UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE COMUNICAÇÕES

AVALIAÇÃO DE DIFERENTES TÉCNICAS PARA RECONHECIMENTO DE FALA

JOSÉ ANTÔNIO MARTINS

Orientador: Prof. Doutor FÁBIO VIOLARO

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas - UNICAMP, como parte dos requisitos exigidos para a obtenção do título de DOUTOR EM ENGENHARIA ELÉTRICA.

DEZEMBRO - 1997

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

M366a

Martins, José Antônio

Avaliação de diferentes técnicas para reconhecimento de fala / José Antônio Martins.–Campinas, SP: [s.n.], 1997.

Orientador: Fábio Violaro.

Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

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

Resumo

Neste trabalho são avaliadas as seguintes técnicas empregadas em reconhecimento de fala: Modelos Ocultos de Markov (Discreto, Semicontínuo e Contínuo), redes " Multilayer Perceptron " e Sistemas Híbridos. Para essa avaliação, foram implementados vários reconhecedores de palavras isoladas independentes do locutor, sendo usada a mesma base de dados para todos os reconhecedores. O objetivo dessa avaliação é identificar as vantagens e desvantagens de cada técnica de modo a facilitar a escolha da técnica mais apropriada para cada aplicação de reconhecimento de fala considerando diversos aspectos como taxa de acerto, tempo de reconhecimento, tempo de treinamento, quantidade de parâmetros, tipo de algoritmo e outros. Reconhecedores implementados empregando as técnicas descritas também foram utilizados para avaliar algoritmos para detecção de início/fim de palavras, vários tipos de medidas de distorção usados em quantização vetorial e diferentes parâmetros utilizados para representar o sinal de fala. Entre esses parâmetros mostrou-se que existe uma superioridade dos parâmetros calculados usando uma escala não linear de freqüências, independentemente da técnica empregada nos reconhecedores. Discute-se também o uso da subtração da média espectral e diferenciação, as quais melhoram o desempenho dos reconhecedores. Combinando diferentes parâmetros conseguiu-se obter uma taxa de acertos de 99.47% para um reconhecedor de palavras isoladas independente do locutor e um vocabulário de 50 palavras.

Abstract

This work presents an evaluation of speaker independent isolated word recognizers using Hidden Markov Models (Discrete, Continuous and Semicontinuous), Artificial Neural Networks (Multilayer Perceptron) and Hybrid Systems. All the recognizers were evaluated considering the same database. The goal of these comparisons is to identify the advantages and disadvantages of each technique used in speech recognition, considering the following features: training and recognition time, recognition accuracy, complexity of algorithms and others. It is also reported the result of a comparison among different algorithms used in word endpoints detection. Moreover, several distance measures employed in vector quantization were evaluated with regard to recognition performance. In addition, different kinds of parameters used to represent the speech signal such as LPC coefficients, Mel Frequency Cepstrum coefficients, PLP coefficients were considered in the evaluation of recognizers and it was discussed the efects of cepstral mean subtraction in order to improve the recognition accuracy. The best recognizer performance of 99.47% was obtained combining different features.

AOS MEUS PAIS
RUBENS E EULÁLIA

Agradecimentos

Ao meu orientador *Professor Doutor Fábio Violaro*, meu grande agradecimento pela orientação, dedicação, paciência, encorajamento, compreensão e discussões técnicas sem as quais não seria possível a realização deste trabalho.

Agradeço ao *Dr. José Sindi Yamamoto* que possibilitou o meu ingresso na área de reconhecimento de fala.

Meus agradecimentos aos pesquisadores *Fernando C. Tofoli Queiroz* e *Renato Marche* pelo apoio e ajuda na aquisição da base de dados, aos pesquisadores *Fernando Oscar Runstein* e *Eliana De Martino* pelo apoio e discussões técnicas que em muito contribuíram para o desenvolvimento deste trabalho e também às pesquisadoras *Flávia Martinho Ferreira Rocha* e *Margarete Mitiko Iramina* pelo incentivo e grande apoio nos momentos difíceis.

Eu sou muito grato ao Grupo de Suporte em Informática do Departamento de Redes de Telecomunicações do CPqD/TELEBRÁS, *Enéas Jorge da Cruz, Gerson Marconi* e *José Flávio dos Santos* que sempre resolveram com muita eficiência todos os meus problemas computacionais.

Gostaria de agradecer ao pesquisador *Ralph Robert Heinrich*, gerente do departamento de Redes de Telecomunicações do CPqD/TELEBRÁS e ao pesquisador *Eduardo Moreno Marques*, gerente da Seção de Acesso Rádio do CPqD/TELEBRÁS, pelo apoio e incentivo para que esse trabalho fosse realizado.

Aos companheiros de trabalho da Seção de Acesso Rádio do CPqD/TELEBRÁS, agradeço pelo apoio, colaboração, tolerância e compreensão diante do meu enorme uso das estações de trabalho e espaço em disco. Eu sou muito grato ao *Delson, Dinho, Cássio, Cláudio, Ênio, Feris, Fio, Flávio, Geraldo G., Geraldo L., Henrique, Idelma, Ivonete, Ivan, Jadir, Jaime, Júlio, Omar, Paulo, Pedro, Petry, Renato, Shinoda, Wander e Zuffo.*

Também gostaria de agradecer ao pesquisadores *Argemiro O. Sousa Filho, Cláudia Povoas Schmidt, João Luis Alves Mendes* e *Roberto Y. Nakamura* pelo apoio, incentivo, tolerância e compreensão pelo monopólio da estação de trabalho.

Os meus agradecimentos aos amigos do Grupo de Processamento de Fala do Forschungs Institut (FI 13) da Deutsche Bundespost Telekom, pelo apoio e discussões durante o período em que trabalhamos juntos. Muito obrigado ao *Dr. Bernhard Kaspar, Dr. Bernd Lochschmidt, Dr. Georg Fries, Dr. Stefan Feldes, Dr. Antje Wirth, Dr. Ulrich Schultheiß, Andreas Schimm, Christian Fleischmann,*

Wolfgang Bleck, Wolfgang Schrecke, Gerhard Schröder, Karlheinz Schuhmacher, Kurt Wendel, Joachim Stegmann, Margot Engraf e Contanze Seidelmann.

Finalmente, gostaria de expressar os meus sinceros agradecimentos a todos os meus *amigos* que sempre me incentivaram e a todos que colaboraram com suas vozes para a execução das simulações realizadas neste trabalho. A todos o meu *Muito Obrigado*.

Índice

Lista de Figuras

Lista de Tabelas

Capítulo 1

INTRODUÇÃO

A tarefa de um sistema de reconhecimento de fala, o qual tem como entrada a forma de onda acústica produzida pelo ser humano, é produzir como saída a seqüência de fonemas, palavras ou sentenças correspondente ao sinal de entrada.

O principal objetivo das pesquisas na área de reconhecimento de fala é o desenvolvimento de uma máquina que seja capaz de transcrever a fala com uma alta taxa de precisão para todos os locutores e em todos os ambientes, possibilitando dessa a forma a comunicação homem-máquina via fala da mesma forma como é feita entre seres humanos.

Embora esse objetivo ainda esteja um pouco distante de ser alcançado, muitos progressos têm sido feitos nos últimos anos na área de reconhecimento de fala. Isso é devido principalmente ao desenvolvimento da microeletrônica (tecnologia VLSI), a qual possibilitou um rápido crescimento da capacidade dos processadores digitais de sinais, ao mesmo tempo em que ocorria uma diminuição nos preços. Adicionalmente, foram apresentados novos algoritmos usando diferentes técnicas matemáticas para modelar os problemas de reconhecimento de fala [1, 2, 3, 4].

São várias as vantagens em empregar a fala na comunicação homem-máquina. Dentre elas, podem ser citadas [5]:

- Não requer treinamento: o uso da fala não exige que a pessoa seja treinada ou tenha determinadas habilidades.

- Rapidez: através da fala, a informação é fornecida mais rapidamente que quando digitada.

- O usuário pode utilizar o sistema enquanto está se movendo ou fazendo uma outra atividade que requer o uso das mãos.

- A rede telefônica pode ser usada para a passagem da informação, possibilitando dessa forma o acesso remoto ao sistema.

Apesar dessas facilidades, a utilização da fala também apresenta algumas desvantagens como a suscetibilidade à interferência do ambiente, necessitando de um cancelador de ruídos para ambientes muito ruidosos [5]. Além disso, as características do sinal de fala variam de pessoa para pessoa, de acordo com as diferenças na forma de falar, dimensão dos órgãos do trato vocal, velocidade da fala e condições físicas e emocionais dos locutores (doença, stress, estado emocional) [5].

O primeiro trabalho na área de reconhecimento de fala foi publicado em 1952 por Davis, Biddulph e Balashek (Bell Labs) [6] e descrevia um sistema para reconhecer dígitos isolados falados por um único locutor. Desde essa data, muitos pesquisadores, em diferentes países, têm trabalhado nessa área e conseguido expressivos resultados.

Durante as décadas de 50 e 60, vários reconhecedores de dígitos e fonemas foram implementados e bons resultados foram alcançados [7].

No final dos anos 60 e início dos anos 70 foram apresentadas as técnicas de Predição Linear (" Linear Predictive Coding " (LPC)) [8] e " Dynamic Time Warping " (DTW) [8], as quais proporcionaram um grande impulso nas pesquisas em reconhecimento de fala. Nessa mesma época foram apresentados algoritmos para reconhecimento de palavras conectadas e para a fala contínua.

A década de 80 caracterizou-se pela difusão de métodos baseados no modelamento estatístico, por exemplo Modelos Ocultos de Markov (" Hidden Markov Models " (HMM)) [8]. Nesse período também foi introduzido o uso de Redes Neurais para reconhecimento de fala e foi dado um grande impulso para a implementação de sistemas robustos de reconhecimento de fala contínua para grandes vocabulários. Foram obtidos sistemas com altas taxas de reconhecimento. Como exemplos desses sistemas, podem ser citados o sistema BYBLOS [9] com taxa de acerto de 93% e o sistema SPHINX [10] com taxa de acerto de 96.2%.

Devido aos bons resultados conseguidos, aumentou bastante o interesse pelos sistemas de reconhecimento de fala e o uso dos mesmos em diferentes áreas tornou-se mais difundido.

As aplicações do reconhecimento de fala podem ser divididas em duas grandes áreas [11, 12]: telecomunicações e negócios. Como exemplos dessas aplicações podemos citar:

- Acesso à informação

  O usuário deseja ter acesso a algum tipo de informação, a qual está armazenada em uma base de dados. Exemplos: notícias, previsão do tempo, programação de cinemas e teatros, horóscopo, hora certa e outros.

  Neste tipo de aplicação, os sistemas apresentam um menu falado ao usuário e este escolhe a informação desejada através de comandos verbais.

- Operações bancárias

  O usuário realiza operações bancárias como transferência de dinheiro, informações sobre o saldo de sua conta, através de comandos verbais pelo telefone.

- Serviços de compra por telefone

  O usuário pode comprar produtos usando o telefone. Nesse tipo de serviço, cada produto tem um código numérico e o usúario solicita via fala o produto desejado.

- Operação de serviços

  Serviços telefônicos, que usualmente necessitam de um operador humano, são automatizados. Exemplo: atendimento automático, informações sobre lista telefônica, serviço de despertar e outros.

- Envio automático de fax

  O usuário seleciona um documento de uma lista, falando sim ou não e fala o número do aparelho que receberá o fax.

- Correio de Voz

  O usuário pode solicitar a reprodução de mensagens de sua caixa postal ou o envio de mensagens através de comandos verbais.

- Discagem por fala

  O usuário fala o número do telefone desejado. Pode-se também, associar a cada número de telefone o nome de uma pessoa, e neste caso o usuário fala o nome da pessoa.

- Preenchimento de formulários

  O usuário entra com os dados via fala.

- Conversão Fala-Texto

  O usuário fala o texto ao invés de digitá-lo.

Considerando essas aplicações, vários sistemas com diferentes características foram implementados, sendo que muitos estão operando comercialmente. Como exemplos de sistemas de reconhecimento de fala temos:

- Dragon - 1975 - Carnegie-Mellon University [13]: reconhecimento de fala contínua dependente do locutor com vocabulário de 194 palavras com taxa de acerto de 84%.

- Hearsay - 1975 - Carnegie-Mellon University [14]: reconhecimento de fala contínua dependente do locutor com vocabulário de 1011 palavras com taxa de acerto de 87%.

- Harpy - 1976 - Carnegie-Mellon University [15]: reconhecimento de fala contínua dependente do locutor com vocabulário de 1011 palavras com taxa de acerto de 97%.

- Bell Labs - 1982 [16]: reconhecimento de palavras isoladas independente do locutor com vocabulário de 129 palavras com taxa de acerto de 91%.

- Prina - 1982 - Ericson Business Systems [17]: reconhecimento de palavras isoladas dependente do locutor para vocabulário pequeno (menos que 25 palavras).

- Feature - 1983 - Carnegie-Mellon University [18]: reconhecimento de palavras isoladas independente do locutor, com vocabulário constituído pelas letras do alfabeto com taxa de acerto de 90%.

- Tangora - 1985 - IBM [19]: reconhecimento de palavras isoladas dependente do locutor com vocabulário de 5000 palavras com taxa de acerto de 97%.

- Bell Labs - 1988 [20]: reconhecimento de dígitos conectados independente do locutor com taxa de acerto de 97.1%.

- Byblos - 1988 - BBN [9]: reconhecimento de fala contínua dependente do locutor com vocabulário de 997 palavras com taxa de acerto de 93%.

- Sphinx - 1988 - Carnegie-Mellon University [10]: reconhecimento de fala contínua independente do locutor com vocabulário de 997 palavras com taxa de acerto de 96.2%.

- Teleton - 1988 - Deutsche Bundespost Telekom [21]: reconhecimento de palavras isoladas independente do locutor com vocabulário de 12 palavras.
- Babsy - 1990 - Deutsche Bundespost Telekom [22]: reconhecimento de palavras isoladas independente do locutor com vocabulário de 18 palavras com taxa de acerto de 95%.
- Citruf - Deutsche Bundespost Telekom [22]: reconhecimento de palavras isoladas independente do locutor para vocabulário pequeno (menos que 25 palavras).
- Teledialogue - 1992 - Jydsk Telefon [22]: reconhecimento de palavras isoladas independente do locutor para vocabulário pequeno (menos que 25 palavras).
- Audiotex - 1992 - Telefonica I. D. de Espanha [23, 24]: reconhecimento de palavras isoladas independente do locutor com vocabulário de 12 palavras com taxa de acerto de 96%.
- World Window - 1992 - Global Communications Ltd. [22]: reconhecimento de palavras conectadas dependente do locutor com vocabulário de 200 palavras.
- Mairievox - 1990 - France Telecom [25]: reconhecimento de palavras isoladas e conectadas independente do locutor com vocabulário de 21 palavras com taxa de acerto de 88%.
- Les Balandins - 1992 - France Telecom [22]: reconhecimento de palavras isoladas e conectadas independente do locutor com vocabulário de 26 palavras com taxa de acerto de 95%.
- Acueil 14 - 1992 - France Telecom [22]: reconhecimento de palavras isoladas e conectadas independente do locutor com vocabulário de 33 palavras.
- Philips Dictation Systems - 1996 [26]: reconhecimento de palavras isoladas dependente do locutor com vocabulário de 64000 palavras.

Levando-se em conta a grande importância da área de reconhecimento de fala e as diversas técnicas existentes, torna-se necessário uma comparação entre elas para a escolha da melhor técnica de forma a obter o melhor reconhecedor para cada aplicação. Assim, neste trababalho serão apresentadas comparações entre as técnicas Modelos Ocultos de Markov (HMM - " Hidden Markov Models ") e Redes Neurais (" Multilayer Perceptron "), sob diversos aspectos como taxa de acerto,

tempo de reconhecimento, tempo de treinamento, quantidade de parâmetros, topologia e outros. Para isso, foram implementados vários reconhecedores de palavras isoladas independentes do locutor usando um vocabulário de 50 palavras faladas em português do Brasil.

Primeiramente, analisou-se o desempenho dos reconhecedores usando diferentes tipos de parâmetros do sinal de fala. Essa comparação foi feita usando-se as técnicas HMM e " Multilayer Perceptron ".

Considerando-se apenas o método HMM, foram feitas comparações entre HMMs discretos, contínuos e semicontínuos levando-se em conta o desempenho, tempo de treinamento, tempo de reconhecimento e diferentes alternativas de implementação.

De modo a comparar as várias técnicas para implementação de reconhecedores de fala, foram implementados reconhecedores usando HMM e " Multilayer Perceptron " e também foram consideradas e avaliadas algumas propostas de sistemas híbridos que utilizam essas duas técnicas conjuntamente.

No capítulo 2 é descrito um sistema básico de reconhecimento de fala. Esse sistema extrai parâmetros do sinal de fala, faz uma comparação com os padrões armazenados medindo a similaridade entre estes e o padrão a ser reconhecido e usa essa medida para escolher o padrão que melhor representa o sinal de fala. Além disso, são explicadas algumas classificações dos reconhecedores de fala considerando o tipo de locutor, tamanho do vocabulário e tipo de fala.

No capítulo 3 são apresentados aspectos da teoria e implementação da técnica de Modelos Ocultos de Markov e os métodos usados para obter os parâmetros que caracterizam cada modelo. Está incluida a descrição dos algoritmos " Forward ", " Forward-Backward " e Viterbi.

No capítulo 4 é discutida a aplicação de Redes Neurais (" Multilayer Perceptron ") para reconhecimento de fala. É apresentado o algoritmo " Back-Propagation ", o qual é usado para treinar a rede.

No capítulo 5 são descritos sistemas híbridos, os quais combinam as técnicas HMM e Redes Neurais (" Multilayer Perceptron ") .

No capítulo 6 é apresentada uma análise dos diversos tipos de parâmetros que podem ser usados para representar o sinal de fala. A comparação entre esses parâmetros é feita usando-se reconhecedores de palavras isoladas empregando as técnicas HMM e Redes Neurais. Além disso, são descritos alguns algoritmos para detectar o início e fim de cada palavra e são discutidos alguns tipos de distância usados na quantização vetorial.

No capítulo 7 são descritos os reconhecedores implementados usando HMM e são mostradas as comparações entre eles.

No capítulo 8 são apresentados os resultados dos reconhecedores implementados usando " Multilayer Perceptron ".

No capítulo 9 são mostrados os sistemas híbridos implementados e os resultados obtidos com os mesmos.

No capítulo 10 são discutidas as conclusões deste trabalho.

No apêndice A descrevem-se a base de dados, o vocabulário e o ambiente de simulação utilizados para implementar os reconhecedores.

Como contribuição desta tese, deve-se enfatizar que foram implementadas e comparadas diversas técnicas para reconhecimento de fala usando a mesma base de dados e o mesmo ambiente para simulação. A técnicas testadas foram: Modelos Ocultos de Markov (discreto, contínuo e semicontínuo), " Multilayer Perceptron " e Sistemas Híbridos. Além disso, foram discutidos diferentes algoritmos propostos para a implementação de cada fase do reconhecedor de palavras isoladas, sendo avaliados 22 tipos de parâmetros para representar o sinal de fala usando HMM e Redes Neurais. Também são propostas algumas modificações nos algoritmos de detecção de início/fim de forma a melhorar o desempenho dos mesmos. Os resultados dessas comparações tornam mais fácil a escolha do melhor algoritmo para uma dada aplicação.

Outro ponto a ser ressaltado é que fez parte do trabalho o desenvolvimento de todo o software utilizado, incluindo o desenvolvimento da base de dados. Este trabalho serviu de suporte à realização de uma série de outros trabalhos em reconhecimento de fala junto ao Grupo de Processamento Digital de Fala do Departamento de Comunicações da Faculdade de Engenharia Elétrica e de Computação da UNICAMP.

Capítulo 2

SISTEMAS DE RECONHECIMENTO DE FALA

2.1 Introdução

De uma forma geral, os reconhecedores de fala podem ser divididos em três grandes classes de acordo com a técnica utilizada para reconhecimento [8]: reconhecedores por comparação de padrões, reconhecedores baseados na análise acústico-fonética e reconhecedores empregando inteligência artificial.

Os reconhecedores por comparação de padrões são baseados na filosofia de que os sistemas foram treinados para reconhecer determinados padrões [8]. Os sistemas que utilizam esse método apresentam duas fases distintas: treinamento e reconhecimento. Na fase de treinamento, exemplos dos padrões a serem reconhecidos são apresentados ao sistema para que sejam criados representantes para cada um dos padrões (padrões de referência). Na fase de reconhecimento, compara-se o padrão desconhecido com os padrões de referência e calcula-se uma medida de similaridade. O padrão que melhor corresponder ao padrão desconhecido é escolhido como o padrão reconhecido. Independentemente do tipo de padrão a ser reconhecido é necessário a apresentação de uma quantidade suficiente de material para um bom treinamento do sistema. Como exemplos de sistemas pertencentes a essa classe podem ser citados os reconhecedores que utilizam Modelos Ocultos de Markov (HMM).

Em outra classe, estão os sistemas baseados na análise acústico-fonética, os quais decodificam o sinal de fala baseados nas características acústicas do mesmo e nas relações entre essas características [27]. A idéia consiste em identificar as uni-

dades fonéticas que compõem a fala a ser reconhecida, e a partir da concatenação dessas unidades, reconhecer a fala. Neste tipo de análise é importante considerar propriedades acústicas dos sons que sejam invariantes. Entre essas características do sinal de fala podem ser destacadas a classificação sonoro/não sonoro, presença de ressonância nasal, localização dos formantes e outras. Basicamente, um reconhecedor de fala usando a análise acústico-fonética apresenta as seguintes fases: análise espectral, detecção das características que descrevem as unidades fonéticas (formantes, pitch, sonoro/não sonoro e outras), segmentação do sinal de fala e identificação das unidades fonéticas e escolha da palavra que melhor corresponde à seqüência de unidades. A fase mais importante é a fase de segmentação e identificação das unidades fonéticas.

Pode-se considerar uma terceira classe, onde estão os sistemas que utilizam inteligência artificial. Esses reconhecedores exploram conceitos presentes nas duas classes apresentadas anteriormente [8]. De uma forma geral, Redes Neurais [28] podem ser enquadradas nesta classe. Considerando-se as redes " Multilayer Perceptron " [28], tem-se uma matriz de ponderação que representa as conexões entre os nós da rede, e cada saída da rede está associada a uma unidade (por exemplo, palavra) a ser reconhecida.

Até o momento, os sistemas usando a técnica de comparação de padrões apresentaram os melhores resultados. Além disso, esses sistemas apresentam outras vantagens como simplicidade de uso, facilidade de entendimento, boas justificativas matemáticas e robustez para as diferentes características do sinal de fala, vocabulários e locutores [8].

2.2 Tipos de Reconhecedores de Fala

Os sistemas de reconhecimento de fala podem ser classificados de várias formas, de acordo com o tipo de enunciação e o grau de dependência do locutor [5].

Considerando o tipo de enunicação os sistemas podem ser:

- Reconhecedor de Palavras Isoladas: cada palavra é falada de forma isolada.
- Reconhecedor de Palavras Conectadas: o padrão a ser reconhecido é uma seqüência de palavras pertencentes a um vocabulário restrito e faladas de forma contínua. O reconhecimento é feito usando padrões de referência para cada palavra.
- Reconhecedor de Fala Contínua: neste caso, os padrões a serem reconhecidos

são sentenças ou frases. Esses sistemas geralmente envolvem o reconhecimento de unidades básicas como fones, difones e outros. Isso pode implicar na necessidade de segmentação do sinal de fala.

O reconhecimento de fala contínua é uma tarefa muito mais difícil que o reconhecimento de palavras isoladas ou conectadas pois o início e fim de cada palavra não são claros devido aos efeitos da coarticulação.

De acordo com o grau de dependência do locutor os sistemas podem ser classificados em [5]:

- Dependente do Locutor: o sistema reconhece apenas a fala dos locutores para os quais foi treinado.
- Independente do Locutor: o sistema reconhece a fala de qualquer locutor.

Um outro aspecto que deve ser considerado na classificação dos reconhecedores de fala é o tamanho do vocabulário. Existem quatro tipos distintos de vocabulários que são [27]:

- Vocabulário Pequeno: 1 a 20 palavras.
- Vocabulário Médio: 20 a 100 palavras.
- Vocabulário Grande: 100 a 1000 palavras.
- Vocabulário Muito Grande: mais de 1000 palavras.

2.3 Reconhecedor de Fala por Comparação de Padrões

Levando-se em conta que os sistemas que apresentam os melhores resultados em reconhecimento de fala utilizam a técnica de comparação de padrões, os reconhecedores implementados neste trabalho usam essa técnica. Assim, será apresentada a seguir apenas a descrição desse tipo de reconhecedor de fala.

A estrutura básica de um sistema de reconhecimento de fala usando o método de reconhecimento de padrões é mostrada na figura 2.1 [8].

Nesse sistema, as caracteríticas do sinal de fala são comparadas com padrões previamente armazenados e o padrão mais próximo do sinal de entrada é escolhido, sujeito a algumas restrições do pós-processador (como gramática). Cada padrão pode representar uma sentença, palavra ou uma unidade menor tal como o fone.

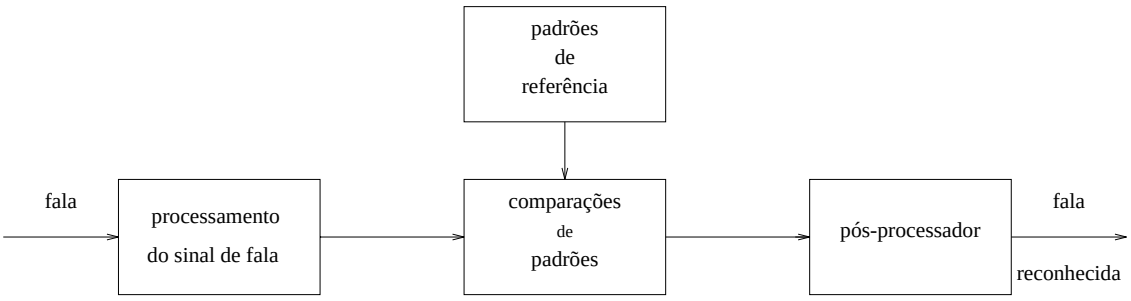

Figura 2.1: Sistema básico de reconhecimento de fala baseado na comparação de padrões.

Neste último caso, cada palavra pode ser representada pela concatenação dos fones de acordo com a transcrição ortográfico-fonética dessas palavras.

O uso de modelos representando palavras é o mais indicado para vocabulários pequenos pois a quantidade de cálculos necessários aumenta à medida que o tamanho do vocabulário torna-se maior. Adicionalmente, quando palavras são usadas como referência, os problemas de coarticulação são evitados, o que leva a altas taxas de reconhecimento.

Para grandes vocabulários e para reconhecimento de fala contínua o uso de unidades menores que palavras é mais apropriado. Uma outra vantagem do uso de unidades sub-lexicais está no fato de que o vocabulário pode ser aumentado ou alterado apenas pela definição da representação fonética das novas palavras.

Analisando a figura 2.1 vemos que o sistema de reconhecimento de fala por comparação de padrões é composto pelas seguintes blocos: processamento do sinal de fala, padrões de referência, comparação de padrões e pós-processador.

2.3.1 Processamento do Sinal de Fala

Nesse bloco o sinal de fala é digitalizado e convertido em um conjunto de parâmetros espectrais e temporais para que seja possível a comparação com os diferentes padrões. O uso de parâmetros deve-se ao fato de que comparações entre formas de onda de sinais de fala são muito complicadas e não relevantes, como por exemplo, distorção de fase altera a forma de onda mas o ouvido humano é insensível a essa distorção.

Para a representação do sinal de fala, um grande número de parâmetros tem sido proposto. Os parâmetros mais usados são os derivados dos coeficientes LPC e os derivados diretamente do espectro do sinal. Para escolher o melhor conjunto de coeficientes pode-se avaliar o desempenho do sistema para cada conjunto.

No caso dos reconhecedores de palavras isoladas é necessária a detecção dos

pontos limitantes de cada palavra. Para isso existem vários algoritmos de detecção de início/fim, os quais utilizam parâmetros como energia e taxa de cruzamento de zero para separar o sinal de fala do ruído.

2.3.2 Padrões de Referência

Os padrões de referência são obtidos a partir de exemplos das unidades a serem reconhecidas. Esse processo é conhecido como treinamento. Para reconhecedores independente do locutor, é necessária a apresentação de vários exemplos de cada unidade, faladas por diferentes locutores e considerando os diferentes sotaques. Dessa forma, conseguem-se sistemas de reconhecimento de fala mais robustos.

Basicamente, dois tipos de padrão são usados como referência [27]. Tem-se o tipo de padrão conhecido como padrão de referência não paramétrico, o qual pode ser algum exemplo do padrão a ser reconhecido ou um padrão médio obtido a partir dos exemplos do padrão a ser reconhecido [27]. O outro tipo, conhecido como modelo estatístico, faz um modelamento estatístico das características dos exemplares do padrão. Como exemplo deste último tipo têm-se os Modelos Ocultos de Markov [8].

Quando o método de Modelos Ocultos de Markov (HMM) é utilizado para criar os padrões de referência, cada padrão é representado por uma rede com N estados, caracterizada por uma função de probabilidade de transição entre estados e um conjunto de funções de probabalidade de símbolos de saída.

2.3.3 Comparação de Padrões

O conjunto de parâmetros representando a fala desconhecida é comparado com os diversos padrões de referência. Esse conjunto de parâmetros é de mesma natureza daquelas usandos durante o treinamento.

Quando são usados padrões de referência não paramétricos, para que seja possível a comparação, é necessário o uso de uma medida de distância e um procedimento de alinhamento temporal. O procedimento mais usado é o " Dynamic Time Warping " (DTW) [8], o qual expande ou comprime o eixo temporal usando técnicas de programação dinâmica. Como resultado dessa comparação, obtém-se as distâncias entre o padrão de entrada e cada um dos padrões de referência.

Quando Modelos Ocultos de Markov são usados para gerar os padrões de referência, o resultado da comparação entre os padrões é a probabilidade de que cada modelo de referência tenha gerado o conjunto de parâmetros de entrada. Para o cálculo das probabilidades, o HMM realiza um alinhamento temporal.

2.3.4 Pós-Processador

Nesta fase, as distâncias ou probabilidades obtidas na comparação com os padrões de referência são usadas para escolher o padrão que melhor corresponde ao padrão desconhecido. Para auxiliar na escolha do melhor candidato pode-se usar restrições sintáticas e semânticas (por exemplo o uso de uma gramática). Dessa forma, candidatos não razoáveis são eliminados.

2.4 Avaliação de Reconhecedores de Fala

O desempenho de um reconhecedor de fala depende de muitos fatores e pode ser avaliado em testes de campo ou em testes em laboratórios. Os testes de campo são realizados em condições mais realistas enquanto os testes em laboratórios são mais baratos e podem ser reproduzidos mais facilmente.

Os fatores que influenciam o desempenho de um reconhecedor de fala podem ser divididos em vários grupos, os quais são [27]:

- Tipo de enunciação: isolada, conectada ou contínua.
- Dependência do locutor: dependende ou independente.
- Vocabulário: tamanho do vocabulário e semelhança entre as palavras que compõem o mesmo.
- Ambiente de gravação: sala quieta, escritório, lugar público (ruas).
- Sistema de transmissão: tipo de microfone, telefone.

Para avaliar o desempenho de um reconhecedor de palavras isoladas, as seguintes medidas podem ser usadas [27]:

- Porcentagem de acerto: porcentagem de palavras que foram reconhecidas corretamente.
- Porcentagem de rejeição: porcentagem de palavras que pertencem ao vocabulário mas foram rejeitadas erroneamente.
- Porcentagem de erro: porcentagem de palavras que foram reconhecidas incorretamente.

Para o caso de fala contínua as seguintes medidas também devem ser consideradas:

- Porcentagem de inserção: porcentagem de palavras extras inseridas na sentença reconhecida.
- Porcentagem de omissão: porcentagem de palavras corretas omitidas na sentença reconhecida.
- Porcentagem de substitiuição: porcentagem de palavras corretas substituidas por palavras incorretas na sentença reconhecida.

Para avaliar os reconhecedores implementados neste trabalho, foram consideradas apenas as porcentagens de acerto, sendo que não foi considerada a possibilidade de rejeição de palavras.

Capítulo 3

MODELOS OCULTOS DE MARKOV

3.1 Introdução

A teoria de Modelos Ocultos de Markov (" Hidden Markov Models " (HMM)) foi introduzida por Baum no final da década de 60 e foi usada pela primeira vez para reconhecimento de fala por Baker e Jelinek no início dos anos 70.

Modelo Oculto de Markov é um poderoso método estatístico que pode ser usado para modelar sinais como um processo aleatório paramétrico [29, 30, 31]. Para estimar os parâmetros dos HMMs existem eficientes algoritmos .

Em reconhecimento de fala, HMMs podem ser usados para representar palavras, sentenças ou unidades menores tais como fones.

Atualmente, HMMs são uma ferramenta muito útil e importante para reconhecimento de fala e bons resultados têm sido encontrados usando esse método.

3.2 Modelos Ocultos de Markov

3.2.1 Definição

Um HMM é um conjunto de estados S conectados por transições. Mudanças entre os estados ocorrem de acordo com uma função de probabilidade (probabilidades de transição). Associada a cada estado ou a cada transição existe uma função de probabilidade dos símbolos de saída, a qual define a probabilidade de emissão de cada símbolo de um alfabeto. Esse alfabeto pode ser finito (HMM discreto) ou

infinito (HMM contínuo). Neste trabalho, considerou-se apenas o caso em que a função de probabilidade dos símbolos de saída está associada a cada estado.

Em cada instante de tempo t existe uma mudança de estado (pode ser para o mesmo estado) e um símbolo é emitido. A seqüência de símbolos emitidos, chamada de seqüência de observação, é a saída do HMM. A seqüência de estados não pode ser observada, é oculta.

Portanto, um HMM é caracterizado por [32, 8]:

- Conjunto de estados $S = \{S_1, \ldots, S_N\}$, onde N é o número de estados.
- Função de probabilidade do estado inicial $\pi = \{\pi_i\}$

$$\pi_i = P\left[q_1 = S_i\right] \quad 1 \leq i \leq N \tag{3.1}$$

  onde q_1 é o estado inicial $(t = 1)$
- Função de probabilidade de transição A
- Função de probabilidade dos símbolos de saída B.

Considerando somente processos nos quais as probabilidades de transição são independentes do tempo e os HMMs são de primeira ordem, o conjunto de probabilidades de transição A é definido por:

$$A = \{a_{ij}\} \tag{3.2}$$

$$a_{ij} = P\left[q_t = S_j | q_{t-1} = S_i\right] \quad 1 \leq i, j \leq N \tag{3.3}$$

onde a_{ij} é a probabilidade de ocorrer uma transição do estado S_i para o estado S_j.

Os coeficientes a_{ij} devem obedecer às seguintes regras:

$$a_{ij} \geq 0 \qquad 1 \leq i, j \leq N \tag{3.4}$$

$$\sum_{j=1}^{N} a_{ij} = 1 \quad 1 \leq i \leq N \tag{3.5}$$

A probabilidade de estar no estado S_j no instante de tempo t depende somente do instante de tempo $t - 1$.

3.2.2 Tipos de HMM quanto à natureza da Função de Probabilidade dos Símbolos de Saída

Dependendo do tipo da função de probabilidade dos símbolos de saída o HMM pode ser discreto ou contínuo.

- **HMM Discreto**:

  No tipo discreto [8], o número de possíveis símbolos de saída é finito (K). A probabilidade de emitir o símbolo V_k no estado S_i é dada por $b_i(k)$.

  A função de probabilidade B tem as seguintes propriedades:

$$b_i(k) \geq 0 \qquad \begin{matrix} 1 \leq i \leq N \\ 1 \leq k \leq K \end{matrix} \tag{3.6}$$

$$\sum_{k=1}^{K} b_i(k) = 1 \quad 1 \leq i \leq N \tag{3.7}$$

- **HMM Contínuo**:

  No tipo contínuo, a função densidade de probabilidade é contínua . Usualmente, uma função densidade elipticamente simétrica ou log-côncava é usada, como por exemplo, função densidade de probabilidade Gaussiana [8, 33, 34, 35].

  A função densidade de probabilidade usada neste trabalho é uma mistura finita da forma [8]:

$$b_j(O_t) = \sum_{m=1}^{M} c_{jm} G(O_t, \mu_{jm}, U_{jm}) \quad 1 \leq j \leq N \tag{3.8}$$

  onde:
  O_t é o vetor de entrada
  M é o número de misturas
  c_{jm} é o coeficiente da m-ésima mistura no estado S_j
  G é uma função densidade de probabilidade Gaussiana multidimensional com vetor média μ_{jm} e matriz covariância U_{jm}

  A função densidade de probabilidade Gaussiana multidimensional é dada por [36]:

$$G(O_t, \mu_{jm}, U_{jm}) = \frac{1}{(2\pi)^{dim/2} |U_{jm}|^{1/2}} \exp\{-(O_t - \mu_{jm}) U_{jm}^{-1} (O_t - \mu_{jm})'/2\} \tag{3.9}$$

  onde:
  dim é a dimensão do vetor O_t

$|U_{jm}|$ é o determinante da matriz covariância U_{jm}
U_{jm}^{-1} é a matriz covariância inversa

Os coeficientes c_{jm} das misturas e a função densidade de probabilidade devem satisfazer as regras:

$$\sum_{m=1}^{M} c_{jm} = 1 \quad 1 \leq j \leq N \tag{3.10}$$

$$c_{jm} \geq 0 \qquad \begin{matrix} 1 \leq j \leq N \\ 1 \leq m \leq M \end{matrix} \tag{3.11}$$

$$\int_{-\infty}^{+\infty} b_j(x)dx = 1 \quad 1 \leq j \leq N \tag{3.12}$$

Quando as misturas (vetor média μ_{jm} e matriz covariância U_{jm}) são iguais para todos os estados o HMM contínuo é denominado HMM " tied " contínuo [8].

