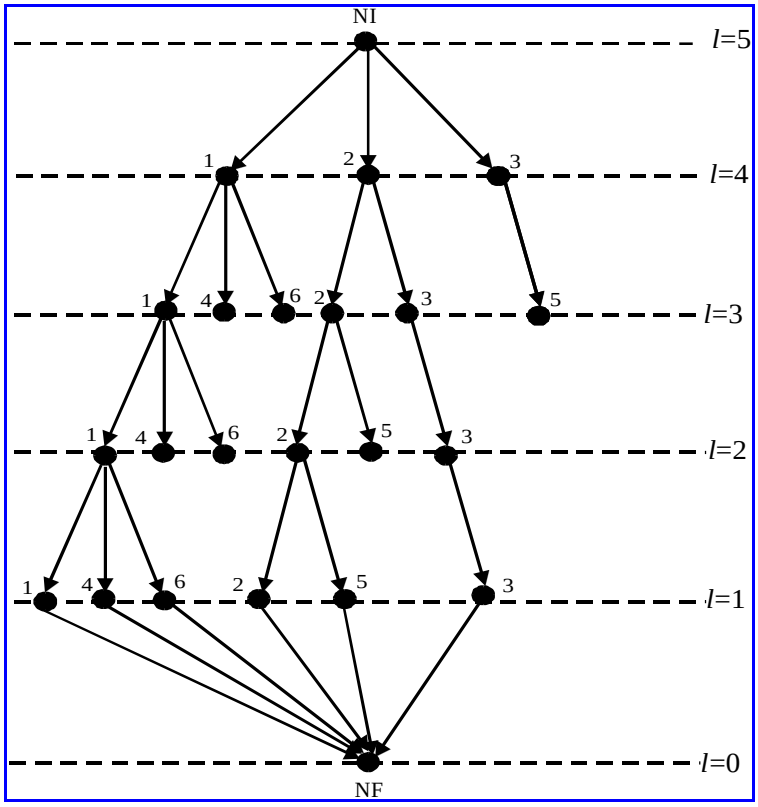

Figura 4.1- Diagrama do procedimento de busca reversa para 6 frases candidatas (N=6 e V=3) e 5 níveis (L=5).

Trata-se, portanto, de um algoritmo que preserva a estrutura dos sistemas que não realizam busca em árvore, simplificando a implementação do procedimento de treinamento discriminativo.

4.3.2.2. Definição da Função Discriminante

A função discriminante para o caso de reconhecimento de fala contínua é definida em relação a uma seqüência de sub-unidades $\{F_l\}_{l=1}^{N_{fon}}$ que compõem uma frase em avaliação, resultando:

$$g(X, F_l; \Lambda) = \max_{q_{F_l}} \log P(X, q_{F_l}, F_l \mid \Lambda) = \log P(X, \bar{q}_{F_l}, F_l \mid \Lambda)$$

onde q_{F_l} é uma seqüência de estados genérica correspondente a seqüência de sub-unidades F_l que compõe uma determinada frase, $\bar{q}_{F_l}$ é a seqüência de estados ótima associada a mesma seqüência F_l e $\Lambda = \{l_l\}_{l=1}^{M}$ é o conjunto dos M modelos de fones que compõem o sistema.

Utilizando a expressão para a verossimilhança obtida a partir do HMM, obtém-se a seguinte definição:

$$g(X, F_r; \Lambda) = \log p_{\bar{q}_0} + \sum_{t=1}^{T} \log\left(a_{\bar{q}_{t-1}\bar{q}_t}\right) + \sum_{t=1}^{T} \log\left[b_{\bar{q}_t} x(t)\right] \tag{4.34}$$

Este valor de verossimilhança corresponde exatamente ao obtido ao final do Level-Building para a elocução *X*.

Uma vez definida a função discriminante, pode-se estabelecer uma medida de erro de classificação $d(X;\Lambda)$:

$$d(X;\Lambda) = -g(X, F_0; \Lambda) + \log\left\{\frac{1}{N_{cand} - 1} \sum_{r=1}^{N_{cand}} \exp\left[g(X, F_r; \Lambda) \cdot h\right]\right\}^{\frac{1}{h}} \tag{4.35}$$

onde N_{cand} é o número de frases candidatas (concorrentes), F_0 é a seqüência correta de sub-unidades (conhecida a priori durante o treinamento) e F_r é a seqüência de sub-unidades correspondente a *r*-ésima frase candidata.

Esta medida possui exatamente as mesmas propriedades descritas anteriormente para a medida definida na equação (4.7). O termo associado às frases candidatas possui sinal oposto a função $g(X,F_0;\Lambda)$, garantindo a propriedade discriminante da medida $d(X;\Lambda)$. Além disto, novamente vale a relação:

$$\lim_{h \to \infty} \left\{\frac{1}{N_{cand} - 1} \sum_{r=1}^{N_{cand}} \exp\left[g(X, F_r; \Lambda) \cdot h\right]\right\}^{\frac{1}{h}} = \max_{1 \le r \le N_{cand}} g(X, F_r; \Lambda)$$

Para construir a função de custo, deve-se obter uma medida restrita preferencialmente ao intervalo (0,1). Utiliza-se, portanto, uma função do tipo sigmóide:

$$l(X; l) = l\left[d(X; l)\right] = \frac{1}{1 + e^{-g \cdot d(X;\Lambda)}} \tag{4.36}$$

4.3.2.3. Estimação dos Parâmetros dos HMM's

Novamente, será utilizado o algoritmo Segmental GPD com a definição de custo médio da seção 4.3.1.1, a fim de estimar os parâmetros $\left\{a_{ij}^{(i)}\right\}_{i,j=1}^{Q}$ e $\left\{b_{jk}^{(i)}\right\}_{k=1}^{N_{simb}}$, correspondentes ao modelo da *i*-ésima subunidade. Desta forma, o processo de otimização será realizado com relação a seguinte função de custo:

$$L(\Lambda) = E_X\left[l(X;\Lambda)\right] = \int l(X;\Lambda)p(X)dX$$

Aplicando o algoritmo Segmental GPD, tem-se:

$$\Lambda_{k+1} = \Lambda_k - e_k \cdot \nabla l(X;\Lambda_k) \tag{4.37}$$

Pode-se ainda utilizar a matriz definida positiva U_k, como discutido na seção 4.3.1.1, resultando no seguinte procedimento alternativo:

$$\Lambda_{k+1} = \Lambda_k - e_k \cdot U_k \cdot \nabla l(X;\Lambda_k)$$

Em seguida, serão descritos de forma detalhada os procedimentos para o ajuste de cada tipo de parâmetros. Vale salientar que, neste caso, serão aplicadas as operações de transformação de parâmetros descritas nas equações (4.18) e (4.19). Desta forma, os gradientes serão calculados em relação aos parâmetros irrestritos e, somente ao final de cada iteração, serão realizadas as normalizações dos parâmetros.

(1) Parâmetros $\left\{b_{jk}^{(i)}\right\}_{k=1}^{N_{simb}}$:

Neste caso, a equação de ajuste, definida a partir da expressão geral (4.37), é dada por:

$$\overline{b}_{jk}^{(i)}(n+1) = \overline{b}_{jk}^{(i)}(n) - e \cdot \frac{\partial l(X;\Lambda)}{\partial \overline{b}_{jk}^{(i)}}$$

A fim de simplificar as expressões dos gradientes, são definidas as seguintes funções auxiliares:

$$z_1(X) = \frac{\partial l(X;\Lambda)}{\partial d(X;\Lambda)} \tag{4.38}$$

$$z_2(X) = \frac{\partial d(X;\Lambda)}{\partial \bar{b}_{jk}^{(i)}} \tag{4.39}$$

$$z_3(X, F_r) = \frac{\partial g(X, F_r;\Lambda)}{\partial \bar{b}_{jk}^{(i)}} \tag{4.40}$$

O gradiente pode, então, ser obtido a partir da seguinte expressão:

$$\frac{\partial l(X;\Lambda)}{\partial \bar{b}_{jk}^{(i)}} = \frac{\partial l(X;\Lambda)}{\partial d(X;\Lambda)} \cdot \frac{\partial d(X;\Lambda)}{\partial \bar{b}_{jk}^{(i)}} = z_1(X) \cdot z_2(X) \tag{4.41}$$

A partir da equação (4.36), obtém-se:

$$z_1(X) = g \cdot l(X;\Lambda) \cdot [1 - l(X;\Lambda)] \tag{4.42}$$

Para encontrar o termo $z_2(X)$, deve-se utilizar a definição (4.35), bem como a função auxiliar $z_3(X, F_r)$, obtendo-se:

$$z_2(X) = -z_3(X, F_0) + \frac{\sum_{r=1}^{N_{cand}} \exp[g(X, F_r;\Lambda) \cdot h] \cdot z_3(X, F_r)}{\sum_{r=1}^{N_{cand}} \exp[g(X, F_r;\Lambda) \cdot h]} \tag{4.43}$$

onde $b_{jk}^{(i)}$ é a probabilidade de emissão do símbolo v_k, para o estado j do modelo da i-ésima subunidade.

As derivadas das funções discriminantes, correspondentes às funções auxiliares $z_3(X, F_r)$, são obtidas a partir da equação (4.34), sendo necessário definir as seqüências $\{t_r(v)\}_{v=1}^{N_{fr}}$ associadas aos delimitadores dos N_{fr} segmentos correspondentes aos fones da *r*-ésima frase candidata. Estes termos podem ser calculados como se segue:

$$z_3(X, F_r) = \frac{\partial}{\partial \bar{b}_{jk}^{(i)}} \left[\sum_{t=1}^{T} \log b_{jk}^{(i)} \right] = \sum_{v=1}^{N_{fr}} \sum_{t=t_r(v-1)+1}^{t_r(v)} d\left(\bar{q}_t^{(r)}, j\right) d\left(F_r(v), i\right) d\left(x_t, v_k\right) \frac{\partial \log b_{jk}^{(i)}}{\partial \bar{b}_{jk}^{(i)}}$$

onde $\bar{q}_t^{(r)}$ é o estado no instante *t* da seqüência ótima de estados associada a *r*-ésima frase candidata F_r.

A derivada $\frac{\partial \log b_{jk}^{(i)}}{\partial \bar{b}_{jk}^{(i)}}$ pode ser calculada de forma análoga à equação (4.33), resultando:

$$\frac{\partial \log b_{jk}^{(i)}}{\partial \bar{b}_{jk}^{(i)}} = 1 - b_{jk}^{(i)}$$

Desta forma, obtém-se as expressões para as funções $z_3(X, F_r)$:

$$z_3(X, F_r) = \sum_{v=1}^{N_{fr}} \sum_{t=t_s(v-1)+1}^{t_s(v)} d\left(\bar{q}_t^{(r)}, j\right) d\left(F_r(v), i\right) d\left(x_t, v_k\right) \left[1 - b_{jk}^{(i)}\right] \qquad (4.44)$$

Finalmente, deve-se realizar a transformação dos parâmetros, definida na expressão (4.19), resultando:

$$b_{jk}^{(i)}(n+1) = \frac{\exp\left[\bar{b}_{jk}^{(i)}(n+1)\right]}{\sum_{k=1}^{N_{simb}} \exp\left[\bar{b}_{jk}^{(i)}(n+1)\right]}, \qquad j = 1,\ldots,Q, \;\; k = 1,\ldots,N_{simb}$$

(2) Parâmetros $\left\{a_{ij}^{(l)}\right\}_{i,j=1}^{Q}$:

A equação de ajuste das probabilidades de transição de estados $a_{ij}^{(l)}$, associadas à subunidade *l*, é obtida também a partir da expressão geral definida na equação (4.37):

$$\bar{a}_{ij}^{(l)}(n+1) = \bar{a}_{ij}^{(l)}(n) - e \cdot \frac{\partial l(X;\Lambda)}{\partial \bar{a}_{ij}^{(l)}}$$

Neste caso, as expressões relativas ao gradiente são obtidas de forma análoga ao item anterior, sendo necessário apenas redefinir as funções auxiliares $z_2(X)$ e $z_3(X,F_r)$:

$$z_2(X) = \frac{\partial d(X;\Lambda)}{\partial \bar{a}_{ij}^{(l)}} \tag{4.45}$$

$$z_3(X,F_r) = \frac{\partial g(X,F_r;\Lambda)}{\partial \bar{a}_{ij}^{(l)}} \tag{4.46}$$

A função $z_1(X)$ permanece inalterada e os gradientes podem, então, ser encontrados a partir das equações (4.41), (4.42) e (4.43):

$$\frac{\partial l(X;\Lambda)}{\partial \bar{a}_{ij}^{(l)}} = z_1(X) \cdot z_2(X)$$

$$z_1(X) = g \cdot l(X;\Lambda) \cdot [1 - l(X;\Lambda)]$$

$$z_2(X) = -z_3(X,F_0) + \frac{\sum_{r=1}^{N_{cand}} \exp[g(X,F_r;\Lambda) \cdot h] \cdot z_3(X,F_r)}{\sum_{r=1}^{N_{cand}} \exp[g(X,F_r;\Lambda) \cdot h]}$$

Neste caso, porém, as funções $z_3(X,F_r)$ devem ser redefinidas. As derivadas das funções discriminantes são obtidas a partir da equação (4.34). Novamente, é necessário encontrar a

segmentação $\{t_r(v)\}_{v=1}^{N_{fr}}$ associada ao caminho de Viterbi para a *r*-ésima frase candidata. Deste modo, tem-se:

$$z_3(X, F_r) = \sum_{v=1}^{N_{fr}} \sum_{t=t_r(v-1)+1}^{t_r(v)} d\left(\bar{q}_{t-1}^{(r)}, i\right) d\left(\bar{q}_t^{(r)}, j\right) d\left(F_r(v), i\right) \frac{\partial \log a_{ij}^{(l)}}{\partial \bar{a}_{ij}^{(l)}}$$

O fator $\dfrac{\partial \log a_{ij}^{(l)}}{\partial \bar{a}_{ij}^{(l)}}$ pode ser calculado como se segue:

$$\frac{\partial \log a_{ij}^{(l)}}{\partial \bar{a}_{ij}^{(l)}} = 1 - a_{ij}^{(l)}$$

Assim, obtém-se:

$$\frac{\partial g(X, F_r; \Lambda)}{\partial \bar{a}_{ij}^{(l)}} = \sum_{v=1}^{N_{fr}} \sum_{t=t_r(v-1)+1}^{t_r(v)} d\left(\bar{q}_{t-1}^{(r)}, i\right) d\left(\bar{q}_t^{(r)}, j\right) d\left(F_r(v), i\right) \left[1 - a_{ij}^{(l)}\right] \tag{4.47}$$

Por fim, realiza-se a transformação descrita em (4.18):

$$a_{ij}^{(l)}(n+1) = \frac{\exp\left[\bar{a}_{ij}^{(l)}(n+1)\right]}{\sum_{j=1}^{Q} \exp\left[\bar{a}_{ij}^{(l)}(n+1)\right]}, \qquad j = 1, \ldots, Q$$

4.3.2.4. Síntese do Algoritmo de Treinamento Discriminativo para Fala Contínua

O algoritmo de Treinamento Discriminativo para o caso dos HMM Discretos, em fala contínua, pode ser resumido como se segue:

1- ***Inicialização*:** Os parâmetros dos HMM são inicializados a partir de outro critério de treinamento (ML, MMI ou MDI). Normalmente, são utilizados os modelos gerados através do algoritmo de Baum-Welch (critério ML);
2- ***Obtenção das N Frases Candidatas:*** Através de um algoritmo de busca em árvore ou de um algoritmo sub-ótimo baseado no Level-Building, são encontradas as N frases mais prováveis fornecidas pelo sistema de reconhecimento;
3- ***Segmentação do Conjunto de Treinamento*:** Utilizando o algoritmo de Viterbi, encontram-se as segmentações de cada elocução de treinamento, a fim de compor a função de custo do algoritmo Segmental GPD;
4- ***Cálculo dos Gradientes das Funções de Custo*:** A partir dos parâmetros atuais, calculam-se os gradientes das funções de custo associadas a cada parâmetro.
5- ***Ajuste dos Parâmetros*:** Uma vez calculados os gradientes, aplicam-se as equações de ajuste provenientes da equação (4.10) a fim de obter os novos parâmetros;
6- ***Transformação dos Parâmetros*:** Após estimados os parâmetros, devem ser realizadas as transformações de parâmetros descritas nas equações (4.16) à (4.21);
7- ***Parada*:** Caso o critério de parada não seja satisfeito, voltar ao passo 2. Normalmente, o critério de parada adotado está associado a um número máximo de épocas de treinamento.

Em seguida, serão descritos os algoritmos de treinamento discriminativo que incluem os parâmetros associados ao Fator de Ponderação Temporal.

4.4. Estimação dos Parâmetros Empíricos do Fator de Ponderação Temporal

No Capítulo 3, descreveu-se uma abordagem alternativa para atenuar os problemas decorrentes do modelamento temporal inconsistente dos HMM's. Para tanto, empregou-se a informação de segmentação $s(t)$, que corresponde a tendência de transição entre os fonemas de uma elocução X.

Foi verificada a necessidade da utilização dos fatores de atenuação empíricos g_g e g_w, estimados a partir de um procedimento manual de verificação dos erros de reconhecimento do sistema. Nos casos em que o vocabulário é reduzido, este procedimento tende a ser suficiente para determinar um conjunto de parâmetros que proporcione melhoria no desempenho do sistema. Entretanto, é possível estabelecer um procedimento automático para a estimação destes parâmetros utilizando as técnicas de treinamento discriminativo descritas nas seções anteriores deste capítulo. Com este procedimento, torna-se possível a utilização dos fatores de ponderação temporal também para médios e grandes vocabulários.

