UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA

GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO

SISTEMAS DE RECONHECIMENTO DE LOCUTOR INDEPENDENTE DE TEXTO

Hector Natan Batista Pinheiro

Trabalho de Graduação

Recife
JANEIRO DE 2013

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA

Hector Natan Batista Pinheiro

SISTEMAS DE RECONHECIMENTO DE LOCUTOR INDEPENDENTE DE TEXTO

Trabalho apresentado ao Programa de GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO do CENTRO DE INFORMÁTICA da UNIVERSIDADE FEDERAL DE PERNAMBUCO como requisito parcial para obtenção do grau de Bacharel em ENGENHARIA DA COMPUTAÇÃO.

Orientador: *Tsang Ing Ren*

Recife
JANEIRO DE 2013

É uma honra dedicar esse trabalho aos meus pais,
Manoela e Sérgio.

Agradecimentos

Minha eterna gratidão

Aos meus pais, Manoela e Sergio, por tudo

Aos mestres Tsang, George e Carlos, pelos ensinamentos, técnicos ou não

Aos amigos de curso Julio e Luís, por tudo que fizemos na graduação

Aos amigos de trabalho Luís e Ivson, pela excelente experiência em fazer pesquisa

Aos amigos, pelo apoio e compreensão

Aos colegas, pela companhia

À música, pelo prazer

E a Deus, por ter me concedido isso tudo.

Resumo

Tradicionalmente, as estratégias de verificação e identificação de um indivíduo são baseadas em algum conhecimento prévio específico, isto é, uma senha ou um número de identificação pessoal. Há também casos em que são utilizados objetos físicos, como um cartão ou uma chave. O maior problema destas estratégias é que senhas podem ser esquecidas, cartões podem ser perdidos e ambos podem ser roubados por um intruso, especialmente em um mundo onde a economia global faz com que tenhamos cada vez mais ambientes e informações compartilhadas.

Nos últimos anos observou-se um crescente avanço tecnológico em sistemas computadorizados e, principalmente, na utilização de dispositivos móveis, como celulares e tablets. Tal fator vem alterando as formas de comunicação e transmissão de informações. Transações bancárias, por exemplo, são realizadas através de ligações telefônicas e a necessidade de garantir a segurança deste tipo de operação é crescente.

Com o objetivo de resolver estes problemas e garantir uma maior segurança em sistemas computadorizados, métodos que utilizam autenticações biométricas vêm sendo estudados e desenvolvidos. Biometria é a ciência que estuda métricas e técnicas para identificar uma pessoa através de características fisiológicas e comportamentais. As formas de biometria mais conhecidas incluem o reconhecimento automático de um indivíduo através de impressões digitais, face, íris, retina, assinatura e voz.

Um Sistema de Verificação de Locutor é um sistema biométrico cujo objetivo é validar a identidade de um indivíduo através de sua voz. Uma vantagem desta biometria é que o sinal de voz pode ser capturado, com qualidade, por praticamente qualquer aparelho eletrônico que contenha um microfone, como um dispositivo móvel.

Neste sentido, este trabalho apresenta o estado da arte de sistemas de verificação de locutor quando não há restrições quanto às locuções utilizadas pelo sistema. Esse tipo de sistema é geralmente referenciado como sistemas independentes de texto. Dada a arquitetura básica de um sistema desse tipo, o trabalho apresenta as principais técnicas utilizadas em cada um dos módulos da arquitetura, que vai desde o pré-processamento do sinal e extração das características até a modelagem dos modelos dos locutores e os testes de locuções utilizando esses modelos.