- **HMM Semicontínuo**:

  HMM semicontínuo [37, 38, 39] é um modelo intermediário entre os HMMs discreto e contínuo.

  Neste caso, a probabilidade de emissão dos símbolos de saída é dada por:

$$b_j(O_t) = \sum_{V_k \in \eta(O_t)} c_j(k) f(O_t|V_k) \quad 1 \leq j \leq N \tag{3.13}$$

  onde:
  O_t é o vetor de entrada
  $\eta(O_t)$ é o conjunto das funções densidade de probabilidade que apresentam os M maiores valores de $f(O_t|V_k)$, $1 \leq M \leq K$
  K é o número de funções densidade de probabilidade (símbolos de saída)
  V_k é o k-ésimo símbolo de saída
  $c_j(k)$ é a probabilidade de emissão do símbolo V_k no estado S_j
  $f(O_t|V_k)$ é o valor da k-ésima função densidade de probabilidade

  No HMM semicontínuo, o conjunto de funções densidade de probabilidade é o mesmo para todos os estados e todos os modelos. O que muda são os coeficientes $c_j(k)$. Os parâmetros das funções densidade de probabilidade, coeficientes das mesmas e probabilidades de transição são calculados e otimizados conjuntamente.

Quando o valor de M é igual a 1, o HMM semicontínuo torna-se um HMM discreto com um " codebook " formado por funções densidade de probabilidade. Neste caso, usa-se apenas a função $f(O_t|V_k)$ que apresentar maior valor para calcular a probabilidade de emissão de símbolos de saída:

$$b_j(O_t) = c_j(k) f(O_t|V_k) \tag{3.14}$$

onde V_k é o símbolo de saída que assegura o maior valor de $f(O_t|V_k)$.

Quando o valor de M é igual a K pode-se considerar o HMM semicontínuo como um HMM contínuo em que todas as misturas (vetor média μ_{jm} e matriz covariância U_{jm}) são iguais para todos os estados e todos os modelos. O que varia de um estado para outro são os valores dos coeficientes das misturas. Portanto, a probabilidade dos símbolos de saída é dada por:

$$b_j(O_t) = \sum_{k=1}^{K} c_j(k) f(O_t|V_k) \quad 1 \leq j \leq N \tag{3.15}$$

onde:
O_t é o vetor de entrada
K é o número de misturas
$c_j(k)$ é a probabilidade de ocorrência da k-ésima mistura no estado S_j
$f(O_t|V_k)$ é o valor da k-ésima mistura

Neste trabalho, foi considerado apenas o caso em que M é igual a K. Além disso, representou-se a função $f(O_t|V_k)$ por uma função de densidade de probabilidade Gaussiana com vetor média μ_k e matriz covariância U_k. Dessa forma, tem-se:

$$f(O_t|V_k) = \frac{1}{(2\pi)^{dim/2}|U_k|^{1/2}} \exp\{-(O_t - \mu_k)U_k^{-1}(O_t - \mu_k)'/2\} \tag{3.16}$$

onde:
dim é a dimensão do vetor O_t
$|U_k|$ é o determinante da matriz covariância U_k
U_k^{-1} é a matriz covariância inversa

O HMM semicontínuo implementado neste trabalho é um variação do HMM semicontínuo pois primeiramente calculou-se os parâmetros das misturas (vetor média μ_{jm} e matriz covariância U_{jm}) e depois foram calculados os coeficientes das misturas e as probabilidades de transição. Dessa forma, todos os parâmetros não foram otimizados conjuntamente.

3.2.3 Topologia

Com relação à topologia, existem diferentes tipos de HMM, tais como ergódico, " left-right " e outros. Para aplicações em reconhecimento de fala, o modelo " left-right " ou modelo de Bakis é o mais usado [8]. Esse modelo está ilustrado na figura 3.1.

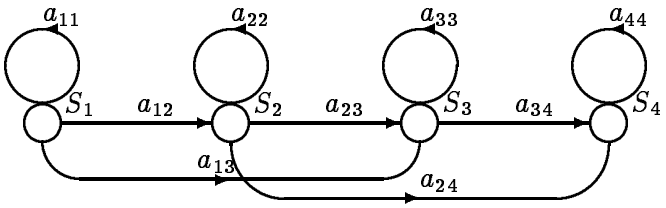

Figura 3.1: Modelo " left-right "

Nesse modelo, à medida que o tempo aumenta, os índices dos estados aumentam ou permanecem os mesmos. Portanto, os coeficientes a_{ij} apresentam a seguinte propriedade:

$$a_{ij} = 0, \quad i > j \tag{3.17}$$

A função de probabilidade do estado inicial é dada por:

$$\pi_i = \begin{cases} 1 & i = 1 \\ 0 & i \neq 1 \end{cases} \tag{3.18}$$

Assim, a seqüência de estados no modelo " left-right " começa sempre no estado 1.

Para evitar grandes mudanças nos índices dos estados, são usadas algumas restrições como:

$$a_{ij} = 0, \quad j > i + \Delta \tag{3.19}$$

No exemplo apresentado na figura 3.1, o valor de Δ é igual a 2.

Neste trabalho utilizou-se apenas o modelo " left-right " e dessa forma não serão apresentadas posteriormente as fórmulas para reestimação da probabilidade do estado inicial.

3.2.4 Três Problemas do HMM

Dado um HMM, existem três problemas básicos de interesse [8]:

1. Dados uma seqüência de observação e um modelo, qual a probabilidade de que esse modelo tenha gerado a seqüência de observação ?

2. Dados uma seqüência de observação e um modelo, qual é a seqüência de estados ótima ?

3. Dada uma seqüência de observação, como podem ser estimados os parâmetros do modelo que gerou essa seqüência ?

Resolvendo-se o problema 1, é possível escolher entre vários modelos, o modelo que apresenta a maior probabilidade de ter gerado a seqüência observada.

Com a solução do problema 2, obtém-se a seqüência de estados mais provável.

Solucionando-se o problema 3, podem ser estimados os parâmetros do modelo que melhor descreve a seqüência de observação dada. A seqüência de observação utilizada para estimar os parâmetros do modelo é chamada de seqüência de treinamento.

3.2.5 Soluções para os Três Problemas Básicos

Para a solução do problema 1 pode-se usar o algoritmo " Forward " ou o algoritmo de Viterbi.

A solução do problema 2 é encontrada usando-se o algoritmo de Viterbi.

O problema 3 pode ser resolvido usando-se o algoritmo " Forward-Backward " ou o procedimento " K-means " juntamente com o algoritmo de Viterbi.

Algoritmo " Forward "

Dada uma seqüência de observação $O = \{O_1, \ldots, O_T\}$, com duração T, e um modelo λ de N estados, deseja-se calcular a probabilidade de ocorrer a seqüência O dado o modelo λ, $P(O|\lambda)$. Isso pode ser feito usando-se o Algoritmo " Forward " [8], o qual está descrito a seguir:

inicialização:

$$\alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \leq i \leq N \tag{3.20}$$

indução:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}) \qquad \begin{array}{l} 1 \leq t \leq T-1 \\ 1 \leq j \leq N \end{array} \tag{3.21}$$

término:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{3.22}$$

A variável $\alpha_t(i)$ é chamada variável " forward " e é definida por:

$$\alpha_t(i) = P\left[O_1 O_2 \ldots O_t, q_t = S_i | \lambda\right] \tag{3.23}$$

Algoritmo de Viterbi

A seqüência de estados ótima para uma dada seqüência de observação pode ser encontrada usando-se o algoritmo de Viterbi [8], que será apresentado em seguida: inicialização:

$$\delta_1(i) = \pi_i b_i(O_1) \quad 1 \leq i \leq N \tag{3.24}$$

$$\psi_1(i) = 0 \tag{3.25}$$

recursão:

$$\delta_t(j) = \max_{1 \leq i \leq N} \left[\delta_{t-1}(i) a_{ij}\right] b_j(O_t) \quad \begin{matrix} 2 \leq t \leq T \\ 1 \leq j \leq N \end{matrix} \tag{3.26}$$

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} \left[\delta_{t-1}(i) a_{ij}\right] \quad \begin{matrix} 2 \leq t \leq T \\ 1 \leq j \leq N \end{matrix} \tag{3.27}$$

término:

$$P^* = \max_{1 \leq i \leq N} \delta_T(i) \tag{3.28}$$

$$q_T^* = \arg \max_{1 \leq i \leq N} \delta_T(i) \tag{3.29}$$

seqüência de estados ótima:

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \quad t = T-1, T-2, \ldots, 1 \tag{3.30}$$

Algoritmo " Forward-Backward "

A solução do problema 3 é dada pelo seguinte algoritmo, o qual é conhecido como Algoritmo " Forward-Backward " ou Algoritmo Baum-Welch [8].

1. Forneça um conjunto inicial de parâmetros $\{A, B, \pi\}$.
2. Calcule $\bar{A}$, $\bar{B}$ de acordo com as fórmulas de reestimação.

3. Faça A igual $\bar{A}$ e B igual $\bar{B}$.

4. Se não ocorrer convergência, volte ao passo 2.

Antes da apresentação das fórmulas de reestimação, é necessário a definição da variável " backward " $\beta_t(i)$:

$$\beta_t(i) = P\left[O_{t+1}O_{t+2}\ldots O_T | q_t = S_i, \lambda\right] \tag{3.31}$$

inicialização:

$$\beta_T(i) = 1 \quad 1 \leq i \leq N \tag{3.32}$$

indução:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \quad \begin{array}{l} t = T-1, T-2, \ldots, 1 \\ 1 \leq i \leq N \end{array} \tag{3.33}$$

As fórmulas para a reestimação das probabilidades de transição e da função de probabilidade dos símbolos de saída são:

- **Probabilidade de Transição a_{ij}:**

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i) \beta_t(i)} \tag{3.34}$$

- **Função de Probabilidade dos símbolos de saída:**

  HMM Discreto:

$$\bar{b}_i(k) = \frac{\sum_{\substack{t=1 \\ O_t = V_k}}^{T} \alpha_t(i) \beta_t(i)}{\sum_{t=1}^{T} \alpha_t(i) \beta_t(i)} \tag{3.35}$$

  HMM Contínuo:

  As fórmulas de reestimação para c_{jm}, μ_{jm} e U_{jm} são as seguintes:

$$\bar{c}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,m)}{\sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_t(j,m)} \tag{3.36}$$

$$\bar{\mu}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,m) O_t}{\sum_{t=1}^{T} \gamma_t(j,m)} \tag{3.37}$$

$$\bar{U}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,m)(O_t - \mu_{jm})(O_t - \mu_{jm})'}{\sum_{t=1}^{T} \gamma_t(j,m)} \tag{3.38}$$

$$\gamma_t(j,m) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{k=1}^{N} \alpha_t(k)\beta_t(k)} \right] \left[\frac{c_{jm} G(O_t, \mu_{jm}, U_{jm})}{b_j(O_t)} \right] \tag{3.39}$$

HMM Semicontínuo:

As fórmulas de reestimação para c_{jm}, μ_m e U_m são as seguintes:

$$\bar{c}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,m)}{\sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_t(j,m)} \tag{3.40}$$

$$\bar{\mu}_m = \frac{\sum_{t=1}^{T} \gamma_t(m) O_t}{\sum_{t=1}^{T} \gamma_t(m)} \tag{3.41}$$

$$\bar{U}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(m)(O_t - \mu_m)(O_t - \mu_m)'}{\sum_{t=1}^{T} \gamma_t(m)} \tag{3.42}$$

$$\gamma_t(m) = \sum_{j=1}^{N} \gamma_t(j,m) \tag{3.43}$$

$$\gamma_t(j,m) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{k=1}^{N} \alpha_t(k)\beta_t(k)} \right] \left[\frac{c_{jm} G(O_t, \mu_m, U_m)}{b_j(O_t)} \right] \tag{3.44}$$

PROCEDIMENTO " K-MEANS "

Uma outra forma de solucionar o problema 3 consiste em usar o algoritmo de Viterbi em conjunto com o procedimento " K-means " [8]. Este procedimento apresenta os mesmos passos que o algoritmo " Forward-Backward " descrito anteriormente. A diferença está no modo de calcular os parâmetros do modelo. Dado um modelo, o algoritmo de Viterbi é usado para segmentar cada observação em uma seqüência de estados. Assim, é possível conhecer os símbolos de saída que ocorreram em cada estado.

Os parâmetros do modelo são calculados da seguinte forma:

- **Probabilidades de Transição a_{ij}:**

  a_{ij}: número de transições do estado S_i para o estado S_j dividido pelo número de transições do estado S_i para todos os estados.

- **Função de Probabilidade dos símbolos de saída:**

  HMM Discreto:

  $b_j(k)$: número de vetores com índice k no estado S_j divido pelo número de vetores no estado S_j.

  HMM Contínuo:

  Nesse caso, o procedimento K-means é usado para classificar os vetores de observação em cada estado em M clusters (M é o número de misturas). A distância Euclidiana pode ser usada como medida de distorção.

  c_{jm}: número de vetores no cluster m do estado S_j dividido pelo número de vetores no estado S_j

  μ_{jm}: vetor médio dos vetores classificados no cluster m do estado S_j

  U_{jm}: matriz covariância dos vetores classificados no cluster m do estado S_j

  HMM Semicontínuo:

  O procedimento " K-means " é usado da mesma forma que o HMM contínuo.

  c_{jm}: número de vetores no cluster m e no estado S_j dividido pelo número de vetores no estado S_j

  μ_m: vetor média dos vetores classificados no cluster m

  U_m: matriz covariância dos vetores classificados no cluster m

O procedimento para terminar o algoritmo é o mesmo apresentado para o algoritmo " Forward-Backward ".

3.2.6 Considerando Estado Final

A definição de HMM apresentada no item 2.2.1 pode ser alterada para incluir um conjunto S_F de estados finais [10]. Neste caso, algumas fórmulas apresentadas anteriormente precisam ser alteradas. Neste trabalho foram realizados alguns experimentos considerando apenas um estado final (último estado). Assim, serão apresentadas as fórmulas modificadas para esse caso.

Algoritmo " Forward "

$$P(O|\lambda) = \alpha_T(i) \quad i = S_F \tag{3.45}$$

Algoritmo de Viterbi

$$P^* = \delta_T(i) \quad i = S_F \tag{3.46}$$

$$q_T^* = \delta_T(i) \quad i = S_F \tag{3.47}$$

Algoritmo " Forward-Backward "

$$\beta_T(i) = \begin{cases} 1 & i = S_F \\ 0 & i \neq S_F \end{cases} \tag{3.48}$$

3.2.7 Normalização

Devido ao fato de que os coeficientes a_{ij} e $b_i(k)$ são menores que 1, à medida que o instante de tempo t torna-se grande, a variável $\alpha_t(i)$ aproxima-se de zero. Para grandes valores de t, o valor de $\alpha_t(i)$ excederá a faixa de precisão de qualquer computador causando underflow. Para evitar isso, é necessário o uso de um fator de normalização[8].

O procedimento de normalização consiste na multiplicação de $\alpha_t(i)$ e $\beta_t(i)$ por um fator que é independente de i. O fator de normalização c_t e os parâmetros normalizados são dados pelas seguintes fórmulas:

$$\tilde{\alpha}_1(i) = \alpha_1(i) \tag{3.49}$$

$$\hat{c}_1 = \frac{1}{\sum_{i=1}^{N} \alpha_1(i)} \tag{3.50}$$

$$\hat{\alpha}_1(i) = \tilde{\alpha}_1(i) \cdot \hat{c}_1 \tag{3.51}$$

$$\tilde{\alpha}_{t+1}(j) = \left[\sum_{i=1}^{N} \hat{\alpha}_t(i) a_{ij}\right] b_j(O_{t+1}) \quad \begin{matrix} 1 \leq t \leq T-1 \\ 1 \leq j \leq N \end{matrix} \tag{3.52}$$

$$\hat{c}_t = \frac{1}{\sum_{i=1}^{N} \tilde{\alpha}_t(i)} \tag{3.53}$$

$$\hat{\alpha}_t(i) = \tilde{\alpha}_t(i) \cdot \hat{c}_t \tag{3.54}$$

$$\hat{\beta}_t(i) = \tilde{\beta}_t(i) \cdot \hat{c}_t \tag{3.55}$$

$$c_t = \prod_{s=1}^{t} \hat{c}_s \tag{3.56}$$

$$\hat{\alpha}_t(i) = \alpha_t(i) \cdot c_t \tag{3.57}$$

$$\hat{\beta}_t(i) = \beta_t(i) \cdot c_t \tag{3.58}$$

Levando-se em conta o fator de normalização, o procedimento para calcular $P(O|\lambda)$ é modificado. Calcula-se $\log(P(O|\lambda))$ em vez de $P(O|\lambda)$, o qual é dado pela seguinte fórmula:

$$\log(P(O|\lambda)) = -\sum_{t=1}^{T} \log \hat{c}_t \tag{3.59}$$

Quando o algoritmo de Viterbi é utilizado, o underflow é evitado trabalhando-se com o logaritmo das probabilidades.

3.2.8 Seqüência de Treinamento com Múltiplas Observações

Para a obtenção de uma boa estimativa dos parâmetros do modelo, uma seqüência com uma única observação não é suficiente. Assim, seqüências com múltiplas observações devem ser usadas. Em reconhecimento de fala, seqüência de treinamento com múltiplas observações significa que a seqüência de treinamento é composta por mais de uma observação da mesma palavra .

Quando um HMM é treinado usando seqüências com múltiplas observações, os parâmetros do modelo são reestimados após a apresentação de todas as observações.

Todas as fórmulas apresentadas até aqui são para seqüências com uma única observação. Quando seqüências com múltiplas observações (D observações) são empregadas, as seguintes fórmulas devem ser usadas:

- **Probabilidades de Transição a_{ij}:**

$$\bar{a}_{ij} = \frac{\sum_{d=1}^{D}\sum_{t=1}^{T_d-1} \hat{\alpha}_t^d(i) a_{ij} b_j(O_{t+1}^d) \hat{\beta}_{t+1}^d(j)}{\sum_{d=1}^{D}\sum_{t=1}^{T_d-1} \hat{\alpha}_t^d(i)\hat{\beta}_t^d(i)/\hat{c}_t^d} \tag{3.60}$$

- **Função de Probabilidade dos símbolos de saída:**

  HMM Discreto:

$$\bar{b}_i(k) = \frac{\sum_{d=1}^{D}\sum_{\substack{t=1 \\ O_t = V_k}}^{T_d} \hat{\alpha}_t^d(i)\hat{\beta}_t^d(i)/\hat{c}_t^d}{\sum_{d=1}^{D}\sum_{t=1}^{T_d} \hat{\alpha}_t^d(i)\hat{\beta}_t^d(i)/\hat{c}_t^d} \tag{3.61}$$

  HMM Contínuo:

$$\bar{c}_{jm} = \frac{\sum_{d=1}^{D}\sum_{t=1}^{T_d} \hat{\alpha}_t^d(j)\hat{\beta}_t^d(j) N_t^d(j,m)/\hat{c}_t^d}{\sum_{d=1}^{D}\sum_{t=1}^{T_d} \hat{\alpha}_t^d(j)\hat{\beta}_t^d(j)/\hat{c}_t^d} \tag{3.62}$$

$$\bar{\mu}_{jm} = \frac{\sum_{d=1}^{D}\sum_{t=1}^{T_d} \hat{\alpha}_t^d(j)\hat{\beta}_t^d(j) N_t^d(j,m) O_t^d/\hat{c}_t^d}{\sum_{d=1}^{D}\sum_{t=1}^{T_d} \hat{\alpha}_t^d(j)\hat{\beta}_t^d(j) N_t^d(j,m)/\hat{c}_t^d} \tag{3.63}$$

$$\bar{U}_{jm} = \frac{\sum_{d=1}^{D}\sum_{t=1}^{T_d} \hat{\alpha}_t^d(j)\hat{\beta}_t^d(j)N_t^d(j,m)(O_t^d - \mu_{jm})(O_t^d - \mu_{jm})'/\hat{c}_t^d}{\sum_{d=1}^{D}\sum_{t=1}^{T_d} \hat{\alpha}_t^d(j)\hat{\beta}_t^d(j)N_t^d(j,m)/\hat{c}_t^d} \tag{3.64}$$

onde:

D é o número de observações na seqüência de treinamento e

$$N_t(j,m) = \frac{c_{jm}G(O_t, \mu_{jm}, U_{jm})}{\sum_{k=1}^{M} c_{jk}G(O_t, \mu_{jk}, U_{jk})} \tag{3.65}$$

HMM Semicontínuo:

$$\bar{c}_{jm} = \frac{\sum_{d=1}^{D_h}\sum_{t=1}^{T_{d_h}} \hat{\alpha}_t^d(j)\hat{\beta}_t^d(j)N_t^d(j,m)/\hat{c}_t^d}{\sum_{d=1}^{D_h}\sum_{t=1}^{T_{d_h}} \hat{\alpha}_t^d(j)\hat{\beta}_t^d(j)/\hat{c}_t^d} \tag{3.66}$$

$$\bar{\mu}_m = \frac{\sum_{h=1}^{H}\sum_{d=1}^{D_h}\sum_{t=1}^{T_{d_h}}\sum_{j=1}^{N} \hat{\alpha}_t^d(j)\hat{\beta}_t^d(j)N_t^d(j,m)O_t^d/\hat{c}_t^d}{\sum_{h=1}^{H}\sum_{d=1}^{D_h}\sum_{t=1}^{T_{d_h}}\sum_{j=1}^{N} \hat{\alpha}_t^d(j)\hat{\beta}_t^d(j)N_t^d(j,m)/\hat{c}_t^d} \tag{3.67}$$

$$\bar{U}_m = \frac{\sum_{h=1}^{H}\sum_{d=1}^{D_h}\sum_{t=1}^{T_{d_h}}\sum_{j=1}^{N} \hat{\alpha}_t^d(j)\hat{\beta}_t^d(j)N_t^d(j,m)(O_t^d - \mu_m)(O_t^d - \mu_m)'/\hat{c}_t^d}{\sum_{h=1}^{H}\sum_{d=1}^{D_h}\sum_{t=1}^{T_{d_h}}\sum_{j=1}^{N} \hat{\alpha}_t^d(j)\hat{\beta}_t^d(j)N_t^d(j,m)/\hat{c}_t^d} \tag{3.68}$$

onde:

$$N_t(j,m) = \frac{c_{jm}G(O_t, \mu_m, U_m)}{\sum_{k=1}^{M} c_{jk}G(O_t, \mu_k, U_k)} \tag{3.69}$$

H é o número de modelos e

D_h é o número de observações na seqüência de treinamento do modelo h

3.2.9 Variação do HMM Semicontínuo

Na implementação do HMM semicontínuo neste trabablho, primeiramente foram calculados os parâmetros μ_m e U_m usando as seguintes fórmulas:

$$\bar{\mu}_m = \frac{\sum_{t=1}^{T} N_t^d(m) O_t^d}{\sum_{t=1}^{T} N_t^d(m)} \tag{3.70}$$

$$\bar{U}_m = \frac{\sum_{t=1}^{T} N_t^d(m)(O_t^d - \mu_m)(O_t^d - \mu_m)'}{\sum_{t=1}^{T} N_t^d(m)} \tag{3.71}$$

$$\bar{Pr}_m = \frac{1}{T} \sum_{t=1}^{T} N_t^d(m) \tag{3.72}$$

onde:

$$N_t(m) = \frac{Pr_m G(O_t, \mu_m, U_m)}{\sum_{k=1}^{M} Pr_k G(O_t, \mu_k, U_k)} \tag{3.73}$$

T é o número de vetores considerando as seqüências de treinamento de todos os modelos.

Em uma segunda fase, os parâmetros μ_m e U_m foram mantidos fixos e os coeficientes c_{jm} foram calculados segundo a equação 3.66 e as probabilidades de transição foram obtidas de acordo com a equação 3.60.

3.2.10 Efeitos devido à Seqüência de Treinamento Insuficiente

Freqüentemente o tamanho da seqüência não é suficiente para uma boa estimativa dos parâmetros do modelo. Uma forma de minimizar esse problema é garantir que todos os parâmetros estimados fiquem acima de um determinado limiar [8], com a finalidade de se evitar a ocorrência de probabilidade de emissão de símbolos nula. Os parâmetros que estiverem abaixo desse limiar terão seus valores alterados.

Assim, para HMM discreto usa-se a seguinte regra:

$$b_i(k) = \begin{cases} \delta_b & \text{se } b_i(k) < \delta_b \qquad 1 \leq i \leq N \\ b_i(k) & \text{caso contrário} \qquad 1 \leq k \leq K \end{cases} \tag{3.74}$$

e para HMM contínuo:

$$U_{jm}(r,r) = \begin{cases} \delta_U & \text{se } U_{jm}(r,r) < \delta_U \\ U_{jm}(r,r) & \text{caso contrário} \end{cases} \quad \begin{matrix} 1 \le j \le N \\ 1 \le m \le M \end{matrix} \tag{3.75}$$

$$c_{jm} = \begin{cases} \delta_c & \text{se } c_{jm} < \delta_c \\ c_{jm} & \text{caso contrário} \end{cases} \quad \begin{matrix} 1 \le j \le N \\ 1 \le m \le M \end{matrix} \tag{3.76}$$

Quando estas alterações são realizadas, os outros parâmetros envolvidos devem ser recalculados de forma a garantir as propriedades das funções de probabilidade. No caso do HMM discreto, deve-se recalcular os valores de $b_i(k)$ para satisfazer $\sum_{k=1}^{K} b_i(k) = 1$. Para HMM contínuo e semicontínuo dever-se garantir $\sum_{m=1}^{M} c_{jm} = 1$.

3.2.11 Incorporação da Duração de Estado em um HMM

A probabilidade de ocorrerem d observações consecutivas no estado S_i de um HMM é dada por [8]:

$$P_i(d) = (a_{ii})^{d-1}(1 - a_{ii}) \tag{3.77}$$

Para melhorar o desempenho de um HMM, a duração de estado pode ser modelada explicitamente. Isso pode ser feito modificando-se o algoritmo " Forward-Backward " e estimando-se $P_i(d)$. O esforço computacional requerido para realizar esses cálculos é muito grande e a melhora no desempenho do sistema não é significativa. Devido a isso, outras formas para modelar a duração de estado estão sendo estudadas [40, 32].

Uma maneira de incorporar a duração de estado é o uso de um pós-processador. Neste caso, a duração de estado pode ser calculada a partir da seqüência de treinamento segmentada usando-se o algoritmo de Viterbi. Após a segmentação, conta-se o número de observações em cada estado e as probabilidades são estimadas. No reconhecedor, mede-se a duração de estado usando o algoritmo de Viterbi e a probabilidade de duração de estado é combinada com a probabilidade do algoritmo de Viterbi da seguinte forma [8] :

$$\log P(q, O|\lambda) = \log P^*(q, O|\lambda) + w \sum_{j=1}^{N} \log(P_j(d_j)) \tag{3.78}$$

onde:
w é um fator de ponderação
d_j é o número de observações no estado S_j
N é o número de estados do modelo

$P_j(d_j)$ é a probabilidade de ocorrerem d_j observações no estado S_j
$P^*(q, O|\lambda)$ é a probabilidade obtida no algoritmo de Viterbi.

Para representar a duração de estado podem ser utilizadas funções de probabilidade não paramétricas ou funções de densidade de probabilidade paramétricas como distribuição Gaussiana, distribuição de Laplace, distribuição Gamma ou distribuição de Rayleigh. Essas distribuições são dadas pelas seguintes fórmulas [36]:

- Função Densidade de Probabilidade Gaussiana

$$G(d) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-(d-\overline{d})^2/2\sigma^2\right) \tag{3.79}$$

  onde σ^2 é a variância e $\overline{d}$ é a média.

- Função Densidade de Probabilidade de Laplace

$$G(d) = \frac{1}{\sqrt{2\sigma^2}} \exp\left(-\sqrt{2}|d|/\sigma\right) \tag{3.80}$$

  onde σ^2 é a variância. A variável d apresenta média igual a zero.

- Função Densidade de Probabilidade Gamma

$$G(d) = \frac{1}{\beta^\alpha \Gamma(\alpha)} d^{(\alpha-1)} \exp\left(-d/\beta\right) \tag{3.81}$$

  onde $\beta = \sigma^2/\overline{d}$, $\alpha = (\overline{d})^2/\sigma^2$, σ^2 é a variância e $\overline{d}$ é a média e $\Gamma(\alpha)$ é a função Gamma.

- Função Densidade de Probabilidade de Rayleigh

$$G(d) = \frac{d}{\alpha^2} \exp\left(-d^2/2\alpha^2\right) \tag{3.82}$$

  onde $\alpha = (2/\pi)^{1/2}\overline{d}$.

Não existe uma fórmula para calcular o fator de ponderação w ótimo, e assim esse fator de ponderação é obtido por tentativa e erro.

Neste trabalho é proposto um método para a obtenção do fator de ponderação. Nesse método, cada modelo usa um fator diferente, o qual é obtido empregando-se a seguinte fórmula:

$$w_k = \frac{0.25}{D} \sum_{i=1}^{D} \frac{\log P_i^*(q, O|\lambda)}{\sum_{j=1}^{N} \log(P_{ij}(d_{ij}))} \tag{3.83}$$

onde D é o número de observações na seqüência de treinamento.

Essa fórmula foi obtida a partir da análise dos valores das probabilidades dos HMMs e valores das probabilidades de duração de estado. Usou-se o fator 0.25 de forma que na equação 3.78 o valor de $\log P^*(q, O|\lambda)$ seja o valor dominante no resultado, diminuindo a influência de $w \sum_{j=1}^{N} \log(P_j(d_j))$.

O esforço computacional requerido para incorporar a duração de estado em um HMM usando um pós processador pode ser considerado desprezível.

3.2.12 Inicialização dos Parâmetros de um HMM

O algoritmo " Forward-Backward " não garante a obtenção de um ponto de máximo global. Dessa forma estimativas iniciais ruins podem levar a um ponto de máximo local ruim.

Existem várias maneiras de inicializar um HMM. Para as probabilidades de transição podem-se usar valores aleatórios ou distribuição uniforme como estimativas iniciais. Neste trabalho empregou-se este último, o qual é um método de fácil implementação e bons resultados foram alcançados. Para a estimativa inicial das probabilidades dos símbolos de saída são usados segmentação manual da seqüência de treinamento, procedimento " K-means " com algoritmo de Viterbi e outros.

Para HMMs discretos não é fundamental uma boa escolha dos parâmetros iniciais pois bons modelos podem ser construídos a partir de estimativas iniciais ruins. Entretanto, boas estimativas das probabilidades dos símbolos de saída são muito úteis [8].

Para HMMs contínuos é necessária uma boa inicialização das médias das misturas para a obtenção de bons modelos. Os modelos contínuos são menos sensíveis a erros na estimativa inicial das matrizes covariância. Erros nas escolhas iniciais das probabilidades de transição e coeficientes das misturas influenciam muito pouco na obtenção de bons modelos [41]. Portanto, para obter-se bons HMMs com densidade contínua, necessita-se de uma boa escolha das médias das misturas, independentemente das estimativas dos outros parâmetros.

3.3 Comparação entre os Tipos de HMM

- **HMM Discreto**:

  Para o uso do HMM discreto, quando os vetores de observação são contínuos como no caso do sinal de fala, é necessário o uso da quantização vetorial (seção

6.4) para torná-los discretos.

Vantagens:

- Não são feitas suposições sobre o tipo de distribuição dos símbolos observados.
- Procedimentos simples para treinamento e reconhecimento.
- Poucos parâmetros para estimar, não sendo necessária uma seqüência de treinamento muito longa.
- Não requer esforço computacional elevado.

Desvantagens:

- É necessário o uso de quantização vetorial.
- Não é possível recuperar os erros devido à quantização.
- Menor flexibilidade.

- **HMM Contínuo**:

HMM contínuo é usado para aplicações onde os vetores da seqüência de observação são contínuos.

Vantagens:

- Modelamento direto dos parâmetros do sinal de fala.
- Não usa quantização vetorial.

Desvantagens:

- Suposição sobre o tipo de distribuição dos símbolos observados (geralmente soma de Gaussianas).
- Procedimentos de treinamento e reconhecimento complexos.
- Requer grande esforço computacional.
- Muitos parâmetros para estimar. Necessita de uma grande seqüência de treinamento (muito maior que o discreto).

- **HMM Semicontínuo**:

Vantagens:

- Procedimento de treinamento mais simples que o HMM contínuo.

- Quantidade média de parâmetros para estimar.

Desvantagens:

- Maior número de parâmetros para estimar que o HMM discreto.
- Reconhecedor mais complexo que o HMM discreto.

3.4 HMM para Reconhecimento de Fala

HMM pode ser usado para representar qualquer unidade da fala desde palavras a unidades menores que palavras, tais como fones, difones e outros [8, 10, 42]. O fone é a realização física de um fonema, o qual é uma unidade distintiva de som. Quando são criados modelos para essas unidades, o modelo que representa a palavra pode ser construído concatenando-se os modelos das unidades que formam a palavra.

Uma das vantagens do uso da palavra como a unidade que será representada pelo modelo está no fato de conservar parte considerável dos efeitos da coarticulação (dentro da palavra).

Para vocabulários pequenos ou médios é comum o uso de modelos para as palavras como um todo, mas isso torna-se intratável para grandes vocabulários. Como cada palavra deve ser tratada individualmente e dados não podem ser compartilhados (dados para uma palavra são usados apenas para gerar o modelo que representa essa palavra), tornam-se necessárias uma enorme seqüência de treinamento e uma grande capacidade de memória para armazenar os dados. Para grandes vocabulários, uma forma de reduzir essas necessidades com memória e treinamento é a utilização de unidades sublexicais como fones, difones e trifones. Como essas unidades aparecem em mais de uma palavra, o material necessário para o treinamento dos modelos é fornecido por várias palavras. Dessa forma, pode-se reduzir o tamanho da seqüência de treinamento.

Para aplicar HMMs em reconhecimento de fala, o sinal de fala deve ser representado por uma seqüência temporal de vetores $O = \{O_1, \ldots, O_T\}$. Para isso, o mesmo é dividido em quadros e obtém-se um conjunto de parâmetros caracterizando cada quadro. Assim, o vetor O_i representando o sinal de fala no tempo $t = i$ é o conjunto de parâmetros que representa o quadro i. Fica claro que o número de vetores de observação O_i é o número de quadros (T).

Para o modelo discreto, cada vetor O_i deve ser representado por um símbolo pertencente a um conjunto de K símbolos, tornando-se necessário o uso da quantização vetorial.

O tipo de parâmetro escolhido para representar o sinal de fala é habitualmente um conjunto de coeficientes derivados dos coeficientes LPC, energia e outros que forneçam uma representação adequada do sinal de fala [8, 43].

O modelo " left-right " é o mais apropriado para representar o sinal de fala pois os estados do modelo podem ser colocados em correspondência com a organização temporal da fala. Além disso, cada estado ou conjunto de estados pode corresponder a um evento fonético.

Para a escolha do número de estados ótimo, não há regras. Uma alternativa consiste em fazer o número de estados igual ao número de fonemas da palavra. Neste caso, modelos representando palavras diferentes apresentarão número de estados diferentes. Usualmente, o mesmo número de estados é empregado para representar todas as palavras e escolhe-se um valor maior que o número de fonemas da maior palavra.

Quanto à escolha do número de símbolos de saída (HMM discreto) ou número de misturas (HMM contínuo), deve-se estabelecer um compromisso entre o tamanho da seqüência de treinamento e o número de parâmetros a estimar, de modo a se conseguir boas estimativas dos mesmos.

3.4.1 HMM para Reconhecimento de Palavras Isoladas

- Treinamento:

  Para cada palavra no vocabulário, é criado um modelo usando-se todos os exemplos dessa palavra na seqüência de treinamento. Para uma boa estimativa dos parâmetros do modelo, são necessárias várias repetições de cada palavra [44].

  Assim, uma seqüência de treinamento consistindo de D seqüências de observação é dada por $O = \{O^1 O^2 ... O^D\}$, onde $O^i = \{O^i_1, O^i_2, ..., O^i_{T_i}\}, 1 \leq i \leq D$ é o conjunto de vetores representando uma seqüência de observação.

  O treinamento pode ser feito usando-se o algoritmo " Forward-Backward " ou procedimento " K-means " e algoritmo de Viterbi. O primeiro requer um maior esforço computacional.

- Reconhecimento:

  Os parâmetros da palavra desconhecida são usados como entrada em cada modelo para calcular a probabilidade $P(O|\lambda)$. A palavra representada pelo modelo com a maior probabilidade é escolhida como sendo a palavra reconhe-

cida. O algoritmo " Forward " ou o algoritmo de Viterbi podem ser usados para calcular $P(O|\lambda)$.

Este tipo de procedimento é aplicado para vocabulários médios e pequenos pois o tempo necessário para reconhecer cada palavra é pequeno, permitindo este tipo de busca exaustiva.

3.4.2 HMM para Reconhecimento de Fala Contínua

Para o reconhecimento de fala contínua, devido ao grande número de palavras no vocabulário, é recomendada a utilização de modelos para representar unidades menores que palavras. Neste caso, para a construção dos modelos das palavras é necessária a utilização de um léxico que fornece a transcrição fonética de cada palavra em função das unidades escolhidas. O léxico pode conter mais de uma representação de cada palavra, considerando as diferentes formas de pronúncia [8, 10, 45, 46, 47].