Inicialmente, será descrito um procedimento de estimação próprio para vocabulários médios, onde ainda é viável a utilização do fator empírico g_w, que está associado à palavra w do vocabulário. Em seguida, este procedimento será redefinido para ser aplicado a grandes vocabulários. Neste caso, os fatores g_w passarão a ser associados às subunidades (fones) e serão denominados g_f.

4.4.1. Vocabulários Médios

O procedimento para a estimação dos parâmetros g_g e g_w é obtido como uma modificação do algoritmo de treinamento descrito para o caso de Fala Contínua, com HMM's Discretos e Modelos de Subunidades (Caso B). Neste caso os fatores g_w estão associados às palavras do vocabulário.

Inicialmente, a função discriminante $g(X,F_r;L)$ deve ser redefinida, a fim de introduzir a influência dos fatores de ponderação temporal no valor da verossimilhança obtida ao final do algoritmo Level-Building.

Utilizando a expressão descrita no capítulo anterior, tem-se:

$$g(X, F_r; \Lambda) = \log\left\{ P\left(X, \bar{q}_{F_r}, F_r; \Lambda\right) \cdot \left[\prod_{l=1}^{L} \left(\prod_{j=1}^{N_{fl}} s\left(t_{lj}\right) \right)^{\frac{g_{wl}}{N_{fl}}} \right]^{g_g} \right\} \tag{4.48}$$

onde F_r é a *r*-ésima frase candidata, $\bar{q}_{F_r}$ é a seqüência de estados ótima associada à frase F_r, $\{t_{lj}\}$ são os instantes associados às fronteiras do *j*-ésimo segmento correspondente ao caminho ótimo no *l*-ésimo nível, *L* é o número de níveis e N_{fl} é o número de fones da *l*-ésima palavra da frase F_r.

Esta função discriminante é bastante conveniente para o sistema de reconhecimento que emprega o Fator de Ponderação Temporal, uma vez que corresponde exatamente à verossimilhança encontrada ao final do Level-Building, que, neste caso, se baseia em caminhos de Viterbi ponderados pela informação de segmentação *s*(*t*).

É importante observar que os parâmetros g_{wl} aparecem sempre divididos pelo número de fones da palavra no *l*-ésimo nível (N_{fl}). Trata-se de um artifício para reduzir o efeito de ponderação excessiva de palavras com maior número de fones. Desenvolvendo o logaritmo, tem-se:

$$g(X, F_r; \Lambda) = \log P\left(X, \bar{q}_{F_r}, F_r; \Lambda\right) + g_g \cdot \sum_{l=1}^{L} \frac{g_{wl}}{N_{fl}} \cdot \sum_{j=1}^{N_{fl}} \log s\left(t_{lj}\right)$$

Definindo:

$$g_1(X, F_r; \Lambda) = \log P\left(X, \bar{q}_{F_r}, F_r; \Lambda\right) = \log p_{\bar{q}_0} + \sum_{t=1}^{T} \log\left(a_{\bar{q}_{t-1}\bar{q}_t}\right) + \sum_{t=1}^{T} \log\left[b_{\bar{q}_t} x(t)\right] \tag{4.49}$$

$$g_2(X, F_r; \Lambda) = g_g \cdot \sum_{l=1}^{L} \frac{g_{wl}}{N_{fl}} \cdot \sum_{j=1}^{N_{fl}} \log s\left(t_{lj}\right) \tag{4.50}$$

Resulta, então, a seguinte função discriminante:

$$g(X, F_r; \Lambda) = g_1(X, F_r; \Lambda) + g_2(X, F_r; \Lambda) \tag{4.51}$$

A função $g_1(X, F_r; \Lambda)$ corresponde exatamente à definição na equação (4.34) e depende apenas dos parâmetros que compõem os modelos HMM. Por outro lado, função $g_2(X, F_r; \Lambda)$ depende apenas dos parâmetros g_g e g_w, bem como da função de segmentação $s(t)$.

Uma vez definida a função discriminante $g(X, F_r; \Lambda)$, pode-se obter as equações de ajuste para os parâmetros g_g e g_w. Vale salientar que, a princípio, a única restrição imposta a estes parâmetros é que sejam não negativos. Entretanto, neste trabalho, estes parâmetros serão restritos ao intervalo (0,1). Para tanto, utiliza-se uma função sigmoidal, resultando:

$$g_g = \frac{1}{1+\exp(-\bar{g}_g)} \tag{4.52}$$

$$g_w = \frac{1}{1+\exp(-\bar{g}_w)} \tag{4.53}$$

(1) Parâmetro $\{g_g\}$

Para ajustar o parâmetro g_g, toma-se como ponto de partida a expressão geral mostrada na equação (4.37). A equação de ajuste, obtida em relação aos parâmetros irrestritos, é dada por:

$$\bar{g}_g(n+1) = \bar{g}_g(n) - e \cdot \frac{\partial l(X;\Lambda)}{\partial \bar{g}_g}$$

Para encontrar as expressões para o gradiente serão empregadas as funções auxiliares $z_1(X)$, $z_2(X)$ e $z_3(X, F_r)$, definidas, neste caso, como se segue:

$$z_1(X) = \frac{\partial l(X;\Lambda)}{\partial d(X;\Lambda)} \tag{4.54}$$

$$z_2(X) = \frac{\partial d(X;\Lambda)}{\partial \bar{g}_g} \tag{4.55}$$

$$z_3(X, F_r) = \frac{\partial g(X, F_r;\Lambda)}{\partial \bar{g}_g} \tag{4.56}$$

Desta forma, as equações (4.41), (4.42) e (4.43) também se aplicam neste caso, resultando:

$$\frac{\partial l(X;\Lambda)}{\partial \bar{g}_g} = z_1(X)\cdot z_2(X)$$

$$z_1(X) = g\cdot l(X;\Lambda)\cdot[1-l(X;\Lambda)]$$

$$z_2(X) = -z_3(X,F_0) + \frac{\sum_{r=1}^{N_{cand}} \exp[g(X,F_r;\Lambda)\cdot h]\cdot z_3(X,F_r)}{\sum_{r=1}^{N_{cand}} \exp[g(X,F_r;\Lambda)\cdot h]}$$

As derivadas das funções $g(X,F_r;\Lambda)$, correspondentes às funções $z_3(X,F_r)$, são obtidas utilizando a equação (4.51):

$$z_3(X,F_r) = \frac{\partial g_2(X,F_r;\Lambda)}{\partial g_g}\cdot\frac{\partial g_g}{\partial \bar{g}_g} = g_g(1-g_g)\cdot\sum_{l=1}^{L}\frac{g_{wl}}{N_{fl}}\cdot\sum_{j=1}^{N_{fl}}\log s(t_{lj}) \qquad (4.57)$$

Por fim, aplica-se a transformação definida na equação (4.52), obtendo-se:

$$g_g(n+1) = \frac{1}{1+\exp[-\bar{g}_g(n+1)]}$$

(2) Parâmetros {g_w}

O procedimento para efetuar o ajuste dos parâmetros g_w é muito semelhante ao descrito para os parâmetros γ_g. Partindo novamente da equação geral (4.33), a seguinte equação de ajuste é obtida:

$$\bar{g}_w(n+1) = \bar{g}_w(n) - e\cdot\frac{\partial l(X;\Lambda)}{\partial \bar{g}_w}$$

Novamente, o gradiente será obtido a partir das funções auxiliares $z_1(X)$, $z_2(X)$ e $z_3(X,F_r)$, sendo necessário redefinir $z_2(X)$ e $z_3(X,F_r)$ como se segue:

$$z_2(X) = \frac{\partial d(X;\Lambda)}{\partial \bar{g}_w} \tag{4.58}$$

$$z_3(X, F_r) = \frac{\partial g(X, F_r;\Lambda)}{\partial \bar{g}_w} \tag{4.59}$$

Deste modo, as expressões para o cálculo dos gradientes são dadas por:

$$\frac{\partial l(X;\Lambda)}{\partial \bar{g}_w} = z_1(X) \cdot z_2(X)$$

$$z_1(X) = g \cdot l(X;\Lambda) \cdot [1 - l(X;\Lambda)]$$

$$z_2(X) = -z_3(X, F_0) + \frac{\sum_{r=1}^{N_{cand}} \exp[g(X, F_r;\Lambda) \cdot h] \cdot z_3(X, F_r)}{\sum_{r=1}^{N_{cand}} \exp[g(X, F_r;\Lambda) \cdot h]}$$

As funções $z_3(X, F_r)$, correspondentes às derivadas das funções discriminantes, são calculadas segundo a seguinte equação:

$$z_3(X, F_r) = \frac{\partial g_2(X, F_r;\Lambda)}{\partial \bar{g}_w} \cdot \frac{\partial g_w}{\partial \bar{g}_w} = g_g \cdot g_w (1 - g_w) \cdot \sum_{l=1}^{L} \frac{d(W_r(l), w)}{N_{fl}} \cdot \sum_{j=1}^{N_{fl}} \log s(t_{lj}) \tag{4.60}$$

Finalmente, é aplicada a transformação definida na equação (4.53), obtendo-se:

$$g_w(n+1) = \frac{1}{1 + \exp[-\bar{g}_w(n+1)]}$$

4.4.2. Vocabulários Extensos

No caso de vocabulários extensos, é necessário associar os parâmetros empíricos do Fator de Ponderação Temporal (cap. 1) às subunidades lingüísticas (fones), a fim de reduzir o número de parâmetros a serem estimados. Neste caso, os fatores g_w passam a ser associados aos fones, sendo denominados g_f .

Para tanto, a função discriminante passa a ser definida como:

$$g(X, F_r; \Lambda) = \log\left\{ P(X, \bar{q}_{F_r}, F_r; \Lambda) \cdot \left[\prod_{l=1}^{L} \left(\prod_{j=1}^{N_{fl}} s(t_{lj})^{g_{fj}} \right)^{\frac{1}{N_{fl}}} \right]^{g_g} \right\} \tag{4.61}$$

onde F_r é a r-ésima frase candidata, $\bar{q}_{F_r}$ é a seqüência de estados ótima associada à frase F_r, $\{t_l\}_{l=1}^{N_{fl}}$ é a seqüência formada pelas marcas de segmentação em fones da elocução *X* e N_{fr} é o número de fones da *r*-ésima frase candidata.

Vale salientar que, neste caso, também se realiza um procedimento de normalização a partir do expoente $1/N_{fl}$, a fim de evitar que palavras com maior número de fones sejam excessivamente penalizadas, mesmo nos casos de alinhamento coerente com a informação de segmentação *s*(*t*).

Desenvolvendo a equação acima, obtém-se:

$$g(X, F_r; \Lambda) = \log P(X, \bar{q}_{F_r}, F_r; \Lambda) + g_g \cdot \sum_{l=1}^{L} \frac{1}{N_{fl}} \cdot \sum_{j=1}^{N_{fl}} g_{fl} \log s(t_{lj})$$

O primeiro termo da soma, denominado $g_1(X, F_r; \Lambda)$, foi definido na equação (4.49) e permite a estimação dos parâmetros do modelo HMM. O segundo termo será denominado $g_2(X, F_r; \Lambda)$, sendo utilizado para obter os parâmetros relacionados ao Fator de Ponderação Temporal. Assim:

$$g_2(X, F_r; \Lambda) = g_g \cdot \sum_{l=1}^{L} \frac{1}{N_{fl}} \cdot \sum_{j=1}^{N_{fl}} g_{fl} \log s(t_{lj})$$

Resulta, então, a seguinte relação:

$$g(X, F_r; \Lambda) = g_1(X, F_r; \Lambda) + g_2(X, F_r; \Lambda) \tag{4.62}$$

Os parâmetros g_g e g_f podem ser estimados utilizando procedimento análogo ao empregado no caso de vocabulários médios. Novamente, os parâmetros serão restritos ao intervalo (0,1), resultando na utilização da função sigmoidal para realizar a transformação de parâmetros.

$$g_g = \frac{1}{1 + \exp(-\bar{g}_g)} \tag{4.63}$$

$$g_f = \frac{1}{1 + \exp(-\bar{g}_f)} \tag{4.64}$$

(1) Parâmetro {g_g}

O parâmetro g_g é estimado a partir da equação (4.37). Novamente, as equações de ajuste serão definidas em relação aos parâmetros irrestritos $\bar{g}_g$:

$$\bar{g}_g(n+1) = \bar{g}_g(n) - e \cdot \frac{\partial l(X; \Lambda)}{\partial \bar{g}_g}$$

As expressões para o gradiente são encontradas através da funções auxiliares $z_1(X)$, $z_2(X)$ e $z_3(X, F_r)$, redefinidas como se segue:

$$z_1(X) = \frac{\partial l(X; \Lambda)}{\partial d(X; \Lambda)} \tag{4.65}$$

$$z_2(X) = \frac{\partial d(X;\Lambda)}{\partial \bar{g}_g} \tag{4.66}$$

$$z_3(X, F_r) = \frac{\partial g(X, F_r;\Lambda)}{\partial \bar{g}_g} \tag{4.67}$$

Desta forma, as equações (4.41), (4.42) e (4.43) também se aplicam neste caso, resultando:

$$\frac{\partial l(X;\Lambda)}{\partial \bar{g}_g} = z_1(X) \cdot z_2(X)$$

$$z_1(X) = g \cdot l(X;\Lambda) \cdot [1 - l(X;\Lambda)]$$

$$z_2(X) = -z_3(X, F_0) + \frac{\sum_{r=1}^{N_{cand}} \exp[g(X, F_r;\Lambda) \cdot h] \cdot z_3(X, F_r)}{\sum_{r=1}^{N_{cand}} \exp[g(X, F_r;\Lambda) \cdot h]}$$

As derivadas das funções discriminantes, que correspondem às funções $z_3(X, F_r)$, são calculadas a partir da equação (4.62):

$$z_3(X, F_r) = \frac{\partial g_2(X, F_r;\Lambda)}{\partial g_g} \cdot \frac{\partial g_g}{\partial \bar{g}_g} = g_g(1 - g_g) \cdot \sum_{l=1}^{L} \frac{1}{N_{fl}} \cdot \sum_{j=1}^{N_{fl}} g_{fl} \log s(t_{lj}) \tag{4.68}$$

Por fim, aplica-se a transformação definida na equação (4.63), obtendo-se:

$$g_g(n+1) = \frac{1}{1 + \exp[-\bar{g}_g(n+1)]}$$

(3) Parâmetros {g_f}

Os parâmetros γ_f podem ser estimados de forma análoga ao parâmetro g_g, resultando:

$$\bar{g}_f(n+1) = \bar{g}_f(n) - e \cdot \frac{\partial l(X;\Lambda)}{\partial \bar{g}_f}$$

O gradiente será calculado utilizando as funções auxiliares $z_1(X)$, $z_2(X)$ e $z_3(X,F_r)$, tornando-se necessário, neste caso, redefinir $z_2(X)$ e $z_3(X,F_r)$ como se segue:

$$z_2(X) = \frac{\partial d(X;\Lambda)}{\partial \bar{g}_f} \tag{4.69}$$

$$z_3(X,F_r) = \frac{\partial g(X,F_r;\Lambda)}{\partial \bar{g}_f} \tag{4.70}$$

Obtém-se, então, as expressões definidas nas equações (4.41), (4.42) e (4.43):

$$\frac{\partial l(X;\Lambda)}{\partial \bar{g}_f} = z_1(X)\cdot z_2(X)$$

$$z_1(X) = g\cdot l(X;\Lambda)\cdot[1-l(X;\Lambda)]$$

$$z_2(X) = -z_3(X,F_0) + \frac{\sum_{r=1}^{N_{cand}} \exp[g(X,F_r;\Lambda)\cdot h]\cdot z_3(X,F_r)}{\sum_{r=1}^{N_{cand}} \exp[g(X,F_r;\Lambda)\cdot h]}$$

As funções $z_3(X,F_r)$ são obtidas de acordo com a seguinte equação:

$$\frac{\partial g(X,F_r;\Lambda)}{\partial \bar{g}_f} = \frac{\partial g_2(X,F_r;\Lambda)}{\partial g_f}\cdot\frac{\partial g_f}{\partial \bar{g}_f} = g_g\cdot g_f\left(1-g_f\right)\cdot\sum_{l=1}^{L}\frac{1}{N_{fl}}\cdot\sum_{j=1}^{N_{fl}} d\left(F_r(l),f\right)\log s\left(t_{lj}\right) \tag{4.71}$$

Aplica-se, em seguida, a transformação definida na equação (4.64), resultando na seguinte expressão:

$$g_f(n+1) = \frac{1}{1+\exp[-\bar{g}_f(n+1)]}$$

4.4.3. Síntese do Algoritmo de Treinamento Discriminativo para Fala Contínua utilizando Fator de Ponderação Temporal

O algoritmo de Treinamento Discriminativo para o caso dos HMM Discretos, em fala contínua e empregando o Fator de Ponderação Temporal, pode ser resumido como se segue:

1- ***Inicialização*:** Os parâmetros dos HMM são inicializados a partir de outro critério de treinamento (ML, MMI ou MDI). Normalmente, são empregados os modelos gerados através do algoritmo de Baum-Welch (critério ML). Os parâmetros g_g e g_w podem ser inicializados uniformemente ou a partir dos resultados do procedimento de estimação manual descrito no capítulo 3;

2- ***Obtenção das N Frases Candidatas:*** Através de um algoritmo de busca em árvore ou de um algoritmo sub-ótimo baseado no Level-Building, encontra-se as N frases mais prováveis fornecidas pelo sistema de reconhecimento;

3- ***Segmentação do Conjunto de Treinamento*:** Utilizando o algoritmo de Viterbi, encontram-se as segmentações de cada elocução de treinamento, a fim de compor a função de custo do algoritmo Segmental GPD;

4- ***Cálculo dos Gradientes das Funções de Custo*:** A partir dos parâmetros atuais, calcula-se os gradientes das funções de custo associadas a cada parâmetro. Neste caso, devem ser incluídos os parâmetros empíricos g_g e g_w ou g_f associados ao Fator de Ponderação Temporal;

5- ***Ajuste dos Parâmetros*:** Uma vez calculados os gradientes, aplica-se as equações de ajuste provenientes da equação (4.10) a fim de obter os novos parâmetros;

6- ***Transformação dos Parâmetros*:** Neste passo devem ser realizadas as transformações sobre os parâmetros dos HMM's, descritas nas equações (4.16) à (4.21). Adicionalmente, devem ser efetuadas as transformações descritas nas equações (4.48) e (4.49);

7- ***Parada*:** Caso o critério de parada não seja satisfeito, voltar ao passo 2. Normalmente, o critério de parada adotado está associado a um número máximo de épocas de treinamento.