Outro ponto que deve ser considerado em reconhecimento de fala contínua é a incorporação de conhecimentos da lingua. Isso é feito através de uma gramática na qual estão definidas as regras para formar as sentenças (seqüência de palavras permitidas), levando-se em conta a probabilidade de ocorrência de cada seqüência de palavras.

Para representar a gramática pode-se usar uma rede, onde cada transição corresponde a uma palavra, e na qual estão representadas todas as possíveis sentenças. A probabilidade de ocorrência de uma determinada seqüência de palavras corresponde tipicamente à freqüência de aparecimento dessa seqüência na seqüência de treinamento.

Quando a gramática não é usada, assume-se que qualquer palavra no vocabulário pode seguir qualquer outra palavra no vocabulário. Com o uso da gramática é possível reduzir os cálculos na fase de reconhecimento.

Para medir o grau de complexidade de uma gramática é usada uma medida denominada perplexidade, a qual pode ser considerada como o número médio de palavras que pode seguir outra palavra [10].

- Treinamento:

  Para estimar os parâmetros do HMM para fala contínua, cada palavra na sentença é representada por um modelo, o qual pode ser formado pela concatenação de outros modelos (fones, difones e outros) de acordo com o léxico. Os modelos representando cada palavra são concatenados seguindo a ordem em que aparecem na sentença. Modelos para representar o silêncio entre palavras

podem ser usados. Dessa forma, obtém-se um grande modelo que representa toda a sentença e o qual é treinado usando-se as sentenças correspondentes. Para esse treinamento não é necessário determinar os pontos limitantes de cada palavra pois os mesmos serão automaticamente determinados pelos algoritmos.

Como resultado do treinamento obtém-se modelos que representam cada uma das unidades pertencente ao conjunto de unidades utilizado.

Novamente pode-se utilizar o algoritmo " Forward-Backward " ou procedimento " K-means " e algoritmo de Viterbi.

- Reconhecimento:

  Na fase de reconhecimento, os modelos de cada palavra são obtidos concatenando-se os modelos das unidades como especificado pelo léxico. Com o uso da gramática testam-se as possíveis combinações de palavras. Pode-se observar que o uso da gramática reduz o número de testes a serem feitos e com isso os custos computacionais também são reduzidos.

  Existem vários algoritmos como " Viterbi-based Search ", " Level Building ", " Stack Decoding " e outros que podem ser utilizados para reconhecimento de fala contínua [8, 10].

  Para usar o algoritmo de Viterbi para fala contínua, primeiramente deve-se numerar todos os estados de todas as palavras do vocabulário. Depois, usa-se o algoritmo para as transições dentro das palavras e em uma segunda etapa testam-se as transições entre palavras. Para vocabulários muito grandes, esse algoritmo requer um grande esforço computacional. Uma forma de minimizar esse problema é o uso de um limiar para selecionar os candidatos mais prováveis em cada instante de tempo.

Capítulo 4

REDES NEURAIS

4.1 Introdução

O termo Rede Neural originalmente referia-se a uma rede de neurônios interconectados. Hoje esse termo é usado para designar qualquer arquitetura computacional que consiste de interconexões paralelas de simples elementos de processamento. Também são usados os termos Modelos Conexionistas e Modelos de Processamento Paralelo Distribuído.

Os primeiros modelos de redes neurais foram propostos por McCulloch e Pitts nos anos 40 [48]. Nas décadas seguintes outros modelos foram apresentados como por exemplo, " Perceptron " de Rosenblatt [49, 50], " Adaline " de Widrow e Hoff [51] e " Learning Matrix " de Steinbuch [52].

Após um período de esquecimento, o uso de redes neurais ganhou um novo impulso nos anos 80 devido à evolução dos computadores e aparecimento de novos algoritmos.

Redes Neurais tem sido usadas em diferentes áreas como processamento de sinais, classificação de padrões, problemas combinatoriais e outros. Além disso, apresentam grande potencial para o uso em áreas que requerem altas taxas de processamento computacional como reconhecimento de fala e imagem [53, 54].

Entre as vantagens do uso de redes neurais nestas áreas estão a alta capacidade de processamento paralelo, habilidade para adaptação das conexões com o tempo possibilitando o aprendizado das características do sinal de entrada e a adaptação às mudanças do mesmo.

4.2 Definição

Uma rede neural é formada por um grande número de elementos não lineares interconectados. A figura 4.1 apresenta um exemplo desses elementos, o qual é denominado nó ou neurônio, devido à semelhança com o neurônio biológico.

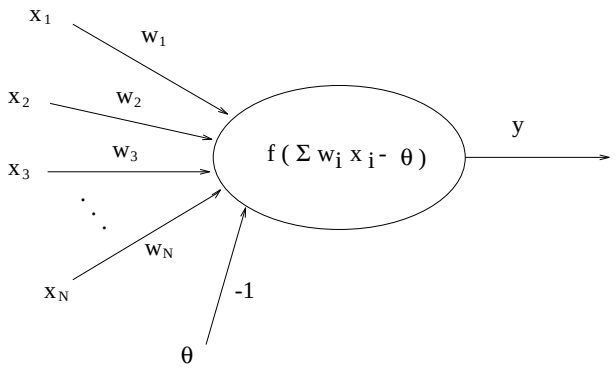

Figura 4.1: Ilustração de um neurônio ou nó.

A saída de um neurônio é uma função não linear da soma das entradas multiplicadas por coeficientes de ponderação.

$$y = f\left(\sum_{i=1}^{N} w_i x_i - \theta\right) \tag{4.1}$$

onde:
y é a saída do neurônio
$f(.)$ é uma função não-linear
x_i é a i-ésima entrada do neurônio
w_i é o coeficiente de ponderação da i-ésima entrada do neurônio
N é o número de entradas do neurônio
θ é o limiar de ativação do neurônio

A função $f(.)$ pode ser do tipo [8]:

- Função " Hard Limiter "

$$f(x) = \begin{cases} -1 & x \leq 0 \\ +1 & x > 0 \end{cases} \tag{4.2}$$

- Funções Sigmóide

$$f(x) = \frac{1}{1 + \exp\{-\beta x\}} , \qquad \beta > 0 \tag{4.3}$$

ou

$$f(x) = \tanh(\beta x) , \qquad \beta > 0 \tag{4.4}$$

As funções Sigmóide são as mais usadas por serem contínuas e diferenciáveis. Na função definida pela equação 4.3 o valor de β controla a inclinação da região semi-linear. Para pequenos valores de β, a função sigmóide aproxima-se de uma função linear e para grandes valores de β a função sigmóide torna-se uma função degrau.

Neste trabalho apenas a função sigmóide definida pela equação 4.3 com $\beta = 1$ foi usada como função não-linear. Assim, a equação 4.3 será chamada de função sigmóide.

A definição de uma rede neural consiste na especificação de:

- Um conjunto finito de neurônios N
- Um grafo de conectividade G
- Um conjunto de coeficientes de ponderação $W = \{w_{ij}\}$ formado por números reais
- Um conjunto de limiares $\theta = \{\theta_i\}$ formado por números reais

4.3 Topologia

A topologia de uma rede neural é a forma como são conectados os elementos que compõem a mesma. Vários modelos tem sido apresentados e os mais utilizados para reconhecimento de fala são [55]:

- Redes Recorrentes
- Kohonen ou Rede Auto-organizadas
- " Multilayer Perceptron "

Redes Recorrentes são sistemas em que as entradas de cada elemento consistem de uma combinação das entradas da rede com as saídas de outros elementos da rede. Essas redes são apropriadas para sistemas com entradas que podem ser representadas por valores binários.

Redes de Kohonen são redes que se auto-organizam a partir da apresentação seqüencial dos vetores de entrada. Essa rede pode ser usada em reconhecimento de fala como um quantizador vetorial.

O tipo de rede mais comumente usado em reconhecimento de fala é o " Multilayer Perceptron ", o qual será descrito na próxima seção [56, 57, 58, 59, 60, 61, 62]. Esse foi o tipo de rede utilizado neste trabalho.

4.4 " Multilayer Perceptron "

" Multilayer Perceptrons " (MLP) são redes com uma ou mais camadas de neurônios entre as camadas de entrada e saída da rede [63]. Essas camadas são chamadas de camadas escondidas. As saídas dos neurônios de uma determinada camada formam as entradas dos neurônios da próxima camada. Na figura 4.2 é apresentado um exemplo de uma rede " Multilayer Perceptron " com duas camadas escondidas.

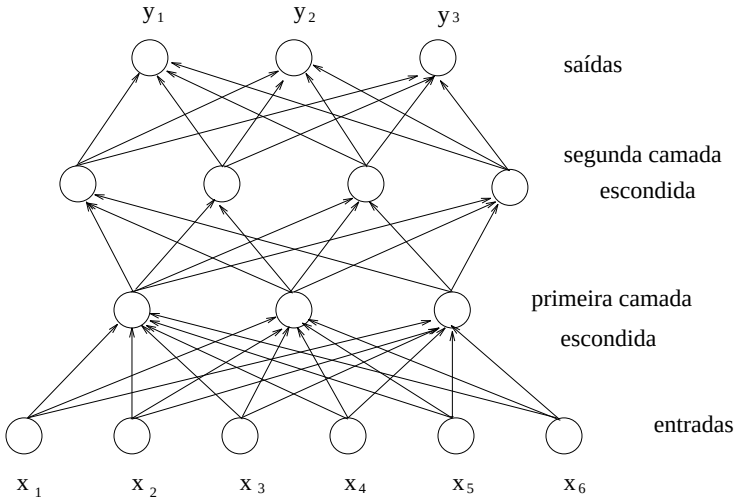

Figura 4.2: " Multilayer Perceptron " com duas camadas escondidas.

Pode ser provado que redes " Multilayer Perceptron " com uma camada escondida e número suficiente de neurônios nessa camada podem realizar qualquer tipo de mapeamento entrada/saída [63].

Para a escolha de parâmetros da rede como número de camadas, número de neurônios por camada e número de entradas, não existe uma regra definida. Deve-se manter um compromisso entre a precisão desejada da rede e o tempo gasto para treiná-la. Ambos aumentam com o aumento do tamanho da rede.

A escolha do tipo de entrada é muito importante pois o desempenho da rede depende fortemente desse parâmetro.

O conjunto de coeficientes de ponderação e limiares são obtidos usando-se o algoritmo " Back Propagation ". Esses parâmetros são calculados de modo a associar um determinado vetor de saída desejado a um determinado tipo de vetor de entrada.

4.5 Algoritmo " Back Propagation "

Esse algoritmo é uma generalização do algoritmo dos mínimos quadrados e busca minimizar o erro quadrático médio entre a saída atual da rede e a saída desejada usando a técnica do gradiente [64].

O procedimento de cálculo dos coeficientes w_{ij} e limiares θ_j é um método iterativo, sendo os coeficientes ajustados após a apresentação de cada padrão de entrada. Esse ajuste é feito de forma recursiva começando pela camada de saída e terminando na primeira camada escondida. O limiar θ_j pode ser considerado como um coeficiente de ponderação para uma entrada com valor igual a -1.0.

A descrição do algoritmo " Back Propagation " [64] é a seguinte:

1. Inicialização

   Todos os coeficientes de ponderação e limiares são inicializados com pequenos valores aleatórios (entre -0.1 e $+0.1$)

2. Apresentação da entrada e da saída desejada

   Apresente o vetor de entrada $x = \{x_1, x_2, ..., x_N\}$ e a correspondente saída desejada $d = \{d_1, d_2, ...d_M\}$. Faça $x_0 = -1.0$.

3. Cálculo das saídas

   Seja $y = \{y_1, y_2, ..., y_M\}$ a saída da rede, $h_i = \{h_{i1}, h_{i2}, ..., h_{iM_i}\}$ a saída da i-ésima camada escondida e K o número de camadas escondidas.

   O cálculo das saídas começa pela primeira camada escondida e vai seqüencialmente até a camada de saída. Assim, temos:

$$h_{1j} = f\left(\sum_{i=0}^{N} w_{1ij}x_i\right) \tag{4.5}$$

$$h_{kj} = f\left(\sum_{i=0}^{N} w_{kij} h_{(k-1)i}\right) , \quad 2 \leq k \leq K \tag{4.6}$$

$$y_j = f\left(\sum_{i=0}^{N} w_{(K+1)ij} h_{Ki}\right) \tag{4.7}$$

4. Adaptação

   A adaptação dos coeficientes de ponderação inicia-se pela camada de saída e segue até a primeira camada escondida.

   Para nós na camada de saída:

$$w_{(K+1)ij}(t+1) = w_{(K+1)ij}(t) + \eta \delta_{(K+1)j} h_{Ki}(t) \tag{4.8}$$

$$\delta_{(K+1)j} = y_j(1-y_j)(d_j - y_j) \tag{4.9}$$

   Para nós nas camadas escondidas $(k = K, ..., 2)$:

$$w_{kij}(t+1) = w_{kij}(t) + \eta \delta_{kj} h_{(k-1)i}(t) \tag{4.10}$$

$$\delta_{kj} = h_{kj}(1-h_{kj}) \sum_{l} \delta_{(k+1)l} w_{(k+1)jl} \tag{4.11}$$

   onde a somatória é realizada considerando todos os nós que estão conectados ao nó j na camada $k+1$.

   Para nós na primeira camada escondida:

$$w_{1ij}(t+1) = w_{1ij}(t) + \eta \delta_{1j} x_i(t) \tag{4.12}$$

$$\delta_{1j} = h_{1j}(1-h_{1j}) \sum_{l} \delta_{2l} w_{2jl} \tag{4.13}$$

   onde a somatória é realizada considerando todos os nós que estão conectados ao nó j na segunda camada escondida (ou camada de saída se existir apenas uma camada escondida).

5. Volte ao passo 2 e repita o procedimento até que todos os vetores de entrada tenham sido apresentados. Feito isto, completa-se uma iteração.

6. Repita o procedimento por um determinado número de iterações ou até que algum critério de convergência tenha sido atingido.

O variável η é denominada taxa de aprendizagem ou fator de ganho e assume valores entre 0 e 1. A velocidade de convergência do algoritmo depende do valor de η. Valores muito pequenos aumentam o tempo de convergência enquanto valores muito altos podem fazer a rede oscilar.

Uma forma de aumentar a velocidade de convergência do algoritmo consiste em usar um termo de momento α. Assim, em todas as fórmulas de adaptação deve ser acrescentado o termo $\alpha(w_{kij}(t) - w_{kij}(t-1))$, onde $0 < \alpha < 1$.

4.6 " Multilayer Perceptron " em Reconhecimento de Fala

Redes " Multilayer Perceptron " têm sido usadas para reconhecimento de fala contínua e palavras isoladas. Como este trabalho restringiu-se apenas ao caso de palavras isoladas, somente este será discutido aqui.

Para palavras isoladas, o número de nós de saída da rede deve ser igual ao número de palavras no vocabulário, associando-se cada saída a uma palavra.

Dessa forma, a saída que corresponde à palavra correta deverá apresentar valor igual a 1 e as outras saídas da rede deverão apresentar valores iguais a zero.

Os vetores de entrada, com valores contínuos, podem ser qualquer parâmetro usado para representar o sinal de fala, como por exemplo, algum tipo de coeficiente derivado dos coeficientes LPC.

Para a obtenção desses parâmetros, o sinal de fala deve ser segmentado em quadros, devido ao fato do sinal de fala não ser estacionário. Como o número de entradas da rede é o mesmo para qualquer palavra e as durações das palavras são variáveis, o tamanho do quadro e da janela de análise será diferente para cada palavra mas usando-se o mesmo índice de superposição entre janelas. Uma outra forma de calcular esses parâmetros é usar janelas de análise de mesmo tamanho, mas com superposição variável entre elas.

Para cada quadro do sinal de fala obtém-se um conjunto de parâmetros. Esses conjuntos devem ser concatenados em um único vetor que será a entrada da rede.

Quando a função sigmóide é usada como função não-linear, recomenda-se o uso de parâmetros que apresentem média zero e variância igual a 0.25 para acelerar a convergência da rede [65]. Isso garante que 95% dos valores dos parâmetros estará na região entre -1 e $+1$.

O procedimento para usar " Multilayer Perceptron " em reconhecimento de fala é dividido em duas fases: treinamento e reconhecimento.

Na fase de treinamento, os vetores representando cada palavra no vocabulário e as correspondentes saídas desejadas são apresentados à rede de forma seqüencial. Com o uso do algoritmo " Back Propagation " calculam-se os coeficientes de ponderação e os limiares de ativação. O conjunto de todos os vetores de entrada é denominado seqüencia de treinamento e deve conter exemplares de todas as palavras do vocabulário. Os coeficientes de ponderação e limiares de ativação são atualizados após a apresentação de cada vetor da seqüência de treinamento. Quando todos os vetores da seqüência de treinamento foram apresentados, concluiu-se uma iteração.

Na fase de reconhecimento, os parâmetros representando a palavra desconhecida são apresentados à rede e a saída desta que apresentar o maior valor é escolhida como sendo a palavra reconhecida.

4.7 Vantagens do Uso de " Multilayer Perceptron "

- Uso de processamento paralelo
- Não são feitas suposições sobre as propriedades do sinal de fala
- Treinamento discriminativo, isto é, ao mesmo tempo em que se maximiza o valor da saída desejada, minimiza-se os valores das outras saídas da rede
- Pode-se produzir qualquer função não-linear das entradas
- Coeficientes da rede podem ser adaptados em tempo real para melhorar o desempenho da mesma

Capítulo 5

SISTEMAS HÍBRIDOS

5.1 Introdução

No capítulo 3 foi apresentada a técnica de Modelos Ocultos de Markov (HMM). Reconhecedores de fala utilizando HMMs têm apresentado resultados muito bons e têm sido largamente utilizados. Isso é devido principalmente a algumas características dos HMMs como: facilidade de modelamento dos aspectos temporais do sinal de fala, existência de bons algoritmos para o treinamento e reconhecimento e topologia flexível. Entretanto, essa técnica ainda apresenta algumas limitações como suposições sobre o comportamento do sinal de fala e pobre poder discriminativo pois maximiza-se a probabilidade do modelo ter gerado uma dada seqüência mas não se preocupa em minimizar a probabilidade do modelo gerar as seqüências correspondentes às demais palavras.

No capítulo 4 indicou-se a utilidade de Redes Neurais em reconhecimento de fala, principalmente pela alta capacidade de processamento paralelo e poder de discriminação de padrões. Entre os pontos fracos desta técnica destaca-se o fato de alguns tipos de redes não serem invariantes ao deslocamento das entradas, não serem apropriadas para o reconhecimento de fala contínua e apresentarem dificuldades em tratar a duração variável do sinal de fala.

Dessa forma, parece ser interessante combinar as características dessas duas técnicas para a obtenção de reconhecedores de fala mais robustos e com melhor desempenho.

Nesse sentido, vários sistemas híbridos combinando essas duas técnicas de diferentes maneiras têm sido propostos [66, 67, 68, 69, 70]. Alguns usam redes neurais

como pós-processadores e outros como pré-processadores. Neste capítulo serão descritos alguns dos sistemas propostos.

5.2 HMM usando MLP como Pós-Processador

Nestes sistemas, a Rede Neural é usada como um pós-processador para os HMMs, explorando o seu poder discriminativo.

Primeiramente calcula-se a probabilidade do HMM representando cada palavra no vocabulário ter gerado a seqüência de parâmetros que representam a palavra desconhecida. O vetor formado pelas probabilidades de todos os modelos é usado como entrada da rede neural e esta determina a palavra reconhecida.

Observa-se que nos sistemas que utilizam apenas HMM, o modelo que apresenta a maior probabilidade é escolhido como a palavra reconhecida. Nos sistemas híbridos nem sempre isso acontece pois leva-se em conta todo o conjunto de probabilidades e não apenas o maior valor. Isso melhora o desempenho do reconhecedor.

Na figura 5.1 é apresentada a arquitetura de um sistema híbrido combinando HMMs e Redes Neurais.

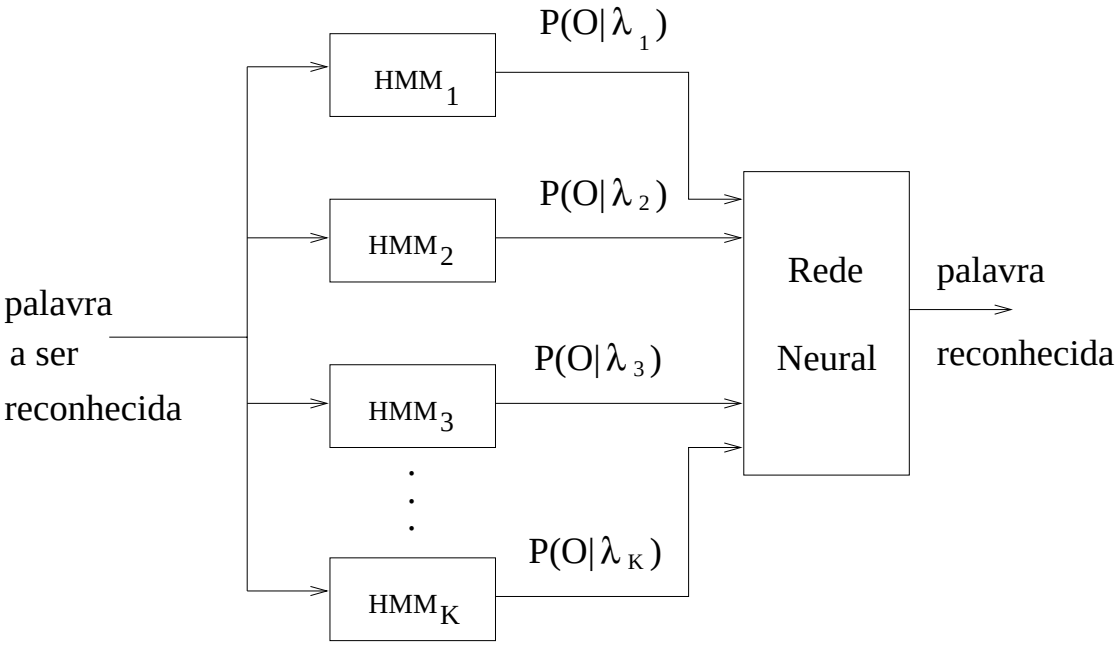

Figura 5.1: Sistema Híbrido usando Rede Neural como pós-processador para HMMs.

Embora esses sistemas apresentem um melhor desempenho, eles requerem esforços computacionais muito maiores para o treinamento dos mesmos. Em uma

primeira fase são treinados os HMMs e depois a Rede Neural. Portanto, o treinamento apresenta duas fases.

Na fase de reconhecimento, o acréscimo computacional é minimo, chegando a ser desprezível.

Esse sistema também pode ser usado para combinar as probalidades dos HMMs e as probalidades de duração de estados. No capítulo 3 foi apresentada uma forma de combinação dessas probabilidades usando um coeficiente empírico w.

$$\log P(q, O|\lambda) = \log P^*(q, O|\lambda) + w \sum_{j=1}^{N} \log(P_j(d_j)) \tag{5.1}$$

Nesta proposta, o cálculo do coeficiente w será feito pela rede neural.

Neste tipo de sistema, a entrada da rede é um vetor formado pelas probabilidades de cada modelo e as probabilidades de duração de estado.

O tipo de Rede Neural que tem sido empregado nestes sistemas híbridos é o " Multilayer Perceptron ".

5.3 HMM como Segmentador para MLP

Este sistema híbrido combina " Hidden Markov Models " (HMM) e redes " Multilayer Perceptron " (MLP) em uma maneira similar ao sistema proposto por Katagiri e Lee [71]. A idéia central desse algoritmo é gerar um vetor com dimensão fixa, o qual será a entrada da rede MLP, a partir do sinal de fala que apresenta duração variável. Para isso usa-se a técnica HMM (densidade discreta ou contínua) e Algoritmo de Viterbi como segmentador. Assim, são utilizadas as potencialidades dos HMMs como segmentadores, aliadas ao grande poder discriminativo das Redes Neurais.

Embora esse algoritmo possa ser utilizado em reconhecedores de fala contínua ou palavras isoladas, apenas este último caso será discutido neste trabalho.

Estes sistemas híbridos podem ser divididos em 3 partes:

- Segmentação usando HMM
- Normalização Temporal
- Reconhecimento usando MLP

Na fase de segmentação utiliza-se HMM e algoritmo de Viterbi para a obtenção da seqüência de estados ótima. Os HMMs foram previamente treinados usando-se os algoritmos padrões (" Forward-Backward " ou procedimento " K-means " e

algoritmo de Viterbi). Cada palavra do vocabulário é representada por um HMM, sendo que todos apresentam o mesmo número de estados. Isso é necessário para que se possa conseguir um vetor normalizado para a entrada da rede neural.

Após a segmentação, calcula-se o vetor média dos vetores agrupados no mesmo estado. Assume-se que esses vetores possuem características comuns. Assim, cada estado será representado por um vetor de parâmetros. Concatenando-se esses vetores, tem-se um vetor de dimensão $N.C$, onde N é o número de estados e C é o número de coeficientes do vetor de parâmetros utilizado para representar o sinal de fala. Observa-se que a dimensão do vetor normalizado independe da duração do sinal de fala.

O vetor normalizado é utilizado como entrada da rede MLP que indicará a palavra reconhecida.

Como os outros sistemas apresentados até agora, este também possui duas fases: treinamento e reconhecimento.

Na fase de treinamento, criam-se os HMMs usando-se um dos algoritmos descritos no capítulo 3. Depois, cada padrão da seqüência de treinamento é segmentado utilizando-se o HMM correspondente e o algoritmo de Viterbi. Feito isto, calcula-se o vetor média dos parâmetros associados a cada estado e assim, obtém-se o vetor normalizado para cada exemplar da seqüência de treinamento. Esses vetores normalizados são usados para treinar a rede neural. Pode-se utilizar seqüências de treinamento diferentes para os HMMs e redes MLP.

Para o caso de reconhecimento de K palavras isoladas, na fase de reconhecimento tem-se K modelos disponíveis para segmentar a palavra desconhecida. Para a escolha do modelo que será usado para a segmentação existem várias possibilidades como:

- Uso do modelo mais provável (figura 5.2)
- Uso de todos os modelos (figura 5.3)

Quando todos os modelos são usados, utiliza-se a rede MLP para os parâmetros normalizados através de cada um dos K HMMs e em seguida as saídas da rede MLP nos K casos podem ser combinadas utilizando-se coeficientes de ponderação. Dessa forma, pode-se escrever:

$$Y_j = 1/K \sum_{i=1}^{K} w_i y_j^i \tag{5.2}$$

onde:
y_j^i é a j-ésima saída da rede usando o i-ésimo HMM

w_i é o i-ésimo coeficiente de ponderação
K é o número de palavras no vocabulário

As probabilidades dos HMMs normalizadas podem ser usadas como os coeficientes de ponderação.

Após a combinação a saída com o maior valor é escolhida como sendo a palavra reconhecida.

Nas figuras 5.2 e 5.3 são apresentados diagramas em blocos dessas duas possibilidades de implementação deste sistema híbrido, combinando HMMs e redes MLP.

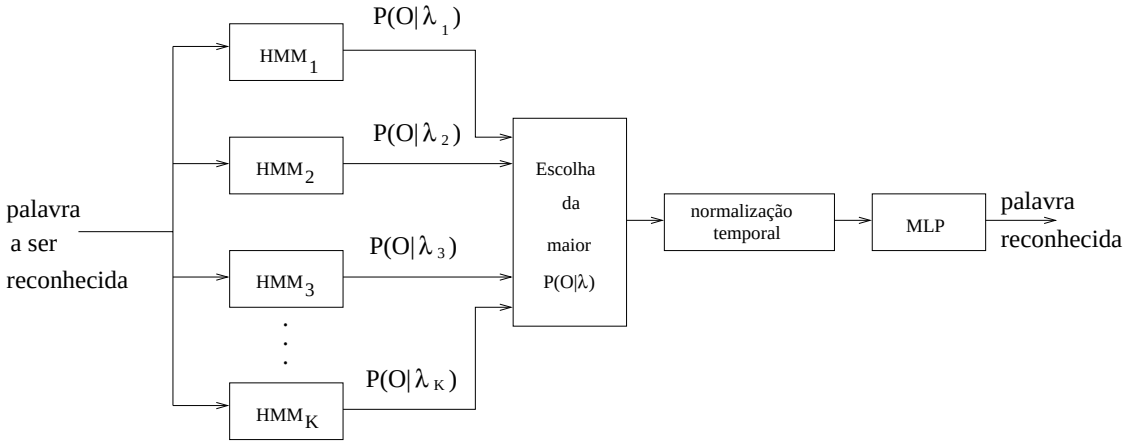

Figura 5.2: Sistema Híbrido usando o HMM com maior probabilidade como segmentador para Rede Neural.

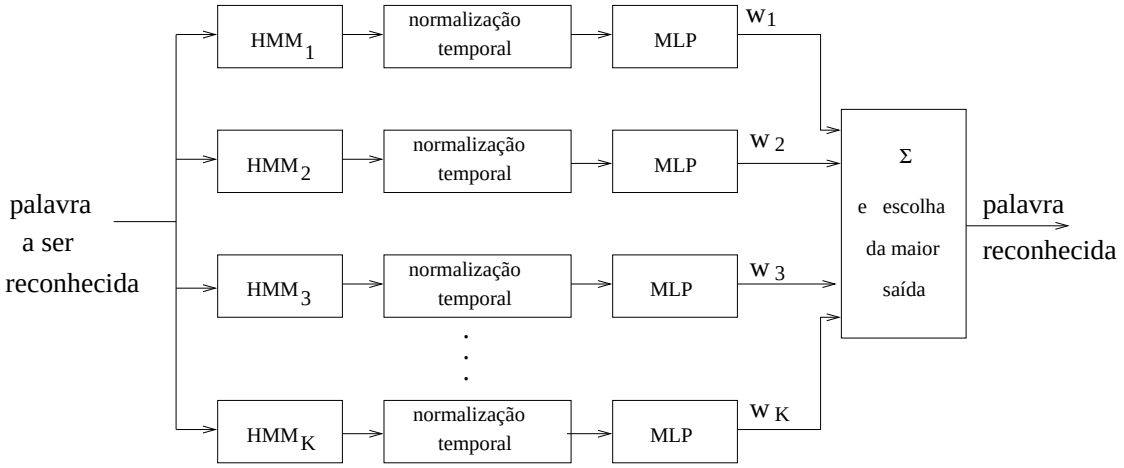

Figura 5.3: Sistema Híbrido usando HMMs como segmentadores para rede MLP.

5.4 Estimação de Probabilidades a Posteriori usando MLP

Os sistemas pertencentes a este grupo usam redes " Multilayer Perceptron " como um estimador de probabilidades a posteriori. Essas probabilidades são utilizadas para calcular as probabilidades de emissão de símbolos nos HMMs [72, 73].

Seja $Q = \{q_k\}$ um conjunto de K classes de padrões. A probabilidade a posteriori da classe q_k representada por $P(q_k|x)$ é a probabilidade de um padrão pertencer à classe q_k condicionada à ocorrência do vetor x.

Considerando-se uma rede " Multilayer Perceptron " com K nós na camada de saída, pode ser mostrado [74] que os valores das saídas da rede são estimativas da distribuição de probabilidade das classes condicionada à entrada da rede.

$$y_k(x) = P(q_k|x) \tag{5.3}$$

Para o treinamento dessa rede pode-se usar como saída desejada os próprios valores da probabilidade a posteriori obtidos pelo uso de HMMs previamente treinados e utilizar o erro quadrático médio como a função a ser minimizada durante o treinamento.

As probabilidades de emissão de símbolos em um HMM com K estados podem ser estimadas a partir das saídas da rede MLP aplicando-se a regra de Bayes:

$$\frac{p(x|q_k)}{p(x)} = \frac{P(q_k|x)}{P(q_k)} \tag{5.4}$$

Levando-se em conta que o fator $p(x)$ é constante para todas as classes, a densidade de probabilidade de emissão de símbolos $p(x|q_k)$ pode ser obtida dividindo-se as saídas da rede pela probabilidade de ocorrência da classe q_k. Essa probabilidade é dada por:

$$P(q_k) = \sum_{x_j \in X} P(q_k|x_j)p(x_j) \tag{5.5}$$

Considerando-se os símbolos x_j como equiprováveis, tem-se:

$$P(q_k) \approx 1/M \sum_{x_j \in X} P(q_k|x_j) \tag{5.6}$$

onde M é o número de símbolos na seqüência de treinamento.

Portanto, a densidade de probabilidade de emissão de símbolos em um HMM pode ser calculada por:

$$p(x|q_k) = \frac{P(q_k|x)}{1/M \sum_{x_j \in X} P(q_k|x_j)} \tag{5.7}$$

A partir da obtenção da probabilidade de emissão de símbolos, usa-se HMM e os algoritmos tradicionais para o reconhecimento.

O uso de redes MLP para a estimativa das probabilidades de emissão de símbolos pode incorporar informação contextual usando-se como entrada da rede não apenas o vetor x_j mas também os vetores vizinhos a este. Assim, o vetor de entrada da rede seria $X = \{x_{j-e}, ..., x_j, ..., x_{j+d}\}$, considerando-se e vetores à esquerda e d vetores à direita.

O procedimento descrito nesta seção tem sido utilizado para reconhecimento de fala contínua, sendo que as saídas da rede correspondem a unidades acústicas como por exemplo fones.

Na figura 5.4 é apresentado um diagrama em blocos deste sistema.

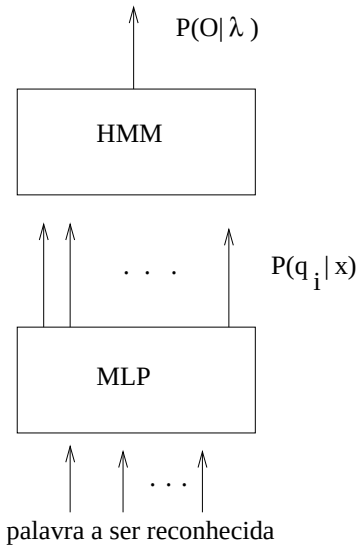

Figura 5.4: Sistema Híbrido usando HMMs e Redes Neurais como estimadores de probabilidades a posteriori.

Neste trabalho foi feita uma adaptação desse algoritmo para o cálculo da densidade de probabilidade de emissão de símbolos para um reconhecedor de palavras isoladas. Neste caso, cada saída da rede corresponde a um estado do HMM e criou-se uma rede para cada HMM representando uma palavra diferente. Para o treinamento da rede neural usou-se como saída desejada os valores de $\gamma_t(i)$ obtidos durante o treinamento dos HMMs e o qual é dado por:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \tag{5.8}$$

O valor de $\gamma_t(i)$ é a probabilidade a posteriori $P(q_t = i|O, \lambda)$, a qual é a probabalidade de estar no estado i no instante de tempo t, dada a seqüência de observação O e o modelo λ.

Deve-se observar que no cálculo de $\gamma_t(i)$ é considerada toda a seqüência de observação $O = \{O_1, \ldots, O_T\}$ em todos os instantes de tempo, pois no cálculo de $\alpha_t(i)$ considera-se parte da seqüência de observação O $(O_1, \ldots, O_t)$ e no cálculo de $\beta_t(i)$ leva-se em conta o restante da seqüência de observação O $(O_{t+1}, \ldots, O_T)$. Para o treinamento da rede, apenas parte da seqüência de observação é usada em cada instante de tempo .

Esta proposta apresenta um método para cálculo da densidade de probabilidade de emissão de símbolos que não requer suposições sobre o modelamento dos parâmetros, como por exemplo, considerar uma mistura de gaussianas.

Capítulo 6

PROCESSAMENTO DO SINAL DE FALA

6.1 Introdução

Em um sistema de reconhecimento de fala, o processamento do sinal de fala tem por objetivo converter esse sinal em um conjunto de parâmetros mais apropriado para o uso nos estágios seguintes do reconhecedor. Esta fase envolve a realização de várias funções como detecção de início/fim das palavras e análise e cálculo de parâmetros do sinal. Esta última função é muito importante para a obtenção de um sistema de reconhecimento robusto. Assim, várias técnicas de processamento de fala e extração de parâmetros têm sido apresentadas e novos tipos de parâmetros têm sido propostos. Muitas dessas técnicas têm se mostrado confiáveis e contribuído para melhorar o desempenho dos reconhecedores de fala.

Na figura 6.1 são apresentadas as funções que compõem este estágio do reconhecedor, as quais serão descritas neste capítulo.

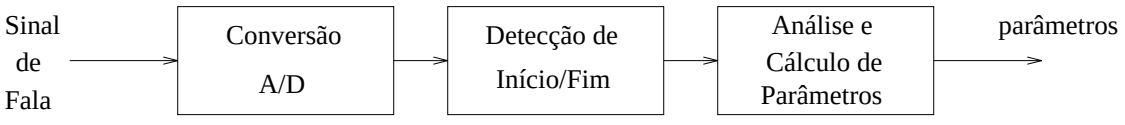

Figura 6.1: Processamento do sinal de fala em um reconhecedor de fala.

6.2 Conversão A/D

O conversor A/D converte o sinal analógico em sinal digital.

Neste trabalho, o sinal de fala foi filtrado por um filtro passa-faixa com freqüências de corte (3 dB) em 300 Hz e 3.4 kHz. A seguir o sinal foi amostrado com uma freqüência de 8 kHz e sofreu uma codificação linear com 16 bits por amostra. O sistema de aquisição de dados utilizado e a descrição da base de dados estão no apêndice A.

Devido à eliminação das baixas freqüências, o sinal de fala digitalizado apresentava média zero.