---

5. Análise dos Resultados

5.1. Considerações Iniciais

Neste capítulo são descritos os resultados obtidos para as diversas configurações do sistema de Reconhecimento de Fala, incluindo:

- *Caracterização do Sistema Básico*: neste caso, várias configurações são implementadas a fim de selecionar a que proporciona melhor desempenho, que será, então, utilizada nos demais experimentos envolvendo segmentação automática. As configurações avaliadas englobaram uma variedade de combinações entre tipos de parâmetros de entrada, métodos de inicialização do algoritmo de treinamento, inclusão de modelo de duração de palavras, etc.;
- *Segmentação Utilizando Filtragem Paramétrica*: são realizados os experimentos envolvendo a introdução da informação de segmentação gerada através do método da Filtragem Paramétrica. Neste caso, diferentes combinações dos parâmetros que caracterizam o filtro paramétrico são utilizadas para verificar a respectiva influência no desempenho do sistema;

- *Segmentação Utilizando MLP*: neste caso, são obtidos os resultados relativos à inclusão, no sistema de reconhecimento, da informação de segmentação gerada por uma MLP. São realizadas várias combinações envolvendo a arquitetura da rede neural, o tipo de alvo de treinamento, etc.;
- *Aspectos Práticos do Algoritmo de Treinamento Discriminativo*: neste item são discutidos alguns aspectos práticos relacionados com o algoritmo de Treinamento Discriminativo, os quais forma obtidos a partir dos experimentos descritos na literatura.

Vale salientar que o desempenho do sistema é avaliado a partir das taxas de acerto de frase e palavras, bem como por uma medida de distorção que reflete a precisão da segmentação automática obtida a partir dos métodos propostos.

5.2. Caracterização do Sistema Básico

O Sistema de Reconhecimento de Fala Contínua do LPDF (Laboratório de Processamento Digital de Fala – DECOM – UNICAMP), foi utilizado como base para a execução das simulações que compõem este trabalho. Foram realizados vários testes a fim de obter dados a respeito do comportamento do sistema em diferentes situações, de modo a permitir a avaliação da influência das técnicas implementadas sobre o seu desempenho.

O sistema empregado fundamenta-se no emprego de HMM's Discretos e modelos de sub-unidades. Os modelos adotados para as sub-unidades são mostrados na figura (5.1):

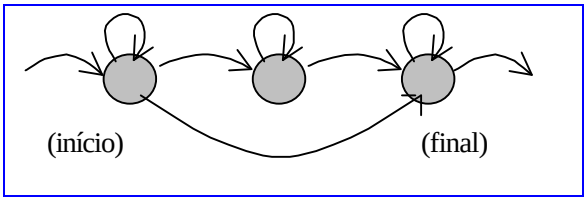

Figura (5.1)- Modelo de fone adotado

Os modelos das palavras são então construídos concatenando-se os modelos dos fones que constituem cada palavra. No Apêndice D, tem-se a lista dos fones empregados no sistema.

As técnicas de segmentação propostas neste trabalho implicam na necessidade de obter as fronteiras dos segmentos associados aos fones da elocução. Estas marcas de segmentação são encontradas ao longo do algoritmo de busca, desde que exista correspondência entre os estados dos modelos adotados e os fones da língua. Portanto, apesar de se tratar de um problema de Reconhecimento de Palavras Conectadas, decidiu-se pela abordagem mais genérica empregada em Reconhecimento de Fala Contínua, que se baseia nos modelos de sub-unidades. Vale salientar que o emprego das técnicas propostas neste trabalho implica na utilização obrigatória dos fones como sub-unidades. Esta limitação pode ser suavizada case se utilize uma técnica de segmentação automática do tipo Lingüisticamente Restrita, que se baseie nas sub-unidades desejadas (sílabas, por exemplo). Contudo, esta técnica deve apresentar propriedades que a diferencie das técnicas de segmentação associadas a algoritmos de busca (por exemplo, o Level-Bulding).

Adotou-se como vocabulário os dígitos em português e a aplicação selecionada consistiu no reconhecimento de dígitos conectados, independentemente do locutor. A motivação para a escolha desta estratégia é que sistemas assim estruturados têm sido utilizados em diversos trabalhos envolvendo novas técnicas que atuam sobre o modelamento acústico representado pelos HMM's [Rabiner89][Juang97][Bush87]. De fato, uma melhor avaliação do desempenho do modelo acústico construído é obtida no caso dos dígitos, uma vez que não existe, neste caso, influência das restrições impostas pela gramática da língua.

A base de dados empregada para o treinamento e teste do sistema foi composta por 440 elocuções pronunciadas por 23 locutores masculinos e 17 femininos. As elocuções são frases constituídas por 8 dígitos pronunciados sem pausa (conectados). Para estimar os parâmetros dos HMM's, utilizou-se um conjunto de treinamento formado por 341 elocuções pronunciadas por 18 locutores masculinos e 13 femininos. Os testes se realizaram sobre um conjunto de teste composto por 99 elocuções pronunciadas por 5 locutores masculinos e 4 femininos. No Apêndice B tem-se a lista das frases empregadas no sistema.

Os parâmetros de entrada empregados no sistema são calculados a partir de quadros de 10 ms obtidos por meio de janelas de Hamming de 20 ms com superposição de 50%. A freqüência de amostragem F_s adotada foi de 11.025 Hz. Foram utilizados os parâmetros Mel-Cepstrais (12 coeficientes), Delta-Mel Cepstrais (12 coeficientes, janela de análise de comprimento igual a 2), Delta-Delta Mel-Cepstrais (12 coeficientes, janela de análise de comprimento igual a 2), Log-Energia Normalizada, Delta Log-Energia Normalizada (janela de análise de comprimento igual a 2) e Delta-

Delta Log-Energia Normalizada (janela de análise de comprimento igual a 2). Utilizou-se, ainda, um procedimento de Subtração Espectral sobre os coeficientes Mel e Energia, bem como pré-ênfase, com Cp = 0,95.

O Quantizador Vetorial empregado foi implementado com base no algoritmo LBG, sendo adotados "codebooks" de tamanho igual a 256, obtidos para cada um dos tipos de parâmetros de entrada. Para a construção dos "codebooks", foi utilizada toda a base de dados disponível (440 elocuções).

Os HMM's foram treinados utilizando o algoritmo Baum-Welch, com inicializações do tipo Uniforme, K-Means ou a partir de modelos pré-treinados. Verificou-se que, para esta aplicação, a inicialização do tipo Uniforme proporcionou melhores resultados que a do tipo K-Means. O critério de parada consistiu na comparação com um limiar pré estabelecido e em uma medida de distorção que quantifica o decréscimo da verossimilhança média do conjunto de treinamento, de uma época em relação à época anterior.

O algoritmo de decodificação acústica utilizado foi o Level-Building, com o número de níveis fixo e igual a 10, pois foram empregadas para treinamento e teste, seqüências de 8 dígitos, com trechos de silêncio no início e no final de cada elocução, sem gramática ou modelo de Duração de Palavras.

Os fones empregados neste trabalho são mostrados no Apêndice D e correspondem estritamente aos fones da língua portuguesa necessários para formar os dígitos que compuseram a aplicação proposta.

Na tabela (5.1), tem-se os melhores resultados obtidos. A taxa de erro de palavras é obtida segundo a expressão abaixo [Pessoa99]:

$$\%ErroPalavras = \frac{S + D + I}{N} * 100$$

onde N é o número total de palavras nas frases de teste e $S+D+I$ corresponde ao número total de erros de substituição (S), exclusão (D) e inserção (I) de palavras.

Verifica-se que o melhor desempenho geral para o Sistema Básico foi obtido com os conjuntos (Mel + dMel) e (Mel + dMel + ddMel).

Parâmetros	Acerto Palavras (%)	Acerto Frases (%)	Erros de Inclusão (%)	Erros de Exclusão (%)
Mel	95,00	66,70	0,25	0,88
Mel+dMel	97,70	81,80	0,24	0,51
Mel+dMel+ddMel	97,50	79,80	0,38	0,76
Mel + dMel + dEnergia	97,70	81,80	0,25	0,51

Tabela (5.1)- Resultados para o Sistema Básico sem Modelo de Duração de Palavras e com Inicialização Uniforme

Foram realizados também experimentos com o intuito de avaliar a segmentação gerada pelos HMM's treinados com diferentes combinações de parâmetros de entrada. Esta avaliação é realizada por meio de uma medida de distorção definida a partir da comparação entre a segmentação automática e a segmentação manual (assumida como correta) da elocução. Obtém-se, então, a distorção para um determinado conjunto de elocuções e finalmente encontra-se a distorção média que é efetivamente adotada para representar a qualidade da segmentação. O algoritmo empregado para calcular a distorção é descrito abaixo:

1- Encontrar os limites $limi[k]$ e $lims[k]$ de janelas de análise em torno da k-ésima marca de segmentação manual:

   1.1- Se $Segm[k] + M < Segm[k+1] - M$, tem-se que $lims[k] = Segm[k] + M$;

   1.2- Se $Segm[k]+M \geq Segm[k+1]-M$, tem-se que $lims[k] = (Segm[k] + Segm[k+1])/2$;

   1.3- Se $Segm[k] - M > Segm[k-1]+M$, tem-se que $limi[k] = Segm[k] - M$;

   1.4- Se $Segm[k] - M \leq Segm[k-1]+M$, tem-se que $limi[k] = (Segm[k] + Segm[k-1])/2$;

2- Verificar o número de marcas de segmentação automática em cada uma das janelas e contabilizar os erros como se segue:

   2.1- *Erros de Exclusão*: ocorrem quando uma janela de análise não contém qualquer marca de segmentação automática;

   2.2- *Erros de Inserção*: ocorrem quando existe mais de uma marca de segmentação automática contida em uma janela de análise ou quando existem marcas de segmentação automática nos intervalos entre janelas adjacentes.

   2.3- *Erro Total*: é obtido somando-se os Erros de Exclusão e os Erros de Inserção.

onde:

Sega[*n*] = seqüência das marcas de segmentação automática;

Na = número de elementos da seqüência *Sega*[*n*];

Segm[*n*] = seqüência das marcas de segmentação manual;

Nm = número de elementos da seqüência *Segm*[*n*] ;

*2*M + 1* = comprimento máximo da janela de análise;

Com base neste algoritmo, tem-se as seguintes definições:

$$Dist.Segmentação = \frac{ND + NI}{NT}$$

$$Dist.Deleção = \frac{ND}{NT}$$

$$Dist.Inserção = \frac{NI}{NT}$$

onde *ND* é o número de erros de exclusão, *NI* é o número de erros de inserção e *NT* é o número total de marcas de segmentação manual da elocução.

De acordo com este critério, foram obtidos os resultados mostrados na tabela (5.2)

Parâmetros	Dist. Exclusão	Dist. Inserção	Dist. Segmentação
Mel	0,217	0,428	0,645
Mel+dMel	0,194	0,442	0,635
Mel+dMel+ddMel	0,206	0,441	0,647
Mel + dMel + dEnergia	0,248	0,492	0,740

Tabela (5.2)- Resultados relativos à precisão da segmentação automática empregada no Sistema Básico

Como podemos observar, a configuração ótima para o sistema base consistiu na utilização dos parâmetros Mel-Cepstrais e Delta-Mel Cepstrais como parâmetros de entrada, uma vez que este conjunto apresentou uma segmentação mais precisa.

Foram realizados alguns testes com Modelos de Palavras, obtendo-se desempenho superior (em temos de taxa de acerto de frases) ao do Sistema Básico. Isto pode ser verificado comparando-se a tabela referente ao desempenho de sistema baseado em sub-unidades (tabela (5.1)) com os resultados obtidos para o sistema baseado em modelos de palavras (vide tabela (5.3)). Entretanto, optou-se pelos modelos de sub-unidades, pois permitem uma pré-avaliação de técnicas referentes ao

modelo acústico, em aplicações de Reconhecimento de Fala Contínua, bem como a implementação das técnicas propostas neste trabalho.

Parâmetros	Acerto de Palavras (%)	Acerto de Frases (%)	Erros de Inclusão (%)	Erros de Exclusão (%)
Mel	96,34	74,75	0,76	0,13
Mel+Dmel	97,6	84,8	0,76	0,26

Tabela (5.3)- Resultados para o sistema empregando Modelos de Palavras

5.3. Segmentação Utilizando Filtragem Paramétrica

Os resultados obtidos utilizando a informação de segmentação gerada pelo método da Filtragem Paramétrica podem ser descritos em função da taxa de acerto de reconhecimento (palavras e frases) e da distorção de segmentação, como foi realizado no Sistema Básico.

A fim de avaliar a influência da Filtragem Paramétrica, foram construídos vários conjuntos de parâmetros h,q e K (vide seção 3.3.3.1 do Capítulo 3), que compõem este método de segmentação. Vale salientar que o parâmetro K corresponde ao tamanho da janela empregada para o cálculo dos parâmetros Delta-Energia, que são combinados com os parâmetros gerados a partir da Filtragem Paramétrica. Tais conjuntos são descritos na tabela a seguir:

BestFP:	$q = \left\{ \frac{2p \cdot 200}{F_s}, \frac{2p \cdot 400}{F_s}, \frac{2p \cdot 600}{F_s}, \frac{2p \cdot 800}{F_s} \right\}$	$h \in [0.1,0.9]$, $m = 4$, $K = 1$
Teta1:	$q = \left\{ \frac{2p \cdot 200}{F_s}, \frac{2p \cdot 800}{F_s}, \frac{2p \cdot 1600}{F_s}, \frac{2p \cdot 2400}{F_s} \right\}$	$h \in [0.1,0.9]$, $m = 4$, $K = 1$
Teta2:	$q = \left\{ \frac{2p \cdot 100}{F_s}, \frac{2p \cdot 200}{F_s}, \frac{2p \cdot 300}{F_s}, \frac{2p \cdot 400}{F_s} \right\}$	$h \in [0.1,0.9]$, $m = 4$, $K = 1$
Teta3:	$q = \left\{ \frac{2p \cdot 400}{F_s}, \frac{2p \cdot 1200}{F_s}, \frac{2p \cdot 2000}{F_s}, \frac{2p \cdot 3500}{F_s} \right\}$	$h \in [0.1,0.9]$, $m = 4$, $K = 1$
Eta1:	$q = \left\{ \frac{2p \cdot 200}{F_s}, \frac{2p \cdot 400}{F_s}, \frac{2p \cdot 600}{F_s}, \frac{2p \cdot 800}{F_s} \right\}$	$h \in [-0.5,0.5]$, $m = 4$, $K = 1$
Eta2:	$q = \left\{ \frac{2p \cdot 200}{F_s}, \frac{2p \cdot 400}{F_s}, \frac{2p \cdot 600}{F_s}, \frac{2p \cdot 800}{F_s} \right\}$	$h \in [-0.1,0.9]$, $m = 4$, $K = 1$

Eta3:	$q = \left\{ \frac{2p \cdot 200}{F_s}, \frac{2p \cdot 400}{F_s}, \frac{2p \cdot 600}{F_s}, \frac{2p \cdot 800}{F_s} \right\}$	$h \in [0.4,0.8]$, $m = 4$, $K = 1$
Delta1:	$q = \left\{ \frac{2p \cdot 200}{F_s}, \frac{2p \cdot 400}{F_s}, \frac{2p \cdot 600}{F_s}, \frac{2p \cdot 800}{F_s} \right\}$	$h \in [0.1,0.9]$, $m = 4$, $K = 2$
Delta2:	$q = \left\{ \frac{2p \cdot 200}{F_s}, \frac{2p \cdot 400}{F_s}, \frac{2p \cdot 600}{F_s}, \frac{2p \cdot 800}{F_s} \right\}$	$h \in [0.1,0.9]$, $m = 4$, $K = 3$

Na tabela (5.4), tem-se as taxas de acerto de palavras e frases do sistema empregando Filtragem Paramétrica, bem como as taxas de erro de exclusão e de inserção. Neste caso, adotou-se a Freqüência de Amostragem (F_s) de 11,025 kHz. Adicionalmente, encontra-se no Apêndice E a lista dos coeficientes g_w e g_g para o caso da Filtragem Paramétrica.