6.3 Detecção de Início/Fim

Detecção de início/fim é o processo de separar o sinal de fala dos sinais de ruído. Um bom detector de início/fim deve apresentar um processamento simples e eficiente, precisão na localização dos pontos limitantes do sinal de fala e capacidade de aplicação para diferentes valores de relação sinal/ruído. Uma detecção de início/fim precisa é muito importante para os sistemas de reconhecimento de fala devido principalmente a duas razões:

- O desempenho do reconhecedor depende da precisão do detector de início/fim.
- O processamento computacional do reconhecedor diminui quando os pontos iniciais e finais da palavra são corretamente determinados.

Detectar os pontos limitantes de uma palavra com precisão não é uma tarefa fácil, principalmente quando a relação sinal/ruído não é muito alta.

Como o detector de início/fim é um componente essencial em um reconhecedor de palavras isoladas, diferentes algoritmos para a realização dessa tarefa têm sido propostos. A maioria deles usa alguns dos seguintes parâmetros: energia do sinal, taxa de cruzamento de zero, duração do sinal e outros.

De modo a comparar o desempenho dos detectores de início/fim, foram implementados reconhecedores de palavras isoladas usando diferentes algoritmos para a detecção dos pontos limitantes das palavras. Os algoritmos utilizados serão descritos a seguir.

6.3.1 Detector " Endpoints "

Esse algoritmo proposto por Rabiner e Sambur no artigo " An Algorithm for Determining the Endpoints of Isolated Utterances " [75] detecta os pontos limitantes do sinal de fala baseado nas medidas de energia e taxa de cruzamento de zero do sinal, sendo apropriado para trabalhar com relação sinal/ruído de no mínimo 30 dB.

Basicamente o algoritmo consiste em dividir o sinal de fala em quadros, calcular a energia e a taxa de cruzamento de zero para cada quadro, comparar com limiares pré-estabelecidos e assim definir a localização dos pontos inicial e final das palavras. Esses limiares definem os valores característicos dos parâmetros para o silêncio ou ruído de fundo e são obtidos analisando-se os quadros iniciais do sinal gravado. Considera-se que o começo do intervalo de gravação contém apenas silêncio ou ruído. Inicialmente os valores de energia de cada quadro são comparados com os limiares e são definidos candidatos a serem os pontos limitantes da palavra. A seguir, analisa-se a taxa de cruzamento de zeros em um intervalo de N quadros a partir dos pontos candidatos a início e fim de forma a melhorar a localização dos mesmos para palavras que comecem ou terminem com consoantes fricativas.

A figura 6.2 apresenta um diagrama em blocos desse detector de início/fim.

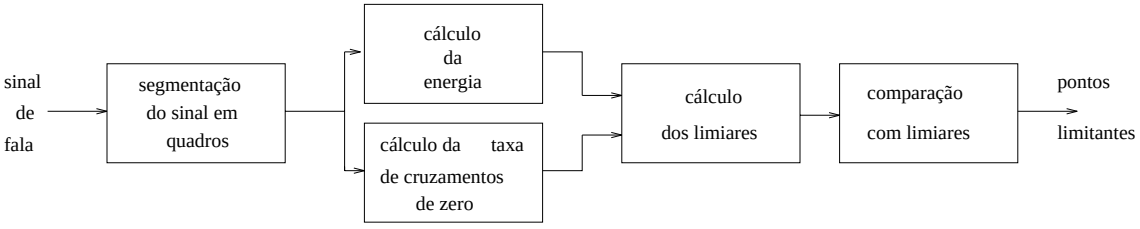

Figura 6.2: Diagrama em blocos do detector de início/fim de Rabiner e outros.

O algoritmo implementado neste trabalho apresenta algumas modificações em relação ao algoritmo proposto por Rabiner. Nesta implementação, o sinal de fala foi segmentado em quadros de 10 ms (80 amostras) e para o cálculo dos limiares do silêncio/ruído foram utilizados os primeiros e os últimos 100 ms do intervalo de gravação. O segmento que apresentar o menor valor de energia é escolhido como representando o silêncio/ruído e a partir dele são calculados os limiares do silêncio/ruído. Para o limiar fixo da taxa de cruzamento de zero usou-se o valor 20 e utilizou-se um limiar superior de energia igual a 4 vezes o limiar inferior de energia. Para analisar os valores da taxa de cruzamento de zero usou-se N = 15 quadros (150 ms) e o ponto inicial ou final era alterado sempre que a taxa de cruzamento superasse o limiar por três quadros consecutivos.

6.3.2 Detector " Bottom-Up "

Esse algoritmo proposto por Lamel e outros no artigo " An Improved Endpoint Detector for Isolated Word Recognition " [76] usa a energia do sinal e a duração dos pulsos de energia para a detecção dos pontos inicial e final da palavra. Na figura 6.3 é mostrado um diagrama desse detector.

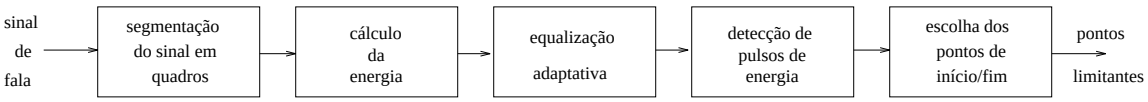

Figura 6.3: Diagrama em blocos do detector de início/fim "Bottom-up ".

O equalizador adaptativo faz uma normalização da energia de cada quadro a partir do valor mínimo de energia do sinal. Como resultado dessa normalização os períodos de silêncio apresentam energia normalizada em torno de 0 dB enquanto que os períodos que contêm sinal de fala apresentam energia superior.

A partir da energia normalizada são definidos quatro limiares, mostrados na figura 6.4, que são usados para determinar os pulsos de energia.

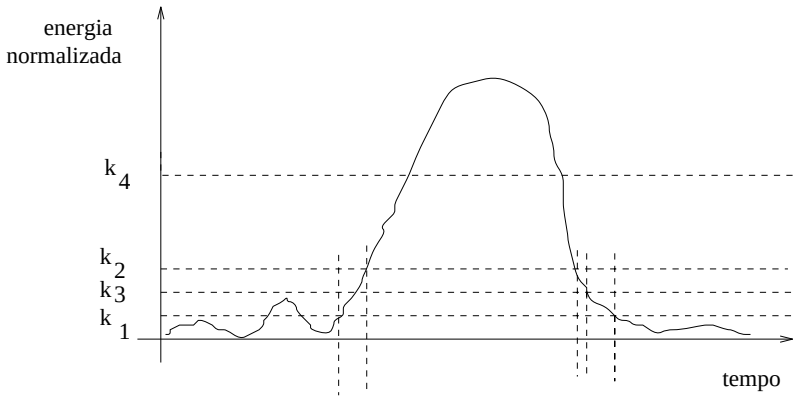

Figura 6.4: Limiares usados para a detecção de pulsos de energia.

Após a detecção dos pulsos de energia, mede-se a duração dos mesmos e a distância entre eles. Se essa distância for menor que um limiar, assume-se que esses pulsos pertencem à mesma palavra. A duração total dos pulsos combinados deve estar acima de um determinado limiar. Se isto acontecer, conseguiu-se detectar uma palavra e o ponto inicial do primeiro pulso juntamente com o ponto final do último pulso são definidos como os pontos limitantes da palavra.

Esse algoritmo é denominado " bottom-up " devido à comparação da energia com os limiares se iniciar com o limiar de menor valor, detectando-se primeiramente

os quadros com menor energia.

Na implementação deste algoritmo usou-se quadros de 10 ms (80 amostras) e os seguintes valores de limiares de energia:

- $K_1 = 3$ dB
- $K_2 = 10$ dB
- $K_3 = 5$ dB
- $K_4 = 15$ dB

Como limiares de duração foram usados os seguintes valores:

- Duração mínima dos pulsos de energia: 70 ms
- Separação máxima entre os pulsos: 150 ms
- Intervalo para determinar a presença ou ausência de qualquer ruído nos pontos inicial e final: 70 ms

6.3.3 Detector " Top-Down "

Este algoritmo, descrito no artigo " An Improved Word-detection Algorithm for Telephone-Quality Speech Incorporating Both Syntactic and Semantic Constraints ", Wilpon e outros [77], é similar ao algoritmo " bottom-up " descrito anteriormente. Aqui também calcula-se a energia normalizada, detectam-se pulsos de energia e faz-se uma combinação destes para encontrar os pontos limitantes da palavra.

A diferença entre os algoritmos está no fato deste inicialmente encontrar o quadro com maior energia e a partir deste quadro começar a fazer as comparações com os limiares de energia para a localização dos pulsos de energia. Essa é a razão para o nome " top-down ".

Após a detecção dos pulsos, eles são ordenados em ordem decrescente de energia e, começando com o pulso de maior energia, tenta-se combinar os pulsos adjacentes seguindo algumas regras. Essas regras apresentam diferenças em relação às regras utilizadas no algoritmo " bottom-up ".

A figura 6.5 apresenta um diagrama do procedimento utilizado para a detecção dos pulsos de energia.

Na implementação deste algoritmo também usou-se quadros de 10 ms (80 amostras) e os valores de limiares de energia foram os mesmos usados na implementação do algoritmo " bottom-up ".

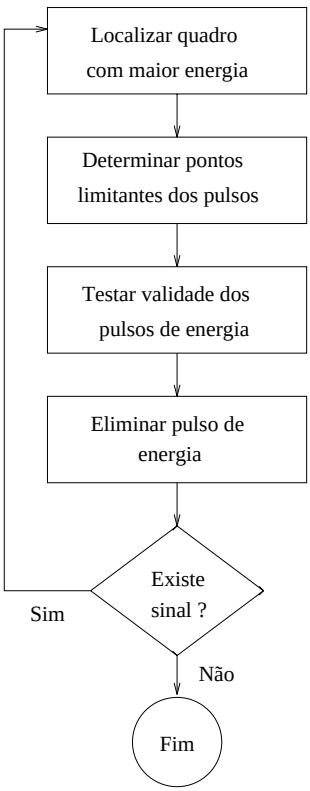

Figura 6.5: Procedimento para detecção de pulsos de energia no algoritmo " top-down ".

6.3.4 Comparação entre os Detectores de Início/Fim Implementados

Para avaliar os algoritmos de detecção de início/fim, foram utilizados reconhecedores de palavras isoladas independentes do locutor empregando as técnicas HMM e MLP, sendo analisados os desempenhos dos reconhecedores para cada um dos algoritmos descritos. Na avaliação do desempenho dos reconhecedores foi usado vocabulário com 50 palavras (apêndice A) e a seqüência de treinamento era formada por uma elocução de cada palavra.

Utilizando a técnica HMM, foram implementados reconhecedores usando HMM discreto, 10 estados e coeficientes " Mel Frequency Cepstrum " com subtração da média espectral (16 coeficientes por vetor) (seção 6.5). Para o treinamento dos reconhecedores foi utilizado o algoritmo " Forward-Backward " e o reconhecimento foi realizado usando o algoritmo " Forward ". Foi utilizada a mesma seqüência de treinamento para criar o " codebook " e treinar os HMMs. Na criação do " codebook " usou-se o algoritmo LBG com distância Euclidiana como medida de distorção (seção 6.4).

Para a comparação, variou-se o número de estados dos HMMs e também o tamanho do " codebook ", os quais foram combinados e resultaram em diferentes reconhecedores.

Os reconhecedores implementados usando MLP apresentavam 1280 entradas, uma camada escondida com 100 neurônios e 50 saídas. Foram utilizados coeficientes " Mel Frequency Cepstrum " (16 coeficientes por vetor) (seção 6.5). Para o treinamento dos reconhecedores foi empregado o algoritmo " Back Propagation ".

Os algoritmos " bottom-up " e " top-down " foram implementados com e sem a utilização de pré-ênfase. Quando esta foi utilizada, o valor do coeficiente de pré-ênfase empregado foi de 0.95.

As tabelas 6.1 e 6.2 apresentam os desempenhos dos reconhecedores em termos de porcentagem de acerto, para cada um dos algoritmos de detecção de início/fim implementados.

Analisando-se os resultados das tabelas 6.1 e 6.2 observa-se que:

- A ordem de desempenho dos algortimos foi a mesma para as duas técnicas utilizadas pelos reconheceodres. Em todos os casos o algoritmo " bottom-up " com pré-ênfase mostrou-se superior aos demais.

- A utilização da pré-ênfase melhorou consideralmente o desempenho dos algoritmos " bottom-up " e " top-down ". O uso da pré-ênfase torna mais precisa a detecção dos pontos inicial/final das palavras que apresentam fricativas nessas regiões.

- O desempenho dos detectores " bottom-up " e " top-down " é bastante parecido. Mais uma vez isso é justificado pelo baixo nível do ruído de fundo. As diferenças ocorrem para palavras começando ou terminando com fricativas.

- O bom desempenho do detector " endpoints " é explicado pelo uso da taxa de cruzamento de zero e baixo nível de ruído de fundo. Quando a relação sinal/ruído é baixa, a taxa de cruzamento de zero deixa de ser um bom parâmetro para detectar fricativas.

- Variando-se os parâmetros dos reconhecedores (números de estados e tamanho do " codebook "), o desempenho dos mesmos variou da mesma forma para todos os detectores de início/fim. Isso mostra que essa mudança de desempenho é devida às alterações na topologia dos reconhecedores. Pode-se concluir que não é possivel recuperar os erros do detector de início/fim alterando-se a topologia do reconhecedor.

algoritmo de detecção de início/fim	tamanho do codebook	número de estados	% de acerto
" endpoints "	64	5	89.29
		10	91.06
		15	92.65
	128	5	89.65
		10	91.47
		15	92.00
" top-down "	64	5	83.29
		10	84.18
		15	86.06
	128	5	84.82
		10	86.82
		15	87.06
" top-down " com pré-ênfase	64	5	89.35
		10	90.29
		15	91.35
	128	5	90.29
		10	91.29
		15	91.94
" bottom-up "	64	5	87.18
		10	87.41
		15	89.12
	128	5	86.82
		10	87.41
		15	88.12
" bottom-up " com pré-ênfase	64	5	90.34
		10	91.88
		15	92.76
	128	5	90.76
		10	91.59
		15	92.00

Tabela 6.1: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto para os diferentes algoritmos de detecção de início/fim.

algoritmo de detecção de início/fim	% de acerto
" endpoints "	91.29
" top-down "	85.59
" top-down " com pré-ênfase	91.65
" bottom-up "	89.06
" bottom-up " com pré-êmfase	93.00

Tabela 6.2: Desempenho de reconhecedores de palavras isoladas independente do locutor usando MLP para os diferentes algoritmos de detecção de início/fim.

A partir dos resultados das tabelas 6.1 e 6.2 e levando-se em conta que os erros na detecção de início/fim de cada algoritmo ocorriam em palavras diferentes, resolveu -se combinar dois dos algoritmos para a obtenção de reconhecedores mais robustos. Como os detectores " bottom-up " e " top-down " são similares e o desempenho do primeiro foi levemente superior, combinou-se o algoritmo " bottom-up " com pré-ênfase com o algoritmo " endpoints ". Esse resultado é apresentado nas tabelas 6.3 e 6.4.

algoritmo de detecção de início/fim	tamanho do codebook	número de estados	% de acerto
" bottom-up " com pré-ênfase e " endpoints "	64	5	90.82
		10	93.24
		15	93.41
	128	5	90.94
		10	92.88
		15	93.53

Tabela 6.3: Desempenho dos reconhecedores de palavras isoladas independente do locutor usando HMM discreto e combinando dois algoritmos de início/fim.

algoritmo de detecção de início/fim	% de acerto
" bottom-up " com pré-ênfase e " endpoints "	92.00

Tabela 6.4: Desempenho dos reconhecedores de palavras isoladas independente do locutor usando MLP combinando dois algoritmos de início/fim.

A partir dos resultados das tabelas 6.3 e 6.4 observa-se que houve um aumento na taxa de acerto apenas para os reconhecedores usando HMM, enquanto para os reconhecedores usando MLP ocorreu uma diminuição.

Em uma tentativa de melhorar o desempenho dos reconhecedores, adiantou-se o ponto inicial em 5 quadros e atrasou-se o ponto final também em 5 quadros. As tabelas 6.5 e 6.6 apresentam os resultados obtidos.

algoritmo de detecção de início/fim	tamanho do codebook	número de estados	% de acerto
" bottom-up " com pré-ênfase e " endpoints " e deslocamento de 5 quadros	64	5	92.65
		10	93.76
		15	94.24
	128	5	92.59
		10	94.12
		15	94.82

Tabela 6.5: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto e combinando dois algoritmos de início/fim e deslocamento de 5 quadros nos pontos inicial/final.

algoritmo de detecção de início/fim	% de acerto
" bottom-up " com pré-ênfase e " endpoints " e deslocamento de 5 quadros	93.41

Tabela 6.6: Desempenho dos reconhecedores de palavras isoladas independente do locutor usando MLP e combinando dois algoritmos de início/fim e deslocamento de 5 quadros nos pontos inicial/final.

Analisando-se os resultados obtidos, pode-se perceber que pequenas alterações nos pontos de início/fim influenciam o desempenho dos reconhecedores. Isso mostra a grande importância de uma detecção precisa dos pontos limitantes de uma palavra.

O melhor desempenho dos detectores de início/fim ocorreu quando combinou-se os detectores " endpoints " e " bottom-up " com pré-ênfase e deslocou-se os pontos limitantes em 5 quadros. Assim, adotou-se esse detector em todos os reconhecedores que foram implementados neste trabalho.

6.4 Quantização Vetorial

Quantização é o processo de aproximação de sinais de amplitudes contínuas por sinais de amplitudes discretas.

Um quantizador vetorial K-dimensional de N níveis é um processo que determina para cada vetor $\boldsymbol{x} = (x_1, ..., x_K)$, um vetor de reprodução $\hat{\boldsymbol{x}}_i = q(\boldsymbol{x})$, o qual pertence a um alfabeto de reprodução $\hat{\mathrm{A}} = \{\hat{\boldsymbol{x}}_i\}$, $i = 1, ..., N$. O alfabeto $\hat{\mathrm{A}}$ é denominado " codebook ", N é o número de vetores código e cada $\hat{\boldsymbol{x}}_i = (\hat{x}_{i1}, ..., \hat{x}_{iK})$ é um vetor código [78, 79, 80, 81, 82, 83].

No processo de quantização, cada vetor $\boldsymbol{x}$ é comparado com cada vetor código $\hat{\boldsymbol{x}}_i$ usando-se uma medida de distorção. O vetor código que resultar em uma menor distorção é escolhido para representar o vetor $\boldsymbol{x}$. Assim, é importante a escolha de uma medida de distorção apropriada.

6.4.1 Medidas de Distorção

As seguintes medidas de distorção foram empregadas na geração de " codebooks ":

Distância Euclidiana

É a mais comum medida de distorção, principalmente devido à sua simplicidade. Essa medida é definida por [84]:

$$d_2(x, \hat{x}_i) = \sum_{j=1}^{K} (x_j - \hat{x}_{ij})^2 \tag{6.1}$$

Distância de Mahalanobis

Nessa medida, para cada coeficiente do vetor x podem ser atribuídos pesos diferentes e assim eles contribuem de forma diferenciada para a distorção. Define-se essa medida como [84]:

$$d_w(x, \hat{x}_i) = (x - \hat{x}_i) W^{-1} (x - \hat{x}_i)' \tag{6.2}$$

onde W é uma matriz definida positiva, a qual pode ser diagonal ou cheia.

É usual o uso da matriz Covariância como matriz de ponderação.

Funçao Densidade de Probabilidade Gaussiana

Neste caso cada vetor código é representado por um vetor média μ_i e matriz covariância U_i. A distância entre o vetor x e um vetor código é dada por [37]:

$$d(x, \hat{x}_i) = Pr_{\hat{x_i}} G(x, \mu_i, U_i) \tag{6.3}$$

onde:

$Pr_{\hat{x_i}}$ é a probabilidade de ocorrer o vetor código $\hat{x_i}$

$G(x, \mu_i, U_i)$ é a função densidade de probabilidade Gaussiana

A função densidade de probabilidade Gaussiana multidimensional é dada por:

$$G(x, \mu_i, U_i) = \frac{1}{(2\pi)^{dim/2} |U_i|^{1/2}} \exp\{-(x - \mu_i) U_i^{-1} (x - \mu_i)'/2\} \tag{6.4}$$

onde:

dim é a dimensão do vetor x

$|U_i|$ é o determinante da matriz covariância U_i

U_i^{-1} é a matriz covariância inversa

Funçao Densidade de Probabilidade Gaussiana Simplificada

Esta distância é uma simplificação da distância anterior, trabalhando-se com o logaritmo. Assim, considerando todos os vetores código equiprováveis e desprezando-se o termo $\ln(1/(2\pi)^{dim/2}|U_i|^{1/2})$, pode-se definir essa distância como [37]:

$$d(x, \hat{x}_i) = (x - \mu_i)U_i^{-1}(x - \mu_i)' \tag{6.5}$$

6.4.2 Projeto do " Codebook "

A geração de um " codebook ", o qual minimiza uma medida de distorção sobre uma grande seqüência de treinamento, requer um processo iterativo. Um método bastante utilizado para o projeto de " codebooks " é o algoritmo iterativo conhecido como algoritmo LBG (Linde-Buzo-Gray) o qual utiliza o procedimento " K-means " e a técnica " splitting " [78].

Esse algoritmo divide a sequência de treinamento em N células C_i , satisfazendo as condições necessárias para ser considerado ótimo.

A figura 6.6 apresenta um diagrama do procedimento do algoritmo LBG .

Os " codebooks " implementados neste trabalho usaram o algoritmo LBG com busca exaustiva, sendo que o treinamento dos mesmos era finalizado quando a diferença entre as distorções médias de duas iterações consecutivas fosse menor que 1%. Utilizou-se a mesma seqüência de treinamento para todos os " codebooks ".

6.4.3 Comparação entre as Medidas de Distorção

Para avaliar as diferentes medidas de distorção, foram utilizados reconhecedores de palavras isoladas independentes do locutor empregando a técnica HMM (discreto) e os " codebooks " foram criados usando as diferentes medidas de distorção.

Novamente, os reconhecedores de fala implementados usavam HMM discreto, 10 estados e coeficientes " Mel Frequency Cepstrum " com subtração da média espectral (16 coeficientes por vetor). Para o treinamento dos reconhecedores foi utilizado o algoritmo " Forward-Backward " e o reconhecimento foi realizado usando o algoritmo " Forward ". Foi empregada detecção de início/fim. Na avaliação do desempenho dos reconhecedores foi usado vocabulário com 50 palavras (apêndice A). Foi utilizada a mesma seqüência de treinamento para criar os " codebooks " e treinar os HMMs, a qual era formada por uma elocução de cada palavra do vocabulário.

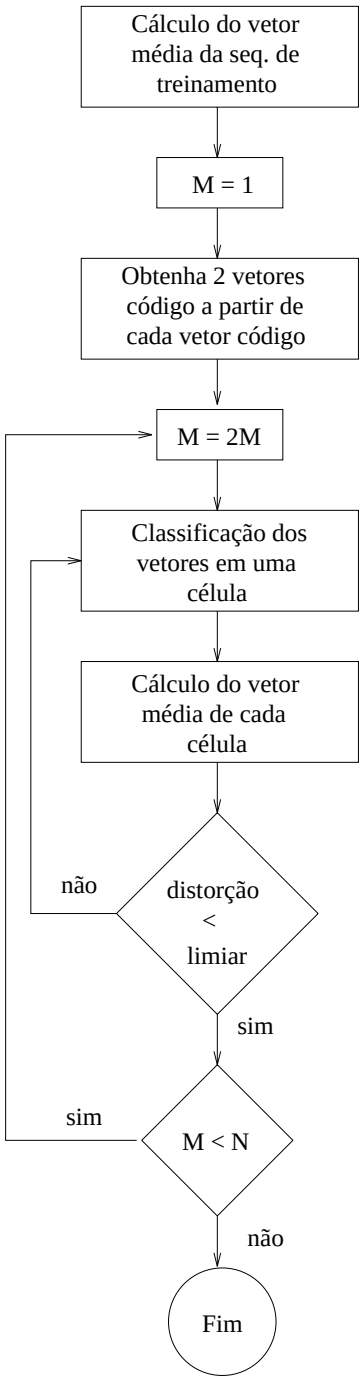

Figura 6.6: Procedimento do algoritmo LBG

Visando uma comparação mais ampla variou-se o número de estados dos HMMs e também o tamanho do " codebook ", os quais foram combinados e resultaram em diferentes reconhecedores. Os resultados obtidos são mostrados na tabela 6.7.

tipo de distância	tamanho do codebook	número de estados	% de acerto
Euclidiana	64	10	93.76
		15	94.24
	128	10	94.12
		15	94.82
Mahalanobis com matriz covariância diagonal	64	10	81.71
		15	82.65
	128	10	84.35
		15	84.76
Mahalanobis com matriz covariância cheia	64	10	80.41
		15	81.88
	128	10	82.35
		15	81.35
pdf gaussiana simplificada com matriz covariância diagonal	64	10	93.82
		15	93.71
	128	10	93.41
		15	93.71
pdf gaussiana simplificada com matriz covariância cheia	64	10	94.18
		15	94.00
	128	10	93.29
		15	93.12
pdf gaussiana com matriz covariância diagonal	64	10	93.59
		15	93.41
	128	10	93.47
		15	94.29
pdf gaussiana com matriz covariância cheia	64	10	94.29
		15	95.12
	128	10	93.59
		15	93.71

Tabela 6.7: Desempenho de reconhecedores de fala independente do locutor para codebooks usando HMM discreto e diferentes tipos de medidas de distorção.

Analisando-se os resultados da tabela 6.7 observa-se:

- Para " codebooks " com 64 vetores código, o melhor desempenho foi obtido pelos reconhecedores utilizando a função densidade de probabilidade gaussiana com matriz covariância cheia, seguido pelos reconhecedores que usam a função densidade de probabilidade gaussiana simplificada. Isso evidencia a superioridade dessas medidas em relação às outras.

- Para " codebooks " com 128 vetores código, a distância Euclidiana foi superior, vindo a seguir a distância função densidade de probabilidade gaussiana com matriz diagonal.

- A degradação de desempenho dos reconhecedores que utilizam a função densi-

dade de probabilidade gaussiana com matriz covariância cheia com o aumento do tamanho do " codebook " é explicada pelo fato de que aumentou-se o número de parâmetros a ser estimado e manteve-se a mesma seqüência de treinamento. Isso acarreta erro principalmente na estimativa da matriz covariância. Com o uso da matriz covariância diagonal, nota-se uma alteração menor no desempenho pois esta apresenta um número menor de parâmetros para ser estimado.

- Não foram obtidos bons resultados empregando-se a distância de Mahalanobis. Neste caso, usou-se como matriz de ponderação a matriz covariância de toda a seqüência de treinamento. Isso não é apropriado pois dessa forma não são enfatizadas as características dos vetores de cada célula. A melhor solução é utilizar uma matriz covariância para cada célula como mostram os resultados obtidos com a distância função densidade de probabilidade gaussiana simplificada.

Os esforços computacionais associados ao uso da função densidade de probabilidade gaussiana com matriz cheia são enormes pois envolvem a inversão de matrizes e a cálculo de muitos parâmetros. Para efeito de comparação, o tempo de CPU (estações Sun SPARK 20) utilizado para gerar um " codebook " com 64 vetores código utilizando-se função densidade de probabilidade gaussiana com matriz cheia é da ordem de 24 horas, enquanto para função densidade de probabilidade gaussiana com matriz diagonal gasta-se aproximadamente 1 hora. Para criar esse mesmo " codebook " utilizando-se a distância Euclidiana são necessários 30 minutos.

Na criação dos " codebooks " incluiu-se nos algoritmos um procedimento para a eliminação de células vazias ou que continham apenas 1 vetor. Isso é necessário quando se usam distâncias que incluem a matriz covariância pois neste caso podem ocorrer erros durante a inversão dessas matrizes. Para evitar esse problema é necessário o aumento da seqüência de treinamento à medida que aumenta-se o tamanho do " codebook ".

Finalizando, reconhecedores que usaram " codebooks " com distância Euclidiana apresentaram um bom desempenho e os esforços computacionais requeridos foram muito menores. Dessa forma, resolveu-se adotar a distância Euclidiana como medida de distorção na geração de " codebooks " para os HMMs discretos. Pode-se afirmar que as perdas devido a essa escolha foram mínimas.

6.5 Análise Espectral

A escolha do tipo de parâmetro para representar o sinal de fala em um sistema de reconhecimento não é uma tarefa fácil e é muito importante pois o desempenho do reconhecedor depende do tipo de parâmetro escolhido. Nesse sentido, várias formas de análise espectral do sinal de fala para cálculo de parâmetros têm sido estudadas. As mais usadas em reconhecimento de fala são os métodos de Bancos de Filtros, Predição Linear e Transformada de Fourier.

O processo de obtenção dos parâmetros envolve as seguintes etapas: pré-ênfase, segmentação e janelamento do sinal de fala e cálculo de parâmetros. Na figura 6.7 é mostrada a seqüência dessas operações.

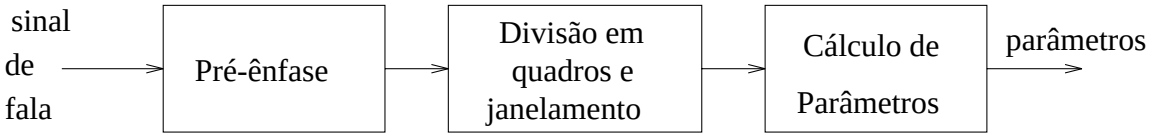

Figura 6.7: Procedimento para a obtenção de parâmetros do sinal de fala.

A pré-ênfase tem por função compensar a atenuação nas altas freqüências devido aos pulsos glotais e ao efeito da radiação dos lábios. A pré-ênfase é realizada por um filtro FIR com a seguinte transformada Z [85]:

$$h(z) = 1 - \mu z^{-1} \tag{6.6}$$

onde μ assume valores próximos a 1. Neste trabalho usou-se $\mu = 0.95$.

Devido à não estacionariedade do sinal de fala, este é dividido em quadros para o cálculo dos parâmetros. A duração dos quadros é escolhida de forma que os parâmetros sejam válidos durante o período de duração dos mesmos (considera-se o sinal de fala dentro do quadro como quase estacionário) e representa um compromisso entre a taxa de mudança do espectro e a complexidade do sistema.

Para o cálculo dos parâmetros, é usual empregar-se intervalos de análise maiores que os quadros, com superposição de intervalos de análise adjacentes como uma forma de se evitar grandes flutuações dos parâmetros calculados. Nesse caso, uma parte das amostras finais do intervalo anterior e uma parte das amostras iniciais do intervalo posterior fazem parte do intervalo atual de análise. Os intervalos de análise são multiplicados por uma janela, usualmente a janela de Hamming, que produzirá uma suavização dos parâmetros. Essa janela dá maior ênfase às amostras localizadas no centro da janela e é representada pela equação [85]:

$$h(n) = \begin{cases} 0.54 - 0.46\cos(2\pi n/(N-1)) & \text{se } 0 \leq n \leq N-1 \\ 0 & \text{caso contrário} \end{cases} \tag{6.7}$$

onde N é o número de amostras da janela

Em reconhecimento de fala usam-se quadros com duração entre 10 e 20 ms e janelas com duração entre 20 e 30 ms. Neste trabalho adotou-se o valor de 10 ms para os quadros e 20 ms para a janela de Hamming.

A figura 6.8 ilustra o processo de superposição de janelas.

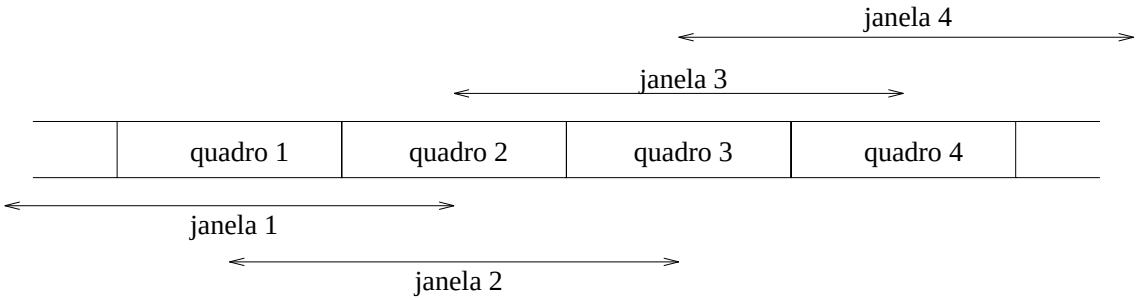

Figura 6.8: Processo de superposição de janelas para o cálculo de parâmetros.

Para representar o sinal de fala, foram calculados e analisados os seguintes parâmetros:

- Coeficientes LPC

  São os coeficientes a_k do preditor na análise LPC. Para o cálculo desses coeficientes foi usado o método de autocorrelação e o algoritmo de Durbin [86, 87, 88].

- Coeficientes Parcor

  Os coeficientes parcor k_i são obtidos como um subproduto do cálculo dos coeficientes LPC pelo método de autocorrelação. Esses coeficientes podem ser obtidos diretamente dos coeficientes do preditor usando-se a seguinte forma recursiva [86]:

$$k_i = a_i^{(i)} \tag{6.8}$$

$$a_j^{(i-1)} = \frac{a_j^{(i)} + a_i^{(i)} a_{i-j}^{(i)}}{1 - k_i^2}, \quad 1 \leq j \leq i-1 \tag{6.9}$$

  onde i varia de P a 1 e inicialmente tem-se:

$$a_j^{(P)} = a_j, \quad 1 \leq j \leq P \tag{6.10}$$

sendo P a ordem do preditor.

- Coeficientes Razão Log-Área

  Os coeficientes razão log-área g_i são iguais ao logaritmo da razão das áreas de seções adjacentes do modelo de tubos sem perdas, o qual modela o trato vocal, e são calculados a partir dos coeficientes parcor usando a seguinte fórmula [86]:

$$g_i = \log \frac{1 - k_i}{1 + k_i} \tag{6.11}$$

- Coeficientes LPC Cepstrais

  Os coeficientes cepstrais podem ser obtidos por um método iterativo diretamente dos coeficientes LPC, usando-se as equações [86]:

$$c_1 = a_1 \tag{6.12}$$

$$c_n = a_n + \sum_{m=1}^{n-1} \frac{m}{n} c_m a_{n-m} \ , \quad 2 \leq n \leq P \tag{6.13}$$

$$c_n = \sum_{m=1}^{n-1} \frac{m}{n} c_m a_{n-m} \ , \quad n > P \tag{6.14}$$

- Coeficientes FFT Cepstrais

  Para o cálculo desses coeficientes, calcula-se a Transformada Discreta de Fourier do sinal de fala, a seguir o logaritmo da magnitude da Transformada Discreta de Fourier e depois a Transformada Discreta de Fourier Inversa do logaritmo da magnitude [85].

$$c(n) = \frac{1}{M} \sum_{k=0}^{M-1} \log_{10} |S(k)| e^{(2\pi/M)kn} \ , \quad 1 \leq n \leq N_c \tag{6.15}$$

  onde:
  M é o número de pontos da DFT
  N_c é o número de coeficientes cepstrais

- Coeficientes " Liftered Cepstral "

  São calculados multiplicando-se os coeficientes cepstrais por uma janela de ponderação, a qual de-enfatiza os coeficientes nas extremidades devido à maior sensibilidade destes ao ruído [8].

$$c_l(n) = c(n) \cdot W_c(n) \ , \quad 1 \leq n \leq N_c \tag{6.16}$$

onde:

$$W_c(n) = 1 + (N_c/2)\sin(\pi n/N_c) \tag{6.17}$$

- Coeficientes " Bilinear Transformed Cepstral "

  Esses coeficientes são obtidos a partir dos coeficientes LPC cepstrais usando uma transformação bilinear. Usualmente a transformação bilinear produz uma compressão na escala de freqüências [85]. Os coeficientes " bilinear transformed " são dados pela recursão:

$$c_{bt}^{(i)}(0) = \alpha_{bt} c_{bt}^{(i-1)}(0) + c(N_c - i) \tag{6.18}$$

$$c_{bt}^{(i)}(1) = \alpha_{bt} c_{bt}^{(i-1)}(1) + (1 - \alpha_{bt}^2) c_{bt}^{(i-1)}(0) \tag{6.19}$$

$$c_{bt}^{(i)}(n) = \alpha_{bt}(c_{bt}^{(i-1)}(n) - c_{bt}^{(i)}(n-1)) + c_{bt}^{(i-1)}(n-1), 2 \leq n \leq N_{bt} \tag{6.20}$$

  onde i varia de 0 a N_c e no final tem-se:

$$c_{bt}(n) = c_{bt}^{(N_c)}(n) \tag{6.21}$$

  N_c é o número de coeficientes ceptrais
  N_{bt} é o número de coeficientes Bilinear Transformed
  α_{bt} é o coeficiente da transformação bilinear

- Coeficientes " Mel Frequency Cepstrum "

  O cálculo desses coeficientes envolve o uso de um banco de filtros espaçados na escala Mel e o cálculo do logaritmo da energia na saída de cada filtro seguido de uma Transformada Discreta do cosseno (DCT) [89]:

$$c(n) = \sum_{k=1}^{M} \log_{10} X(k) \cos(n(k - 1/2)\pi/M) \ , \ 1 \leq n \leq N \tag{6.22}$$

  onde:
  $X(k)$ é a energia na saída do k-ésimo filtro
  M é o número de filtros
  N é o número de coeficientes

  A escala Mel é apresentada na tabela 6.8 [85]. O banco de filtros normalmente é implementado como um banco de filtros triangulares com largura de faixa BW e posicionados nas freqüências centrais discriminadas.