Como se pode observar, os conjuntos BestFP, Teta1 e Teta3 proporcionaram as mesmas taxas de acerto de frases (85,86%) e de palavras (98,11%). Entretanto, o conjunto BestFP apresentou menores taxas de erros de inserção e exclusão, concluindo-se que este constitui o melhor resultado para o sistema empregando Filtragem Paramétrica. Note que houve uma melhoria em relação ao melhor desempenho do Sistema Básico (vide tabela (5.1) e tabela (5.2)).

	Acerto de Frases (%)	Acerto de Palavras (%)	Erros de Exclusão (%)	Erros de Inserção (%)
BestFP	85,86	98,11	0,25	0,13
Teta1	85,86	98,11	0,25	0,25
Teta2	85,86	97,98	0,38	0,13
Teta3	85,86	98,11	0,38	0,13
Eta1	83,84	97,85	0,51	0,13
Eta2	83,84	97,85	0,51	0,25
Eta3	84,85	97,98	0,51	0,13
Delta1	83,84	97,85	0,51	0,13
Delta2	83,84	97,85	0,51	0,25

Tabela (5.4)- Resultados relativos ao desempenho do sistema de reconhecimento empregando a informação de segmentação automática gerada através do método da Filtragem Paramétrica

Verifica-se, ainda, que não houve grande variação do desempenho entre os conjuntos Delta1 e Delta2, indicando pouca sensibilidade às variações do parâmetro *K*. Entretanto, maiores variações no desempenho são percebidas ao variar-se os parâmetros q e h, indicando que o sistema apresenta maior dependência da escolha destes parâmetros.

	Dist. Exclusão	Dist. Inserção	Dist. Segmentação
BestFP	0,201	0,427	0,628
Teta1	0,215	0,443	0,658
Teta2	0,218	0,441	0,659
Teta3	0,202	0,427	0,629
Eta1	0,199	0,430	0,629
Eta2	0,212	0,568	0,780
Eta3	0,200	0,430	0,630
Delta1	0,201	0,426	0,627
Delta2	0,199	0,432	0,632

Tabela (5.5)- Resultados relativos à precisão da segmentação para o sistema empregando Filtragem Paramétrica

Observa-se que o conjunto Delta1 proporcionou a segmentação mais precisa. Entretanto a segmentação obtida empregando-se o conjunto BestFP apresentou maior precisão em relação aos conjuntos Teta1 e Teta3, que apresentaram o mesmo desempenho de reconhecimento (vide tabela(5.4)). Conclui-se, portanto, que o conjunto BestFP apresentou o melhor resultado geral para o sistema empregando Filtragem Paramétrica.

Note, ainda, que novamente os conjuntos Delta1 e Delta2 não provocaram variações muito significativas na medida de distorção de segmentação, confirmando a pouca sensibilidade ao parâmetro *K*. Uma maior oscilação na distorção de segmentação é verificada com as variações dos parâmetros q e h.

Por fim, deve-se notar que os conjuntos Delta2 e Eta1 apresentaram as menores medidas de Distorção de Exclusão, enquanto que o conjunto Delta1 apresentou a menor Distorção de Inserção.

No Apêndice C, tem-se a comparação entre algumas frases reconhecidas com o Sistema Básico e com o sistema empregando o Fator de Ponderação Temporal baseado na Filtragem Paramétrica.

Na figura (5.2) pode-se visualizar melhor a influência da informação de segmentação obtida através da Filtragem paramétrica na segmentação gerada pelo HMM. O sinal analisado corresponde a uma elocução da frase "NOVE DOIS", pronunciada por um locutor feminino.

A figura (5.2-a) mostra as marcas das segmentações manual (linhas pontilhadas) e automática obtida utilizando o HMM obtido no Sistema Básico. A figura (5.2-b) mostra as marcas das segmentações manual (linhas pontilhadas) e automática obtidas através do HMM empregando

Filtragem Paramétrica (conjunto BestFP). Pode-se observar que na figura (5.2-a), existem erros de inserção em torno da amostra 10000, enquanto na figura (5.2-b), estes erros foram corrigidos.

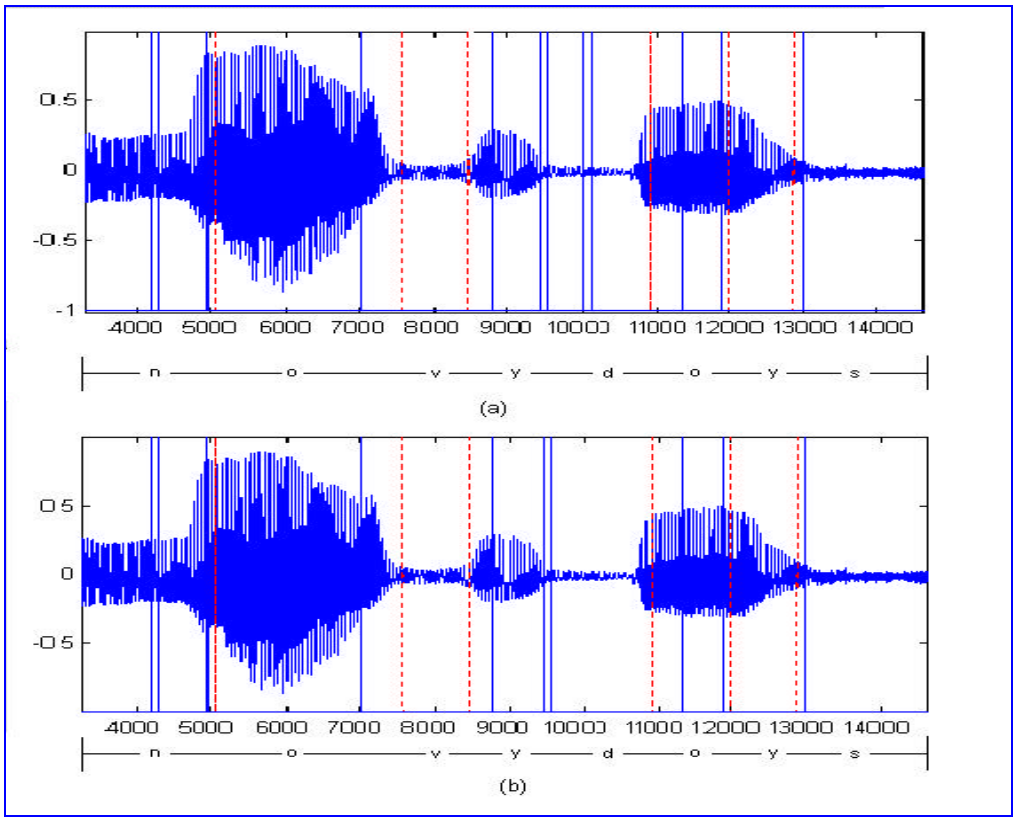

Figura (5.2)- (a) Segmentação manual (marcas pontilhadas) e a segmentação automática gerada pelo HMM do Sistema Básico (marcas contínuas). (b) Segmentação manual (marcas pontilhadas) e a segmentação automática gerada pelo HMM empregando Filtragem Paramétrica (marcas contínuas).

5.4. Segmentação Utilizando MLP

Nesta seção são analisados os resultados relacionados ao sistema empregando a informação de segmentação gerada por meio das Redes Multi-Layer Perceptron (MLP). Novamente, a influência desta informação será avaliada através do desempenho no reconhecimento e da precisão da segmentação.

Os pesos das matrizes W_i e W_s (matrizes de pesos sinápticos das camadas de entrada e intermediária, respectivamente) foram inicializados aleatoriamente com distribuição uniforme no intervalo $[-b,b]$. Vários valores distintos para b foram avaliados, obtendo-se melhores resultados para $b=10^{-7}$. Os parâmetros a e b das funções de ativação das camadas intermediária e de saída foram ajustados independentemente, a fim de minimizar os problemas de saturação dos neurônios.

Os valores que proporcionaram melhor resultado foram:

$a_s = 2.5; b_s = 0.1$
$a_i = 1.8; b_i = 0.1$

A arquitetura adotada consistiu em 1 camada de entrada, 1 camada escondida com N_i neurônios e 1 camada de saída com 1 neurônio. O conjunto de saída foi construído a partir dos Alvos Abruptos e Suaves, como descrito na seção 3.3.3.2. do Capítulo 3.

O treinamento das MLP's foi do tipo lote e o conjunto de treinamento foi composto por 115 elocuções formadas por 8 dígitos conectados, pronunciadas por 7 locutores femininos e 6 masculinos. O conjunto de teste foi composto por 28 elocuções pronunciadas por 3 locutores femininos e dois masculinos. Vale salientar que à entrada da rede MLP devem ser apresentados os vetores acústicos compostos pelos parâmetros Mel-Cepstrais, como descrito na seção 3.3.3.1 do Capítulo 3. Obteve-se, então, os seguintes valores para o Erro Quadrático Médio (EQM) de treinamento e teste, mostrados na tabela abaixo:

Tipo de Alvos	N_i	EQM Treinamento	EQM Teste
Abruptos	30	0,146	0,148
Abruptos	50	0,145	0,147
Abruptos	80	0,144	0,146
Suaves	30	0,186	0,188
Suaves	50	0,186	0,188
Suaves	80	0,178	0,179

Tabela (5.6)- Resultados relativos ao Erro Quadrático Médio (EQM) verificado para o treinamento e teste, bem como para os Alvos Abruptos e Suaves.

Uma vez treinadas as redes, foram geradas as seqüências de variação espectral $s(t)$ a serem utilizadas nos testes de reconhecimento e segmentação automática. Naturalmente, estas seqüências devem passar por um processo de normalização, de modo que $s(t_i) \in [0,1]$, para todo t_i. Finalmente, tem-se no Apêndice E os parâmetros g_w e g_g estimados para as redes de 30, 50 e 80 neurônios na camada intermediária.

Os resultados relativos ao desempenho do sistema são mostrados na tabela (5.7). Observa-se, inicialmente, que o melhor desempenho foi obtido para o caso dos Alvos Abruptos, com 50 neurônios na camada escondida. Pode-se também verificar que, como esperado, o aumento no número de

neurônios da camada intermediária tende a melhorar o desempenho. Tem-se, ainda, que os Alvos Suaves apresentaram taxa de acerto de palavras equivalentes às taxas obtidas com os Alvos Abruptos, apesar de terem ocasionado um Erro Quadrático Médio mais elevado que os Alvos Abruptos, durante as fases de treinamento e de teste da MLP.

	Acerto de Frases (%)	Acerto de Palavras (%)	Erros de Exclusão (%)	Erros de Inserção (%)
Alvos Abruptos, Ni=30	87,88	98,23	0,63	0,13
Alvos Abruptos, Ni=50	87,88	98,23	0,38	0
Alvos Abruptos, Ni=80	86,87	98,36	0,38	0,13
Alvos Suaves, Ni=30	86,87	98,23	0,63	0,13
Alvos Suaves, Ni=50	**87,88**	**98,36**	**0,63**	**0,13**
Alvos Suaves, Ni=80	85.85	98,23	0,51	0

Tabela (5.7)- Resultados relativos ao desempenho do sistema com a utilização da informação de segmentação gerada através das Redes Multi-Layer Perceptron (MLP).

Em seguida, tem-se os resultados associados à influência da segmentação por meio das MLP's sobre a medida de Distorção de Segmentação:

	Dist. Exclusão	Dist. Inserção	Dist. Segmentação
Alvos Abruptos, Ni=30	0,193	0,426	0,619
Alvos Abruptos, Ni=50	0,199	0,431	0,632
Alvos Abruptos, Ni=80	0,198	0,419	0,617
Alvos Suaves, Ni=30	0,190	0,428	0,618
Alvos Suaves, Ni=50	**0,192**	**0,435**	**0,627**
Alvos Suaves, Ni=80	0,192	0,425	0,617

Tabela (5.8)- Resultados relativos à precisão de segmentação com a utilização da informação de variação espectral gerada através das Redes Multi-Layer Perceptron (MLP).

Observa-se que as menores medidas de Distorção de Segmentação são obtidas para os Alvos Suaves ou Abruptos, com N_i = 80. Entretanto, analisando as tabelas (5.7) e (5.8), conclui-se que o melhor resultado, empregando-se MLP's, é obtido através dos Alvos Suaves, com N_i = 50.

Finalmente, comparando as tabelas (5.1), (5.2), (5.4), (5.5), (5.7) e (5.8), pode-se concluir que o melhor resultado geral é obtido com a utilização de uma MLP com 50 neurônios na camada escondida e conjunto de saídas desejadas gerado a partir de Alvos Suaves. Na figura (5.3), tem-se um exemplo das segmentações automáticas obtidas. Neste caso, analisa-se uma elocução da frase "TRÊS

SEIS CINCO", pronunciada por um locutor feminino. A figura (5.3-a) mostra as marcas de segmentação manual (pontilhadas) e as marcas de segmentação automática geradas através do HMM do Sistema Básico (contínuas). Verifica-se erros de inserção em torno da amostra 20.800, justificando o erro de reconhecimento ocorrido, que consistiu na substituição do SEIS pelo TRÊS. A figura (5.3-b), por sua vez, mostra as marcas de segmentação manual (pontilhadas) e as marcas de segmentação automática (linhas contínuas) geradas através do HMM com informação de segmentação gerada pela MLP com Alvos Suaves e $N_i = 80$. Verificou-se que, neste caso, os erros de inserção foram corrigidos, assim como o erro de substituição do SEIS pelo TRÊS.

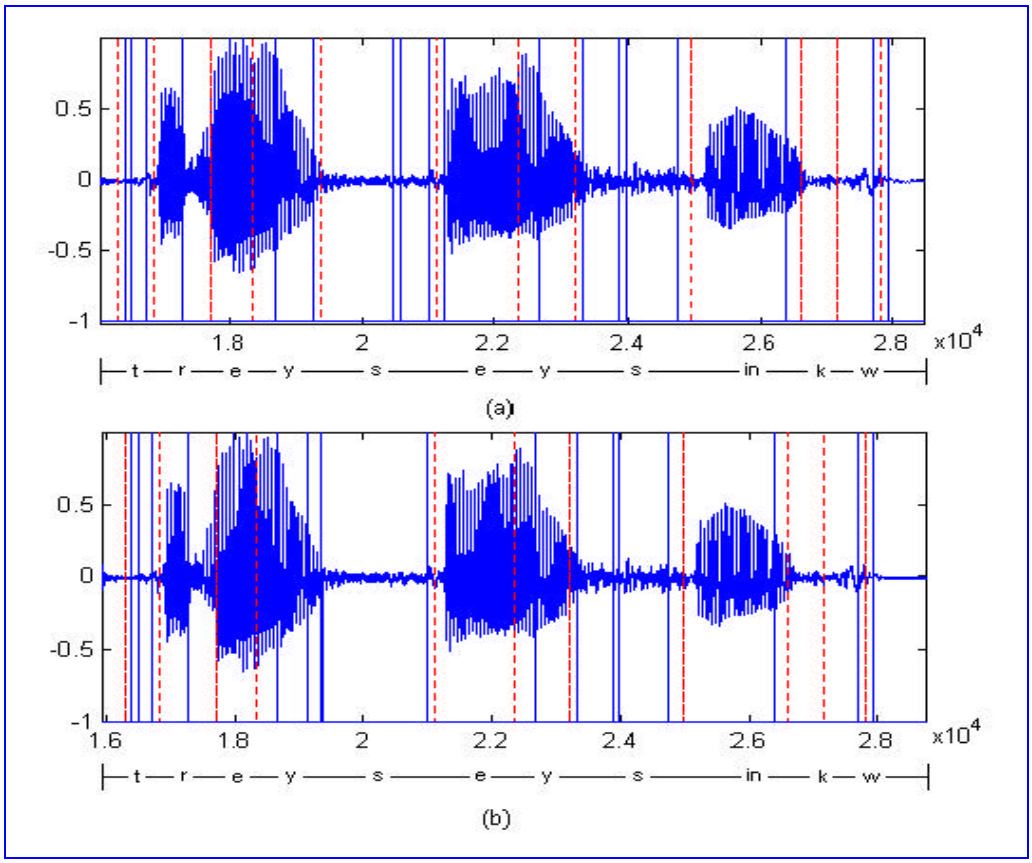

Figura (5.3)- (a) Segmentação manual (marcas pontilhadas) e a segmentação automática gerada pelo HMM do Sistema Básico (marcas contínuas). (b) Segmentação manual (marcas pontilhadas) e a segmentação automática gerada pelo HMM empregando MLP com Alvos Suaves e Ni = 80 (marcas contínuas).

Com relação ao tempo de reconhecimento, verificou-se o acréscimo médio em torno de 22% sobre o tempo do sistema padrão. Este aumento no tempo de reconhecimento é menor que o obtido ao introduzir-se um novo parâmetro de entrada no sistema.

5.5. Aspectos Práticos do Algoritmo de Treinamento Discriminativo

Nesta seção são descritos alguns aspectos práticos relacionados com o treinamento discriminativo de HMM's. Para tanto, são utilizados resultados experimentais relatados na literatura relacionados com aplicações envolvendo dígitos conectados. Inicialmente, em [Chou92] foram realizados experimentos com dígitos conectados, para a Língua Inglesa. Foi empregada a base da Texas Instruments (TI-Database), que é composta por frases de 1 a 7 dígitos conectados. Esta base contém 8565 elocuções para treinamento e 8578 frases para teste. Empregou-se o modelamento acústico baseado em HMM's Contínuos e Modelos de Palavras, com misturas de 64 gaussianas. Como parâmetros de entrada, foram utilizados 12 coeficientes Cepstrais, 12 Delta-Cesptrais e 12 Delta-Delta Cepstrais. Realizou-se o treinamento pelo método ML até obter-se o melhor desempenho possível, gerando os modelos para a inicialização do algoritmo de treinamento discriminativo. Em seguida, executou-se o algoritmo Segmental GPD e obteve-se uma redução de 8% na taxa de erros de frases. Vale salientar que foi adotado o critério de minimização da taxa de erros de palavras para implementar o algoritmo GPD.