Índice	Freqüência Central (Hz)	BW (Hz)
1	100	100
2	200	100
3	300	100
4	400	100
5	500	100
6	600	100
7	700	100
8	800	100
9	900	100
10	1000	124
11	1149	160
12	1320	184
13	1516	211
14	1741	242
15	2000	278
16	2297	320
17	2639	367
18	3031	422
19	3482	484
20	4000	556

Tabela 6.8: Banco de Filtros na Escala Mel.

- Coeficientes " Mel Frequency Cepstrum " Normalizados

  São calculados pela seguinte equação [90]:

$$c(n) = \sum_{k=1}^{M} \log_{10}(X(k)/L_k) \cos(n(k-1/2)\pi/M) \ , \quad 1 \leq n \leq N \qquad (6.23)$$

  onde:
  $X(k)$ é a energia na saída do k-ésimo filtro
  L_k é a largura de faixa do k-ésimo filtro em Hz
  M é o número de filtros.
  N é o número de coeficientes

- Coeficientes " Bark Frequency Cepstrum "

  Esses coeficientes são obtidos da mesma forma que os coeficientes Mel mas utilizando um banco de filtros espaçados segundo a escala Bark.

  A escala Bark é apresentada na tabela 6.9 [85].

- Coeficientes " Linear Frequency Cepstrum "

  Repete-se o mesmo procedimento descrito para os coeficientes Mel e Bark, mas usando-se um banco de filtros linearmente espaçados. O espaçamento entre as

Índice	Freqüência Central (Hz)	BW (Hz)
1	50	100
2	150	100
3	250	100
4	350	100
5	450	110
6	570	120
7	700	140
8	840	150
9	1000	160
10	1170	190
11	1370	210
12	1600	240
13	1850	280
14	2150	320
15	2500	380
16	2900	450
17	3400	550
18	4000	700

Tabela 6.9: Banco de Filtros na Escala Bark.

freqüências centrais dos filtros é de 100 Hz e a faixa dos mesmos também é de 100 Hz.

- Coeficientes " Line Spectrum Frequencies "

  Esses coeficientes são as freqüências dos zeros no círculo unitário dos seguintes polinômios [5]:

$$P(z) = A(z) + z^{-(P+1)}A(z^{-1}) \tag{6.24}$$

$$Q(z) = A(z) - z^{-(P+1)}A(z^{-1}) \tag{6.25}$$

  onde: $A(z)$ é o filtro inverso do preditor dado por:

$$A(z) = 1 - \sum_{k=1}^{P} a_k z^{-k} \tag{6.26}$$

  Uma forma de se obter esses coeficientes é calcular a DFT dos coeficientes dos polinômios $P(z)$ e $Q(z)$. Os zeros podem ser estimados por interpolação. Os zeros w_i de $P(z)$ e $Q(z)$ ocorrem alternadamente e tem-se: $0 < w_1 < w_2 < \ldots < w_P < \pi$.

- Coeficientes " Mel Frequency "

  São iguais ao logaritmo da energia na saída de cada filtro do Banco de Filtros na escala Mel.

- Coeficientes " Bark Frequency "

  São iguais ao logaritmo da energia na saída de cada filtro do Banco de Filtros na escala Bark.

- Coeficientes PLP

  Esses coeficientes são obtidos usando o método de análise PLP (" Perceptually-based Linear Prediction Analysis ") , o qual é baseado em conceitos psico-acústicos do processamento do sistema periférico auditivo. Esse método apresenta os seguintes passos [91]:

  1. Cálculo da Transformada Discreta de Fourier do sinal de fala
  2. Cálculo do espectro de potência.

  $$P(w) = Re[S(w)]^2 + Im[S(w)]^2 \tag{6.27}$$

  3. Cálculo das saídas de um banco de filtros na escala Bark usando o espectro de potências como entrada.
  4. Pré-ênfase utilizando curva de equalização de sonoridade

  $$\Xi(\Omega(\omega)) = E(\omega)\Theta(\Omega(\omega)) \tag{6.28}$$

  onde:

  $$E(\omega) = \frac{(\omega^2 + 56.8\ 10^6)\omega^4}{(\omega^2 + 6.3\ 10^6)^2(\omega^2 + 0.38\ 10^9)} \tag{6.29}$$

  $\Theta(\Omega(\omega))$ é a saída do k-ésimo filtro
  w é a freqüência em rad/s
  $\Omega(\omega)$ é a freqüência na escala BARK correspondente à freqüência w

  5. Compressão Intensidade - Sonoridade

  $$\Phi(\Omega) = \Xi(\Omega)^{1/3} \tag{6.30}$$

  6. Cálculo da Transformada Discreta de Fourier Inversa
  7. Os primeiros $M + 1$ valores da transformada inversa são usados como valores de autocorrelação no método de predição linear. A resolução dessas equações fornece os coeficientes PLP.

- Coeficientes PLP Parcor

  São os coeficientes Parcor obtidos pelo método de predição linear na análise PLP.

- Coeficientes PLP Cepstrais

  Esses coeficientes são obtidos a partir dos coeficientes PLP usando-se as equações 6.12 a 6.14.

- Coeficientes RASTA-PLP

  O cálculo desses coeficientes envolve o uso da técnica RASTA (" Relative Spectral ") incorporada à análise PLP. Os passos desse procedimento serão descritos a seguir [92]:

  1. Cálculo da Transformada Discreta de Fourier do sinal de fala
  2. Cálculo do espectro de potência.
  3. Cálculo das saídas de um banco de filtros na escala Bark usando o espectro de potências como entrada.
  4. Transformação da amplitude spectral (saídas dos filtros) por uma compressão não linear (logaritmo)
  5. Filtragem da componente espectral transformada. Pode-se usar o seguinte filtro:
  $$H(z) = 0.1z^4 \frac{2 + z^{-1} - z^{-3} - 2z^{-4}}{1 - 0.98z^{-1}} \tag{6.31}$$
  6. Transformação da saída do filtro por uma expansão não linear (função exponencial)
  7. Pré-ênfase utilizando curva de igual sonoridade
  8. Compressão Intensidade - Sonoridade
  9. Cálculo da Transformada de Fourier Inversa
  10. Os primeiros $M + 1$ valores da transformada inversa são usados como valores de autocorrelação no método de predição linear. A resolução dessas equações fornece os coeficientes RASTA-PLP.

- Coeficientes RASTA-MEL

  Para a obtenção dos coeficientes RASTA-MEL usa-se o mesmo procedimento descrito para os coeficientes RASTA-PLP, substituindo-se o banco de filtros na escala BARK por um banco de filtros na escala MEL.

- Energia

A energia é calculada usando a seguinte equação [85]:

$$E = \sum_{i=0}^{N-1} s^2(i) \tag{6.32}$$

onde N é o número de amostras da janela de análise.

Neste trabalho usou-se a energia normalizada , dada por:

$$E_N = log(E) - log(E_{max}) \tag{6.33}$$

onde E_{max} corresponde à janela de máxima energia para uma dada palavra.

6.5.1 Comparação entre os Diferentes Parâmetros

Para comparar os diferentes parâmetros foram implementados reconhecedores de palavaras isoladas independentes do locutor usando as técnicas HMM e Redes Neurais. Foi utilizado vocabulário com 50 palavras na avaliação dos reconhecedores.

Os reconhecedores HMM apresentavam as seguintes características:

- 10 estados
- Densidade de probabilidade discreta com 128 símbolos
- Treinamento: algoritmo " Forward-Backward "
- Reconhecimento: algoritmo " Forward "

Para esta comparação implementou-se reconhecedores HMM sem detecção de início/fim (HMM 1) e com o uso do detector de início/fim (HMM 2). No primeiro caso, deixou-se um grande intervalo de silêncio no início e fim de cada palavra de forma que nenhuma parte da mesma fosse cortada. A duração desse intervalo variou de palavra para palavra, sendo a duração total do sinal gravado igual para todas as palavras.

A Rede Neural implementada tinha as seguintes características:

- Rede " Multilayer Perceptron "
- 100 neurônios na camada escondida
- 50 neurônios na camada de saída correspondendo a cada uma das palavras do vocabulário
- Número de entradas igual a 80 vezes o número de coeficientes por vetor
- Usou-se detecção de início/fim

Para cálculos dos parâmetros foram usados os seguintes valores:

- Coeficiente de pré-ênfase: 0.95
- Tamanho do quadro: 10 ms
- Tamanho da janela: 20 ms

- Ordem LPC: 12
- Número de pontos para cálculo da FFT: 512
- Número de filtros Mel: 19
- Número de fitros BARK: 17
- Número de filtros Lineares: 39
- Coeficiente da Transformação Bilinear: 0.6
- No cálculo dos coeficientes PLP e RASTA-PLP foi usado um banco de filtros triangulares com 17 filtros na escala BARK
- No cálculo dos coeficientes RASTA-MEL foi usado um banco de filtros triangulares com 17 filtros na escala MEL

Como o número de entradas da rede neural é fixo, usou-se janelas de 20 ms com superposição variável para cálculo dos parâmetros, mantendo-se fixo o número de quadros.

Os resultados obtidos estão apresentados na tabela 6.10.

Tipo de Coeficiente	Número de coeficientes	% de Acerto		
		HMM 1	HMM 2	MLP
LPC	12	80.00	82.41	86.24
PARCOR	12	85.76	86.12	86.18
Razão Log Area	12	82.76	87.41	86.18
Cepstral	12	87.12	87.88	91.00
Bilinear transformed Cep.	13	89.94	90.94	91.06
liftered Cep.	12	88.24	88.88	91.06
Mel frequency Cep.	12	92.06	91.94	92.29
Mel frequency Cep.	16	92.82	92.24	93.41
Mel frequency Cep.	18	92.65	92.29	92.76
Bark frequency Cep.	12	92.12	92.82	92.41
FFT Cep.	12	88.71	89.47	91.41
Line Spectrum frequencies	12	71.35	82.41	73.82
Linear frequency Cepstrum	12	89.65	90.76	91.88
Mel frequency Cep. normalizado	12	92.06	92.65	92.18
Mel frequency	12	85.35	87.35	92.29
Bark frequency	12	85.12	86.71	90.71
PLP	5	87.53	89.65	90.06
PLP parcor	5	88.06	90.41	88.65
PLP cepstral	5	89.06	89.29	89.59
RASTA-PLP	5	87.00	86.06	89.06
PLP	12	86.71	88.65	92.47
PLP parcor	12	89.53	93.35	92.12
PLP cepstral	12	90.12	91.35	92.53
RASTA-PLP	12	85.88	88.06	89.35
RASTA-MEL	12	87.53	85.94	81.35

Tabela 6.10: Desempenho de reconhecedores de palavras isoladas independente do locutor utilizando diferentes tipos de parâmetros, onde: HMM 1 = HMM discreto sem detecção de início/fim, HMM 2 = HMM discreto com detecção de início/fim e MLP = " Multilayer Perceptron ".

Analisando os resultados obtidos observamos que:

- Para as duas técnicas (HMM e MLP) os melhores desempenhos foram apresentados pelos reconhecedores usando coeficientes " Mel Frequency Cepstrum " e " Bark Frequency Cepstrum ", com uma pequena superioridade deste último.

- A ordem de desempenho dos outros coeficientes foi parecida para as duas técnicas.

- De uma forma geral, os coeficientes obtidos usando banco de filtros em escalas não lineares ou variações destes apresentaram os melhores resultados, mostrando que parâmetros que representam o espectro do sinal de fala obtidos pelo uso de uma escala de freqüência não linear, que procura reproduzir a resposta em freqüência do sistema periférico auditivo, são os mais indicados para o uso em reconhecedores.

- Comparando-se as duas formas de obtenção dos coeficientes cepstrais (LPC e FFT) observa-se um melhor desempenho quando se usa a FFT para o cálculo dos mesmos. Conseguiu-se um melhor desempenho aplicando-se a transformação bilinear aos coeficientes cepstrais. Isso reforça a necessidade de usar uma escala de freqüências não linear.

- Entre os coeficientes derivados dos coeficientes LPC, estes apresentaram o pior desempenho independentemente da técnica utilizada pelos reconhecedores.

- Para a maioria dos reconhecedores empregando HMM, ocorreu um aumento na taxa de acerto quando usou-se a detecção de início/fim, indicando a necessidade do uso de detectores de início/fim no modelo HMM. Entretanto, alguns tipos de coeficientes são mais sensíveis aos erros do detector de início/fim, acarretando uma diminuição na taxa de acerto.

- A partir dos resultados pode-se afirmar que os coeficientes " Line Spectrum Frequencies " são os menos apropriados para uso em reconhecimento de fala.

- As técnicas PLP e RASTA mostraram-se superiores à técnica LPC, indicando a necessidade de um melhor tratamento do sinal de fala. Entretanto, o desempenho dessas técnicas foi inferior ao obtido usando-se os coeficientes Mel e Bark. Apenas o coeficiente PLP parcor apresentou um bom desempenho quando usou-se o detector de início/fim.

A partir dessas considerações, pode-se concluir que para sistemas com baixo nível de ruído, o coeficiente " Mel Frequency Cepstrum " é um bom parâmetro para representar o sinal de fala para fins de reconhecimento. Assim, decidiu-se usar esse tipo de parâmetro (16 coeficientes por vetor) nas demais simulações para avaliação dos reconhecedores.

6.5.2 Transformações dos Coeficientes

A partir dos resultados da análise anterior, testou-se algumas modificações nos coeficientes Mel de forma a melhorar o desempenho dos reconhecedores.

- Suavização dos Coeficientes " Mel Frequency Cepstrum "

  Esse coeficientes são obtidos calculando-se a média ponderada dos coeficientes de $2k+1$ quadros consecutivos, como mostra a seguinte equação [93]:

$$\hat{c}_i(n) = \sum_{j=-k}^{+k} \frac{c_{i+j}(n)}{2k+1} \tag{6.34}$$

- Subtração da Média Espectral (SME)

  Esse método usa o princípio de que a média do espectro de um sinal de fala representa a distorção do canal. Assim, removendo-se a média espectral remove-se a distorção do canal [55, 94, 95].

  A aplicação desta técnica consiste em estimar a média de cada coeficiente espectral de um sinal e subtraí-la dos valores dos coeficientes de cada quadro. Dessa forma, tem-se:

$$c_i^{'}(n) = c_i(n) - \bar{c}(n) \tag{6.35}$$

- Variação Temporal

  Como uma forma de melhor caracterizar as variações do sinal de fala ao longo do tempo estão sendo usadas as derivadas temporais dos parâmetros propostos .

  Para cálculo das derivadas, basicamente três métodos têm sido usados [85, 8]:

$$delta(n) = \sum_{k=-K}^{K} ks_{i-k}(n)/(2K+1) \tag{6.36}$$

$$delta(n) = s(n) - s(n-1) \tag{6.37}$$

$$delta(n) = s(n-K) - s(n+K) \tag{6.38}$$

Novamente, para a comparação desses parâmetros foram implementados reconhecedores de palavras isoladas empregando as técnicas HMM e Redes Neurais com as mesmas características descritas no item anterior.

Inicialmente, variou-se o valor de K na equação 6.36 e e avaliou-se o desempenho de reconhecedores usando HMM discreto, sem detecção de início/fim e coeficientes " Mel frequency Cepstrum " com subtração da média. Os resultados da comparação são mostrados na tabela 6.11.

Valor de K	% Acerto
2	89.41
3	93.18
5	96.00
7	95.82
9	94.71

Tabela 6.11: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto com coeficientes delta-Mel com remoção da média, calculados segundo a equação 6.36.

Observa-se que o melhor desempenho ocorreu para $K = 5$. Isso mostra que 110 ms (11 quadros de 10 ms) é um bom intervalo para capturar as variações temporais. Assim, adotou-se nas demais simulações esse valor para o cálculo de coeficientes delta usando a equação 6.36.

Na tabela 6.12 são mostrados os desempenhos de reconhecedores de palavras isoladas com HMM discreto usando coeficientes delta calculados segundo as equações 6.36 (1), 6.37 (2) e 6.38(3) a partir de coeficientes Mel e coeficientes Mel com remoção da média. Na equação 6.36 usou-se $K = 5$ e na equação 6.38 usou-se $K = 2$.

Coeficientes	% de Acerto
Delta-Mel 1	95.47
Delta-Mel 2	31.94
Delta-Mel 3	89.18
Delta-Mel-SME 1	96.00
Delta-Mel-SME 2	31.12
Delta-Mel-SME 3	88.76

Tabela 6.12: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto com coeficientes delta, calculados segundo as equação 6.36 (1), 6.37 (2) e 6.38 (3).

Analisando-se a tabela 6.12 nota-se que os reconhecedores com melhores desempenhos usaram coeficientes delta calculados pela equação 6.36. Dessa forma, adotou-se esse método para cálculo de coeficientes delta neste trabalho.

Escolhido o método para calcular os coeficientes delta, foram implementados reconhecedores de palavras isoladas usando HMM discreto sem detecção de início/fim (HMM D 1), HMM discreto com detecção de início/fim (HMM D 2), HMM semicontínuo (HMM SC), HMM contínuo (HMM C) e " Multilayer Perceptron " (MLP) usando coeficientes obtidos a partir dos coeficientes Mel. Os reconhecedores usando HMM discreto e MLP são os mesmos descritos anteriormente e os outros apresentam as seguintes características:

HMM Semicontínuo:

- 10 estados
- " Codebook " com 128 vetores código com matriz covariância diagonal
- Treinamento: algoritmo " Forward-Backward "
- Reconhecimento: algoritmo " Forward "
- Usou-se detecção de início/fim

HMM Contínuo:

- 10 estados
- 3 misturas com matriz covariância diagonal
- Treinamento: algoritmo " Forward-Backward "
- Reconhecimento: algoritmo " Forward "
- Usou-se detecção de início/fim

Os resultados são apresentados na tabela 6.13.

Tipo de Coeficiente	% de Acerto				
	HMM D 1	HMM D 2	HMM SC	HMM C	MLP
Mel freq. Cep.	92.82	92.24	93.47	90.29	93.41
Mel freq. Cep.- SME	95.35	94.12	95.41	95.71	89.59
Mel freq. Cep. suavizado	89.88	89.82	89.71	86.18	92.94
Delta-Mel-SME	96.00	94.47	96.94	94.00	86.65
Delta-Mel	95.47	93.24	96.88	93.47	89.35

Tabela 6.13: Desempenho de reconhecedores de palavras isoladas com coeficientes obtidos a partir dos coeficientes Mel.

A partir dos resultados das tabela 6.13 observou-se que a remoção da média melhorou muito o desempenho dos reconhecedores utilizando a técnica HMM enquanto o desempenho dos reconhecedores utilizando MLP piorou. Além disso, deve ser ressaltado o bom desempenho dos coeficientes Delta para os HMMs. O uso de suavização apresentou bom resultado apenas para os reconhecedores usando MLP. Dessa forma, resolveu-se usar coeficientes Mel com remoção da média para os reconhecedores implementados usando HMM e coeficientes Mel sem remoção da média para os reconhecedores implementados usando " Multilayer Perceptron ".

Capítulo 7

AVALIAÇÃO DE RECONHECEDORES USANDO HMM

7.1 Introdução

De modo a avaliar o uso da técnica de Modelos Ocultos de Markov (HMM) em reconhecimento de palavras isoladas, foram implementados reconhecedores independentes do locutor usando HMM discretos, contínuos e semicontínuos. Para essa avaliação foram considerados os seguintes aspectos: taxa de acerto, tempo de reconhecimento, tempo de treinamento, quantidade de parâmetros, tipo de algoritmo para reconhecimento e treinamento, modelamento da duração de estado, tamanho da seqüência de treinamento e tamanho do vocabulário. Neste capítulo serão apresentados os resultados dessas avaliações.

O vocabulário utilizado nas simulações era formado por 50 palavras, algumas bastante similares tais como : sim e cinco, nove e norte, não e leão, certo e sete. As palavras foram faladas por 69 locutores, sendo 43 homens e 26 mulheres.

O número de locutores empregado na seqüência de treinamento era igual a 35, sendo 22 homens e 13 mulheres.

Foram utilizadas três diferentes seqüências de treinamento:

- Seqüência 1: formada por uma elocução de cada palavra por locutor.
- Seqüência 2: formada por duas elocuções de cada palavra por locutor.

- Seqüência 3: formada por três elocuções de cada palavra por locutor.

Todas a simulações foram realizadas em estações Sun SPARC 20.

A descrição completa da Base de Dados e Ambiente de Simulação está apresentada no Anexo A.

7.2 Reconhecedores Implementados

Os reconhecedores implementados utilizaram o modelo " left-rigth " com $\Delta = 2$. Um exemplo desse modelo é apresentado na figura 7.1.

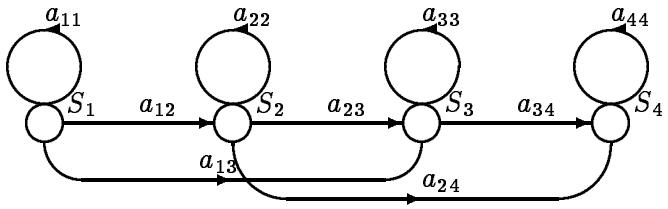

Figura 7.1: Modelo " left-right " com $\Delta = 2$.

Foram realizadas implementações permitindo que qualquer estado fosse o estado final e também restringindo o estado final ao último estado.

Em muitas simulações realizadas usou-se o mesmo número de estados para todas as palavras, mas em outras foram usados números de estados diferentes para cada palavra. Neste caso, o número de estados é aproximadamente o número de fonemas das palavras mais dois. Na tabela 7.1 estão mostradas as palavras do vocabulário e o correspondente número de estados utilizado nas simulações em que se usaram números de estados diferentes por palavra.

7.2.1 Treinamento

O treinamento dos modelos foi feito usando-se o algoritmo " Forward-Backward " ou alternadamente o procedimento " K-means " com o algoritmo de Viterbi.

Os modelos representando cada palavra foram treinados separadamente e o treinamento era finalizado quando a diferença entre as probabilidades médias $P(O|\lambda)$ de duas iterações consecutivas fosse menor que 0.1%.

7.2.2 Reconhecimento

No reconhecimento foram utilizados os algoritmos " Forward " ou Viterbi. Escolheu-se como palavra reconhecida, a palavra representada pelo modelo que apre-

Palavra	Número de estados	Palavra	Número de estados
zero	6	nordeste	10
um	4	sul	5
dois	6	sudeste	9
três	6	centro-oeste	13
quatro	8	esportes	10
cinco	7	departamento	14
seis	6	divisão	9
sete	6	seção	7
oito	6	coordenação	13
nove	6	imagem	8
meia	6	voz	5
sim	5	áries	7
não	5	touro	7
terminar	10	câncer	8
repetir	9	leão	6
continuar	11	gêmeos	8
voltar	8	virgem	8
avançar	9	libra	7
certo	9	escorpião	11
errado	8	capricórnio	13
opções	8	sagitário	11
dólar	7	aquário	9
real	6	peixes	8
tempo	7	horóscopo	11
norte	7	ajuda	7

Tabela 7.1: Número de estados usado para representar cada palavra nas simulaçẽos em que usaram-se números de estados diferentes por palavra.

sentava a maior probabilidade $P(O|\lambda)$.

Para avaliar os reconhecedores, foram utilizadas palavras faladas por 34 locutores (21 homens e 13 mulheres) diferentes dos locutores usados no treinamento. Utilizou-se apenas uma elocução de cada palavra por locutor.

7.3 HMM Discreto

Foram implementados reconhecedores com e sem detecção de início/fim, para avaliar a influência dos erros do detector de início/fim. O algoritmo utilizado para a detecção de início/fim é a combinação dos algoritmos " endpoints " e " bottom-up " com pré-ênfase descritos no capítulo 6. Para a avaliação dos reconhecedores foi utilizado o vocabulário com 50 palavras

Para a criação dos " codebooks " usou-se o algoritmo LBG, com distância Euclidiana. A criação do " codebook " era finalizada quando a diferença entre as distorções de duas iterações consecutivas fosse menor que 1%. O tempo necessário

para a criação do " codebook " não está incluído nos tempos de treinamento que serão apresentados no decorrer do trabalho.

7.3.1 Tipos de Algoritmos

Inicialmente comparou-se os diferentes algoritmos empregados para treinamento e reconhecimento. Para isso foram implementados reconhecedores com as seguintes características:

- 10 estados
- " Codebook " com 128 vetores código
- Coeficientes Mel-SME (16 coef. por vetor)
- Seqüência de treinamento 1

Na tabela 7.2 são apresentados os resultados obtidos sem a detecção de início/fim e a tabela 7.3 mostra os resultados considerando o detector de início/fim.

Estado Final	Algoritmo Trein.	Algoritmo Rec.	% de Acerto	Tempo médio de trein. (1 modelo)(s)	Tempo médio de rec. (1 palavra)(s)
qualquer estado	FB	F	95.35	20	0.44
	FB	VIT	95.41	20	0.78
	VIT	F	94.24	3	0.44
	VIT	VIT	94.06	3	0.78
último estado	FB	F	94.88	19	0.45
	FB	VIT	94.94	19	0.76
	VIT	F	94.35	3	0.45
	VIT	VIT	94.18	3	0.76

Tabela 7.2: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto, sem detecção de início/fim, considerando diferentes algoritmos para treinamento e reconhecimento.

Estado Final	Algoritmo Trein.	Algoritmo Rec.	% de Acerto	Tempo médio de trein. (1 modelo)(s)	Tempo médio de rec. (1 palavra)(s)
qualquer estado	FB	F	94.12	11	0.27
	FB	VIT	93.82	11	0.49
	VIT	F	93.76	2.2	0.27
	VIT	VIT	93.41	2.2	0.49
último estado	FB	F	94.18	11.2	0.27
	FB	VIT	94.00	11.2	0.49
	VIT	F	93.76	2.4	0.27
	VIT	VIT	93.41	2.4	0.49

Tabela 7.3: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto, com detecção de início/fim, considerando diferentes algoritmos para treinamento e reconhecimento.

Analisando-se os resultados observa-se que o algoritmo " Forward-Backward " requer um esforço computacional muito maior que o algoritmo de Viterbi para o treinamento, enquanto no reconhecimento o algoritmo de Viterbi gasta quase o dobro do tempo do algoritmo " Forward ". Os desempenhos dos reconhecedores são bastante próximos, com uma leve superioridade da combinação " Forward-Backward " para treinamento e " Forward " para reconhecimento. Deve-se ressaltar também, que com a utilização do detector de início/fim diminui-se consideravelmente o tempo de processamento. Considerando a detecção de início/fim, o uso do último estado como estado final apresentou uma ligeira melhora sobre o uso de qualquer estado como estado final.

7.3.2 Tamanho da Seqüência de Treinamento

Para essa avaliação foram consideradas as três diferentes seqüências de treinamento e usados reconhecedores com as seguintes características:

- Coeficientes Mel-SME (16 coef. por vetor)
- Sem detecção de início/fim
- Treinamento: algoritmo " Forward-Backward "
- Reconhecimento: algoritmo " Forward "
- Estado final: qualquer estado

Na tabela 7.4 são mostrados os desempenhos de reconhecedores com diferentes números de estados e utilizando " codebook " com 128 vetores código. A tabela 7.5 apresenta os resultados de reconhecedores com 10 estados e tamanhos variados de " codebooks ".

A partir dos resultados observa-se que aumentando o tamanho da seqüência de treinamento, melhora-se o desempenho dos reconhecedores. Isso ocorre devido a uma melhor estimativa dos parâmentros dos modelos. Também nota-se que à medida que se aumenta o número de parâmetros a serem estimados necessita-se de uma seqüência de treinamento maior para melhorar o desempenho do reconhecedor.

7.3.3 Quantidade de Parâmetros

Para verificar a influência da quantidade de parâmetros no desempenho dos reconhecedores, foram implementados reconhecedores variando-se o número de estados e o tamanho do " codebook ", com e sem o uso de detectores de início/fim e utilizando diferentes seqüências de treinamento. Em todos os reconhecedores implementados o conjunto de possíveis estados finais era formado por todos os estados,

Criação do " codebook "	Treinamento dos HMMs	Número de estados	% de Acerto
seqüência de treinamento 1	seqüência de treinamento 1	5	93.94
		10	95.35
		15	95.71
seqüência de treinamento 1	seqüência de treinamento 2	5	94.65
		10	96.06
		15	96.59
seqüência de treinamento 1	seqüência de treinamento 3	5	94.76
		10	96.47
		15	97.06
seqüência de treinamento 2	seqüência de treinamento 2	5	94.65
		10	95.94
		15	96.47
seqüência de treinamento 3	seqüência de treinamento 3	5	95.24
		10	96.41
		15	96.47

Tabela 7.4: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto com diferentes números de estados e considerando seqüências de treinamento de vários tamanhos. Utilizou-se " codebook " com 128 vetores código. Não foi utilizado detector de início/fim.

Criação do " codebook "	Treinamento dos HMMs	Tamanho do " codebook "	% de Acerto
seqüência de treinamento 1	seqüência de treinamento 1	64	95.53
		128	95.35
		256	94.41
seqüência de treinamento 1	seqüência de treinamento 2	64	94.88
		128	96.06
		256	96.29
seqüência de treinamento 1	seqüência de treinamento 3	64	96.00
		128	96.47
		256	96.59
seqüência de treinamento 2	seqüência de treinamento 2	64	95.82
		128	95.94
		256	95.76
seqüência de treinamento 3	seqüência de treinamento 3	64	96.59
		128	96.41
		256	97.18

Tabela 7.5: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto com " codebooks " de vários tamanhos e considerando diferentes seqüências de treinamento. Os reconhecedores apresentavam 10 estados. Não foi utilizado detector de início/fim.

permitindo que qualquer estado fosse o estado final. No treinamento e reconhecimento foram usados os seguintes algoritmos:

- Treinamento: algoritmo " Forward-Backward "
- Reconhecimento: algoritmo " Forward "

Na tabela 7.6 são apresentados os desempenhos de reconhecedores usando coeficientes Mel e Mel-SME e vários tamanhos de " codebook ". Não foi usado o detector de início/fim e para o treinamento foi empregada a seqüência 1.

Número de estados	Tipo de Coeficiente	Tamanho do " codebook "	% de Acerto
10	Mel	64	93.24
		128	92.82
		256	92.53
10	Mel-SME	64	95.53
		128	95.35
		256	94.41

Tabela 7.6: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto com " codebooks " de vários tamanhos.

Analisando-se os resultados da tabela 7.6 observa-se que o desempenho do reconhecedor piora com o aumento do tamanho do " codebook ", independentemente do tipo de coeficiente utilizado. Isso ocorre devido ao aumento do número de parâmetros a estimar. Para melhorar o desempenho seria necessária uma seqüência de treinamento maior.

A tabela 7.7 mostra os desempenhos de reconhecedores usando coeficientes Mel-SME, diferentes números de estados e vários tamanhos de " codebook ". Também não foi usado o detector de início/fim e para o treinamento foi empregada a seqüência 1.

Número de estados	Tamanho do " codebook "	% de Acerto	Tempo médio de reconhecimento (1 palavra) (s)
5	64	93.71	0.14
	128	93.94	0.15
	256	93.65	0.15
10	64	95.53	0.43
	128	95.35	0.44
	256	94.41	0.44
15	64	95.18	0.89
	128	95.71	0.91
	256	95.00	0.91

Tabela 7.7: Desempenho de reconhecedores de palavras isoladas independente do locutor usando usando HMM discreto, coeficientes Mel-SME com " codebooks " de vários tamanhos e diferentes números de estados.

Considerando-se os desempenhos dos reconhecedores da tabela 7.7 nota-se que o aumento do número de estados melhora a taxa de acerto do reconhecedor para

todos os " codebooks " implementados. Isso mostra a importância do número de estados em HMM. Além disso, nota-se também que com o aumento do número de parâmetros, o tempo de processamento aumenta consideravelmente.

Para avaliar melhor a influência do número de estados no desempenho do reconhecedor, foram implementados vários reconhecedores com diferentes números de estados, utilizando coeficientes Mel-SME. A tabela 7.8 mostra os resultados obtidos sem o uso de detecção de início/fim, enquanto a tabela 7.9 apresenta o desempenho de reconhecedores empregando um detector de início/fim e utilizando a seqüência de treinamento 1.

Seq. de treinamento	Tamanho do " codebook "	Número de estados	% de Acerto	Tempo médio de trein. (1 modelo)(s)	Tempo médio de rec. (1 pal.)(s)
1	128	5	93.94	6	0.15
		10	95.35	20	0.44
		15	95.71	45	0.91
		20	95.59	1:19	1.52
		30	96.00	2:54	3.21
		40	96.24	5:13	5.67
		50	96.18	8:28	8.80
		60	96.47	12:36	12.96
		70	95.41	20:05	18.78
		80	94.18	27:29	22.69
		90	93.53	35:17	29.87
		100	91.53	44:03	44.28
3	256	5	95.18	26	0.15
		10	97.18	1:33	0.44
		15	96.94	3:06	0.91
		20	97.12	5:04	1.52
		30	97.06	11:12	3.23
		40	97.47	18:39	5.82
		50	98.12	27:01	8.80
		60	98.35	49:54	12.96
		70	98.18	1:02:28	18.36
		80	97.76	1:21:15	22.67
		90	98.00	1:46:56	29.11
		100	97.76	2:24:34	44.70

Tabela 7.8: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto, coeficientes Mel-SME e diferentes números de estado. Não foi usado detector de início/fim.

Comparando-se os resultados das tabelas observa-se que existe um limite para o aumento do número de estados e conseqüente aumento do número de parâmetros do HMM e que esse limite é dependente do tamanho da seqüência de treinamento. Também fica claro que ocorre uma diminuição do tempo de processamento com o uso do detector de início/fim, devido à uma diminuição do tamanho dos arquivos

Tamanho do " codebook "	Número de estados	% de Acerto	Tempo médio de trein. (1 modelo)(s)	Tempo médio de rec. (1 palavra) (s)
128	5	92.59	3	0.09
	10	94.12	11	0.27
	15	94.82	27	0.56
	20	94.88	51	0.95
	30	96.06	2:02	1.99
	40	94.94	3:55	3.55

Tabela 7.9: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto, coeficientes Mel-SME e diferentes números de estado. Foi usado detector de início/fim.

da seqüência de treinamento. Outro ponto a ser ressaltado é que aumentando-se o tamanho da seqüência de treinamento, pode-se conseguir melhores resultados aumentando o tamanho do " codebook ". Esse aumento do tamanho do " codebook " eleva consideravelmente o tempo de treinamento mas não causa impacto no tempo de reconhecimento.

7.3.4 Incorporação da Duração de Estado

De modo a avaliar o desempenho dos reconhecedores incorporando a duração de estado, esta foi modelada usando as seguinte funções de probabilidade: não paramétrica, Gamma, Rayleigh, Gaussiana e Laplace. A duração de estado foi incorporada ao HMM através do uso de um pós-processador. Os reconhecedores implementados tinham as seguintes características:

- " Codebook " com 128 vetores código
- Coeficientes Mel-SME (16 coef. por vetor)
- Treinamento: " Forward-Backward " e algoritmo de Viterbi para segmentação
- Reconhecimento: algoritmo de Viterbi (será indicado quando for usado o " Forward ")
- Estado final: qualquer estado (será indicado quando for usado apenas o último estado)
- Seqüência de treinamento 1
- Uso de detector de início/fim

A tabela 7.10 apresenta o desempenho de reconhecedores com 10 estados considerando as diferentes funções de probabilidade para modelar a duração de estado e vários valores de coeficientes de ponderação, os quais são empíricos.

A partir dos resultados da tabela 7.10 percebe-se que para cada tipo de função existe um determinado coeficiente de ponderação ótimo. Entre todas as funções o

Prob. de dur. de estado	Coeficiente de dur. de estado	% de Acerto
Não-Paramétrica	0	93.82
	0.5	94.71
	1.0	94.82
	1.5	94.65
	2.0	93.88
	3.0	92.12
Gamma	0	93.82
	0.001	94.00
	0.005	94.06
Rayleigh	0	93.82
	0.001	94.00
	0.005	94.00
Gaussiana	0	93.82
	1.0	95.18
	1.5	95.53
	2.0	95.71
	2.5	95.65
	3.0	95.59
Laplace	0	93.82
	1.0	95.24
	2.0	95.76
	3.0	95.76
	4.0	95.82
	5.0	95.76
	6.0	95.82
	6.5	95.94
	7.0	95.88
	8.0	95.82
	9.0	95.82
	12.0	95.47

Tabela 7.10: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto com 10 estados e considerando as diferentes funções de probabilidade para modelar a duração de estado e vários valores de coeficientes de ponderação.

melhor desempenho foi conseguido utilizando-se a função densidade de probabilidade de Laplace para modelar a duração de estado. Em todos os casos ocorre um aumento da taxa de acerto ao incorporar-se a duração de estado.

Na tabela 7.11 são apresentados resultados de reconhecedores com 10 e 15 estados considerando-se as diferentes funções densidade de probabilidade para modelar a duração de estado. Os coeficientes de ponderação são empíricos.

Analisando os resultados da 7.11 observa-se que o custo computacional para a incorporação da duração de estado é desprezível e que a função densidade de probabilidade de Laplace apresenta o melhor resultado independentemente do número

Núm. de estados	Prob. de dur. de estado	Coef. de dur. de estado	% de Acerto	Tempo médio de trein. (1 mod.)(s)	Tempo médio de rec. (1 pal.)(s)
10	-	-	93.82	11.0	0.49
	Não Paramétrica	1.0	94.82	12.3	0.51
	Gaussiana	2.0	95.71	11.5	0.50
	Gamma	0.005	94.06	11.5	0.50
	Rayleigh	0.005	94.00	11.5	0.50
	Laplace	6.5	95.94	11.5	0.50
15	-	-	94.29	27.0	0.93
	Não Paramétrica	0.3	95.06	27.0	0.93
	Gaussiana	3.0	95.65	26.9	0.97
	Gamma	0.001	94.53	26.9	0.93
	Rayleigh	0.001	94.53	26.9	0.92
	Laplace	7.0	96.06	26.9	0.93

Tabela 7.11: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto e considerando as diferentes funções de probabilidade para modelar a duração de estado.

de estados.