Uma experiência análoga é descrita em [Juang97]. Neste caso, porém, utilizou-se o algoritmo GPD para estimar os parâmetros dos HMM's de modo a minimizar a taxa de erros de frases do sistema, e não mais da taxa de erros da palavras. Foi adotada a base de dados TI (Texas Instruments) de dígitos conectados, bem como Modelos de Palavras. Obteve-se os seguintes resultados:

Sistema	Taxa de Erros de Frases (%)	Número de Erros de Frases	Redução na Taxa de Erros (%)
Sistema Básico (ML)	1,4	120	-
Sistema com GPD para minimizar a Taxa de erros de frases	0,95	82	31,6

O critério de minimização da taxa de erros de frases, realizado através do algoritmo Segmental GPD, é uma abordagem descrita em [Chou93]. Neste caso, o algoritmo para encontrar as *N*-Candidatas é empregado com o intuito de gerar as frases concorrentes necessárias para garantir a propriedade discriminativa desejada para o procedimento de estimação dos parâmetros. Novamente foi utilizada a base de dados TI e verificou-se que a taxa de erros de frases de reduziu de 1,3% para 1%, representando uma redução de 23%.

Em [Chen94], tem-se uma descrição mais detalhada do algoritmo de treinamento discriminativo baseado na busca das *N* frases candidatas. A aplicação de reconhecimento de dígitos conectados foi implementada utilizando Modelos de Palavras e uma base de dados de dígitos na Língua Mandarim, pertencente à Telecommunication Laboratories. Esta base é composta por 4000 frases contendo de 2 a 7 dígitos, pronunciadas por 100 locutores, sendo 50 masculinos e 50 femininos. Os resultados referentes à influência do número de frases candidatas são mostrados nas tabelas abaixo:

No. de Frases Candidatas	**Erro % Treinamento**	**Erro % Teste**
1	1,8	6,3
2	1,6	5,9
3	1,8	6,3
5	1,9	6,4
10	2,0	6,4
Critério ML	9,3	10,4

Tabela (5.10)- Resultados referentes a uma aplicação de dígitos conectados empregando Treinamento Discriminativo para HMM's Contínuos com mistura de 1 gaussiana [Chen94].

No. de Frases Candidatas	**Erro % Treinamento**	**Erro % Teste**
1	0,4	5,9
2	0,4	5,6
3	0,4	5,2
5	0,4	5,2
10	0,3	5,1
Critério ML	2,7	8,2

Tabela (5.10)- Resultados referentes a uma aplicação de dígitos conectados empregando Treinamento Discriminativo para HMM's Contínuos com mistura de 2 gaussianas [Chen94].

Pode-se verificar que, no caso do modelo composto por uma gaussiana (tabela (5.9)), existe um número ótimo N^* de frases candidatas que proporciona o melhor desempenho para o sistema. (neste caso, $N^* = 2$). Na Tabela (5.10) o aumento do número de frases candidatas tende a melhorar o desempenho do sistema. Vale salientar que o melhor desempenho foi obtido para o caso de 10 frases candidatas e mistura de 2 gaussianas (taxa de erro de teste igual a 5,1%).

6 Conclusão

6.1. Discussão Geral

Neste trabalho, foram abordados alguns dos principais problemas que limitam o desempenho dos sistemas de Reconhecimento de Fala baseados em HMM's.

Para permitir a realização dos experimentos, definiu-se um Sistema Básico de reconhecimento baseado em HMM Discreto, com modelo de sub-unidades (fones) composto por 3 estados. Adotou-se, ainda, uma aplicação consistindo no reconhecimento de Dígitos Conectados, em português, com frases compostas por 8 dígitos. No Capítulo 5, observou-se que o melhor resultado para o Sistema Básico foi obtido utilizando os parâmetros Mel-Cepstrais e Delta-Mel-Cepstrais. Obteve-se, neste caso, uma taxa de acerto de frases de 81,8% e uma taxa de acerto de palavras de 97,7%. Verificou-se, ainda, que a Medida de Distorção de Segmentação, para este caso, foi 0,635.

Primeiramente, foram analisados os problemas relacionados com o Modelo de Duração de Estados inadequado, bem como com a inconsistência da hipótese de independência entre quadros. Para atenuar estes problemas, foi proposta uma técnica alternativa que consiste na introdução de um Fator de Ponderação Temporal ao longo do processo de busca. Este fator tem a função de penalizar os modelos que gerarem segmentações desalinhadas com os picos de uma função de variação espectral obtida a partir de métodos de Segmentação Automática. Foram, então, implementados dois algoritmos de segmentação: Filtragem Paramétrica e MLP's.

O método baseado na Filtragem Paramétrica consistiu, basicamente, na implementação de um banco de filtros especial, cujas saídas apresentam propriedades de caracterização da estrutura de correlação do sinal de entrada. A informação de variação espectral foi, então, obtida a partir de uma Medida de Distorção apropriada, que avalia a distância entre vetores acústicos adjacentes, correspondentes às saídas do banco de filtros paramétricos, para cada quadro da elocução. Verificou-se (vide Capítulo 5) que, introduzindo-se a informação de segmentação proveniente da Filtragem Paramétrica, o melhor desempenho, em termos de taxa de acertos de reconhecimento e precisão de segmentação, foi obtido para o conjunto de parâmetros denominado BestFP. Neste caso, a taxa de acerto de frases foi de 85,86%, a taxa de acerto de palavras foi de 98,11% e Medida de Distorção de Segmentação foi 0,628. A introdução desta informação de segmentação resultou, portanto, em uma redução de 22,3% na taxa de erro de frases e uma redução de 17,8% na taxa de erro de palavras.

O método baseado nas redes Multi-Layer Perceptron (MLP) consistiu na utilização de uma rede neural para avaliar as variações espectrais ao longo de uma elocução. Realizou-se a segmentação manual das elocuções de treinamento e teste, a fim de permitir a determinação das saídas desejadas Empregou-se o algoritmo Back-Propagation para realizar o treinamento e a arquitetura adotada consistiu em 1 camada de entrada, 1 camada intermediária, com N_i neurônios, e 1 camada de saída, com 1 neurônio. Adotou-se dois tipos de saída desejada (vide seção 3.3.3.2): Alvos Abruptos e Alvos Suaves. A partir dos resultados obtidos no Capítulo 5, verifica-se que os melhores resultados são obtidos com 50 neurônios na camada intermediária (N_i = 50) e Alvos Suaves. Com esta configuração, obteve-se 87,88% de taxa de acerto de frases, 98,36% de taxa de acerto de palavras e Medida de Distorção de Segmentação igual a 0,627. A introdução da informação de segmentação obtida através das MLP's resultou em uma redução de 33,4% na taxa de erro de frases e 28,7% na taxa de erro de palavras.

Realizando-se uma análise sobre os resultados obtidos para todos os experimentos realizados, pode-se concluir que a obtenção de taxas elevadas de acerto de reconhecimento nem sempre resultará em aumento na precisão de segmentação. Entretanto, verifica-se que a obtenção de menores distorções de segmentação está associada a elevações nas taxas de acerto do sistema. Desta forma, a Medida de Distorção de Segmentação representa um parâmetro adicional que permite uma avaliação mais completa da qualidade do modelo acústico construído. De fato, obter um modelo acústico que fornece segmentações precisas, além de baixas taxas de erro de reconhecimento, pode ser um passo inicial importante para tornar o sistema mais robusto, principalmente no que se refere às variabilidades temporais do padrão de voz.

Finalmente, abordou-se no Capítulo 4 os problemas encontrados com o algoritmo de treinamento dos HMM's baseado no critério da Máxima Verossimilhança (ML). Foi então proposto um algoritmo de Treinamento Discriminativo, que se baseia no critério do Erro Mínimo de Classificação (MCE). Foram descritos os algoritmos para o caso de Reconhecimento de Palavras Isoladas, com HMM Contínuo e Modelos de Palavras, bem como para o caso de Reconhecimento de Fala Contínua, com HMM Discreto e Modelos de Sub-Unidades. Como subproduto do algoritmo de treinamento discriminativo, foi proposto, ainda, um método para estimar de forma automática os parâmetros que compõem o Fator de Ponderação Temporal. No Capítulo 5, foram relatados alguns resultados experimentais extraídos da literatura, que mostram a superioridade dos algoritmos discriminativos quando comparados ao algoritmo Baum-Welch. Pode-se, então, concluir que implementar o algoritmo de Treinamento Discriminativo é uma excelente estratégia para melhorar o desempenho do sistema, sem a necessidade de introduzir mais parâmetros de entrada e sem aumentar o tempo de reconhecimento.

Vale salientar que não chegamos a implementar o Treinamento Discriminativo, devido à necessidade de algoritmos de busca do tipo depth-first (Stack, A*), que geram as N frases candidatas com exatidão. Infelizmente, estes algoritmos não estão disponíveis no LPDF e sua implementação demandaria tempo acima do disponível para a conclusão desta tese.

6.2. Contribuições

Através deste trabalho, esperamos reativar as pesquisas relacionadas com os problemas do Modelo de Duração de Estados dos HMM's, uma vez que normalmente se concentra grande parte do esforços apenas no modelamento das variabilidades espectrais dos padrões de voz. Além disto, acreditamos ter esclarecido vários aspectos teóricos e práticos relacionados ao problema da Segmentação Automática da Fala e sua relação com o problema de Reconhecimento de Fala. Finalmente, esperamos ter despertado o interesse dos leitores para a abordagem do Treinamento Discriminativo de HMM's, no sentido de melhorar o desempenho de sistemas de reconhecimento já montados com base em algoritmos de treinamento do tipo Baum-Welch. Adicionalmente, podemos citar as seguinte contribuições principais:

- Levantamento bibliográfico sobre os principais métodos de Segmentação Automática, bem como sobre algoritmos de Treinamento Discriminativo;

- Implementação de dois algoritmos de segmentação automática, a fim de extrair medidas de variação espectral ao longo do tempo;
- Construção de um segmentador automático mais preciso que o Sistema Básico, utilizando um sistema de reconhecimento baseado em HMM Discreto, adicionando as informações de variação espectral ao longo da elocução;
- Elaboração e teste de um método eficiente para incorporar fontes adicionais de conhecimento durante o processo de decodificação acústica. Este método foi representado neste trabalho pelo Fator de Ponderação Temporal;
- Elaboração de um método para a estimação automática, com base no critério MCE, dos parâmetros que compõem o Fator de Ponderação Temporal.

6.3. Sugestões para Trabalhos Futuros

Como sugestões para trabalhos futuros, teríamos:

- Implementar modelos acústicos mais avançados que os HMM's, tais como os Modelos Segmentais, que modelam melhor a variabilidade temporal dos padrões de voz [Ostendorf89][Ostendorf97];
- Aplicar as técnicas propostas para o caso de Reconhecimento de Fala com vocabulários extensos;
- Melhorar o desempenho do segmentador automático obtido, através de incrementos nos métodos aqui apresentados ou através de novas técnicas de Segmentação Automática Irrestrita;
- Implementar algoritmos de busca do tipo *depth-first* (tais como o Herman-Ney, A*), de modo a obter-se as *N* frases candidatas de forma mais simples e exata;
- Implementar o algoritmo de Treinamento Discriminativo de HMM's, tanto para o caso de HMM Discreto quanto para o caso de HMM Contínuo;
- Utilizar a técnica representada pelo Fator de Ponderação Temporal para a introdução de outras fontes de conhecimento durante uma busca integrada. Como exemplo, poderíamos citar a introdução de um fator de ponderação relativo a traços acústicos, que poderiam ser classificados ao longo da elocução através de redes neurais ou neuro-fuzzy.

AN ISOLATED WORD SPEECH RECOGNITION SYSTEM BASED ON KOHONEN NETWORK

Fabrício L. Figueiredo and Fábio Violaro
DECOM - FEEC – UNICAMP
P.O. Box 6101
13083-970 – Campinas – SP – BRAZIL
e-mail {flf, fabio}@decom.fee.unicamp.br

Abstract- This paper describes an algorithm for isolated words recognition using Kohonen Neural Network. This procedure performs a temporal normalization by using a segmentation algorithm, necessary to allow a Multilayer Perceptron Neural Network to recognize the spoken words.

1. INTRODUCTION

The application of Neural Networks to Speech Recognition tasks has shown to be a limited approach due to its inability to deal with nonstationary dynamic patterns. This happens mostly as a consequence of the static architectures of traditional ANN's. Furthermore, there is an inherent generalization difficulty due to the variability at the acoustic patterns for different speakers.

In order to overcome these problems, a possible approach is to apply a nonuniform segmentation algorithm that groups the speech frames in variable length segments, using a spectral distance measure. In other words, the most correlated frames are grouped in nonuniform segments. Thus, this procedure is a temporal normalization that generates a reduced dimension representation of the spoken word, preserving the acoustic properties.

In this paper, the Kohonen Self-Organized Feature Mapping algorithm was modified to build a nonuniform segmenter. A moving neighborhood mechanism for competition was coupled in order to approximate a segmentation algorithm based on LBG vector quantization [2].

The proposed temporal normalization mechanism could also be used in Continuos Word Speech Recognition. However, it should be necessary to adopt an architecture with variable number of neurons, because the limits of each word in the phrase are not determined. This flexibility in the architecture should increase the computational complexity and dealing with this problem is beyond the scope of this work.

2. NONUNIFORM SEGMENTATION

A speech signal representation usually adopted is a sequence of acoustic vectors, each vector being composed by mel-frequency cepstral coefficients [1,6]:

$$X = [\bar{x}_1, \bar{x}_2, ..., \bar{x}_N]$$

The acoustic vector $\bar{x}_i$ is associated with a frame of the speech signal and the number of acoustic vectors N changes according to the word duration.

A segmentation algorithm attempts to map the representation X into another representation Y with fixed number of components :

$$Y = f\{X\} = [\bar{y}_1, \bar{y}_2, ..., \bar{y}_K], \qquad K \le N$$

This problem should be viewed as a K level vector quantization problem, with the number N of elements in the input space varying according to the signal duration, so that:

$$Y = f\{X\} = q\{X\}$$

where q{X} is a mapping to the discrete space Y.

A variety of strategies has been proposed to implement this mapping. In [3], a technique based on joint segmentation and quantization was used. In the present work, the adopted strategy for segmentation was based on the LBG algorithm.

The LBG is a vector quantization algorithm that, by means of an iterative process, builds a codebook $\hat{A}=\{yi;i=1,...,K\}$ in an output space Y associated with a partition $S=\{Si;i=1,...,K\}$ of the input space X. The codebook $\hat{A}$ is built from the centroids of the classes $Si=\{x \in X:q(x)=y_i\}$ that compose a partition S and the partition S is built using the nearest neighbor rule. This procedure is suggested by the following inequalities [4]:

$$D\{\hat{A}, S\} \ge D\{\hat{A}, P(\hat{A})\}$$

$$D\{\hat{A}, S\} \ge D\{q^*(S), S\}$$

where D{.} is a distortion measure, $P(\hat{A})$ is a minimum distortion partition and q*(S) is the minimum distortion codebook.

The partition S is associated with a segmentation of the speech signal and each class Si⊂S is related to a segment. The number of elements (frames) in each class Si is variable, imposing that the segmentation is nonuniform. Besides that, each segment is represented by the centroid of class Si.

The partitioning mechanism of the input space X defines the word segmentation, i.e., the nearest neighbor rule may be used in the segmentation process. However , the order of the acoustic events represented by $\bar{x}_i \in X$ or $\bar{y}_i \in Y$ must be preserved in the partitioning mechanism. Furthermore, the sequence of elements $\bar{y}_i \in Y$ must be directly determined by the sequence of the vectors $\bar{x}_i \in X$ and thus the mapping q*{X} must include this relation.

In [2], the nearest neighbor rule is modified and adapted to the segmentation problem. The input vectors $\bar{x}_i$ are only compared (in the same order of the acoustic events) to the code vectors $\bar{y}_i$ and $\bar{y}_{i+1}$. This mechanism imposes that the elements $\bar{x}_i \in X$ and $\bar{y}_i \in Y$ represent the same sequence of acoustic events. This partitioning mechanism defines the nonuniform segmentation of the representation X as follows:

$$l_i = \min_{n>l_{i-1}}[n-1] \;|\, d(x_n, y_i) > d(x_n, y_{i+1}) \qquad (1)$$

where l_i is an index that points to the last frame (limiter) in segment i.

3- SEGMENTATION USING KOHONEN NETWORK

In this paper, a Self-Organized Neural Network trained using the Kohonen unsupervised algorithm has been proposed to handle the nonuniform segmentation of a speech representation X and to generate the representation Y.

The Kohonen algorithm is based on the Hebbian learning paradigm related to the competitive learning in the process of adjusting the synaptic weights. This training algorithm is named Self-Organized Feature Mapping (SOFM) and works as follows:

1- Randomly initialize the weight matrix of the network;

2- For each input pattern x(n), a winning neuron I{x(n)} is selected according to the minimum Euclidean distance criterion:

$$I\{x[n]\} = \arg\min_{1\le j\le K} \{x[n] - wj[n]\}$$

3- A neighborhood $\Lambda_{I(x)}$ is defined around the winning neuron I[x(n)];

4- Apply the weight adjusting equation to this neighborhood:

$$w_j[n+1] = \begin{cases} w_j[n] + h \cdot \{x[n] - wj[n]\}, & \text{if } j \in \Lambda_{I\{x\}}[n] \\ w_j[n], & \text{if } j \notin \Lambda_{I\{x\}}[n] \end{cases}$$

5- Go back to step 2, until the stop point is reached. The stop point is defined as the desired maximum number of iterations, large enough to allow the topological adjustment of the network.

An interesting property of this algorithm is that it approximates the input space (number of elements N) by a discrete space (number of elements K). Besides that, the network topology is adapted to the probability distribution of the input space elements.

So, the Kohonen algorithm can be viewed as K level vector quantization algorithm.

In fact, [7] shows that the Kohonen algorithm in batch mode and with no neighborhood corresponds to the LBG algorithm. In this case there is a clear relationship between the nearest neighbor rule, in LBG, and the selection of the winning neuron, in SOFM. There is also a relationship between the code vectors y_i, in LBG, and the weights w, in SOFM. Finally, the procedure for calculating the centroids of classes Si, in LBG, corresponds to the weight adjusting rule, in SOFM.

The last relationship becomes more clear if the Kohonen Network is treated as a multidimensional adaptive filter, where each weight vector $\underline{w}_i[n]$, associated to neuron i, is adjusted in order to minimize de error signal $e_i[n]$ (fig..

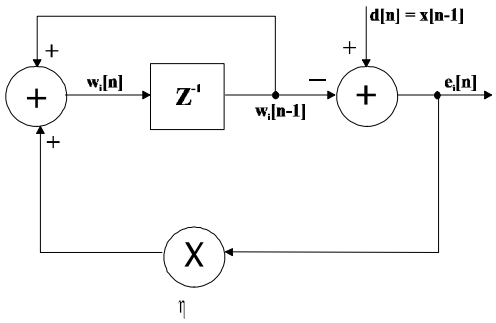

Fig 1: Kohonen weight adjusting equation viewed as an adaptive filtering mechanism.

This proceeding clearly minimizes the mean square error (MSE) and intends to reach the Wiener solution:

$$e_i[k] = x_i[n-1] - w_i[n-1]$$
$$x_i = E\{e_i^2[n]\}$$
$$x_i = E\{x_i[n-1]^2 - 2 \cdot w_i[n-1] \cdot x_i[n-1] + w_i[n-1]^2\}$$

By differentiating ξ with respect to w[n-1], it follows that:

$$g_{w_i} = -2 \cdot E\{x_i[n-1]\} + 2 \cdot E\{w_i[n-1]\} \Rightarrow$$
$$g_{w_i} = -2 \cdot \overline{x_i} + 2 \cdot \overline{w_i}$$

By equating the gradient g_{wi} to zero, it is possible to find the optimum solution (Wiener solution) that minimizes the MSE:

$$w_o = \overline{w_i} = \overline{x_i}$$

The convergence of the algorithm can also be analyzed as follows:

$$w_i[n] = w_i[n-1] + h \cdot \{x_i[n-1] - w_i[n-1]\} \Rightarrow$$
$$w_i[n] = (1-h) \cdot w_i[n-1] + h \cdot x_i[n-1]$$

The difference weight vector is defined as:

$$\Delta w_i[n] = w_i[n] - w_o$$

So, the following relations are obtained:

$$\Delta w_i[n] = (1-h) \cdot \Delta w_i[n-1] + h \cdot \{x_i[n-1] - w_o\} \Rightarrow$$
$$\overline{\Delta w_i[n]} = (1-h) \cdot \overline{\Delta w_i[n-1]} + h \cdot \{\overline{x_i} - w_o\} \Rightarrow$$
$$\overline{\Delta w_i[n]} = (1-h) \cdot \overline{\Delta w_i[n-1]} = (1-h)^{k+1} \cdot \overline{\Delta w_i[0]}$$

Thus, for steady-state convergence, it's necessary to satisfy:

$$0 < h < 1$$

The transient behavior obeys the following time constant:

$$t_{w_i} = \frac{1}{h} \qquad (2)$$

Each neuron representing the code vector y_i filters the input data clustered by class Si. This filtering may be viewed as an average process, imposing that each weight vector moves toward the average vector (centroid) of class Si.

Based on these observations, a nonuniform segmentation algorithm based on the LBG can be implemented using the Kohonen network. In this case, however, the mechanism for selecting the winning neuron should be changed in order to become adapted to the mechanism described by (6).

This modified mechanism can be viewed in figure 2 and consists of imposing restrictions over the competing neurons, at each moment. Data are presented to the network in the same sequence of the acoustic events of the spoken word.

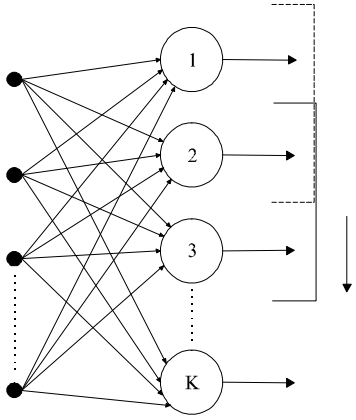

Fig 2: Mechanism for following the acoustic events sequence applied to Kohonen Network.

Initially, only the first neuron is allowed to compete and it must be associated with the first class S_1 which contains the first input vector $\bar{x}_2$. Then the second input vector $\bar{x}_2$ is presented and neurons 1 e 2 are allowed to compete. If neuron 1 wins, the next input vector $\bar{x}_3$ is presented. If neuron 2 wins, the limiter l_1 is fixed in frame 1 and it is imposed that, in the next iteration, only neurons 2 and 3 can compete. This mechanism corresponds to a moving neighborhood for competition.

Using these modifications on the SOFM, the algorithm for nonuniform segmentation proposed in this work is obtained:

1- The weight matrix is initialized using Quasi-Uniform Segmentation [2]. Initialize counter c (c=1), associate the first input data x(1) with neuron 1 and adjust n =2;

2- For each input pattern x(n), a neuron is selected according to the minimum Euclidean distance criterion. The winning neuron I is selected as follows:

$$I\{x[n]\} = \arg \min_{c \le j \le c+1} \{x[n] - wj[n]\}$$

If $I[x(n)] \neq I[x(n-1)]$, c=c+1 and l_i=n-1;

3- Weight adjusting equation is applied as follows:

$$w_j(n+1) = w_j(n) + h \cdot \{x(n) - wj(n)\}$$

4- Go back to step 2, until the stop point is reached, using the desired maximum number of iterations as the stopping criterion. If end of epoch, initializes n=2, c=1 and x(1) is associated with neuron 1.

Finally, the algorithm for training Kohonen network is defined so that the centroids, the classes Si (segments) associated with each neuron and the limiters l_i are obtained and jointly define the word segmentation.

4- IMPLEMENTATION OF THE ISOLATED WORD SPEECH RECOGNITION SYSTEM

The segmentation algorithm proposed in this work was used to build an Isolated Word Speech Recognition System (fig. 3). The selected vocabulary were the digits in portuguese and the adopted sample rate was 8 kHz.

The first block detects the beginning and end of the spoken words, using short-time energy measures and statistical techniques, in order to deal with noise.

The second block extracts the acoustic vectors, that are composed by 12 mel-cepstral parameters, obtained by using temporal windows with 20 ms and a 50% degree of superposition.

The segmenter corresponds to block 3, and is used as a preprocessor to implement the temporal normalization.

The fourth block normalizes the training and testing data just to minimize saturation problems in the neurons of the MLP. In this normalization process, each representation Y (previously segmented) is viewed as a K dimensional random variable and a new random variable Z with zero average and variance 0.25 is defined:

$$Z = A.[X - E(X)]$$

where $A=1/(2.\sqrt{s_Y^2})$ is a constant adjusted to guarantee the desired variance.

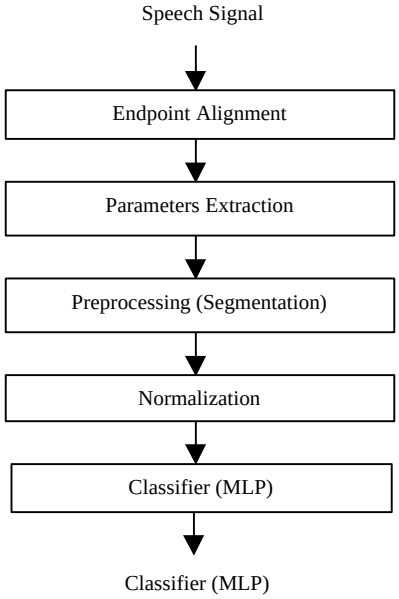

Fig. 3: Block diagram representing the adopted Isolated Speech Recognition System using nonuniform segmentation.

In the last block, a Multilayer Perceptron network was used as the classifier. In this MLP the Back-Propagation Algorithm in batch mode was used for training, and the network was built with 50 neurons in the hidden layer and 10 neurons in the output layer. The number of input nodes is determined by the desired number of segments K and the dimension of the acoustic vectors. In this work, acoustic vectors with dimension 12 were adopted. The recognized word is associated with the output neuron presenting the highest output level.

5- RESULTS

5.1- Segmentation Results.

In the segmentation algorithm based on the Kohonen network , the learning rate was fixed in η=0.1 and it was necessary 15,000 iterations for convergence (average distortion below 0.005). In figure (4), some examples of a word segmentation with 15 and 20 segments are shown. In general, the most correlated frames are grouped in the same segments.

The used database was composed by 34 male speakers and 26 female speakers. Each speaker repeated three times each word. The adopted vocabulary were the digits, from 0 to 9, in portuguese. The total number of words in the database was 1800.

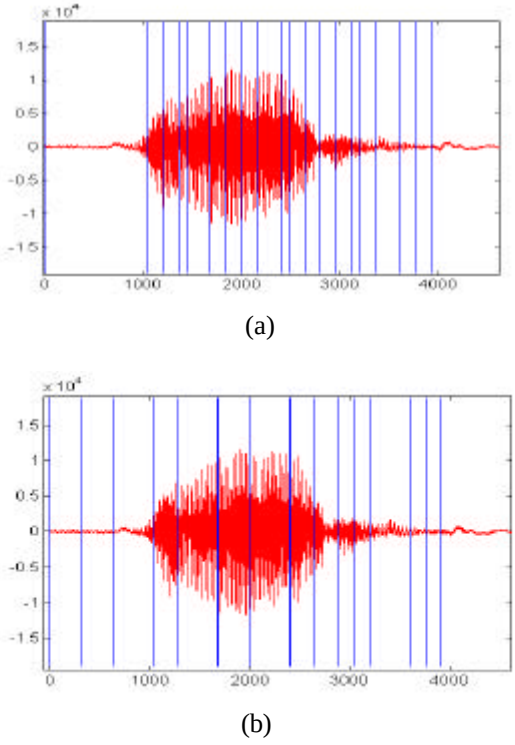

Fig. 4. a) Spoken word 'zero', with 20 segments through Kohonen algorithm. b) Spoken word 'zero', with 15 segments using Kohonen algorithm.

In figure (5), there is a comparison between the nonuniform segmentation algorithms based on the Kohonen network and on the LBG algorithm. It is also shown the spectrogram of the speech signal, useful to identify the acoustic events along the time. The word "zero" is used in the example. It was spoken by a male speaker and was segmented in 7 parts.

Figures (5.a), (5.b) and (5.c) show that the resulting segmentations are different. In figure (5.a), the LBG based segmentation is shown. Figure (5.b) is the Kohonen based segmentation with learning rate equal to 0.01. The behavior of both algorithms are very similar. However in figure (5.c) the Kohonen segmentation was obtained by adjusting the learning rate η to a larger value (η = 0.2), thus reducing the time constant τ (eq.2). As shown in the spectrogram (fig. 5.d), the segmenter became more sensible to non-stationarities and grouped them in isolated segments, just improving the detection of phoneme boundaries. This property is very important in applications related to speech recognition.

So, the Kohonen segmentation allows the control of the clustering tendency by adjusting the learning rate η. In practice, this parameter shall be trimmed in order to allow an adaptation to the acoustic events of the experiment.

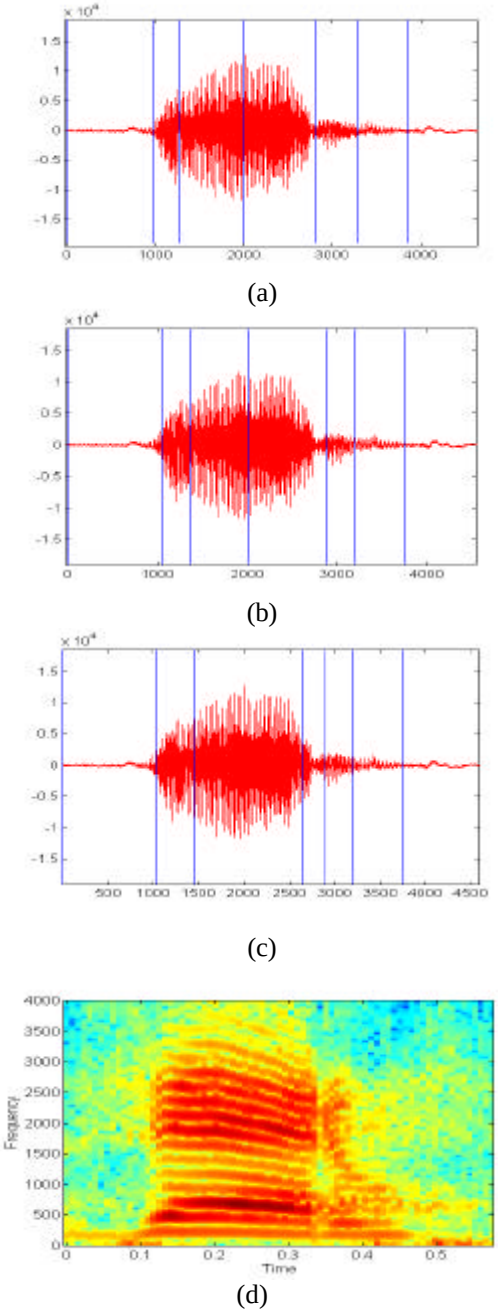

Fig. 5. a) Speech signal of word 'zero', with 5 segments using LBG algorithm. b) Speech signal of word 'zero', with 5 segments using the Kohonen network with η=0.01. c) Speech signal of word 'zero', with 5 segments using the Kohonen network with η=0.2. d) Spectrogram of the original speech signal.

5.2- Classifier Results

Initially, the MLP was trained using 20 segments. The training set (including the validation set) was composed by 1350 words and the test set was composed by 450 words. The training consisted of

272 epochs and the error rate for the test set was 4.6%.

Using 15 segments, the training set was composed by 1320 words and the test set by 480 word. In this case the training consumed 227 epochs for convergence and the error rate for the test set was 5.6%.

6- CONCLUSIONS

In this work, it was possible to notice how useful a nonuniform segmentation algorithm can be in an Isolated Word Speech Recognition system. It was shown that the Kohonen network can be used to implement such segmentation.

It was also verified that it's possible to control de clustering tendency of the algorithm by adjusting the leraning rate η.

Finally, an important result observed is that the representation used in the segmentation process is acoustically consistent and assures a good performance for the MLP. This performance can be improved by using more input parameters, as well as better classifier architectures and training algorithms.

REFERENCES

[1]L.R. Rabiner e B.H. Juang, "Fundamentals of Speech Recognition", *Prentice Hall*, 1993.

[2] J.G. Guimarães, E.C. Negreiros, L.M. Silva e A.R.S.Romariz, "Comparação de Técnicas de Ajuste Temporal para Reconhecimento de Palavras Isoladas com Redes Neurais", *XV Simpósio de Telecomunicações*, Recife, PE, pp. 336-339, 1997.

[3] Y. Shiraki e M. Honda, "LPC Speech Coding Based on Variable-Length Segment Quantization", *IEEE Trans. Acoust. and Speech, Signal Processing*, vol. 36, no. 9, pp.1437-1444, 1988.

[4] Y. Linde, A. Buzo e R.M. Gray, "An algorithm for Vector Quantizer Design", *IEEE Trans. On Communications*, vol. 28, pp. 84-95, 1980.

[5] J. Tebelskis, "Speech Recognition Using Neural Networks", *Carnegie Mellon University*, 1995.

[6] N.B. Yoma, "Reconhecimento Automático de Palavras Isoladas: Estudo e Aplicação dos Métodos Determinístico e Estocástico", *State University of Campinas*, 1993.

[7] S. Haykin, Neural Networks: a Comprehensive Foundation, *Prentice Hall*, New Jersey, E.U.A, 1994.