Nos resultados apresentados até aqui, os coeficientes de ponderação foram obtidos por tentativa e erro. No capítulo 3 foi proposta uma fórmula para calcular esses coeficientes de ponderação, sendo que cada modelo teria um coeficiente diferente. A tabela 7.12 mostra os resultados dos reconhecedores implementados usando 10 estados e fator de ponderação dado pela equação 3.83.

Prob. de duração de estado	% de Acerto	Tempo médio de rec. (1 palavra) (s)
-	93.82	0.49
Não Paramétrica	80.35	0.49
Gaussiana	95.35	0.43
Gamma	48.35	0.44
Rayleigh	48.82	0.44
Laplace	95.88	0.43

Tabela 7.12: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto e considerando as diferentes funções de densidade de probabilidade para modelar a duração de estado. Foram empregados coeficientes de ponderação calculados pela equação 3.83.

Comparando-se as tabelas 7.11 e 7.12 nota-se que para as funções densidade de probabilidade Gaussiana e de Laplace os resultados foram praticamente iguais, considerando a taxa de acerto e o tempo de processamento. Portanto, pode-se considerar bom o método proposto para cálculo dos coeficientes de ponderação. Já para as demais densidades o uso de fatores de ponderação obtidos segundo esse método é extremamente prejudicial ao desempenho dos reconhecedores.

Na tabela 7.13 são apresentados resultados de reconhecedores com diferentes

números de estados, modelando-se a duração de estado com a função densidade de probabilidade de Laplace e usando-se coeficientes de ponderação empíricos. Para o reconhecimento foi utilizado o algoritmo de Viterbi e o algoritmo " Forward ". Neste último caso, utilizou-se também o algoritmo de Viterbi para segmentar as palavras e calcular a duração de estado.

Algoritmo reconhecimento	Número de estados	Coef. de dur. de estado	% de Acerto	Tempo médio de rec. (1 palavra) (s)
Viterbi	5	-	92.59	0.18
	10	-	93.82	0.49
	15	-	94.29	0.93
	20	-	94.76	1.47
	30	-	95.53	2.99
	40	-	94.59	5.12
	5	3.5	94.29	0.18
	10	6.5	95.94	0.50
	15	7.0	96.06	0.93
	20	3.5	96.41	1.52
	30	2.0	96.29	2.99
	40	1.5	95.53	5.03
Forward	5	-	92.59	0.09
	10	-	94.12	0.27
	15	-	94.82	0.56
	20	-	94.88	0.95
	30	-	96.06	1.99
	40	-	94.94	3.55
	5	3.5	94.18	0.28
	10	6.5	96.06	0.78
	15	7.0	96.12	1.56
	20	3.5	96.41	2.59
	30	2.0	96.41	5.03
	40	1.5	95.71	8.65

Tabela 7.13: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto e considerando diferentes números de estados. A probabilidade de duração de estado foi modelada usando fdp de Laplace.

A partir dos resultados, verifica-se que em todos os casos o desempenho do reconhecedor melhorou ao incorporar a duração de estado. As taxas de acerto obtidas empregando-se o algoritmo " Forward " são melhores, mas o custo computacional é maior, devido ao fato de ser necessário o uso do algoritmo de Viterbi para a segmentação. Também observa-se que quanto maior a taxa de acerto do reconhecedor sem incorporação da duração de estado, menor é o ganho obtido com a incorporação da mesma, mostrando que existe um limite para a melhora do desempenho do reconhecedor. Outro ponto a ser considerado é que o ganho devido à incorporação da duração de estado também diminui com o aumento do número de estados, pois

quando este aumenta, automaticamente incorpora melhor a duração de estado.

Também avaliou-se a influência da duração de estados em reconhecedores considerando como estado final o último estado. Os desempenhos desses reconhecedores modelando a duração de estado com a função densidade de probabilidade de Laplace e usando coeficientes de ponderação empíricos estão na tabela 7.14.

Número de estados	Coef. de dur. de estado	% de Acerto	Tempo médio de rec. (1 palavra) (s)
10	7.5	96.12	0.50
15	7.0	95.82	0.97
20	5.0	96.41	1.50
30	2.0	96.12	3.02

Tabela 7.14: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto, considerando diferentes números de estados e o último estado com estado final. A duração de estado foi modelada pela função densidade de probabilidade de Laplace e os coeficientes de duração de estado são empíricos.

Comparando-se as tabelas 7.13 e 7.14 observa-se que em alguns casos conseguiu-se melhorar o desempenho do reconhecedor considerando apenas o último estado como estado final. Em outros casos, ocorreu uma diminuição da taxa de acerto.

7.3.5 Combinando Diferentes Tipos de Coeficientes

Em uma tentativa de melhorar o desempenho dos reconhecedores foram testadas combinações de diferentes tipos de coeficientes. Em todos os casos foram utilizados " codebooks " independentes para quantizar cada coeficiente. Os reconhecedores implementados apresentavam as seguintes características:

- Treinamento: algoritmo " Forward-Backward "
- Reconhecimento: algoritmo " Forward "
- Estado final: qualquer estado
- Detecção de início/fim
- Seqüência de treinamento 1

Foram testados os seguintes coeficientes: " Mel Frequency Cepstrum " com subtração da média espectral (Mel-SME), Delta(Mel-SME) (D), Delta-Delta(Mel-SME) (DD), Delta-Delta-Delta(Mel-SME) (DDD), " Bark Frequency Cepstrum " (Bark), PLP parcor, Energia (E), Delta-Energia (DE) e Delta-Delta-Energia (DDE).

A tabela 7.15 apresenta o resultado de combinações para alguns desses coeficientes, usando-se reconhecedores com 10 estados e " codebooks " com 128 vetores código.

Tipo de Coeficiente	% de Acerto
Mel-SME+D	97.12
Mel-SME+D+DD	98.00
Mel-SME+D+BARK	97.06
Mel-SME+D+PLP PARCOR	97.71

Tabela 7.15: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto e combinando diferentes tipos de coeficientes.

Analisando os resultados da tabela 7.15 observa-se que os melhores desempenhos foram obtidos pela combinação de coeficientes Mel-SME e suas derivadas. Assim, decidiu-se estudar melhor essas combinações.

Foram implementados reconhecedores com 10 e 15 estados combinando os coeficientes Mel-SME e suas derivadas de várias formas. Os resultados são mostrados nas tabelas 7.16 e 7.17 respectivamente.

Tamanho do " codebook "	Tipo de Coeficiente	% de Acerto	Tempo médio de trein. (1 mod.) (s)	Tempo médio de rec. (1 pal.) (s)
64	Mel-SME	93.76	9	0.27
	D(Mel-SME)	94.29	9	0.27
	Mel-SME+D	96.94	13.6	0.31
	Mel-SME+D+DD	97.29	16.8	0.32
	Mel-SME+D+DD+DDD	97.76	21.5	0.34
128	Mel-SME	94.12	11	0.27
	D(Mel-SME)	94.47	12	0.27
	Mel-SME+D	97.12	18.4	0.32
	Mel-SME+D+DD	98.00	27.3	0.33
	Mel-SME+D+DD+DDD	97.82	32.3	0.35

Tabela 7.16: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto com 10 estados, combinando coeficientes Mel-SME e suas derivadas.

Analisando-se os resultados das tabelas 7.16 e 7.17 verifica-se que em todos os casos, o desempenho do reconhecedor foi aumentando à medida que acrescentava-se mais um coeficiente Delta. Ao acrescentar-se cada coeficiente Delta, fornece-se ao reconhecedor mais informações sobre a palavra a ser reconhecida e assim consegue-se melhorar o desempenho dos reconhecedores. Entretando, ocorre um aumento de processamento, elevando dessa forma o tempo de treinamento e o tempo necessário para o reconhecimento de uma palavra.

Um outro parâmetro que também pode ser combinado com os coeficientes Mel-SME é a energia. A tabela 7.18 apresenta os desempenhos de reconhecedores usando 10 estados, coeficientes Mel-SME, Delta(Mel-SME), Delta-Delta(Mel-SME), quantizado com " codebooks " independentes com 128 vetores código e energia. Os

Tamanho do " codebook "	Tipo de Coeficiente	% de Acerto	Tempo médio de trein. (1 mod.) (s)	Tempo médio de rec. (1 pal.) (s)
64	Mel-SME	94.24	21.8	0.55
	D(Mel-SME)	94.35	19.6	0.55
	Mel-SME+D	97.29	26.8	0.60
	Mel-SME+D+DD	97.65	36.0	0.63
	Mel-SME+D+DD+DDD	97.82	43.6	0.67
128	Mel-SME	94.82	27.2	0.56
	D(Mel-SME)	94.76	25.1	0.55
	Mel-SME+D	97.35	38.3	0.59
	Mel-SME+D+DD	98.00	59.3	0.63
	Mel-SME+D+DD+DDD	97.88	74.2	0.73

Tabela 7.17: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto com 15 estados, combinando coeficientes Mel-SME e suas derivadas.

diferentes " codebooks " apresentados nesta tabela foram usados para quantizar a energia.

Tamanho do " codebook "	% de Acerto	Tempo médio de rec. (1 palavra) (s)
16	97.41	0.32
32	97.47	0.33
64	97.41	0.32
128	97.00	0.33

Tabela 7.18: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto com 10 estados, combinando coeficientes Mel-SME, suas derivadas e energia. Foram usados " codebooks " com vários tamanhos para quantizar a energia.

Os resultados da tabela 7.18 indicam que o melhor " codebook " para a quantização da energia apresentou 32 vetores código. Assim, este " codebook " foi utilizado para quantizar a energia em todos os outros reconhecedores implementados.

Na tabela 7.19 são mostrados os desempenhos de reconhecedores combinando os coeficientes Mel-SME e suas derivadas (quantizados com " codebooks " independentes com 128 vetores código) com energia e derivadas (quantizados com " codebooks " independentes com 32 vetores código). Os reconhecedores foram implementados com 10 estados e foram usadas as seqüências 1 e 3 para treinamento dos mesmos.

A partir dos resultados da tabela 7.19 observa-se que o uso da energia e suas derivadas aumenta a taxa de acerto dos reconhecedores e pode-se concluir que a energia e derivadas contém informações que não estão presentes nos coeficientes Mel-SME e derivadas. Além disso, o custo computacional devido ao acréscimo da

Seqüência de treinamento	Tipo de Coeficiente	% de Acerto	Tempo médio de rec. (1 palavra) (s)
1	Mel-SME+D +E	97.47	0.33
	Mel-SME+D+E+DE	97.47	0.35
	Mel-SME+D+E+DE+DDE	97.76	0.38
	Mel-SME+D+DD+E	97.94	0.35
	Mel-SME+D+DD+E+DE	97.94	0.38
	Mel-SME+D+DD+E+DE+DDE	98.65	0.38
3	Mel-SME	96.47	0.26
	D(Mel-SME)	97.06	0.26
	Mel-SME+D	98.18	0.28
	Mel-SME+D+DD	98.71	0.30
	Mel-SME+D+DD+E	99.00	0.34
	Mel-SME+D+DD+E+DE	99.00	0.35
	Mel-SME+D+DD+E+DE+DDE	99.06	0.37

Tabela 7.19: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto com 10 estados, combinando coeficientes Mel-SME, suas derivadas e energia.

energia e derivadas é mínimo. Novamente, aumentando-se o tamanho da seqüência de treinamento, melhores resultados foram conseguidos.

A combinação de coeficientes constitui-se numa forma simples e eficiente de aumentar a taxa de acerto dos reconhecedores, sem a necessidade de aumentar a seqüência de treinamento. O aumento da seqüência de treinamento não é uma tarefa fácil e está associada a altos custos computacionais.

7.3.6 Combinando Diferentes Modelos de Palavra

Para melhorar os resultados foi testada a combinação de dois modelos. Isso foi implementado da seguinte forma: para cada palavra no vocabulário foram criados dois modelos usando tipos de coeficientes diferentes. Na fase de reconhecimento, as probabilidades de cada modelo foram combinadas da seguinte forma:

$$\log P(O|\lambda) = w_1 \log P(O|\lambda_1) + w_2 \log P(O|\lambda_2) \tag{7.1}$$

onde w_i são coeficientes de ponderação.

Neste trabalho, foram usados $w_1 = w_2 = 0.5$. Todos os HMMs implementados apresentavam as seguintes características:

- 10 estados
- " Codebooks " com 128 vetores código
- Treinamento: " Forward-Backward "
- Reconhecimento: " Forward "
- Estado final: qualquer estado
- Seqüência de treinamento 1

A tabela 7.20 mostra os resuldados obtidos.

Tipo de modelos	Coef. dos. modelos	% de Acerto	Tempo médio de rec. (1 palavra) (s)
BARK e PLP	0.5	94.06	0.46
BARK e PLP parcor	0.5	94.76	0.46
BARK e Mel norm.	0.5	95.12	0.46
Mel norm. e PLP parcor	0.5	95.12	0.46
BARK e Mel-SME	0.5	96.29	0.46
Mel-SME e PLP parcor	0.5	96.59	0.46
Mel-SME e Mel norm.	0.5	95.94	0.46
BARK e D(Mel-SME)	0.5	96.94	0.46
PLP parcor e D(Mel-SME)	0.5	97.18	0.46
Mel norm. e D(Mel-SME)	0.5	96.76	0.46
Mel-SME e D(Mel-SME)	0.5	97.35	0.46

Tabela 7.20: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto com 10 estados, combinando modelos com coeficientes diferentes.

Os resultados da tabela 7.20 mostram que a combinação de modelos resulta em um aumento na taxa de acerto dos reconhecedores. Entretando o desempenho é inferior ao obtido quando combinam-se diferentes coeficientes em um mesmo modelo (tabela 7.19). Além disso, o tempo de processamento deste último também é menor. De qualquer forma, esses resultados são uma indicação de que reconhecedores com melhores desempenhos podem ser obtidos com o uso de mais de um tipo de coeficiente para representar o sinal de fala.

7.3.7 Combinando Várias Características

Até o momento, conseguiu-se aumentar a taxa de acerto dos reconhecedores incorporando a duração de estado como pós-processador e combinando diferentes tipos de coeficientes separadamente. Assim, resolveu-se implementar reconhecedores unindo essas duas formas de aumento da taxa de acerto. Os reconhecedores usaram coeficientes Mel-SME, Delta(Mel-SME), Delta-Delta(Mel-SME), quantizados com " codebooks " independentes com 128 vetores código e Energia, Delta-Energia e Delta-Delta-Energia, quantizados com " codebooks " independentes com 32 vetores código. Para modelar a duração de estado foi usada a função densidade de probabilidade de Laplace e os coeficientes de duração de estado são empíricos. Foi empregada a detecção de início/fim e foram utilizados os seguintes algoritmos:

- treinamento: " Forward-Backward "
- reconhecimento: " Viterbi "

Os resultados estão apresentados na tabela 7.21.

Treinamento	Estado final	Número de estados	Coef. de dur. de estado	% de Acerto	Tempo médio de rec. (1 palavra) (s)
seq. de treinamento 1	qualquer estado	10	-	98.59	0.90
			6.5	98.94	0.90
		15	-	98.47	1.54
			7.0	99.06	1.54
		diferente para cada palavra	-	98.47	0.71
			7.0	98.94	0.71
	último estado	10	-	98.88	0.90
			7.0	98.94	0.90
		15	-	99.12	1.52
			8.0	99.18	1.54
		diferente para cada palavra	-	98.47	0.67
			8.0	98.94	0.68
seq. de treinamento 3	qualquer estado	10	-	99.06	0.89
			7.0	99.29	0.89
		15	-	99.00	1.56
			8.0	99.41	1.56
		diferente para cada palavra	-	99.06	0.68
			8.0	99.29	0.73
	último estado	10	-	99.06	0.88
			7.0	99.29	0.88
		15	-	98.94	1.53
			8.0	99.41	1.54
		diferente para cada palavra	-	98.88	0.68
			8.0	99.35	0.68

Tabela 7.21: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto, combinando vários tipos de coeficientes e incorporando duração de estado.

Analisando-se os resultados obtidos, verifica-se que foram conseguidas altas taxas de acerto, todas por volta de 99%. Os erros ocorreram para as palavras cinco (confundida com sim), nove (confundida com norte) e certo (confundida com sete). Nota-se que os erros aconteceram para palavras muito similares, mostrando que com a união do modelamento da duração de estado e combinação de parâmetros pode-se conseguir reconhecedores bastante robustos. Observa-se também que o fato de se considerar como estado final apenas o último estado ou qualquer estado não influenciou no desempenho do reconhecedor (diferenças entre resultados é muito pequena). Novamente, conseguiu-se melhores desempenhos ao aumentar-se o tamanho da seqüência de treinamento.

7.3.8 Diferentes Vocabulários

Para avaliar a dependência dos reconhecedores em relação ao vocabulário, foram avaliados reconhecedores usando diferentes vocabulários. Os reconhecedores

usaram 10 estados, coeficientes Mel-SME, Delta(Mel-SME), Delta-Delta(Mel-SME), quantizados com " codebooks " independentes com 128 vetores código e Energia, Delta-Energia e Delta-Delta-Energia, quantizados com " codebooks " independentes com 32 vetores códgio. Para modelar a duração de estado foi usada a função densidade de probabilidade de Laplace e coeficientes de duração de estado empíricos. O conjunto de possíveis estados finais era formado por todos os estados. Além disso, empregou-se detecção de início/fim e usou-se a seqüência de treinamento 1. Foram utilizados os seguintes algoritmos:

- Treinamento: " Forward-Backward "
- Reconhecimento: " Viterbi "

Os resultados são mostrados na tabela 7.22.

Vocabulário	Número de palavras	coef. de dur. de estado	% de Acerto	Tempo médio de rec. (1 palavra) (s)
todas as pal.	50	6.5	98.94	0.90
dígitos	11	6.5	100.0	0.23
comandos	10	6.5	100.0	0.26
opções	5	10	100.0	0.13
regiões	5	6.5	100.0	0.15
signos	12	6.5	100.0	0.32

Tabela 7.22: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto, combinando vários tipos de coeficientes e incorporando duração de estado (fdp Laplace) como pós-processador para diferentes vocabulários.

Os resultados obtidos mostram que o desempenho dos reconhecedores depende do tipo e tamanho do vocabulário. Para vocabulários pequenos e sem palavras similares pode-se conseguir altas taxas de acerto sem a necessidade de sofisticação dos reconhecedores. Ressalta-se também que à medida que se aumenta o tamanho do vocabulário aumenta-se o tempo de processamento do reconhecedor.

7.3.9 Considerando Unidades Menores que Palavras

Para comparar as formas de representação de palavras por HMMs, foram implementados os seguintes reconhecedores:

- reconhecedores usando um modelo para cada palavra, obtido considerando a palavra como a unidade a ser representada
- reconhecedores usando um modelo para cada palavra, obtido a partir da concatenação de modelos de unidades menores

Neste trabalho, usou-se o fone como unidade menor. Para representar cada fone usou-se o modelo " left-right " com $\Delta = 1$. Para a formação do modelo de

cada palavra, os modelos dos fones que formam a palavra foram concatenados e acrescentaram-se um estado inicial e um estado final para representar o silêncio. Para o treinamento dos modelos dos fones, foi considerada uma seqüência de treinamento formada por exemplos de todas as palavras e para representar cada palavra utilizaram-se os modelos obtidos pela concatenação dos fones.

Todos os reconhecedores implementados apresentaram as seguintes características:

- Treinamento: " Forward-Backward "
- Reconhecimento: " Forward "
- " Codebook " com 128 vetores código
- Coeficientes Mel-SME
- Seqüência de treinamento 1

Os reconhecedores implementados considerando a palavra como unidade a ser representada apresentavam diferentes números de estados por palavra, os quais eram aproximadamente o número de fonemas das palavras mais dois. A tabela 7.23 apresenta o desempenho dos reconhecedores.

Unidade	Número de estados	% de Acerto
palavra	diferente por palavra	93.76
fone	3 por fone	85.00
	4 por fone	84.76

Tabela 7.23: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto e considerando unidades menores que palavras (fones).

Como pode ser observado na tabela 7.23, os melhores resultados são conseguidos considerando-se a palavra como a unidade a ser representada pelos HMMs. Deve-se ressaltar que o fone não é a melhor unidade para ser usada em reconhecimento, pois não leva em conta os efeitos da coarticulação. Outras unidades como trifones e trifones generalizados têm sido empregadas com melhores resultados.

7.4 HMM Semicontínuo

Todos os reconhecedores implementados utilizaram detecção de início/fim empregando a combinação dos algoritmos " endpoints " e " bottom-up " com pré-ênfase descritos no capítulo 6. Para a avaliação dos reconhecedores foi utilizado o vocabulário com 50 palavras.

Os tempos correspondentes ao treinamento dos modelos, que serão apresentados a seguir, incluem apenas o tempo para calcular as probabilidades de transição e os coeficientes das misturas, não considerando o tempo utilizado para calcular o vetor média e a matriz covariância de cada mistura.

7.4.1 Tipos de Algoritmos

Os diferentes algoritmos empregados para treinamento e reconhecimento foram comparados usando-se reconhecedores com as seguintes características:

- 10 estados
- " Codebook " com 128 misturas
- Coeficientes Mel-SME (16 coef. por vetor)
- Estado final: qualquer estado
- Seqüência de treinamento 1

Na tabela 7.24 são apresentados os resultados obtidos considerando matriz covariância diagonal e matriz covariância cheia.

Matriz Covariância	Algoritmo Trein.	Algoritmo Rec.	% de Acerto	Tempo médio de trein. (1 modelo)(s)	Tempo médio de rec. (1 palavra) (s)
diagonal	FB	F	95.41	1:52	3.46
	FB	VIT	95.06	1:52	3.40
	VIT	F	95.65	1:14	3.41
	VIT	VIT	95.18	1:14	3.40
cheia	FB	F	96.59	4:50	4.32
	FB	VIT	96.35	4:50	4.21
	VIT	F	96.24	4:28	4.34
	VIT	VIT	95.88	4:28	4.17

Tabela 7.24: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM semicontínuo e considerando diferentes algoritmos para treinamento e reconhecimento.

Analisando-se os resultados nota-se que o algoritmo " Forward-Backward " exige um maior esforço computacional para o treinamento, enquanto no reconhecimento o tempo de processamento é praticamente o mesmo para os dois algoritmos. Os desempenhos dos reconhecedores são bastante próximos, sendo que o uso do algoritmo de Viterbi na fase de reconhecimento resultou nas menores taxas de acerto.

Comparando-se o desempenho dos reconhecedores quanto ao tipo de matriz covariância utilizado, observa-se que os reconhecedores que empregaram a matriz cheia obtiveram maiores taxas de acerto, mas necessitaram de maiores esforços computacionais.

7.4.2 Quantidade de Parâmetros

Foram implementados reconhecedores com as seguintes características:

- Treinamento: algoritmo " Forward-Backward "
- Reconhecimento: algoritmo " Forward "
- Coeficientes Mel-SME (16 coef. por vetor)
- Estado final: qualquer estado
- Seqüência de treinamento 1

A tabela 7.25 mostra os desempenhos de reconhecedores para diferentes números de estados e vários tamanhos de " codebook ".

Matriz covariância	Número de estados	Tamanho do " codebook "	% de Acerto	Tempo médio de trein. (1 modelo)(s)	Tempo médio de rec. (1 palavra) (s)
diagonal	5	64	93.12	38	0.91
		128	94.18	1:17	1.73
		256	95.12	2:29	3.49
	10	64	95.12	56	1.82
		128	95.41	1:52	3.46
		256	96.35	3:43	6.55
	15	64	96.00	1:19	2.86
		128	95.53	2:22	5.22
		256	96.59	4:58	9.87
cheia	5	64	94.12	1:58	1.48
		128	94.76	3:43	2.64
		256	95.65	6:54	5.19
	10	64	95.65	2:51	2.46
		128	96.59	4:50	4.32
		256	96.41	9:54	8.38
	15	64	96.06	3:21	3.31
		128	96.82	5:57	6.12
		256	96.71	11:59	12.05

Tabela 7.25: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM semicontínuo e considerando diferentes números de estados e " codebooks " com vários tamanhos.

Considerando-se os desempenhos dos reconhecedores da tabela 7.25 verifica-se que o aumento do número de estados melhora a taxa de acerto do reconhecedor. O mesmo ocorre ao aumentar-se o tamanho dos " codebooks ". Isso é válido para os dois tipos de matriz covariância: diagonal e cheia. Além disso, nota-se também que com o aumento do número de parâmetros, o tempo de processamento aumenta consideravelmente.

A tabela 7.26 mostra os resultados obtidos para reconhecedores com diferentes números de estados e utilizando " codebook " com 128 misturas.

Como já observado na 7.25, aumentando-se o número de estados aumenta-se

Matriz Covariância	Número de estados	% de Acerto	Tempo médio de trein. (1 modelo)(s)	Tempo médio de rec. (1 palavra) (s)
diagonal	5	94.18	1:17	1.73
	10	95.41	1:52	3.46
	15	95.53	2:22	5.22
	20	96.18	3:15	7.03
	30	96.00	4:33	10.9
	40	96.53	7:45	15.16
cheia	5	94.76	3:43	2.64
	10	96.59	4:50	4.32
	15	96.82	5:57	6.12
	20	96.71	7:00	8.10
	30	96.88	8:14	11.79
	40	97.29	10:34	15.79

Tabela 7.26: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM semicontínuo e considerando diferentes números de estados e " codebook " com 128 misturas.

a taxa de acerto do reconhecedor. Novamente o desempenho dos reconhecedores usando matriz covariância cheia é superior ao desempenho dos reconhecedores que utilizam matriz covariância diagonal.

7.4.3 Incorporação da Duração de Estado

Considerando os resultados obtidos para HMMs discretos, o desempenho dos reconhecedores incorporando a duração de estado através do uso de um pós-processador foi avaliado usando as seguintes funções densidade de probabilidade: Gaussiana e Laplace. Considerou-se coeficiente de duração de estado empírico. O reconhecedores implementados tinham as seguintes características:

- 10 estados
- " Codebook " com 128 misturas
- Coeficientes Mel-SME (16 coef. por vetor)
- Treinamento: algoritmo " Forward-Backward "
- Reconhecimento: algoritmo de Viterbi
- Estado final: qualquer estado
- Seqüência de treinamento 1

Na tabela 7.27 são apresentados resultados de reconhecedores considerando-se as diferentes funções de probabilidade para modelar a duração de estado.

Analisando os resultados da tabela 7.27 observa-se que o custo computacional para a incorporação da duração de estado é desprezível e que a função densidade de probabilidade de Laplace apresenta o melhor resultado independentemente do tipo de matriz covariância. A incorporação da duração de estado como um pós-

Matriz Covariância	Prob. de dur. de estado	Coef. de dur. de estado	% de Acerto	Tempo médio de trein. (1 mod.)(s)	Tempo médio de rec. (1 pal.)(s)
diagonal	-	-	95.06	1:52	3:40
	Gaussiana	3.0	96.94	1:45	3.39
	Laplace	3.0	97.59	1:45	3.39
cheia	-	-	96.35	4:50	4.21
	Gaussiana	3.0	97.29	5:38	4.30
	Laplace	3.0	97.82	5:38	4.21

Tabela 7.27: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM semicontínuo e incorporando duração de estado como pós-processador.

processador melhora o desempenho do reconhecedor.

7.4.4 Combinando Diferentes Tipos de Coeficientes

Foram testadas combinações dos seguintes coeficientes: " Mel Frequency Cepstrum " com subtração da média espectral (Mel-SME), Delta(Mel-SME) (D), Delta-Delta(Mel-SME) (DD), Delta-Delta-Delta(Mel-SME) (DDD), Energia (E), Delta-Energia (DE) e Delta-Delta-Energia (DDE).

Em todos os casos foram utilizados " codebooks " independentes para cada tipo de coeficiente. Os reconhecedores implementados apresentavam as seguintes características:

- Treinamento: algoritmo " Forward-Backward "
- Reconhecimento: algoritmo " Forward "
- Weqüência de treinamento 1
- 10 estados
- " Codebook " com 128 misturas para os coeficientes Mel-SME, Delta(Mel-SME) e Delta-Delta(Mel-SME)
- " Codebook " com 32 misturas para os coeficientes Energia, Delta-Energia e Delta-Delta-Energia.
- Estado final: qualquer estado

Na tabela 7.28 são mostrados os desempenhos dos reconhecedores para matriz covariância cheia e matriz covariância diagonal.

A partir dos resultados da tabela 7.28 observa-se que o uso de mais de um tipo de parâmetro aumenta a taxa de acerto do reconhecedor. Entretanto, também aumenta-se o custo computacional. Essas conclusões são válidas para os dois tipos de matriz covariância empregados.

Matriz variância	Tipo de Coeficiente	% de Acerto	Tempo médio de trein. (1 mod.)(s)	Tempo médio de rec. (1 pal.)(s)
diagonal	Mel-SME	95.41	1:52	3.46
	D(Mel-SME)	96.94	1:14	3.39
	Mel-SME+D	98.18	2:41	6.52
	Mel-SME+D+DD	98.53	3:13	9.63
	Mel-SME+D+DD+E	98.76	3:42	11.12
	Mel-SME+D+DD+E+DE	99.06	4:04	12.28
	Mel-SME+D+DD+E+DE+DDE	99.12	4:33	13.45
cheia	Mel-SME	96.59	4:50	4.32
	D(Mel-SME)	97.76	3:43	4.36
	Mel-SME+D	98.35	8:27	8.36
	Mel-SME+D+DD	98.82	9.34	12.56
	Mel-SME+D+DD+E	98.94	10:23	13.38
	Mel-SME+D+DD+E+DE	99.06	9:59	13.95
	Mel-SME+D+DD+E+DE+DDE	99.12	10:26	14.88

Tabela 7.28: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM semicontínuo e combinando vários tipos de coeficientes.

7.4.5 Combinando Várias Características

Os reconhecedores implementados usaram coeficientes Mel-SME, Delta(Mel-SME), Delta-Delta(Mel-SME) (" codebooks " independentes com 128 misturas) e Energia, Delta-Energia e Delta-Delta-Energia (" codebooks " independentes com 32 misturas). Para modelar a duração de estado foi usada a função densidade de probabilidade de Laplace e coeficiente de ponderação empírico. Foram utilizados os seguintes algoritmos:

- treinamento: algoritmo " Forward-Backward "
- reconhecimento: algoritmo de Viterbi

A tabela 7.29 apresenta os resultados obtidos considerando-se matriz covariância cheia e matriz covariância diagonal e seqüência de treinamento 1 enquanto na tabela 7.30 são mostrados os resultados utilizando a seqüência de treinamento 3 e considerando qualquer estado como estado final.

Analisando-se os resultados obtidos, a combinação de diferentes tipos de coeficientes juntamento com a incorporação da duração de estado melhora o desempenho dos reconhecedores. Os reconhecedores que usaram números de estados diferentes para cada palavra, necessitaram de um esforço computacional menor, e a degradação do desempenho foi muito pequena. Quando se usa a seqüência de treinamento 3, o aumento na taxa de acerto devido à incorporação da duração de estados é menor. Outro ponto a ser observado é que foram conseguidos praticamente os mesmos resultados considerando o último estado ou qualquer estado como estado final.

Estado final	Matriz variância	Número estados	Coef. de dur. de estado	% de Acerto	Tempo médio de rec. (1 palavra) (s)
qualquer estado	diagonal	10	-	99.12	11.4
			9.0	99.47	11.4
		15	-	99.18	16.75
			3.0	99.24	16.75
		diferente para cada palavra	-	98.59	9.37
			9.0	99.18	9.38
	cheia	10	-	99.12	14.3
			9.0	99.29	14.8
		15	-	99.12	19.5
			3.0	99.29	19.6
		diferente para cada palavra	-	98.47	12.17
			9.0	99.18	12.19
último estado	diagonal	10	-	99.12	11.4
			9.0	99.41	11.73
		15	-	99.18	16.7
			3.0	99.24	16.76
		diferente para cada palavra	-	98.65	9.40
			9.0	99.18	9.55
	cheia	10	-	99.12	14.7
			9.0	99.29	15.1
		15	-	99.12	19.7
			3.0	99.29	20.3
		diferente para cada palavra	-	98.47	13.6
			9.0	99.18	13.65

Tabela 7.29: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM semicontínuo, com diferentes números de estados, incorporando duração de estado como pós-processador e combinando vários tipos de coeficientes. Utilizou-se a seqüência de treinamento 1.

Matriz Covariância	Número estados	Coef. de dur. de estado	% de Acerto	Tempo médio de rec. (1 palavra) (s)
diagonal	10	-	99.29	11.4
		9.0	99.35	11.4
	15	-	99.24	16.75
		3.0	99.35	16.75
	diferente para cada palavra	-	98.76	9.37
		9.0	99.18	9.39
cheia	10	-	99.24	14.2
		9.0	99.41	14.35
	15	-	99.35	19.7
		3.0	99.35	20.0
	diferente para cada palavra	-	98.82	12.17
		9.0	99.12	12.18

Tabela 7.30: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM semicontínuo, com diferentes números de estados, incorporando duração de estado como pós-processador e combinando vários tipos de coeficientes. Utilizou-se a seqüência de treinamento 3.

7.5 HMM Contínuo

Todos os reconhecedores implementados utilizaram detecção de início/fim empregando a combinação dos algoritmos " endpoints " e " bottom-up " com pré-ênfase descritos no capítulo 6. Além disso, os reconhecedores foram avaliados usando o vocabulário com 50 palavras.

7.5.1 Tipos de Algoritmos

Foram implementados reconhecedores com as seguintes características:

- 10 estados
- 3 misturas com matriz covariância diagonal
- 1 mistura com matriz covariância cheia
- Coeficientes Mel-SME (16 coef. por vetor)
- Estado final: qualquer estado
- Seqüência de treinamento 1 (matriz covariância diagonal)
- Seqüência de treinamento 3 (matriz covariância cheia)

Na tabela 7.31 são apresentados os resultados obtidos considerando matriz covariância diagonal e cheia.

Matriz Covariância	Algoritmo Trein.	Algoritmo Rec.	% de Acerto	Tempo médio de trein. (1 modelo)(s)	Tempo médio de rec. (1 palavra) (s)
diagonal	FB	F	95.71	46.5	4.57
	FB	VIT	95.65	46.5	4.82
	VIT	F	95.35	26.24	4.59
	VIT	VIT	95.41	26.26	4.81
cheia	FB	F	95.53	2:47	5.38
	FB	VIT	95.53	2:47	5.74
	VIT	F	95.29	1:16	5.37
	VIT	VIT	95.24	1:16	5.72

Tabela 7.31: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM contínuo e considerando diferentes algoritmos.

Analisando-se os resultados observa-se mais uma vez que o algoritmo " Forward-Backward " requer um esforço computacional maior para o treinamento, enquanto no reconhecimento as diferenças entre os tempos de processamento são pequenas, com uma leve vantagem para o algoritmo " Forward " . Os desempenhos dos reconhecedores são bastante próximos, com superioridade da combinação " Forward-Backward " para treinamento e " Forward " para reconhecimento. O uso de matriz covariância cheia necessita de maiores esforços computacionais.

7.5.2 Quantidade de Parâmetros

Condiderando a quantidade de parâmetros, foram implementados reconhecedores com as seguintes características:

- Treinamento: algoritmo " Forward-Backward "
- Reconhecimento: algoritmo " Forward "
- Coeficientes Mel-SME
- Estado final: qualquer estado

As tabelas 7.32 e 7.33 mostram os desempenhos de reconhecedores usando diferentes números de estados e misturas e utilizando respectivamente as seqüências de treinamento 1 e 3.

Matriz Covariância	Número de estados	Número de misturas	% de Acerto	Tempo médio de trein. (1 modelo)(s)	Tempo médio de rec. (1 palavra) (s)
diagonal	5	1	91.06	7.12	0.82
		3	94.65	16	2.23
		5	94.71	32.2	3.80
		7	94.24	50.8	5.09
		9	93.35	1:03	6.53
		12	93.47	1:33	8.2
	10	1	95.47	14.5	1.70
		3	95.71	46.5	4.57
		5	94.76	1:18	7.37
		7	93.88	1:57	10.30
		9	93.18	2:36	13.20
		12	91.41	3:26	17.55
	15	1	96.18	26.3	2.70
		3	95.53	1:22	7.0
		5	93.71	2:17	11.44
		7	92.00	3:30	15.91
		9	91.00	4:28	20.27
		12	88.76	6:10	26.57
cheia	5	1	92.94	20.36	2.56
		3	89.59	1:27	7.72
		5	85.71	3:04	12.91
		7	78.86	4:23	17.96
		9	71.94	5:35	23.14
		12	60.76	8:29	30.83
	10	1	93.29	58	5.78
		3	84.00	2:53	15.44
		5	71.29	5:53	25.53
		7	55.29	9:35	36.09
	15	1	92.35	1:16	8.20
		3	76.53	5:53	23.53

Tabela 7.32: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM contínuo e empregando diferentes números de estados e misturas. Foi utilizada a seqüência de treinamento 1.