Apêndice B:

Lista das Frases Empregadas no Sistema

Frases de Teste

UM	CINCO	DOIS	TRÊS	TRÊS	SEIS	CINCO	TRÊS
UM	UM	DOIS	CINCO	TRÊS	SETE	CINCO	CINCO
UM	NOVE	DOIS	QUATRO	TRÊS	OITO	CINCO	QUATRO
UM	DOIS	DOIS	OITO	TRÊS	NOVE	CINCO	OITO
UM	MEIA	DOIS	UM	TRÊS	ZERO	CINCO	UM
MEIA	TRÊS	UM	OITO	TRÊS	MEIA	QUATRO	TRÊS
DOIS	SETE	UM	TRÊS	TRÊS	TRÊS	QUATRO	SETE
DOIS	DOIS	UM	SEIS	TRÊS	DOIS	QUATRO	NOVE
DOIS	NOVE	UM	QUATRO	TRÊS	CINCO	QUATRO	QUATRO
DOIS	MEIA	UM	ZERO	TRÊS	QUATRO	QUATRO	UM
DOIS	DOIS	UM	SEIS	TRÊS	DOIS	QUATRO	NOVE

Frases de Treinamento

QUATRO	OITO	NOVE	SEIS	SETE	OITO	OITO	SEIS
QUATRO	ZERO	NOVE	SETE	SETE	ZERO	OITO	SETE
QUATRO	SEIS	NOVE	MEIA	SETE	SEIS	OITO	OITO
QUATRO	CINCO	NOVE	NOVE	SETE	CINCO	OITO	NOVE
QUATRO	DOIS	NOVE	ZERO	SETE	DOIS	OITO	ZERO
MEIA	CINCO	NOVE	UM	SETE	TRÊS	OITO	UM
CINCO	SETE	SEIS	TRÊS	SETE	SETE	OITO	MEIA
CINCO	DOIS	SEIS	SEIS	SETE	NOVE	OITO	DOIS
CINCO	NOVE	SEIS	QUATRO	SETE	QUATRO	OITO	CINCO
CINCO	SEIS	SEIS	ZERO	SETE	UM	OITO	QUATRO
CINCO	ZERO	SEIS	NOVE	SETE	MEIA	OITO	TRÊS

SEIS	CINCO	ZERO	OITO	UM	CINCO	DOIS	TRÊS
SEIS	UM	ZERO	ZERO	UM	UM	DOIS	CINCO
SEIS	MEIA	ZERO	SEIS	UM	NOVE	DOIS	MEIA
SEIS	DOIS	ZERO	MEIA	UM	DOIS	DOIS	OITO
SEIS	SETE	ZERO	DOIS	UM	MEIA	DOIS	UM
MEIA	QUATRO	ZERO	TRÊS	UM	OITO	DOIS	ZERO
NOVE	TRÊS	ZERO	SETE	UM	TRÊS	DOIS	SETE
NOVE	DOIS	ZERO	NOVE	UM	SEIS	DOIS	DOIS
NOVE	CINCO	ZERO	QUATRO	UM	QUATRO	DOIS	NOVE
NOVE	QUATRO	ZERO	UM	UM	ZERO	DOIS	SEIS
NOVE	OITO	ZERO	CINCO	UM	SETE	DOIS	QUATRO

ZERO	CINCO	MEIA	MEIA	CINCO	SEIS	SEIS	OITO
ZERO	UM	MEIA	UM	CINCO	SETE	SEIS	ZERO
ZERO	NOVE	MEIA	NOVE	CINCO	OITO	SEIS	SEIS
ZERO	DOIS	MEIA	DOIS	CINCO	MEIA	SEIS	CINCO
ZERO	SETE	MEIA	SETE	CINCO	ZERO	SEIS	DOIS
MEIA	ZERO	MEIA	OITO	CINCO	UM	SEIS	TRÊS
TRÊS	SETE	DOIS	ZERO	CINCO	TRÊS	SEIS	SETE
TRÊS	DOIS	CINCO	MEIA	CINCO	DOIS	SEIS	NOVE
TRÊS	NOVE	SEIS	OITO	CINCO	CINCO	SEIS	QUATRO
TRÊS	SEIS	ZERO	OITO	CINCO	QUATRO	SEIS	UM
TRÊS	MEIA	OITO	ZERO	CINCO	NOVE	SEIS	MEIA

OITO	TRÊS	SETE	SEIS	NOVE	OITO	TRÊS	TRÊS
OITO	CINCO	SETE	SETE	NOVE	ZERO	TRÊS	CINCO
OITO	MEIA	SETE	OITO	NOVE	SEIS	TRÊS	QUATRO
OITO	OITO	SETE	NOVE	NOVE	CINCO	TRÊS	OITO
OITO	UM	SETE	ZERO	NOVE	MEIA	TRÊS	UM
MEIA	SEIS	SETE	UM	NOVE	TRÊS	TRÊS	ZERO
SETE	TRÊS	OITO	SETE	NOVE	SETE	ZERO	TRÊS
SETE	DOIS	OITO	DOIS	NOVE	NOVE	ZERO	SEIS
SETE	CINCO	OITO	NOVE	NOVE	QUATRO	ZERO	QUATRO
SETE	QUATRO	OITO	SEIS	NOVE	UM	ZERO	ZERO
SETE	MEIA	OITO	QUATRO	NOVE	DOIS	ZERO	MEIA

Apêndice C:

Lista das Frases Reconhecidas pelo Sistema Básico e das Frases Reconhecidas pelo Sistema com Melhor Desempenho (MLP, Ni=50, Alvos Suaves)

Simbologia:

f = Erro de Exclusão

INS = Erro de Inserção

SUB = Erro de Substituição

, = Silêncio

SISTEMA BÁSICO	SISTEMA COM MELHOR DESEMPENHO
, UM CINCO DOIS TRÊS TRÊS **TRÊS** CINCO TRÊS ,	, UM CINCO DOIS TRÊS TRÊS SEIS CINCO TRÊS ,
, UM UM DOIS CINCO TRÊS SETE CINCO CINCO ,	, UM UM DOIS CINCO TRÊS SETE CINCO CINCO ,
, UM NOVE DOIS QUATRO TRÊS **DOIS** CINCO QUATRO ,	, UM NOVE DOIS QUATRO TRÊS **DOIS** CINCO QUATRO ,
, UM DOIS DOIS , **DOIS** TRÊS NOVE CINCO OITO	, UM DOIS DOIS **DOIS** TRÊS NOVE CINCO OITO
, UM MEIA DOIS UM TRÊS ZERO CINCO UM ,	, UM **UM** MEIA DOIS UM TRÊS ZERO CINCO **f** ,
, MEIA TRÊS UM OITO **SEIS** MEIA QUATRO TRÊS ,	, MEIA TRÊS UM OITO **SEIS** MEIA QUATRO TRÊS ,
, DOIS SETE UM TRÊS TRÊS TRÊS QUATRO SETE **SEIS**	, DOIS , SETE UM TRÊS TRÊS TRÊS QUATRO SETE
, DOIS DOIS UM SEIS TRÊS DOIS QUATRO NOVE ,	, DOIS DOIS UM SEIS TRÊS DOIS QUATRO NOVE ,
, DOIS NOVE UM QUATRO TRÊS CINCO QUATRO QUATRO ,	, DOIS NOVE UM QUATRO TRÊS CINCO QUATRO QUATRO ,
, DOIS MEIA UM ZERO TRÊS QUATRO QUATRO UM ,	, DOIS MEIA UM ZERO TRÊS QUATRO QUATRO UM ,
, DOIS SEIS UM SETE **SEIS** UM QUATRO MEIA ,	, DOIS SEIS UM SETE **SEIS** UM QUATRO MEIA ,
, UM CINCO DOIS TRÊS TRÊS **TRÊS** , CINCO TRÊS	, UM CINCO DOIS , TRÊS TRÊS SEIS CINCO TRÊS
, UM **f** DOIS CINCO TRÊS SETE CINCO CINCO , ,	, UM **f** DOIS CINCO TRÊS SETE CINCO CINCO ,
, UM NOVE **UM** DOIS QUATRO TRÊS OITO CINCO QUATRO	, UM NOVE DOIS QUATRO TRÊS OITO CINCO , QUATRO
, UM DOIS DOIS OITO TRÊS NOVE CINCO OITO ,	, UM DOIS DOIS OITO TRÊS NOVE CINCO OITO ,
, UM MEIA DOIS UM TRÊS ZERO CINCO UM ,	, UM MEIA DOIS UM TRÊS ZERO CINCO UM ,
, MEIA TRÊS UM , OITO TRÊS MEIA QUATRO TRÊS	, MEIA TRÊS UM , OITO TRÊS MEIA QUATRO TRÊS
, DOIS , SETE UM TRÊS TRÊS TRÊS QUATRO SETE	, DOIS , SETE UM TRÊS TRÊS TRÊS QUATRO SETE
, DOIS DOIS UM SEIS TRÊS DOIS QUATRO NOVE ,	, DOIS DOIS UM SEIS TRÊS DOIS QUATRO NOVE ,
, DOIS NOVE UM QUATRO TRÊS CINCO QUATRO QUATRO ,	, DOIS NOVE UM QUATRO TRÊS CINCO QUATRO QUATRO ,
, DOIS MEIA UM ZERO TRÊS QUATRO QUATRO UM ,	, DOIS MEIA UM ZERO TRÊS QUATRO QUATRO UM ,
, DOIS SEIS UM SETE TRÊS UM QUATRO MEIA ,	, DOIS SEIS UM SETE TRÊS UM QUATRO MEIA ,
, UM CINCO DOIS TRÊS TRÊS SEIS CINCO TRÊS ,	, UM CINCO DOIS TRÊS TRÊS SEIS CINCO TRÊS ,
, UM **f** DOIS , CINCO TRÊS SETE CINCO CINCO ,	, UM **f** DOIS , CINCO TRÊS SETE CINCO CINCO ,

, UM NOVE DOIS QUATRO TRÊS OITO CINCO QUATRO ,	, UM NOVE DOIS QUATRO TRÊS OITO CINCO QUATRO ,
, UM DOIS DOIS OITO TRÊS NOVE CINCO OITO ,	, UM DOIS DOIS OITO TRÊS NOVE CINCO OITO ,
, UM MEIA DOIS UM TRÊS ZERO CINCO UM ,	, UM MEIA DOIS UM TRÊS ZERO CINCO UM ,
, MEIA TRÊS UM OITO TRÊS MEIA QUATRO TRÊS ,	, MEIA TRÊS UM OITO TRÊS MEIA QUATRO TRÊS ,
, DOIS SETE UM TRÊS TRÊS TRÊS QUATRO SETE ,	, DOIS SETE UM TRÊS TRÊS TRÊS QUATRO SETE ,
, DOIS DOIS UM SEIS TRÊS DOIS QUATRO NOVE ,	, DOIS DOIS UM SEIS TRÊS DOIS QUATRO NOVE ,
, DOIS NOVE UM QUATRO TRÊS CINCO QUATRO QUATRO ,	, DOIS NOVE UM QUATRO TRÊS CINCO QUATRO QUATRO ,
, DOIS MEIA UM **SETE** TRÊS QUATRO QUATRO UM ,	, DOIS MEIA UM **SETE** TRÊS QUATRO QUATRO UM ,
, DOIS SEIS UM SETE TRÊS UM QUATRO MEIA ,	, DOIS SEIS UM SETE TRÊS UM QUATRO MEIA ,
, UM CINCO DOIS TRÊS TRÊS SEIS CINCO TRÊS ,	, UM CINCO DOIS TRÊS TRÊS SEIS CINCO TRÊS ,
, UM UM DOIS CINCO TRÊS SETE CINCO CINCO ,	, UM UM DOIS CINCO TRÊS SETE CINCO CINCO ,
, UM NOVE DOIS QUATRO TRÊS OITO CINCO QUATRO ,	, UM NOVE DOIS QUATRO TRÊS OITO CINCO QUATRO ,
, UM DOIS DOIS OITO TRÊS NOVE CINCO OITO ,	, UM DOIS DOIS OITO TRÊS NOVE CINCO OITO ,
, UM MEIA DOIS UM TRÊS ZERO CINCO UM ,	, UM MEIA DOIS UM TRÊS ZERO CINCO UM ,
, MEIA TRÊS UM OITO TRÊS MEIA QUATRO TRÊS	, , MEIA TRÊS UM OITO TRÊS MEIA QUATRO TRÊS
, DOIS SETE UM TRÊS TRÊS TRÊS QUATRO SETE ,	, DOIS SETE UM TRÊS TRÊS TRÊS QUATRO SETE ,
, DOIS DOIS UM SEIS TRÊS DOIS QUATRO NOVE ,	, DOIS DOIS UM SEIS TRÊS DOIS QUATRO NOVE ,
, DOIS NOVE **f** QUATRO , TRÊS CINCO QUATRO QUATRO ,	, , DOIS NOVE **f** QUATRO TRÊS CINCO QUATRO QUATRO ,
, DOIS MEIA UM ZERO TRÊS QUATRO QUATRO UM ,	, DOIS MEIA UM ZERO TRÊS QUATRO QUATRO UM ,
, DOIS **TRÊS** UM SETE TRÊS UM QUATRO MEIA ,	, DOIS SEIS UM SETE TRÊS UM QUATRO MEIA ,
, UM CINCO DOIS TRÊS TRÊS SEIS CINCO TRÊS ,	, UM CINCO DOIS TRÊS TRÊS SEIS CINCO TRÊS ,
, UM UM DOIS CINCO TRÊS SETE CINCO CINCO ,	, UM UM DOIS CINCO TRÊS SETE CINCO CINCO ,
, UM NOVE DOIS QUATRO TRÊS OITO CINCO QUATRO ,	, UM NOVE DOIS QUATRO TRÊS OITO CINCO QUATRO ,
, UM DOIS DOIS OITO TRÊS NOVE CINCO OITO ,	, UM DOIS DOIS OITO TRÊS NOVE CINCO OITO ,
, UM MEIA DOIS UM TRÊS ZERO CINCO UM ,	, UM MEIA DOIS UM TRÊS ZERO CINCO UM ,
, MEIA TRÊS UM OITO TRÊS MEIA QUATRO TRÊS ,	, MEIA TRÊS UM OITO TRÊS MEIA QUATRO TRÊS ,
, DOIS SETE UM TRÊS TRÊS TRÊS QUATRO SETE ,	, DOIS SETE UM TRÊS TRÊS TRÊS QUATRO SETE ,
, DOIS DOIS UM SEIS TRÊS DOIS QUATRO NOVE ,	, DOIS DOIS UM SEIS TRÊS DOIS QUATRO NOVE ,
, DOIS NOVE **TRÊS** QUATRO TRÊS CINCO QUATRO QUATRO ,	, DOIS NOVE **TRÊS** QUATRO TRÊS CINCO QUATRO QUATRO ,
, DOIS MEIA UM ZERO TRÊS QUATRO QUATRO UM ,	, DOIS MEIA UM ZERO TRÊS QUATRO QUATRO UM ,
, DOIS SEIS UM SETE TRÊS UM QUATRO MEIA ,	, DOIS SEIS UM SETE TRÊS UM QUATRO MEIA ,
, UM CINCO DOIS TRÊS TRÊS SEIS CINCO TRÊS ,	, UM CINCO DOIS TRÊS TRÊS SEIS CINCO TRÊS ,
, UM **f** DOIS CINCO TRÊS SETE CINCO CINCO ,	, UM **f** DOIS CINCO TRÊS SETE CINCO CINCO ,
, UM NOVE DOIS QUATRO TRÊS OITO CINCO QUATRO ,	, UM NOVE DOIS QUATRO TRÊS OITO CINCO QUATRO ,
, UM DOIS DOIS OITO TRÊS NOVE CINCO OITO ,	, UM DOIS DOIS OITO TRÊS NOVE CINCO OITO ,
, UM MEIA DOIS UM TRÊS ZERO CINCO UM ,	, UM MEIA DOIS UM TRÊS ZERO CINCO UM ,
, MEIA TRÊS UM OITO TRÊS MEIA QUATRO TRÊS ,	, MEIA TRÊS UM OITO TRÊS MEIA QUATRO TRÊS ,
, DOIS SETE UM TRÊS TRÊS TRÊS QUATRO SETE ,	, DOIS SETE UM TRÊS TRÊS TRÊS QUATRO SETE ,
, DOIS DOIS UM SEIS TRÊS DOIS QUATRO NOVE ,	, DOIS DOIS UM SEIS TRÊS DOIS QUATRO NOVE ,
, DOIS NOVE UM QUATRO TRÊS CINCO QUATRO QUATRO ,	, DOIS NOVE UM QUATRO TRÊS CINCO QUATRO QUATRO ,
, DOIS MEIA UM ZERO TRÊS QUATRO QUATRO UM ,	, DOIS MEIA UM ZERO TRÊS QUATRO QUATRO UM ,
, DOIS SEIS UM SETE TRÊS UM QUATRO MEIA ,	, DOIS SEIS UM SETE TRÊS UM QUATRO MEIA ,
, UM CINCO DOIS TRÊS TRÊS SEIS CINCO TRÊS ,	, UM CINCO DOIS TRÊS TRÊS SEIS CINCO TRÊS ,
, UM **f** DOIS CINCO TRÊS SETE CINCO CINCO ,	, UM UM DOIS CINCO TRÊS SETE CINCO CINCO ,
, UM NOVE DOIS QUATRO TRÊS OITO CINCO QUATRO ,	, UM NOVE DOIS QUATRO TRÊS OITO CINCO QUATRO ,
, UM DOIS DOIS OITO TRÊS NOVE CINCO OITO ,	, UM DOIS DOIS OITO TRÊS NOVE CINCO OITO ,
, UM MEIA DOIS UM TRÊS ZERO CINCO UM ,	, UM MEIA DOIS UM TRÊS ZERO CINCO UM ,