Matriz Covariância	Número de estados	Número de de misturas	% de Acerto	Tempo médio de trein. (1 modelo)(s)	Tempo médio de rec. (1 palavra) (s)
diagonal	5	3	96.06	55.42	2.37
		5	95.94	1:31	3.73
		7	95.88	2:15	5.20
		9	95.94	3:18	6.40
		12	95.59	4:08	8.70
	10	3	96.76	2:09	4.55
		5	96.59	3:47	7.37
		7	96.88	5:37	10.32
		9	96.18	7:51	13.26
		12	96.12	10:10	17.66
	15	3	97.12	3:51	7.01
		5	97.47	6:32	11.37
		7	96.35	9:24	15.69
		9	96.47	12:13	19.88
		12	95.59	18:40	26.72
cheia	5	1	94.18	1:01	2.54
		3	94.12	4:20	7.75
		5	93.88	7:33	12.69
		7	92.59	12:36	18.08
	10	1	95.53	2:47	5.38
		3	94.35	10:37	15.51
		5	91.65	18:49	25.91
		7	89.59	28:19	30.63
	15	1	95:53	5:18	8.15
		3	93.24	18:13	23.28

Tabela 7.33: Desempenho de reconhecedores de palavras isoladas independente do locutor empregando HMM contínuo e diferentes números de estados e misturas. Foi utilizada a seqüência de treinamento 3.

Comparando-se os resultados das tabelas observa-se que existe um limite para o aumento do número de parâmetros do HMM e que esse é dependente do tamanho da seqüência de treinamento. Como o número de parâmetros de reconhecedores utilizando matriz covariância cheia é muito maior, eles necessitam de uma seqüência de treinamento maior. De qualquer forma, pode-se melhorar a taxa de acerto dos reconhecedores aumentando o número de estados e o número de misturas até que seja atingido o limite da seqüência de treinamento. Outro ponto a ser ressaltado é que o uso de matriz covariância cheia requer esforços computacionais muito maiores.

Na tabela 7.34 são apresentados os resultados obtidos considerando HMM " tied " contínuo com matriz covariância diagonal e cheia. Foi utilizada a seqüência de treinamento 1.

Comparando-se os resultados das tabelas 7.34 e 7.32 verifica-se que o desempenho dos reconhecedores usando HMM " tied " contínuo é inferior ao desempenho dos reconhecedores que utilizam HMM contínuo, embora o HMM " tied " contínuo

Matriz Covariância	Número de estados	Número de de misturas	% de Acerto	Tempo médio de trein. (1 modelo)(s)	Tempo médio de rec. (1 palavra) (s)
diagonal	5	10	93.53	15.3	1.62
		30	94.06	47.8	4.68
		50	93.00	1:29	7.46
	10	10	94.29	23.5	1.92
		30	95.00	1:01	5.10
		50	93.71	1:53	8.22
	15	10	90.71	31	2.25
		30	95.88	1:17	5.61
		50	94.24	2:17	9.12
cheia	5	10	90.06	50.5	5.1
		30	78.35	3:12	15.23
		50	64.82	5:54	25.27
	10	10	90.24	58.53	5.41
		30	80.94	3:21	15.81
		50	66.88	6:10	25.96
	15	10	90.71	1:10	5.73
		30	80.94	3:41	16.49
		50	68.00	6:01	26.69

Tabela 7.34: Desempenho de reconhecedores de palavras isoladas independente do locutor empregando HMM " tied " contínuo com diferentes números de estados e misturas. Foi utilizada a seqüência de treinamento 1.

permita que seja usado um número maior de misturas nos estados.

7.5.3 Incorporação da duração de Estado

O desempenho dos reconhecedores incorporando a duração de estado, foi avaliado usando as funções densidade de probabilidade Gaussiana e de Laplace e coeficientes de ponderação empíricos. Os reconhecedores implementados tinham as seguintes características:

- Coeficientes Mel-SME
- 10 estados
- Estado final: qualquer estado
- Treinamento: algoritmo " Forward-Backward "
- Reconhecimento: algoritmo de Viterbi

Na tabela 7.35 são apresentados resultados de reconhecedores usando 3 misturas com matriz covariância diagonal e seqüência de treinamento 1 e a tabela 7.36 mostra os resultados dos reconhecedores usando 1 mistura com matriz covariância cheia e seqüência de treinamento 3.

Analisando-se os resultados das tabelas 7.35 e 7.36 observa-se que a inclusão da duração de estado como pós-processador aumenta a taxa de acerto para os dois

Prob. de dur. de estado	Coef. de dur. de estado	% de Acerto	Tempo médio de trein. (1 modelo)(s)	Tempo médio de rec. (1 palavra) (s)
-	-	95.65	46.5	4.82
Gaussiana	3.0	96.71	49.1	5.16
Laplace	3.0	96.71	49.1	4.84

Tabela 7.35: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM contínuo e incorporando duração de estado como pós-processador. Foram usadas 3 misturas com matriz covariância diagonal e seqüência de treinamento 1.

Prob. de dur. de estado	Coef. de dur. de estado	% de Acerto	Tempo médio de trein. (1 modelo)(s)	Tempo médio de rec. (1 palavra) (s)
-	-	95.53	2:47	5.55
Gaussiana	3.0	97.12	2:57	5.57
Laplace	3.0	97.12	2:57	5.55

Tabela 7.36: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM contínuo e incorporando duração de estado como pós-processador. Foi usada 1 mistura com matriz covariância cheia e seqüência de treinamento 3.

tipos de matriz convariância utilizados. O custo computacional para a incorporação da duração de estado pode ser considerado desprezível.

7.5.4 Combinando Diferentes Tipos de Coeficientes

Para combinar os diferentes tipos de coeficientes foram utilizados reconhecedores com as seguintes características:

- Treinamento: algoritmo " Forward-Backward "
- Reconhecimento: algoritmo " Forward "
- Coeficientes: " Mel Frequency Cepstrum " com subtração da média espectral (Mel-SME), Delta(Mel-SME) (D), Delta-Delta(Mel-SME) (DD), Delta-Delta-Delta(Mel-SME) (DDD), Energia (E), Delta-Energia (DE) e Delta-Delta-Energia (DDE)
- Estado final: qualquer estado

Na tabela 7.37 são mostrados os resultados de reconhecedores com 10 estados e 3 misturas com matriz covariância diagonal. A tabela 7.38 apresenta os resultados de reconhecedores com diferentes números de estados por palavra e 5 misturas com matriz covariância diagonal. Nos dois casos foi utilizada a seqüência de treinamento 1.

Na tabela 7.39 são mostrados os resultados de reconhecedores com 10 estados. A tabela 7.40 apresenta os resultados de reconhecedores com diferente número de estados por palavras. Nos dois casos foi utilizada 1 mistura com matriz covariância cheia e seqüência de treinamento 3.

Tipo de Coeficiente	% de Acerto	Tempo médio de trein. (1 modelo)(s)	Tempo médio de rec. (1 palavra) (s)
Mel-SME	95.71	46.5	4.57
D(Mel-SME)	94.00	30.3	4.60
Mel-SME+D	96.47	59.4	8.76
Mel-SME+D+DD	96.29	1:17	13.05
Mel-SME+D+DD+E	96.71	1:35	16.96
Mel-SME+D+DD+E+DE	97.47	1:46	19.77
Mel-SME+D+DD+E+DE+DDE	97.82	1:57	23.05

Tabela 7.37: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM contínuo com 10 estados, 3 misturas com matriz covariância diagonal e combinando vários tipos de coeficientes.

Tipo de Coeficiente	% de Acerto	Tempo médio de rec. (1 palavra) (s)
Mel-SME	95.59	5.96
D(Mel-SME)	94.29	5.80
Mel-SME+D	97.24	11.4
Mel-SME+D+DD	96.29	17.28
Mel-SME+D+DD+E	97.47	21.5
Mel-SME+D+DD+E+DE	97.88	26.86
Mel-SME+D+DD+E+DE+DDE	97.82	30.75

Tabela 7.38: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM contínuo com diferentes números de estados por palavra, 5 misturas com matriz covariância diagonal e combinando vários tipos de coeficientes.

Tipo de Coeficiente	% de Acerto	Tempo médio de trein. (1 modelo)(s)	Tempo médio de rec. (1 palavra) (s)
Mel-SME	95.53	2:47	5.38
D(Mel-SME)	93.18	1:57	5.43
Mel-SME+D	96.06	3:90	10.45
Mel-SME+D+DD	95.12	5:33	15.67
Mel-SME+D+DD+E	95.76	5:21	16.64
Mel-SME+D+DD+E+DE	96.26	5:20	18.04
Mel-SME+D+DD+E+DE+DDE	96.76	5:23	19.21

Tabela 7.39: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM contínuo com 10 estados, 1 mistura com matriz covariância cheia e combinando vários tipos de coeficientes.

Considerando os resultados obtidos, observa-se que novamente, o uso de mais de um tipo de parâmetro aumenta a taxa de acerto do reconhecedor e também o custo computacional. O ganho obtido pela combinação de coeficientes foi maior para a matriz covariância diagonal. Isso pode ser justificado pelo fato de ter sido usada apenas uma mistura nos reconhecedores com matriz covariância cheia. Esse número não é suficiente para um bom modelamento dos coeficientes utilizados.

Tipo de Coeficiente	% de Acerto	Tempo médio de trein. (1 modelo)(s)	Tempo médio de rec. (1 palavra) (s)
Mel-SME	96.12	2:08	4.29
D(Mel-SME)	92.53	1:43	5.37
Mel-SME+D	95.59	3:54	8.28
Mel-SME+D+DD	95.00	4:29	12.65
Mel-SME+D+DD+E	95.53	4:47	13.53
Mel-SME+D+DD+E+DE	96.12	4:47	14.55
Mel-SME+D+DD+E+DE+DDE	96.53	4:43	15.07

Tabela 7.40: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM contínuo com diferentes números de estados por palavra, 1 mistura com matriz covariância cheia e combinando vários tipos de coeficientes.

7.5.5 Combinando Várias Características

Os reconhecedores implementados incorporando a duração de estado como pós-processador e combinando diferentes tipos de coeficientes apresentaram as seguintes características:

- Treinamento: algoritmo " Forward-Backward "
- Reconhecimento: algoritmo de Viterbi
- Coeficientes Mel-SME, Delta(Mel-SME), Delta-Delta(Mel-SME), Energia, Delta-Energia e Delta-Delta-Energia
- Função densidade de probabilidade de Laplace para modelar a duração de estado com coeficiente de ponderação empírico

Na tabela 7.41 são mostrados os resultados para matriz covariância diagonal e seqüência de treinamento 1 e a tabela 7.42 apresenta os resultados matriz covariância cheia e seqüência de treinamento 3.

Estado final	Número de estados	Número de misturas	Coef. de dur. de estado	% de Acerto	Tempo médio de rec. (1 palavra) (s)
qualquer estado	10	3	-	97.82	24.4
			9.0	98.65	24.7
	diferente para cada palavra	5	-	97.82	30.8
			20.0	99.00	30.8
		7	-	97.88	43.8
			20.0	98.88	44.1
último estado	diferente para cada palavra	5	-	98.06	31.8
			9.0	98.65	31.8

Tabela 7.41: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM contínuo com diferentes números de estados, incorporando duração de estado como pós-processador e combinando vários tipos de coeficientes. Utilizou-se a matriz covariância diagonal.

Estado final	Número de estados	Número de misturas	Coef. de dur. de estado	% de Acerto	Tempo médio de rec. (1 palavra) (s)
qualquer estado	10	1	-	96.82	19.74
			9.0	98.35	19.76
	diferente para cada palavra	1	-	96.59	15.48
			9.0	99.06	15.61
		3	-	97.65	45.66
			9.0	98.88	46.0
último estado	diferente para cada palavra	1	-	97.00	15.51
			9.0	99.12	15.83
		3	-	97.65	45.66
			9.0	98.88	46.1

Tabela 7.42: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM contínuo com diferentes números de estados, incorporando duração de estado como pós-processador e combinando vários tipos de coeficientes. Utilizou-se matriz covariância cheia.

Analisando-se as tabelas 7.41 e 7.42 observa-se que os desempenhos dos reconhecedores melhoraram bastante ao combinar-se diferentes coeficientes e ao incorporar a duração de estado. Neste caso, o modelamento da duração de estado conseguiu suprir as deficiências do uso de poucas misturas para modelar os coeficientes. Nota-se também que o fato de considerar-se como estado final apenas o último estado, acarretou um aumento muito pequeno no desempenho do reconhecedor.

7.6 Comparação entre os Tipos de HMMS

Serão apresentadas a seguir tabelas comparativas entre os tipos de HMMs. Todos os reconhecedores implementados apresentaram as seguintes características:

- 10 estados
- " Codebook " com 128 vetores código (HMM discreto) para os coeficientes: Mel-SME, Delta(Mel-SME), Delta-Delta(Mel-SME)
- " Codebook " com 32 vetores código (HMM discreto) para os coeficientes: Energia, Delta-Energia e Delta-Delta-Energia
- " Codebook " com 128 misturas (HMM semicontínuo) para os coeficientes: Mel-SME, Delta(Mel-SME), Delta-Delta(Mel-SME)
- " Codebook " com 32 misturas (HMM semicontínuo) para os coeficientes: Energia, Delta-Energia e Delta-Delta-Energia
- 3 misturas com matriz covariância diagonal (HMM contínuo)
- 1 mistura com matriz covariância cheia (HMM contínuo)
- Treinamento: algoritmo " Forward-Backward "
- Reconhecimento: algoritmo de Viterbi

• Função densidade de probabilidade de Laplace para modelar a duração de estado com coeficiente de ponderação empírico

• Detecção de início/fim

• Estado final: qualquer estado

Na tabela 7.43 são apresentados os desempenhos dos reconhecedores usando coeficientes Mel-SME e seqüência de treinamento 1. Para os reconhecedores utilizando HMM contínuo com matriz cheia foi empregada a seqüência de treinamento 3.

Tipo de HMM	Tipo de matriz cov.	Coef. de dur. de estado	% de Acerto	Tempo médio de trein. (1 modelo)(s)	Tempo médio de rec. (1 palavra)(s)
Discreto	-	-	93.82	11	0.49
		6.5	95.94	11.5	0.50
Semi-contínuo	Diagonal	-	95.06	1:52	3.40
		3.0	97.59	1:45	3.28
	Cheia	-	96.35	4:50	4.21
		3.0	97.82	5:38	4.30
Contínuo	Diagonal	-	95.65	46.5	4.82
		3.0	96.71	49.1	4.84
	Cheia	-	95.53	2:47	5.55
		3.0	97.12	2:47	5.55

Tabela 7.43: Comparação entre o desempenho de reconhecedores de palavras isoladas independente do locutor usando os vários tipos de HMM e coeficientes Mel-SME.

Comparando-se os resultados da tabela 7.43 observa-se que os melhores resultados foram conseguidos usando HMM semicontínuo com matriz cheia e HMM contínuo com matriz diagonal. Deve ser ressaltado que o número de misturas utilizado pelo HMM semicontínuo é muito maior que o empregado pelo HMM contínuo. Isso também justifica o fato do tempo de treinamento do HMM semicontínuo ter sido maior que o tempo de treinamento do HMM contínuo. O desempenho do HMM contínuo com matriz cheia foi bom, mesmo utilizando apenas uma mistura, mas foi necessário utilizar uma seqüência de treinamento maior. Outro ponto a ser destacado é que o HMM discreto requer o menor esforço computacional, mas o seu desempenho é inferior aos outros. Com o uso da incorporação da duração de estado como pós-processador, houve uma melhora no desempenho de todos reconhecedores, sendo que esta foi maior para os reconhecedores que apresentavam pior desempenho. Embora tendo o maior custo computacional, o HMM contínuo apresenta-se como a melhor opção quando se dispõe de grandes seqüências de treinamento e assim pode-se empregar um maior número de misturas.

Na tabela 7.44 são apresentados os desempenhos dos reconhecedores usando coeficientes Mel-SME, Delta(Mel-SME), Delta-Delta(Mel-SME), Energia, Delta-E-

nergia e Delta-Delta-Energia e seqüência de treinamento 1. Para os reconhecedores utilizando HMM contínuo com matriz cheia foi empregada a seqüência de treinamento 3.

Tipo de HMM	Tipo de matriz cov.	Coef. de dur. de estado	% de Acerto	Tempo médio de rec. (1 palavra) (s)
Discreto	-	-	98.59	0.77
		6.5	98.94	0.90
Semi-contínuo	Diagonal	-	99.12	11.4
		9.0	99.47	11.4
	Cheia	-	99.12	14.3
		9.0	99.29	14.3
Contínuo	Diagonal	-	97.82	24.4
		9.0	98.65	24.7
	Cheia	-	96.82	19.74
		9.0	98.35	19.76

Tabela 7.44: Comparação entre o desempenho de reconhecedores de palavras isoladas independente do locutor usando os vários tipos de HMM e combinando diferentes tipos de parâmetros.

A partir dos resultados da tabela 7.44 nota-se que à medida que os reconhecedores vão sendo sofisticados pelo uso de mais de um tipo de parâmetro e pela incorporação da duração de estado, a diferença entre os desempenhos de reconhecedores usando os diferentes tipos de HMMs torna-se menor. Entretanto, o HMM discreto continua apresentando um custo computacional muito menor.

Comparando-se os três tipos de HMM implementados pode-se concluir que:

- HMM Discreto: é o mais simples e utiliza um tempo de processamento menor. Entretanto o seu desempenho é inferior ao outros. Além disso, é necessário o uso de quantização vetorial.

- HMM Semicontínuo: é mais complexo que o discreto mas pode-se conseguir melhores taxas de acerto às custas de um aumento no tempo de processamento

- HMM Contínuo: é o mais complexo dos três e, devido ao grande número de parâmetros, requer uma seqüência de treinamento maior. Não necessita do uso de quantização vetorial e proporciona as melhores taxas de acerto mas apresenta um alto custo computacional.

Dependendo do tipo de aplicação, é mais vantajoso usar o HMM Discreto combinando vários parâmetros e incorporando duração de estado.

7.7 Considerações Finais

Considerando-se os resultados obtidos através da utilização da técnica de Modelos Ocultos de Markov (HMM) em reconhecedores de palavras isoladas tem-se:

- Treinamento: algoritmo " Forward-Backward " requer um maior esforço computacional que o algoritmo de Viterbi e procedimento " K-means "
- Reconhecimento: algoritmo de Viterbi apresenta um maior custo computacional que o algoritmo " Forward "
- Combinando-se os algoritmos " Forward-Backward " no treinamento e " Forward " no reconhecimento, consegue-se reconhecedores com um desempenho levemente superior
- A consideração de apenas um estado como estado final (último estado) praticamente não altera o desempenho do reconhecedor
- O aumento do número de parâmetros (número de estados, tamanho do " codebook ", número de misturas) do reconhecedor melhora o desempenho do mesmo, mas está limitado pelo tamanho da seqüência de treinamento.
- Pode-se melhorar o desempenho dos reconhecedores aumentando-se o tamanho da seqüência de treinamento
- A incorporação da duração de estado como um pós-processador melhora o desempenho dos reconhecedores e a melhor função para modelar a duração de estado é a função densidade de probabilidade de Laplace. O custo computacional do uso do pós-processador pode ser considerado desprezível
- O uso de mais de um tipo de coeficiente também melhora o desempenho do reconhecedor, principalmente quando são adicionadas as derivadas do coeficiente utilizado. Entretanto, isso acarreta um aumento no tempo de processamento.
- Para vocabulários pequenos pode-se conseguir boas taxas de acerto sem a necessidade de reconhecedores muito sofisticados.

Capítulo 8

AVALIAÇÃO DE RECONHECEDORES USANDO MLP

8.1 Introdução

Neste capítulo serão apresentados os resultados obtidos a partir de reconhecedores utilizando redes " Multilayer Perceptron ". Foram implementados reconhecedores com diferentes números de entradas e diferentes números de nós na camada escondida. Além disso, testou-se também a combinação de vários tipos de coeficientes como uma forma de melhorar o desempenho do reconhecedor.

8.2 Reconhecedores Implementados

Todos os reconhecedores implementados usaram redes " Multilayer Perceptron ". Para cálculo dos parâmetros foram utilizadas janelas de Hamming com 20ms e superposição variável. O número de quadros era igual para todas as palavras, independentemente da duração das mesmas. Se a duração do quadro fosse maior que 20 ms, o tamanho da janela era alterado para 1.5 vezes o tamanho do quadro, obtendo-se assim superposição de 25% entre janelas adjacentes. Em todos os reconhecedores implementados usou-se deteçcão de início/fim e os coeficientes utilizados foram calculados da forma descrita no capítulo 6.

O vetor de entrada da rede foi normalizado de modo a apresentar variância igual a 0.25.

8.2.1 Treinamento

Para o treinamento das redes utilizou-se o algoritmo " Back Propagation ", sendo que o treinamento era encerrado quando o número de erros fosse inferior a 0.3% ou o mesmo não diminuisse por 200 iterações consecutivas. Considerou-se como erro o fato da maior saída da rede não corresponder à palavra correta ou ser menor que 0.8. Assim, um acerto era obtido quando a maior saída da rede fosse maior que 0.8 e correspondesse à palavra correta.

Os coeficientes de ponderação da rede eram atualizados após a apresentação de cada exemplo da seqüência de treinamento. Usou-se apenas a seqüência de treinamento 1 (descrita no capítulo 7) e vocabulário de 50 palavras.

8.2.2 Reconhecimento

No reconhecimento, a palavra correspondente à saída de maior valor da rede era escolhida como palavra reconhecida.

Para avaliar os reconhecedores usou-se o mesmo conjunto de palavras descrito no capítulo 7.

8.2.3 Número de Nós na Camada Escondida

Para avaliar esse parâmetro, foram implementados reconhecedores com 1280 entradas e 50 saídas usando coeficientes " Mel Frequency Cepstrum " (16 coef. por quadro). Os desempenhos desses reconhecedores são mostrados na tabela 8.1.

Número de nós na camada escondida	% de Acerto	Tempo médio de rec. (1 palavra) (s)
80	92.18	0.07
100	93.41	0.07
120	93.24	0.11
150	88.94	0.12

Tabela 8.1: Desempenho de reconhecedores de palavras isoladas independente do locutor usando " multilayer perceptron " para diferentes números de nós na camada escondida. Os reconhecedores apresentavam 1280 entradas.

Observa-se que o melhor desempenho foi obtido pela rede com 100 nós na camada escondida. Ao aumentar-se esse número, o desempenho piorou pois aumentou-se muito o número de parâmetros a ser estimado mantendo-se a mesma seqüência de treinamento.

8.2.4 Número de Nós na Camada de Entrada

Foram implementados reconhecedores com 100 nós na camada escondida, 50 nós na camada de saída e novamente foram usados os coeficientes " Mel Frequency Cepstrum " (16 coef. por quadro). A tabela 8.2 mostra a taxa de acerto desses reconhecedores.

Número de nós na na entrada	% de Acerto
960	92.65
1280	93.41
1600	92.59
1920	92.29

Tabela 8.2: Desempenho de reconhecedores de palavras isoladas independente do locutor usando " multilayer perceptron " para diferentes números de nós na camada de entrada. A camada escondida era formada por 100 nós.

Analisando-se os resultados nota-se que a maior taxa de acerto foi obtida usando-se 1280 nós na camada de entrada. Aumentando-se esse número houve um decréscimo na taxa de acerto do reconhecedor. Mais uma vez isso é explicado pelo fato de que aumento do número de parâmetros da rede requer uma seqüência de treinamento maior para a obtenção de uma boa estimativa dos mesmos.

8.2.5 Combinando Diferentes Tipos de Coeficientes

Para tentar melhorar o desempenho dos reconhecedores, testou-se a combinação de coeficientes Mel e Delta-Mel (DMel). Esses parâmetros também foram combinados com a energia do sinal (E).

Na implementação dos reconhecedores foram usados 100 nós na camada escondida e 50 nós na camada de saída. Na tabela 8.3 são apresentados os resultados obtidos.

Tipo de Coeficiente	Número de entradas	% de Acerto	Tempo médio de rec. (1 palavra) (s)
Mel	1280	93.41	0.07
DMel	1280	89.35	0.07
Mel + E	1360	93.82	0.12
Mel +DMel	2560	90.35	0.17
Mel+DMel+E	2640	91.76	0.17

Tabela 8.3: Desempenho de reconhecedores de palavras isoladas independente do locutor usando " multilayer perceptron " e combinações de vários tipos de coeficientes. A camada escondida era formada por 100 nós. E = energia e DMel = Delta-Mel.

A partir dos resultados da tabela 8.3 percebe-se que apenas a combinação Mel e energia aumentou a taxa de acerto do reconhecedor. O uso de coeficientes Mel juntamento com coeficientes Delta-Mel não melhorou o desempenho do reconhecedor pois em redes MLP, a informação dos coeficientes Delta é obtida automaticamente pela combinação linear das entradas. Outro fator que deve ser ressaltado é que ao combinar-se tipos diferentes de coeficientes, aumenta-se o número de parâmetros da rede e assim torna-se necessária uma maior seqüência de treinamento.

8.3 Considerações Finais

De uma forma geral, as redes " Multilayer Perceptron " apresentaram resultados que podem ser considerados apenas razoáveis. Além disso, não dispõem de possibilidades de sofisticação para melhorar a taxa de acerto e o treinamento pode ser bastante demorado (por volta de dias). Também apresentam dificuldades para trabalhar com as variações temporais dos sinal de fala pois é necessário trabalhar com número fixo de quadros ao invés de quadros com duração fixa. Isso implica em receber a palavra inteira para poder começar o processamento.

Como pontos positivos devem ser ressaltados o pequeno tempo necessário para o reconhecimento de uma palavra e facilidade de implementação. Soma-se a isso o fato de não ser necessário o uso de quantização vetorial.

Capítulo 9

AVALIAÇÃO DE RECONHECEDORES HÍBRIDOS

9.1 Introdução

Neste capítulo serão apresentados os resultados das avaliações da várias propostas de sistemas híbridos descritas no capítulo 5.

Como esse sistemas combinam HMM e MLP, foram considerados nas avaliações características dessas duas técnicas como: número de nós nas camadas de entrada e escondida, número de estados, modelamento da duração de estado, densidade de probabilidade discreta e contínua e outros.

9.2 HMM usando MLP como Pós-Processador

Foram implementadas três variações desse sistema.

- Proposta 1: Consiste em apresentar à rede MLP um vetor formado pelas probabilidades de cada modelo ter gerado as observações correspondentes à palavra desconhecida $P(O|\lambda)$.

- Proposta 2: Nesta proposta, o vetor de entrada é formado pelas probabilidades de cada modelo ter gerado a palavra desconhecida e pelas probabilidades de duração de estado. Esse vetor apresenta dimensão igual ao dobro da dimensão do vetor usado na primeira implementação. O objetivo desta proposta é usar

a rede MLP para somar as probabilidades do modelo e da duração de estado sem ser necessário o cálculo de coeficientes de ponderação.

- Proposta 3: A probabilidade de cada modelo ter gerado a palavra desconhecida e a correspondente probabilidade de duração de estado são somadas usando coeficiente de ponderação igual a 1. Após a soma obtém-se um vetor com dimensão igual à dimensão do vetor usado na primeira implementação.

Em todas as propostas usou-se a probabilidade de maior valor para normalizar o vetor de entrada da rede MLP.

Foi usada detecção de início/fim em todos os reconhecedores implementados e os coeficientes utilizados foram calculados como descrito no capítulo 6.

Todos os reconhecedores implementados usando HMM apresentavam as seguintes características:

- HMM Discreto:
  - 10 estados
  - " codebook " com 128 vetores código para os coeficientes Mel-SME, Delta(Mel-SME) (D) e Delta-Delta(Mel-SME) (DD)
  - " codebook " com 32 vetores código para energia (E), Delta-energia (DE) e Delta-Delta-energia (DDE)
  - estado final: qualquer estado

- HMM Contínuo:
  - 10 estados
  - 3 misturas com matriz covariância diagonal
  - estado final: qualquer estado

9.2.1 Treinamento

Inicialmente os HMMs foram treinados usando-se o mesmo procedimento descrito no capítulo 7, sendo utilizado o algoritmo " Forward-Backward ". Para o cálculo da duração de estados usou-se o algoritmo de Viterbi. Após o treinamento de todos os modelos, os mesmos foram usados para o cálculo das probabilidades de cada modelo ter gerado cada exemplo da seqüência de treinamento. As probabilidades correspondentes a cada exemplo foram combinadas em um único vetor e normalizadas pela probabilidade de maior valor. Esses vetores formaram a seqüência de treinamento da rede " multilayer perceptron ".

Para o treinamento das redes utilizou-se o algoritmo " Back Propagation ", sendo que o treinamento era encerrado quando o número de erros fosse inferior a 0.3% ou o mesmo não diminuisse por 200 iterações consecutivas. Considerou-se como erro o fato da maior saída da rede não corresponder à palavra correta ou ser menor que 0.8. Os coeficientes de ponderação da rede eram atualizados após a apresentação de cada exemplo da seqüência de treinamento.

Foram usadas as seqüências de treinamento descritas no capítulo 7 e vocabulário de 50 palavras.

9.2.2 Reconhecimento

No reconhecimento, primeiramente calculava-se as probabilidades dos HMMs usando o algoritmo " Forward ", e depois estas eram normalizadas e apresentadas à rede. A palavra correspondente à saída de maior valor da rede era escolhida como palavra reconhecida. Para os reconhecedores que usaram a probabilidade de duração de estado, empregou-se o algoritmo de Viterbi para o reconhecimento.

Para avaliar os reconhecedores usou-se o mesmo conjunto de palavras descrito no capítulo 7.

9.2.3 Número de Nós na Camada Escondida

Primeiramente avaliou-se o número necessário de nós na camada escondida da rede MLP. Assim, usando-se a proposta 1 foram implementados reconhecedores combinando HMM discreto e coeficientes " Mel Frequency Cepstrum " com subtração da média e rede MLP com 50 nós na camada de entrada e 50 nós na camada de saída. Foi utilizada a seqüência de treinamento 1. A tabela 9.1 apresenta os resultados dessas implementações.

Número de nós na camada escondida	% de Acerto	Tempo médio de rec. (1 palavra) (s)
0	94.94	0.23
50	90.29	0.23
80	91.59	0.26
100	90.88	0.25
150	91.41	0.27
200	90.12	0.28

Tabela 9.1: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM e MLP para diferentes números de nós na camada escondida.

Analisando-se os resultados observa-se que o maior taxa de acerto foi obtida

sem o uso de camada escondida. Dessa forma, os próximos reconhecedores a serem descritos não apresentam camada escondida na rede MLP.

9.2.4 Combinando Diferentes Tipos de Coeficientes

Na tabela 9.2 são apresentados os desempenhos de reconhecedores usando HMM discreto e reconhecedores híbridos seguindo a proposta 1 para vários tipos de coeficientes e combinações dos mesmos. A rede MLP apresentava 50 nós na camada de entrada e 50 nós na camada de saída. Empregou-se a seqüência de treinamento 1 para treinamento dos HMMs e da rede MLP.

Reconhecedor	Tipo de Coeficiente	% de Acerto	Tempo médio de rec. (1 palavra) (s)
HMM discreto	Mel-SME	94.12	0.27
	D(Mel-SME)	94.47	0.27
	Mel-SME+D+DD	98.00	0.33
	Mel-SME+D+DD+E+DE+DDE	98.65	0.38
HMM discreto + MLP	Mel-SME	95.06	0.23
	D(Mel-SME)	96.41	0.23
	Mel-SME+D+DD	98.47	0.36
	Mel-SME+D+DD+E+DE+DDE	99.06	0.38

Tabela 9.2: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto e sistemas híbridos (HMM e MLP) para diferentes tipos de coeficientes.

A partir dos resultados da tabela 9.2 nota-se que os desempenhos dos reconhecedores híbridos foram superiores para todos os tipos de coeficientes. Isso é devido ao fato da rede considerar as probabilidades conjuntamente e não separadamente como fazem os HMMs. Além disso, observa-se que praticamente não houve aumento do tempo necessário para o reconhecimento de uma palavra. Portanto, esses sistemas híbridos são uma boa opção para a obtenção de reconhecedores com melhores taxas de acerto.

Considerando-se a seqüência de treinamento 3 foram implementados reconhecedores com HMM discreto e um sistema híbrido combinando vários tipos de coeficientes. Novamente, a rede MLP apresentava 50 nós na camada de entrada e 50 nós na camada de saída. A tabela 9.3 mostra o resultado da comparação entre os reconhecedores implementados.

Novamente, o desempenho do reconhecedor híbrido foi superior ao reconhecedor HMM. Entretanto, à medida que melhora o desempenho do reconhecedor HMM diminui a diferença entre as taxas de acerto dos dois sistemas.

Na tabela 9.4 são mostrados os desempenhos de reconhecedores usando HMM com densidade de probabilidade contínua. Os reconhecedores empregaram 3 mis-

Reconhecedor	Tipo de Coeficiente	% de Acerto	Tempo médio de rec. (1 palavra) (s)
HMM disc.	Mel-SME+D+DD+E+DE+DDE	99.06	0.38
HMM disc.+MLP	Mel-SME+D+DD+E+DE+DDE	99.18	0.38

Tabela 9.3: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto e sistemas híbridos (HMM e MLP) para diferentes tipos de coeficientes. Os reconhecedores foram treinados usando-se a seqüência de treinamento 3.

turas com matriz covariância diagonal. A rede MLP implementada apresentava 50 nós na camada de entrada e 50 nós na camada de saída. O treinamento foi feito usando-se a seqüência 1.

Reconhecedor	Tipo de Coeficiente	% de Acerto	Tempo médio de rec. (1 palavra) (s)
HMM contínuo	Mel-SME	95.71	4.57
HMM cont.+MLP	Mel-SME	95.94	4.65

Tabela 9.4: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM contínuo e sistemas híbridos (HMM e MLP).

Analisando-se a tabela 9.4 nota-se que também para densidade contínua, o sistema híbrido apresenta uma maior taxa de acerto.

9.2.5 Incorporação da Duração de Estado

Finalmente foram avaliados os sistemas híbridos descritos nas propostas 2 e 3. Nesses reconhecedores, a duração de estado foi modelada usando a função densidade de probabilidade de Laplace e usou-se o algoritmo de Viterbi para reconhecimento. Utilizou-se coeficiente de duração de estado empírico. A rede MLP apresentava 50 nós na camada de saída. Foi usada a seqüência de treinamento 1 para treinar os HMMs e a rede MLP. Os resultados dessas avaliações estão na tabela 9.5.

Os resultados da tabela 9.5 mostram que os sistemas híbridos propostos não superaram o desempenho dos HMMs que combinaram a probabilidade de duração de estado usando um coeficiente de ponderação empírico. Considerando que os resultados da proposta 3 ficaram bastante próximos do reconhecedor de melhor desempenho, pode-se afirmar que esta é uma boa forma de incorporar as probabilidades de duração de estado principalmente pelo fato de eliminar o cálculo do coeficiente de ponderação por tentativa e erro.

Reconhecedor	Número de ent. (MLP)	Tipo de Coeficiente	% de Acerto	Tempo médio rec. (1 pal.)(s)
HMM discreto	-	Mel-SME	94.00	0.43
		Mel-SME+D+DD+E+DE+DDE	98.59	0.77
HMM disc. + dur. de est.	-	Mel-SME	95.88	0.43
		Mel-SME+D+DD+E+DE+DDE	98.94	0.90
HMM disc.+dur. de est.+MLP 2	100	Mel-SME	95.00	0.42
		Mel-SME+D+DD+E+DE+DDE	98.84	0.92
HMM disc.+dur. de est+MLP 3	50	Mel-SME	95.65	0.43
		Mel-SME+D+DD+E+DE+DDE	99.00	0.92

Tabela 9.5: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM discreto e sistemas híbridos (HMM e MLP), considerando o modelamento da duração de estado.

9.3 HMM como Segmentador para MLP

Também foram implementadas três variações para esse sistema híbrido:

- Proposta 1: Na fase de reconhecimento, apenas o HMM com maior probabilidade foi usado para segmentar o sinal de fala.

- Proposta 2: Nesta proposta, consideraram-se os três HMMs com maiores probabilidades para segmentar o sinal de fala.

- Proposta 3: Nesta alternativa, todos os HMMs foram usados para segmentar o sinal de fala.

Os reconhecedores seguindo as propostas 2 e 3 foram implementados com e sem uso de coeficientes de ponderação. Neste último caso, as probabilidades dos HMMs normalizadas pelo valor máximo foram usadas como coeficientes de ponderação.

Após a segmentação e obtenção do vetor de entrada da rede MLP, este foi normalizado de modo a apresentar variância igual a 0.25.

Foi usada detecção de início/fim em todos os reconhecedores implementados e os coeficientes utilizados foram calculados como descrito no capítulo 6.

O reconhecedores implementados usaram os seguintes " codebooks ":

- " codebook " com 128 vetores código para os coeficientes Mel, Mel-SME, Delta(Mel-SME) (D) e Delta-Delta(Mel-SME) (DD)
- " codebook " com 32 vetores código para energia (E), Delta-energia (DE) e Delta-Delta-energia (DDE)

9.3.1 Treinamento

Inicialmente, os HMMs, considerando um último estado como estado final, foram treinados usando o mesmo procedimento descrito no capítulo 7, sendo utilizado o algoritmo " Forward-Backward ". Após o treinamento de todos os modelos, cada exemplo da seqüência de treinamento foi segmentado usando o modelo correspondente e o algoritmo de Viterbi. A seguir calculou-se o vetor média dos vetores agrupados em um mesmo estado e depois esses vetores média foram concatenados em um único vetor. O conjunto de vetores normalizados formou a seqüência de treinamento da rede " multilayer perceptron ".