, MEIA TRÊS UM OITO TRÊS MEIA QUATRO TRÊS ,	, MEIA TRÊS UM OITO TRÊS MEIA QUATRO TRÊS ,
, DOIS SETE UM TRÊS **SEIS** TRÊS QUATRO SETE ,	, DOIS SETE UM TRÊS **SEIS** TRÊS QUATRO SETE ,
, DOIS DOIS UM SEIS TRÊS DOIS QUATRO NOVE ,	, DOIS DOIS UM SEIS TRÊS DOIS QUATRO NOVE ,
, DOIS NOVE UM QUATRO TRÊS CINCO QUATRO QUATRO ,	, DOIS NOVE UM QUATRO TRÊS CINCO QUATRO QUATRO ,
, DOIS MEIA UM ZERO TRÊS QUATRO QUATRO UM ,	, DOIS MEIA UM ZERO TRÊS QUATRO QUATRO UM ,
, DOIS **TRÊS** UM SETE TRÊS UM QUATRO MEIA ,	, DOIS SEIS UM SETE TRÊS UM QUATRO MEIA ,
, UM CINCO DOIS TRÊS TRÊS SEIS CINCO TRÊS ,	, UM CINCO DOIS TRÊS TRÊS SEIS CINCO TRÊS ,
, UM UM DOIS CINCO TRÊS SETE CINCO CINCO ,	, UM UM DOIS CINCO TRÊS SETE CINCO CINCO ,
, UM NOVE DOIS QUATRO TRÊS OITO CINCO QUATRO ,	, UM NOVE DOIS QUATRO TRÊS OITO CINCO QUATRO ,
, UM DOIS DOIS OITO TRÊS NOVE CINCO OITO ,	, UM DOIS DOIS OITO TRÊS NOVE CINCO OITO ,
, UM MEIA DOIS UM TRÊS ZERO CINCO UM ,	, UM MEIA DOIS UM TRÊS ZERO CINCO UM ,
, MEIA TRÊS UM OITO TRÊS MEIA QUATRO TRÊS ,	, MEIA TRÊS UM OITO TRÊS MEIA QUATRO TRÊS ,
, DOIS SETE UM TRÊS TRÊS TRÊS QUATRO SETE ,	, DOIS SETE UM TRÊS TRÊS TRÊS QUATRO SETE ,
, DOIS DOIS UM SEIS TRÊS DOIS QUATRO NOVE ,	, DOIS DOIS UM SEIS TRÊS DOIS QUATRO NOVE ,
, DOIS NOVE UM QUATRO TRÊS CINCO QUATRO QUATRO ,	, DOIS NOVE UM QUATRO TRÊS CINCO QUATRO QUATRO ,
, DOIS MEIA UM ZERO TRÊS QUATRO QUATRO UM ,	, DOIS MEIA UM ZERO TRÊS QUATRO QUATRO UM ,
, DOIS SEIS UM SETE TRÊS UM QUATRO MEIA ,	, DOIS SEIS UM SETE TRÊS UM QUATRO MEIA ,
, UM CINCO DOIS TRÊS TRÊS SEIS CINCO TRÊS	, , UM CINCO DOIS TRÊS TRÊS SEIS CINCO TRÊS
, UM UM DOIS CINCO TRÊS SETE CINCO CINCO ,	, UM UM DOIS CINCO TRÊS SETE CINCO CINCO ,
, UM NOVE DOIS QUATRO TRÊS OITO CINCO QUATRO ,	, UM NOVE DOIS QUATRO TRÊS OITO CINCO QUATRO ,
, UM DOIS DOIS OITO TRÊS NOVE CINCO OITO ,	, UM DOIS DOIS OITO TRÊS NOVE CINCO OITO ,
, UM MEIA DOIS UM TRÊS ZERO CINCO UM ,	, UM MEIA DOIS UM TRÊS ZERO CINCO UM ,
, MEIA TRÊS UM OITO TRÊS MEIA QUATRO TRÊS	, MEIA , TRÊS UM OITO TRÊS MEIA QUATRO TRÊS
, DOIS SETE UM TRÊS TRÊS TRÊS QUATRO SETE ,	, DOIS SETE UM TRÊS TRÊS TRÊS QUATRO SETE ,
, DOIS DOIS UM SEIS TRÊS DOIS QUATRO NOVE ,	, DOIS DOIS UM SEIS TRÊS DOIS QUATRO NOVE ,
, DOIS NOVE UM QUATRO TRÊS CINCO QUATRO QUATRO ,	, DOIS NOVE UM QUATRO TRÊS CINCO QUATRO QUATRO ,
, DOIS MEIA UM ZERO TRÊS QUATRO QUATRO UM ,	, DOIS MEIA UM ZERO TRÊS QUATRO QUATRO UM ,
, DOIS SEIS UM SETE TRÊS UM QUATRO MEIA ,	, DOIS SEIS UM SETE TRÊS UM QUATRO MEIA ,

Apêndice D:

Lista dos Fonemas Empregados e Transcrição Fonética Adotada para os Dígitos em Português

LISTA DOS FONEMAS
a A e E y in o O w un d k m n r s t T v z

TRANSCRIÇÃO FONÉTICA DOS DÍGITOS	
, (silêncio)	#
ZERO	z E r w
UM	un
DOIS	d o y s
DOIS	d o y z
TRÊS	t r e y s
TRÊS	t r e y z
QUATRO	k w a t r w
CINCO	s in k w
SEIS	s e y s
SEIS	s e y z
SETE	s E T y
OITO	o y t w
NOVE	n O v y
MEIA	m e y A

Apêndice E:

Lista dos Parâmetros do Fator de Ponderação Temporal

1. Filtragem Paramétrica $(g_g = 0{,}25)$

Modelo	g_w
#	**1**
z E r w	**1**
un	**0,8**
d o y s	**1**
d o y z	**1**
t r e y s	**0,55**
t r e y z	**0,55**
k w a t r w	**1**
s in k w	**1**
s e y s	**0,2**
s e y z	**0,2**
s E T y	**0,3**
o y t w	**0,1**
n O v y	**1**
m e y A	**0,8**

2. MLP – 30 Neurônios na Camada Intermediária $(g_g = 0{,}25)$

Modelo	g_w
#	**1**
z E r w	**1**
un	**0,8**
d o y s	**1**
d o y z	**1**
t r e y s	**0,62**
t r e y z	**0,62**
k w a t r w	**1**
s in k w	**1**
s e y s	**0,32**
s e y z	**0,32**
s E T y	**0,25**
o y t w	**0,23**
n O v y	**1**
m e y A	**0,8**

3. MLP – 50 Neurônios na Camada Intermediária $(g_g = 0{,}25)$

Modelo	g_w
#	**1**
z E r w	**1**
un	**0,8**
d o y s	**1**
d o y z	**1**
t r e y s	**0,65**
t r e y z	**0,65**
k w a t r w	**1**
s in k w	**1**
s e y s	**0,3**
s e y z	**0,3**
s E T y	**0,25**
o y t w	**0,2**
n O v y	**1**
m e y A	**0,8**

4. MLP – 80 Neurônios na Camada Intermediária $(g_g = 0{,}25)$

Modelo	g_w
#	**1**
z E r w	**1**
un	**1**
d o y s	**1**
d o y z	**1**
t r e y s	**0,9**
t r e y z	**0,9**
k w a t r w	**1**
s in k w	**0,7**
s e y s	**0,35**
s e y z	**0,35**
s E T y	**0,25**
o y t w	**0,2**
n O v y	**1**
m e y A	**0,8**

Referências

[Bakis76] R. Bakis, "Continuous speech word recognition via centisecond acoustic states", Proceedings of ASA Meeting (Washington, DC), April, 1976.

[Bourland85] H. Bourland, Y. Camp, C.J. Wellekens, "Speaker Dependent Connected Speech Recognition Via Phonemic Markov Models", Proc. Int'l. Conf. on Acoust., Speech and Signal Processing, pp. 1213-1216, 1985.

[Bush87] M.A. Bush e G.E. Kopec, "Network-based connected digit recognition", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 35, pp. 1401-1413, 1987.

[Chen94] J.-K. Chen e F.K. Soong, "An N-Best candidates-based discriminative training for speech recognition", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 2(1), pp. 206-216, January 1994.

[Chou92] W. Chou, B.-H. Juang e C.-H.Lee, "Segmental GPD training of HMM based speech recognizer", Proc. Int'l. Conf. on Acoust., Speech and Signal Processing, pp. 473-476, 1992.

[Chou93] W. Chou, C.-H. Lee e B.-H. Juang, "Minimum error rate training based on N-Best string models", Proc. Int'l. Conf. on Acoust., Speech and Signal Processing, pp. 652-655, 1993.

[Chow90] Y.-L. Chow, "Maximum Mutual Information estimation of HMM parameters for continuous speech recognition using the N-Best algorithm", Proc. Int'l. Conf. on Acoust., Speech and Signal Processing, pp. 701-704, 1990.

[Dosierre93] O.B. Dosierre, "Segmentation automatique d'unites acoustiques por las synthese de la parole", Docteur These, L'Universite de Rennes I, 1993.

[Fagundes98] R.D.R. Fagundes, "Abordagem fonético-fonológica em sistemas de reconhecimento de voz de linguagem contínua", Tese de Doutorado, USP, São Paulo, 1998.

[Ferguson80] J.D. Ferguson, "Hidden Markov Analysis: An Introduction", Hidden Markov Models for Speech, Institute for Defense Analyses, Princeton, NJ, 1980;

[Fukada97] T. Fukada, S. Aveline, M. Schuster e Y. Sagisaka, "Segment boundary estimation using recurrent neural networks", Proceedings of EUROSPEECH'97, pp. 2839-2842, 1997.

[Gales87] M. Gales and S. Young, "The theory of segmental hidden Markov models", Cambridge University Engineering Department, Technical Report, CUED / F-IFENG/TR.133, 1987.

[Gibson96] T.-H. Li e J.D. Gibson, "Speech analysis and segmentation by parametric filtering", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 4(3), pp. 203-213, May 1996.

[Glass88] J.R. Glass e V.W. Zue, "Multi-level segmentation of continuous speech", Proc. Int'l. Conf. on Acoust., Speech and Signal Processing, pp. 429-432, 1988.

[Hermert91] J.P.V. Hermert, "Automatic Segmentation of Speech", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 39(4), pp. 1008-1012, April 1991.

[Huo93] Q. Huo e C. Chan, "The Gradient Projection Method for the training of hidden Markov models", Speech Communication, vol. 13, pp. 307-313, 1993.

[Huo95] Q. Huo e C. Chan, "Discriminative training of HMM based speech recognizer with Gradient Projection Method", Proceedings of EUROSPEECH'95, pp. 101-104, 1995.

[Jiménez95] V.M. Jiménez, A. Marzal e J. Monné, "A comparison of two exact algorithms for finding the N-Best sentence hypotheses in continuous speech recognition", Proceedings of EUROSPEECH'95, pp. 1071-1074, 1995.

[Juang97] B. –H. Juang, W. Chou e C. –H. Lee, "Statistical and discriminative methods for speech recognition", Automatic Speech and Speaker Recognition – Advanced Topics, Kluwer Academic Publishers, pp. 109-132, 1997.

[Kenny90] P. Kenny, M. Lennig e P. Mermelstein, "A linear predictive HMM for vector-valued observations with applications to speech recognition", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 38, pp. 220-225, 1990.

[Lee89] C.-H. Lee e L.R. Rabiner, "A frame-synchronous network search algorithm for connected word recognition", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 37(11), pp. 1649-1658, November 1989.

[Lee91] C.–H. Lee, C.–H. Lin e B.H. Juang, "A study on speaker-adaptation of the parameters of continuous density hidden Markov models", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 39, pp. 806-841, April 1991.

[Levinson86] S.E. Levinson, "Continuously variable duration hidden Markov models for automatic speech recognition", Computer Speech and Language, pp. 29-45, March, 1986.

[Li94] T.H. Li e J.D. Gibson, "Discriminat analysis of speech by parametric filtering", Proc. IEEE Conf. Inform. Science and Systems, PP. 575-580, Mar, 1994.

[Li95] T.H. Li, "Robust divergence measures for time series diecrimination", tech. rep. no. 237, Dept. of Statistics, Texas A&M University, College Station, TX, 1995

[Luna90] J.C.S. Luna, J.M.L. Soler, A.P. –Herreros, V.S. –Cale e A.J.R. Ayuso, "Signal segmentation into spectral homogeneous units", Signal Processing V: Theories and Applications, L. Torres, E. Masgrau and M.A. Lagunas (eds.), Elsevier Science Publishers B.V., pp. 1255-1258, 1990.

[Marzal90] A. Marzal, "Nuevos métodos de segmentación para la decodificación acústico-fonética", Facultad de Informática, Universidad Politécnica de Valência, Abril, 1990.

[Morais97] E.S. Morais, "Reconhecimento Automático de Fala Contínua Empregando Modelos Híbridos ANN + HMM", Tese de Mestrado, UNICAMP, 1997.

[Ostendorf89] M. Ostendorf e S. Roukos, "A Stochastic Segment Model for Phoneme-Based Continuous Speech Recognition", IEEE Transactions on Acoustic, Speech and Signal Processing, vol. 37, pp. 1857-1869, Dec. 1989.

[Ostendorf97] M. Ostendorf, "From HMM's to Segment Models: Stochastic Modeling for CSR", Automatic Speech and Speaker Recognition – Advanced Topics, Kluwer Academic Publishers, pp. 185-210, 1997.

[Paul91] D. Paul, "Algorithms for an optimal A* search and linearizing the search in stack decoder", Proc. Int'l. Conf. on Acoust., Speech and Signal Processing, pp. 693-696, 1991.

[Pessoa99] L.A.S. Pessoa, "Modelos da Língua para o Português do Brasil Aplicados ao Reconhecimento de Fala Contínua: Modelos Lineares e Modelos Hierárquicos (Parsing)", Tese Mestrado, UNICAMP, Campinas, 1999.

[Rabiner85] L.R. Rabiner, S.E. Levinson, " A Speaker-Independent, Syntax-Directed, Connected Word Recognition System Based on Hidden Markov Models and Level Building", IEEE Transactions on Acoustic, Speech and Signal Processing, v. 33, n. 3, p. 561-573, Jun., 1985.

[Rabiner89] L.R. Rabiner, J.G. Wilpon and F.K. Soong, "High Performance Connected Digit Recognition Using Hidden Markov Models", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 37, pp. 1214-1225, August 1989.

[Rabiner93] L.R. Rabiner and B.H. Juang, “Fundamentals of Speech Recognition”, Prentice Hall, 1993.

[Reichl95] W. Reichl e G. Ruske, "Discriminative training for continuous speech recognition", Proceedings of EUROSPEECH'95, pp. 537-540, 1995.

[Rubio95] A.J. Rubio e R.G. Reilly, "Preliminary results on speech signal segmentation with recurrent neural networks", Proceedings of EUROSPEECH'95, pp. 2197-2200, 1995.

[Russel85] M. Russel and R. Moore, "Explicit modeling of state occupancy in hidden Markov models for automatic speech recognition", Proc. Int'l. Conf. on Acoust., Speech and Signal Processing, pp. 2376-2379, 1985.

[Russel93] M. Russel, "A segmental HMM for speech pattern matching", Proc. Int'l. Conf. on Acoust., Speech and Signal Processing, vol II, pp. 499-502, 1993.

[Schwartz97] R. Schwartz, L. Nguyen, J. Makhoul, "Multiple-pass search strategies", Automatic Speech and Speaker Recognition – Advanced Topics, Kluwer Academic Publishers, pp. 429-456, 1997.

[Schwartz85] R. Schwartz, Y. Chow, O. Kimball, S. Roucos, M. Krasner e J. Makhoul, "Context-dependent modelling for acoustic-phonetic recognition of continuous speech", Proc. Int'l. Conf. on Acoust., Speech and Signal Processing, pp. 1205-1208, 1988.

[Soong90] F.K. Soong e E.F. Huang, "A Tree-Trellis based fast search for finding the N-Best sentence hypotheses in continuous speech recognition", Proceedings of ICSLP'90, pp. 709-712, 1990.

[Suh96] Y. Suh e Y. Lee, "Phoneme segmentation of continuous speech using multi-layer perceptron", Proceedings of ICSLP'96, pp. 273-275, 1996.

[Svendsen87] T. Svendsen e F.K. Soong, "On the automatic segmentation of speech signals" Proc. Int'l. Conf. on Acoust., Speech and Signal Processing, pp. 77-89, 1987.

[Vidal90] E. Vidal e A. Marzal, "A review and new approaches for automatic segmentation of speech signals", Signal Processing V: Theories and Applications, L. Torres, E. Masgrau and M.A. Lagunas (eds.), Elsevier Science Publishers B.V., pp. 43-53, 1990.

[Wellekens87] C.J. Wellekens, "Explicit time correlation in hidden Markov models for speech recognition", Proc. Int'l. Conf. on Acoust., Speech and Signal Processing, pp. 384-386, 1987.

[Wilpon87] J.G. Wilpon, B.-H. Juang e L.R. Rabiner, "An investigation on the use of acoustic sub-word units for automatic speech recognition", Proc. Int'l. Conf. on Acoust., Speech and Signal Processing, pp. 821-824, 1987.

[Witkin84] A.P. Witkin, "Scale-space filtering: a new approach to multi-scale description", Proc. Int'l. Conf. on Acoust., Speech and Signal Processing, 1987.

[Zelinski83] R. Zelinski e F. Glass, "A segmentation algorithm for connected word recognition based on estimation principles", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 31(4), pp. 818-827, August 1983.

[Zue89] V. Zue, J. Glass, M. Philips e S. Seneff, "Acoustic segmentation and phonetic classification in the SUMMIT system", Proc. Int'l. Conf. on Acoust., Speech and Signal Processing, 1989, pp. 389-392.