Para o treinamento das redes utilizou-se o algoritmo " Back Propagation ", sendo que o treinamento era encerrado quando o número de erros fosse inferior a 0.3% ou o mesmo não diminuisse por 200 iterações consecutivas. Considerou-se como erro o fato da maior saída da rede não corresponder à palavra correta ou ser menor que 0.8. Os coeficientes de ponderação da rede foram atualizados após a apresentação de cada exemplo da seqüência de treinamento.

Usou-se apenas a seqüência de treinamento 1 (descrita no capítulo 7) e vocabulário de 50 palavras.

9.3.2 Reconhecimento

No reconhecimento, os HMMs e o algoritmo de Viterbi foram usados para segmentar o sinal de fala. A palavra correspondente à saída de maior valor da rede era escolhida como palavra reconhecida. Para avaliar os reconhecedores usou-se o mesmo conjunto de palavras descrito no capítulo 7.

9.3.3 Comparação entre as Três Propostas

Para comparar as três propostas apresentadas para a fase de reconhecimento, implementaram-se reconhecedores usando HMM discreto com 10 estados, 128 vetores código (16 coef.por vetor) e coeficientes " Mel Frequency Cepstrum " com subtração da média e rede MLP com 160 nós na camada de entrada, 100 nós na camada escondida e 50 nós na camada de saída. Os resultados obtidos estão na tabela 9.6.

A partir dos resultados da tabela 9.6 percebe-se que a proposta 1 apresentou o melhor resultado. Deve-se ressaltar que para as propostas 2 e 3 ocorreram diferenças entre os procedimentos usados nas fases de treinamento e reconhecimento, pois no primeiro considerou-se apenas o modelo correto para segmentar cada pala-

Algoritmo de Reconhecimento	% de Acerto	Tempo médio de rec. (1 palavra) (s)
1	94.76	0.49
2	90.76	0.53
3	68.18	1.17
2 com coef. de ponderação	92.76	0.53
3 com coef. de ponderação	71.47	1.17

Tabela 9.6: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM como segmentador para MLP considerando diferentes propostas para a fase de reconhecimento.

vra pertencente à seqüência de treinamento. O procedimento correto para a fase de treinamento da rede nas propostas 2 e 3 não foi testado neste trabalho devido ao grande esforço computacional requerido pois para a proposta 3, a seqüência de treinamento seria multiplicada pelo número de modelos utilizados, o que neste caso corresponde a 50. Entretando, observa-se que houve uma melhora no desempenho dos reconhecedores com o uso das probabilidades dos HMMs normalizadas como coeficientes de ponderação. Isso indica que com uma mudança no treinamento da rede, pode-se conseguir melhores resultados. Devido a esses motivos, resolveu-se empregar apenas a proposta 1 nos próximos reconhecedores a serem descritos.

9.3.4 Número de Nós na Camada Escondida

Inicialmente avaliou-se o número de nós na camada escondida da rede MLP. Assim, usando a proposta 1 foram implementados reconhecedores combinando HMM discreto com 10 estados, 128 vetores código (16 coef.por vetor) e coeficientes " Mel Frequency Cepstrum " com subtração da média e rede MLP com 160 nós na camada de entrada e 50 nós na camada de saída. A tabela 9.7 apresenta os resultados dessas implementações.

Número de nós na camada escondida	% de Acerto	Tempo médio de rec. (1 palavra) (s)
0	94.53	0.50
30	94.18	0.56
50	94.76	0.53
80	94.65	0.50
100	94.76	0.49
120	94.65	0.53
150	94.59	0.52
200	94.65	0.51

Tabela 9.7: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM como segmentador para MLP para diferentes números de nós na camada escondida da rede.

Analisando-se os resultados da tabela 9.7 observa-se que a maior taxa de acerto foi obtida para reconhecedores com 100 nós na camada escondida. Dessa forma, os próximos reconhecedores a serem descritos utilizaram 100 nós na camada escondida na rede MLP.

9.3.5 Número de Nós na Camada de Entrada

Para avaliar a influência do número de nós na camada de entrada, foram implementados reconhecedores usando HMM com diferentes números de estados, 128 vetores código (16 coef. por vetor) e coeficientes " Mel Frequency Cepstrum " com subtração da média e rede MLP com 100 nós na camada escondida e 50 nós na camada de saída. Na tabela 9.8 são mostrados os desempenhos dos reconhecedores.

Reconhecedor	Número de estados	Número de nós na camada de entrada	% de Acerto	Tempo médio de rec. (1 palavra) (s)
HMM discreto	10	-	94.00	0.49
	20	-	94.76	1.47
	30	-	95.53	2.99
	40	-	94.59	5.12
HMM + MLP	10	160	94.76	0.50
	20	320	95.59	1.48
	30	480	95.41	3.03
	40	640	95.65	5.20

Tabela 9.8: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM como segmentador para MLP considerando redes com diferentes números de nós na camada de entrada.

Analisando-se os resultados da tabela 9.8 vê-se que com o aumento do número de estados dos HMMs, e conseqüente aumento do número de entradas da rede, consegue-se uma melhora na taxa de acerto dos reconhecedores híbridos, desde que o aumento do número de estados não seja excessivo, o que compromete o treinamento do HMM e da rede MLP. Além disso, observa-se também que estes apresentaram um melhor desempenho que os HMMs, mostrando a capacidade de discriminação das redes MLP. Deve ser lembrado que existe um limite para o aumento do número de estados dos HMMs.

9.3.6 Combinando Diferentes Tipos de Coeficientes

Considerando os vários tipos de coeficientes e combinações dos mesmos foram implementados alguns reconhecedores usando HMM discreto com 10 estados, sendo que os resultados dos mesmos estão apresentados na tabela 9.9.

Reconhecedor	Tipo de de Coeficiente	% de Acerto	Tempo médio de rec. (1 palavra) (s)
MLP	Mel	93.41	0.07
	Mel-SME	89.59	0.08
HMM disc.	Mel	91.18	0.42
	Mel-SME	94.00	0.43
	Mel-SME+D+DD+E+DE+DDE	98.59	0.77
HMM disc. + MLP	Mel	93.12	0.50
	Mel-SME	94.76	0.50
	Mel-SME+D+DD+E+DE+DDE	98.29	0.92

Tabela 9.9: Desempenho de reconhecedores de palavras isoladas independente do locutor usando diferentes tipos de coeficientes.

A partir dos resultados da tabela 9.9 nota-se que para um único tipo de parâmetro os sistemas híbridos obtêm os melhores resultados mas, quando vários tipos de coeficientes são combinados, o desempenho dos sistemas híbrido é inferior ao reconhecedores que usam apenas HMM.

Na tabela 9.10 são mostrados os desempenhos de reconhecedores usando HMM com densidade de probabilidade contínua. Os reconhecedores empregaram 3 misturas com matriz covariância diagonal e coeficientes " Mel Frequency Cepstrum " com subtração da média.

Reconhecedor	Número de estados	Número de nós na camada de entrada	% de Acerto	Tempo médio de rec. (1 palavra) (s)
HMM contínuo	10	-	95.71	4.57
	15	-	97.12	7.01
HMM cont.+MLP	10	160	95.59	4.97
	15	240	95.65	7.46

Tabela 9.10: Desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM contínuo e sistemas híbridos (HMM como segmentador para MLP).

Com o uso de HMMs com densidade de probabilidade contínua, os sistemas híbridos não conseguiram superar os reconhecedores usando HMM contínuo. Isso pode ser explicado pelo fato dos HMM contínuos não usarem quantização vetorial.

9.4 Estimação de Probabilidades a Posteriori usando MLP

Nos sistemas implementados, usou-se a probabilidade a posteriori $\gamma_t(i) = P(q_t = i|O, \lambda)$ como o alvo da rede. Os valores de $\gamma_t(i)$ foram calculados usando HMM discreto, com " codebook " com 128 vetores código e coeficientes

" Mel Frequency Cepstrum " com subtração da média. Nos HMMs implementados, o conjunto de possíveis estados finais era formado por todos os estados.

Foi usada detecção de início/fim em todos os reconhecedores implementados.

Para treinamento e avaliação dos reconhecedores foi utilizado apenas o conjunto de dígitos (uma elocução de cada dígito por locutor).

Para cálculo dos coeficientes foram utilizados quadros de 10 ms e janelas de Hamming de 20 ms com superposição. Observa-se que foram usados quadros de tamanho fixo para todas as palavras.

9.4.1 Treinamento

Inicialmente os HMMs foram treinados usando o mesmo procedimento descrito no capítulo 7, sendo utilizado o algoritmo " Forward-Backward ". Após o treinamento de todos os modelos, foram obtidos os valores de $\gamma_t(i)$ que foram usados para o treinamento da rede " multilayer perceptron ".

Para o treinamento das redes utilizou-se o algoritmo " Back Propagation ". O treinamento era encerrado quando o erro quadrático médio fosse inferior a 0.0001 ou não diminuisse por 200 iterações consecutivas.

9.4.2 Reconhecimento

No reconhecimento, primeiramente calculava-se as probabilidades de emissão de símbolos usando a rede MLP e depois usava-se os valores dessas probabilidades nos HMMs para calcular a probabilidade do modelo ter gerado a palavra desconhecida usando o algoritmo " Forward ".

9.4.3 Número de Nós na Camada de Entrada

Foram avaliados reconhecedores usando redes MLP com diferentes números de nós na camada de entrada. Esses reconhecedores usaram HMM discreto com 10 estados. A rede MLP apresentava 10 nós na camada de saída e não apresentava camada escondida. A tabela 9.11 mostra os resultados obtidos.

Analisando-se os resultados da tabela 9.11 observa-se que ao aumentarmos o número de nós na camada de entrada obtém-se reconhecedores com melhor desempenho.

Número de quadros	Número de nós na na entrada	% de Acerto
3	48	86.10
7	112	93.05
11	176	92.51
15	240	95.72

Tabela 9.11: Desempenho de reconhecedores de palavras isoladas independente do locutor usando redes MLP para estimar a probabilidade de emissão em HMM, para diferentes números de nós na camada de entrada.

9.4.4 Comparação entre Reconhecedores

A tabela 9.12 apresenta uma comparação entre reconhecedores utilizando apenas HMM e os sistemas híbridos. Nestes reconhecedores as redes MLP não apresentavam camada escondida e apresentavam 240 nós na camada de entrada. O número de nós na camada de saída era igual ao número de estados dos HMMs.

Reconhecedor	Número de estados	% de Acerto
HMM disc.	10	99.47
	dif. por palavra	98.66
HMM disc.+MLP	10	95.72
	dif. por palavra	87.97

Tabela 9.12: Comparação entre o desempenho de reconhecedores de palavras isoladas independente do locutor usando HMM e sistemas híbridos usando redes MLP para estimar a probabilidade de emissão em HMM.

A partir dos resultados da tabela 9.12 nota-se que o desempenho dos reconhecedores usando apenas HMM é superior ao desempenho dos sistemas híbridos.

9.5 Considerações Finais

Considerando-se os resultados obtidos pode-se afirmar que os sistemas híbridos apresentam grande potencial para se conseguir altas taxas de acerto pois consegue-se explorar as vantagens das duas técnicas HMM e MLP.

O uso de redes MLP como pós-processador para HMM melhora o desempenho do reconhecedor com um custo computacional desprezível na fase de reconhecimento, mas exige um treinamento mais complexo e demorado.

O método HMM e algoritmo de Viterbi mostram-se muito eficientes como segmentadores, e os reconhecedores híbridos implementados usando esse segmentador superaram em muitos casos o desempenho das redes MLP e reconhecedores HMM. Entretanto, esse sistema necessita de um tempo muito maior que as redes MLP para

reconhecer uma palavra. Deve-se ressaltar que podem-se estudar outras formas de combinar os HMMs na fase de reconhecimento, o que pode resultar em um melhor desempenho dos reconhecedores.

Quanto ao uso de redes MLP para estimar a probabilidade de emissão em HMMs, os resultados obtidos não foram bons. Isso talvez seja uma indicação de que a metodologia usada no treinamento não tenha sido a mais correta. Para o cálculo das probabilidades a posteriori, a técnica HMM considera toda a seqüência de observação, enquanto no procedimento adotado para treinar a rede, apenas parte da seqüência foi considerada. Pode-se dizer que este é um tópico que precisa ser melhor discutido e estudado.

Capítulo 10

CONCLUSÕES

10.1 Considerações Finais

Neste trabalho foram avaliadas diferentes técnicas para reconhecimento de fala, sendo comparados os desempenhos de reconhecedores de palavras isoladas independente do locutor implementados usando-se Modelos Ocultos de Markov, redes " Multilayer Perceptron " e sistemas Híbridos combinando essas duas técnicas. Nestas avaliações foram considerados os vários parâmetros que podem influir no desempenho do reconhecedor. Além disso, também foram avaliados algoritmos propostos para partes do reconhecedor como: detecção de início/fim, quantização vetorial e processamento do sinal de fala.

Considerando-se os algoritmos para detecção de início/fim analisados, pode-se concluir que nenhum é perfeito, entretanto para a maioria dos tipos de coeficientes, o uso de detectores de início/fim acarretou um aumento na taxa de acerto do reconhecedor e uma diminuição no tempo de processamento. Para aproveitar as vantagens de cada algoritmo, pode-se usar uma combinação dos mesmos. Como foi mostrado neste trabalho, essa pode ser uma boa solução para melhorar o desempenho dos reconhecedores.

A respeito da quantização vetorial, pode-se concluir que o uso da distância Euclidiana como medida de distorção traz prejuízos muito pequenos para o desempenho do reconhecedor. Por outro lado, é uma medida simples, de fácil implementação e não requer grande esforço computacional. Se tempo de CPU e espaço em disco não forem problema, recomenda-se o uso da função densidade de probabilidade Gaussiana como medida de distorção pois se conseguirá reconhecedores com uma maior

taxa de acerto.

A partir da análise dos vários tipos de coeficientes propostos para representar os sinal de fala, conclui-se que os melhores são os obtidos aplicando-se a Transformada Cosseno Discreta nas saídas de um banco de filtro usando a escala Mel ou Bark, as quais são escalas não lineares de freqüências. Variações desses coeficientes também produzem bons resultados, independentemente da técnica utilizada pelo reconhecedor.

Para a avaliação da técnica HMM, foram implementados reconhecedores usando HMM discreto, contínuo e semicontínuo. Os resultados mostraram que HMM é uma poderosa ferramenta para reconhecimento de fala. Esse método permite a sofisticação dos reconhecedores para melhorar o desempenho dos mesmos. Entretanto, isso está associado a um custo que é o aumento do tempo de processamento. Assim, o desempenho do reconhecedor pode ser melhorado através do modelamento da duração de estado como um pós-processador (neste caso o aumento do tempo de processamento é desprezível), combinando diferentes tipos de parâmetros e suas derivadas ou até combinando vários modelos. Outro ponto a ser ressaltado é a facilidade que os HMMs têm para trabalhar com a variação temporal do sinal de fala. Bons resultados podem ser obtidos usando-se HMM discretos, apesar do uso da quantização vetorial. Entre os tipos de HMM, este é o mais simples e o que requer o menor esforço computacional. HMMs semicontínuos são uma boa opção quando se dispõe de um tempo de processamento maior e necessita-se de um reconhecedor com melhor desempenho. Quando apenas a taxa de acerto do reconhecedor é importante, recomenda-se o uso de HMM contínuo, pois ele permite alcançar as mais altas taxas de acerto, mas exige grandes esforços computacionais e uma seqüência de treinamento maior. A escolha do melhor tipo de HMM deve ser baseada no compromisso entre o tempo de processamento disponível e a taxa de acerto desejada.

A avaliação das redes " Multilayer Perceptron " mostrou que essa técnica apresenta o menor tempo de processamento na fase de reconhecimento, sendo esta uma grande vantagem. Além disso, as redes são simples e os algoritmos utilizados são de fácil implementação. Entretanto, o desempenho dos reconhecedores deixou um pouco a desejar devido à impossibilidade de ser melhorado sem que para isso seja necessário um aumento da seqüência de treinamento. Outro ponto fraco é o tempo de treinamento, que dependendo dos limiares utilizados é muito grande, superando em muito o tempo necessário para treinar os HMMs e obter reconhecedores com a mesma taxa de acerto. O uso de redes MLP é recomendado para sistemas que necessitem de baixo tempo de reconhecimento e que possam operar com maiores

taxas de erro.

Por fim, foram avaliados os sistemas híbridos, os quais combinam HMM e redes MLP. Alguns desses sistemas mostrararam desempenho superior aos HMMs e redes MLP pois apresentam as boas características das duas técnicas. Dessa forma estão unidas a capacidade discriminativa das redes MLP e a capacidade de modelamento dos HMMs. Mas, estes sistemas também apresentam desvantagens como um aumento do custo computacional. Com a evolução dos processadores de sinais, os sistemas híbridos tendem a se tornar a melhor opção para implementação de reconhecedores de fala.

Finalizando, pode-se concluir que é possível a obtenção de reconhecedores de palavras isoladas com altas taxas de acerto. Em alguns casos pode ser necessário um grande esforço computacional. Considerando-se todas as avaliações e comparações realizadas, mostrou-se que a escolha dos melhores algoritmos dependem da aplicação em que se pretende usar o reconhecedor e que não existe uma soluçaõ que seja ótima para todos os casos.

10.2 Contribuições deste Trabalho

A contribuição mais significativa deste trabalho é mostrar a partir das avaliações e comparações realizadas utilizando uma mesma base de dados, os pontos bons e ruins das técnicas empregadas em reconhecimento de fala. Assim, pode-se ressaltar:

- Comparação entre os algoritmos para detecção de início/fim: " Endpoints ", " Bottom-up " e " Top-down "
- Proposta de combinação desses algoritmos
- Comparação entre as seguintes medidas utilizadas em quantização vetorial: distância Euclidiana, distância de Mahalanobis, função densidade de probabilidade Gaussiana (matriz covariância diagonal e cheia) e função densidade de probabilidade Gaussiana simplificada (matriz covariância diagonal e cheia)
- Comparação entre 22 tipos de coeficientes utilizados para representar o sinal de fala, utilizando reconhecedores usando HMM e redes MLP
- Comparação entre HMM discreto, contínuo e semicontínuo considerando diversos parâmetros

- Comparação entre funções densidade de probabilidade para modelar a duração de estado: não paramétrica, Gamma, Gaussiana, Laplace e Rayleigh
- Proposta de um método para obtenção dos coeficientes de ponderação das probabilidades de duração de estado
- Avaliação de sistema Híbridos e propostas para alguns desses sistemas

10.3 Trabalhos Futuros

Como trabalhos futuros, recomenda-se a avaliação desses algoritmos e técnicas para sinais de fala com diferentes valores de relação sinal/ruído. Sabe-se que o comportamento de alguns algoritmos é muito dependende da relação sinal/ruído e assim torna-se importante essa nova avaliação.

Outro ponto que pode ser melhor estudado e aprofundado é a implementação de reconhecedores híbridos. Como foi mostrado neste trabalho, estes sistemas apresentam grande potencial e precisam ser melhor discutidos.

Apêndice A

BASE DE DADOS E AMBIENTE DE TRABALHO

A.1 Ambiente de Trabalho

As simulações realizadas neste trabalho foram executadas em tempo não real em estações de trabalho SPARCstation 20 da SUN, tendo como ambiente o sistema SunOS versão 4.3.1.

Todos os programas desenvolvidos para a simulação dos reconhecedores foram escritos em linguagem C e não houve preocupação com a otimização dos programas.

A.2 Sistema de Aquisição

Os arquivos de fala utilizados para as simulações foram gravados usando o sistema de aquisição de dados do CPqD-TELEBRÁS. Esse sistema utiliza a placa DSP-16 Data Aquisition Processor, fabricada pela Ariel. Esta placa está acoplada a um microcomputador, apresentando a seguinte configuração para aquisição de dados:

- Os arquivos de fala sob a forma analógica são limitados em faixa entre 100 Hz e 3400 Hz, amostrados a 8000 Hz e quantizados com 16 bits por amostra.

Um conversor A/D de 16 bits permite uma excursão dos sinais no intervalo -32768 a 32767 níveis de quantização.

A.3 Base de Dados

A base de dados utilizada nas simulações era formada por um vocabulário de 50 palavras. A aquisição desta base de dados foi realizada em um ambiente com ruído de escritório e utilizou-se o sistema de aquisição descrito na seção anterior.

As 50 palavras foram faladas por 69 locutores adultos, sendo 43 homens e 26 mulheres. Cada locutor falou cada palavra 3 vezes.

O conjunto de locutores formado por 22 homens e 13 mulheres foi usado no treinamento dos reconhecedores. Os outros locutores foram empregados na avaliação dos reconhecedores.

O vocabulário era formado pelos seguintes grupos de palavras:

- Dígitos: zero, um, dois, três, quatro, cinco, seis, sete, oito, nove, meia
- Comandos: sim, não, terminar, repetir, continuar, voltar, avançar, certo, errado, opções, ajuda
- Regiões: norte, nordeste, sul, sudeste, centro-oeste
- Signos: áries, touro, câncer, leão, gêmeos, virgem, libra, escorpião, capricórnio, sagitário, aquário, peixes
- Opções: horóscopo, dólar, real, tempo, esportes
- Organização: departamento, divisão, seção, coordenação, imagem, voz

Referências Bibliográficas

[1] C.-H. Lee and L. R. Rabiner. Directions in Automatic Speech Recognition. *NTT Review*, 7(2):19–29, Março 1995.

[2] J. G. Wilpon, R. P. Mikkilineni, D. B. Roe, and S. Gokcen. Speech Recognition: From the Laboratory to the Real World. *AT&T Technical Journal*, pages 14–24, Setembro/Outubro 1995.

[3] Tucker P. and D. M. Jones. Voice as Interface: An Overview. *International Journal of Human-Computer Interaction*, pages 145–170, 1991.

[4] R. Reddy. Speech Recognition by Machine: A Review. *Proceedings of the IEEE*, 64(4):501–521, Abril 1976.

[5] S. Furui. *Digital Speech Processing, Synthesis and Recognition.* Marcel Dekker, Inc., 1989.

[6] H. Davis, R. Biddulph, and S. Balashek. Automatic Recognition of Spoken Digits. *The Journal of the Acoustical Society of America*, 24(6):637–642, 1952.

[7] S. Furui. Speech Recognition - Past, Present and Future. *NTT Review*, 7(2):13–18, Março 1995.

[8] L. R. Rabiner and B.-H. Juang. *Fundamentals of Speech Recongnition.* Englewood Cliffs, Prentice-Hall, 1993.

[9] G. F. Kubala, Y. Chow, A. Derr, M. Feng, O. Kimball, J. Makhoul, J. Rohlicek, S. Roucos, R. Schwartz, and J. Vandergrift. Continuous Speech Recognition Results of the Byblos System on the DARPA 1000-word Resource Management Database. *IEEE International Conference on Acoustics, Speech and Signal Processing*, Abril 1988.

[10] K. F. Lee. *Automatic Speech Recognition - The Development of the Sphinx System.* Kluwer Academic Publishers, 1989.

[11] L. R. Rabiner. Applications of Voice Processing to Telecommunications. *Proceedings of the IEEE*, 82(2):199–228, Fevereiro 1994.

[12] B. Wheatley and J. Picone. Voice across America: Toward Robust Speaker-Independent Speech Recognition for Telecommunications Applications. *Digital Signal Processing*, pages 45–63, 1991.

[13] J. K. Baker. The Dragon System - an Overview. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-23:24–29, Fevereiro 1975.

[14] V. R. Lesser, R. D. Tennell, L. D. Erman, and R. D. Reddy. The Hearsay II Speech Understanding System. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-23:11–24, Fevereiro 1975.

[15] B. T. Lowerre. Dynamic Speaker Adaptation in the Harpy Speech Recognition System. *IEEE International Conference on Acoustics, Speech and Signal Processing*, Abril 1977.

[16] J. G. Wilpon, L. R. Rabiner, and A. Bergh. Speaker Independent Isolated Word Recognition Using a 129-word Airline Vocabulary. *The Journal of the Acoustical Society of America*, Agosto 1982.

[17] M. Blomberg, K. Elenius, and F. Lundin. Voice Controlled Dialling in an Intercom system. *Proceedings 10th International Conference of Human Factors in Telecommunications*, pages 233–238, 1983.

[18] R. A. Cole, R. M. Stern, M. S. Phillips, S. M. Brill, P. Specker, and A. P. Pilant. Feature Based speaker Independent Recognition of English Letter. *IEEE International Conference on Acoustics, Speech and Signal Processing*, Outubro 1983.

[19] IBM Speech Recognition Group. A Real-time Isolated-Word Speech Recognition System for Dictation Transcription. *IEEE International Conference on Acoustics, Speech and Signal Processing*, Março 1985.

[20] L. R. Rabiner, J. G. Wilpon, and F. K. Soong. High Performance Connected Digit Recognition using Hidden Markov Models. *IEEE International Conference on Acoustics, Speech and Signal Processing*, Abril 1988.

[21] K. Schuhmacher, B. Lochschmidt, and B. Kaspar. Erfahrungen mit Einem auf Einzelworterkennug Basierenden Sprachdialogsystem. *Internationaler Workshop im Rahmen der BIGTECH'88*, pages 242–249, 1988.

[22] Relatórios do COST-232 - Speech Recognition over the Telephone Line, 1992. European Co-operative for Scientific and Technical Research.

[23] M. J. Poza, C. de la Torre, D. Tapias, and L. Villarubia. An Approach to Automatic Recognition of keywords in Unconstraint Speech using pPrametric Models. *Eurospeech 91*, 1991.

[24] M. J. Poza, J. F. Mateos, and J. A. Siles. Audiotex with Speech Recognition and Text to speech Conversion for the Spanish Telephone Network. *Voice systems Wordwide 91*, 1991.

[25] C. Gagnoulet, D. Jouvet, and J. Damay. Mairievox: A Voice-activated Information System. *Speech Communication*, (10):23–31, 1988.

[26] Speech Recognition Update. (35):7, 1991

[27] A. Nejat Ince, editor. *Digital Speech Processing, Speech Coding, Synthesis and Recognition*. Kluwer Academic Publishers, 1992.

[28] D. P. Morgan and C. L. Scofield. *Neural Networks and Speech Processing*. Kluwer Academic Publishers, 1991.

[29] L. R. Rabiner, S. E. Levinson, and M. M. Sondhi. An Introduction to the Applications of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recogniton. *The Bell System Technical Journal*, 62(4):1035–1105, Abril 1983.

[30] L. R. Rabiner and B. H. Juang. An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*, pages 4–16, Janeiro 1986.

[31] A. B. Poritz. Hidden Markov Models: a Guided Tour. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 7–13, 1988.

[32] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):501–521, Fevereiro 1989.

[33] L. A. Liporace. Maximum Likelihood Estimation for Multivariate Observations of Markov Sources. *IEEE Transactions on Information Theory*, IT-28(5):729–754, Setembro 1982.

[34] B. H. Juang, S. E. Levinson, and M. M. Sondhi. Maximum Likelihood Estimation for Multivariate Observations of Markov Chain. *Transactions on Information Theory*, IT-32(2):307–309, Março 1986.

[35] B. H. Juang. Maximum Likelihood Estimation for Mixtures Multivariate Stochastic Observations of Markov Chains. *AT&T Technical Journal*, 64(6):1263–1249, Julho/Agosto 1985.

[36] I. Miller and J. E. Freund. *Probability and Statistics for Engineers*. Englewood Cliffs, 1985.

[37] X. D. Huang, Y. Ariki, and M. A. Jack. *Hidden Markov Models for Speech Recognition*. Edinburgh University Press, 1991.

[38] X. D. Huang and M. A. Jack. Hidden Markov Modelling of Speech based on a Semicontinuous Model. *Eletronics Letters*, 24(1):6–7, Janeiro 1988.

[39] X. D. Huang and M. A. Jack. Performance Comparison between Semicontinuous and discrete Hidden Markov Models of Speech. *Eletronics Letters*, 24(3):149–151, Fevereiro 1988.

[40] D. Burshtein. Robust Parametric Modelling of Durations in Hidden Markov Models. *IEEE Transactions on Speech and Audio Processing*, 4(3):240–242, Maio 1996.

[41] L. R. Rabiner, B. H. Juang, S. E. Levinson, and M. M. Sondhi. Some Properties of Continuous Hidden Markov Model Representation. *AT&T Technical Journal*, 64(6):1251–1269, Agosto 1985.

[42] L. R. Rabiner and B.-H. Juang. Hidden Markov Models for Speech Recognition. *Technometrics*, 33(3):251–272, Agosto 1991.

[43] J. G. Wilpon, Lee C.-H., and L. R. Rabiner. Improvements in Connected Digit Recognition Using Higher Order Spectral and Energy Features. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 349–352, 1991.

[44] L. R. Rabiner and S. E. Levinson. Isolated and Connected Word Recognition - Theory and Selected Applications. *IEEE Transactions on Communications*, COM-29(5):621–659, Maio 1981.

[45] T. Matsuoka and Y. Minami. Acoustic and Language Processing Technology for speech Recognition. *NTT Review*, 7(2):30–39, Março 1995.

[46] J. Picone. Continuous Speech Recognition using Hidden Markov Models. *IEEE ASSP Magazine*, pages 26–41, Julho 1990.

[47] K. F. Lee, H. W. Hon, M. Y. Hwang, and X. Huang. Speech Recognition using Hidden Markov Models: a CMU Perspective. *Speech Communication*, pages 497–508, 1990.

[48] W. S. McCulloch and W. Pitts. A Logical Calculus of the Ideas Immanent in Neural Nets. *Bulletin of Math. Biophys.*, EC-12:115–127, Maio 1943.

[49] F. Rosenblatt. Two theorems of Statistical Separability in the Perceptron - Mechanization of Thought Processes. *Proceedings of a Symposium held at the National Physical Laboratory*, 1:1421–1426, Novembro 1958.

[50] F. Rosenblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms.* Spartan Books, 1962.

[51] B. Widrow. Generalization and Information Storage in Networks of Adaline Neurons. *Self-Organizing Systems 1962*, pages 435–461, 1962.

[52] K. Steinbuch and V. A. W. Piske. Learning Matrices and their Applications. *IEEE Transactions Electron. Comput.*, pages 846–862, 1963.

[53] B. Widrow and M. Lehr. 30 years of Adaptive Neural Networks: Perceptron, Madaline and Back-Propagation. *Proceedings of the IEEE*, 78(9):1415–1442, Setembro 1990.

[54] R. P. Lippmann. Review of Neural Networks for Speech Recognition. *Neural Computation*, pages 1–38, 1989.

[55] J. C. Junqua and J. P. Haton. *Robustness in Automatic Speech Recogniton.* Kluwer Academic Publishers, 1996.

[56] B. R. Kämmerer and Küpper W. A. Experiments for Isolated-Word recognition with single and two Layer Perceptrons. *Neural Networks*, 3:693–706, 1990.

[57] G. Tatman and R. Jannarone. Alternative Neural Networks for Speech Applications. *The Twenty-Third Southeastern Symposium on System Theory*, pages 591–596, 1991.

[58] S. M. Peeling and R. K. Moore. Isolated Digit Recognition Experiments using the Multilayer Perceptron. *Speech Communications*, 7(4):403–409, Dezembro 1988.

[59] M. Zhu and K. Fellbaum. A Connectionist Model for Speader Independent Isolated Word Recognition. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 529–532, 1990.

[60] P. C. Woodland. Isolated Word Speech Recognition based on Connectionist Techniques. *British Telecom Technology Journal*, 8(2):61–66, 1990.

[61] N. Botros, Z. Deiri, and I. Wattar. Neural Networks: Algorithm and Instrumentation for Isolated Word Recognition. *Proceedings of the International Conference on Mathematical and Computer Modelling*, pages 279–283, 1990.

[62] C. J. Wellekens. Speech Recognition using Connectionist Methods. *Connectionism in Perspective*, pages 102–111, 1989.

[63] R. P. Lippmann. An Introduction to Computing with Neural Nets. *IEEE ASSP Magazine*, 4(2):4–22, Abril 1987.

[64] K. Knight. Connectionist Ideas and Algorithms. *Communications of the ACM*, 33(11):59–74, Novembro 1990.

[65] J. Tebelskis. *Speech Recognition using Neural Networks*. PhD thesis, School of Computer Science, Carnegie Mellon University, 1995.

[66] L. T. Niles and H. F. Silverman. Combining Hidden Markov Model and Neural Network Classifiers. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 417–420, 1990.

[67] H. Morgan, N.and Bourlard. Continuous Speech Recognition using Multilayer Perceptrons with Hidden Markov Models. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 413–416, 1990.

[68] H. Bourlard. How Connectionist Models could Improve Markov Models for Speech Recognition. *Advanced Neural Computers*, pages 247–254, 1990.

[69] Y. Bengio, R. Cardin, R. De Mori, and Y. Normandin. A Hybrid Coder for Hidden Markov Models using a Recurrent Neural Network. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing,*, pages 537–540, 1990.

[70] Y.Q. Gao, T. Y. Huang, and D. W. Chen. Hmm based Warping in Neural Networks. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 501–504, 1990.

[71] S. Katagiri and Lee C.-H. A new Hybrid Algorithm for Speech Recognition based on HMM Segmentation and Learning Vector Quantization. *IEEE Transactions on Speech and Audio Processing*, 1(4):421–430, Outubro 1993.

[72] H. Bourlard and J. Wellekens. Links between Markov Models and Multilayer Perceptrons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12):1167–1178, Dezembro 1990.

[73] N. Morgan and H. Bourlard. Continuous speech Recognition. *IEEE Signal Processing Magazine*, pages 25–42, Maio 1990.

[74] N. Morgan and H. Bourlard. Neural Networks for Statistical Recognition of Continuous Speech. *Proceedings of the IEEE*, 83(5):25–42, Maio 1995.

[75] L. R. Rabiner and M. R. Sambur. An Algorithm for Determining the Endpoints of Isolated Utterances. *The Bell System Technical Journal*, pages 297–315, Fevereiro 1975.

[76] L. F. Lamel, L. Rabiner, and J. G. Rosemberg, A. E.and Wilpon. An Improved Endpoint Detector for Isolated Word Recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-29(4):777–785, Agosto 1981.

[77] J. G. Wilpon, L. R. Rabiner, and T. Martin. An Improved Word Detection Algorithm for Telephone Quality Speech incorporating both Syntactic and Semantic Constraints. *AT&T Technical Journal*, 63(3):479–497, Março 1984.

[78] J. Linde, A. Buzo, and R. M. Gray. An Algorithm for Vector Quantizer. *IEEE Transactions on Communications*, Com. 28(1):84–94, Janeiro 1980.

[79] R. Viswanathan and J. Makhoul. Quantization Properties of Transmission Parameters in Linear Predictive Systems. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-23(3):309–321, Junho 1975.

[80] J. Makhoul, S. Roucos, and H. Gish. Vector Quantization in Speech Coding. *Proceedings of the IEEE*, 73(11):1551–1588, Novembro 1985.

[81] R. M. Gray. Vector Quantization. *IEEE ASSP Magazine*, pages 4–29, Abril 1984.

[82] A. Buzo, A. H. Gray Jr., R. M. Gray, and J. D. Markel. Speech Coding upon Vector Quantization. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-28(5):562–574, Outubro 1980.

[83] J. A. Martins. Vocoder LPC com Quantização Vetorial. Master's thesis, FEEC - Universidade de Campinas, 1991.

[84] D. O' Shaughnessy. *Speech Communication: Human and Machine.* Addison-Wesley Publishing Company, 1987.

[85] J. W. Picone. Signal Modeling Techniques in Speech Recognition. *Proceedings of the IEEE*, 81(9):1215–1247, Setembro 1993.

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

[87] J. D. Markel and A. H. Gray Jr. *Linear Prediction of Speech.* Springer-Verlag, 1982.

[88] J. Makhoul. Linear Prediction: A Tutorial Rewiew. *Proceedings of IEEE*, 63(4):561–580, Abril 1975.

[89] S. B. Davis and P. Mermelstein. Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-28(4):357–366, Agosto 1980.

[90] B. A. Carlson and M. A. Clements. A Weighted Projection Measure for Robust Speech Recognition. *IEEE Proceedings Southeastcom*, pages 94–98, 1990.

[91] H. Hermansky. Perceptual Linear Predictive (PLP) Analysis of Speech. *Journal Acoustical Society of America*, 87(4):1738–1752, Abril 1990.

[92] H. Hermansky and N. Morgan. RASTA Processing of Speech. *IEEE Transactions on Speech and Audio Processing*, 2(4):578–589,, Outubro 1994.

[93] F. Mihelic, L. Gyergyek, and N. Pavesic. Comparison of Features and Classification Rules for Acoustic-phonetic Transcription of Slovene Speech. *Digital Signal Processing*, pages 1453–1457, 1991.

[94] H. Hermansky. Exploring Temporal Domain for Robustness in Speech Recognition.

[95] H. Hermansky. Speech beyond 100 milliseconds (Temporal Filtering in Feature Domain).

[96] L. R. Bahl, P. F. Brown, P. V. Souza and R. L. Mercer. Speech Recognition with Continuous Parameter Hidden Markov Models. *Computer Science*, 1987.

[97] R. Pieraccini. Speaker Independent Recognition of Italian Telephone Speech with Mixture Density Hidden Markov Models. *Speech Communication*, pages: 105–115, 1991.

[98] A.Waibel, T. Hanazawa, G. Hinton, K. Shikano and K. Lang. Phoneme Recognition Using Time-Delay Neural Networks. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(3): 328-338, Março 1989.

[99] K. M. Ponting and S. M. Peeling. The Use of Variable Frame Rate Analysis in Speech Recognition. *Computer Speech and Language*, pages: 169–179, 1991.

[100] S. Nakagawa and Y. Hirata. Comparison Among Time-Delay Neural Networks, LQVQ2, Discrete Parameter HMM and Continuous Parameter HMM. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages: 509–512, 1990.

Índice Remissivo