As técnicas de pré-processamento incluem a utilização de filtros para redução de ruído, além de estratégias de normalização de canal como a Subtração da Média Cepstral (*Cepstral Mean Subtraction - CMS*) e a Filtragem RASTA. As características úteis para a análise da voz correspondem às chamadas características espectrais, no espaço das frequências. Atualmente, vem sendo observado a utilização dos Coeficientes Cepstrais na escala Mel (*Mel-Frequency Cepstral Coefficients - MFCC*) em sistemas que lidam com informações presentes nos sinais

de voz. Entre as técnicas de modelagem de locutores, temos uma abrangente utilização de Modelos de Misturas Gaussianas (*Gaussian Mixture Models - GMM*). Atualmente, as principais técnicas incluem a utilização de um modelo específico para a modelagem dos impostores. Damos um foco especial à utilização dos chamados Modelos Universais de Fundo (*Universal Background Model - UBM*). Outraos modelos mais robustos são estudados como GMMs Adaptativas e Supervetores de GMMs combinados com Máquinas de Vetores Suporte (*Support Vector Machine - SVMs*. Finalmente, estudamos a forma como os *scores* produzidos pelos modelos são utilizados para realizar a decisão final. O mais conhecido método é conhecido como Teste da Razão das Verossimilhanças.

Palavras-chave: Reconhecimento de locutor, Sistemas de verificação de locutor, Biometria de voz, Processamento de sinais, Subtração de Média Cepstral, Filtragem RASTA, Coeficientes Cepstrais na escala Mel, Modelos de misturas Gaussianas, Modelos de Misturas Gaussianas Adaptativas, Supervetores de GMMs, Máquinas de vetores suporte, Teste da Razão das Verossimilhança.

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1

Introdução

Com a massiva presença de dispositivos móveis capazes de acessar a Internet, a segurança de sistemas informatizados é prioritária para garantir com que dados de usuários e acessos a determinados ambientes sejam protegidos e acessíveis apenas por pessoas autorizadas. Compras com cartão de crédito, acesso restrito a áreas e recursos, e transações bancárias são apenas alguns exemplos de operações onde se faz necessário verificar a identidade do indivíduo.

Tradicionalmente, as estratégias de verificação de um indivíduo são baseadas em algum dado específico, como uma senha ou um número de identificação pessoal. Há também casos em que são utilizados objetos físicos, como um cartão ou uma chave. O maior problema destas estratégias é que senhas podem ser esquecidas, cartões podem ser perdidos e ambos podem ser roubados ou falsificados. De fato, os meios tradicionais de verificação de indivíduos podem se apresentar inseguros de diversas formas, especialmente em um mundo onde a economia global faz com que tenhamos cada vez mais informações compartilhadas.

Com o objetivo de resolver estes problemas e garantir uma maior segurança, métodos que utilizam autenticações biométricas vêm sendo desenvolvidos nas últimas décadas. Biometria é a ciência que estuda as métricas e técnicas para identificar uma pessoa através de características fisiológicas e comportamentais. Em função de que muitas destas características são únicas para cada pessoa, a identificação biométrica é mais segura e robusta quando comparada com os métodos tradicionais [FR04].

Uma biometria ideal deve ser capaz de fornecer a identificação de uma pessoa de forma exata e barata. Dessa maneira, podemos dizer que uma biometria ideal deve ser:

- Universal, de modo que cada pessoa possua uma característica biométrica única;
- Permanente, de modo que não ocorra mudanças na característica, com o passar do tempo;
- Coletável, de modo que ela possa ser facilmente capturada e representada pelo sistema.

As formas de biometria mais conhecidas incluem o reconhecimento automático de um indivíduo através de sua impressão digital, face, íris, retina, assinatura ou voz.

1.1 Biometria de voz

Tratando-se de sinais de voz, o processo cognitivo da audição humana é dividido em três etapas:

- Captação do sinal de voz;
- Verificação das características presentes no sinal;

- Decodificiação da mensagem e compreensão da informação recebida.

Além de entender a informação contida no sinal de voz, a estrutura cognitiva também é capaz de reconhecer quem está emitindo o sinal, ou seja, é capaz de identificar o locutor da mensagem com base nas características acústicas do sinal [RJ93].

Baseando-se nesta capacidade, pode-se propor um sistema computacional que seja capaz de fazer o reconhecimento de um locutor, tendo o sinal de voz como elemento a ser processado. O sistema capta este sinal, extrai as características sonoras e, por meio de reconhecimento de padrões e modelagem de sinais, faz o reconhecimento do locutor.

O reconhecimento de um locutor realizado por um sistema computacional, é definido como o processo de reconhecer automaticamente um indivíduo utilizando apenas informações presentes na voz do mesmo. Cada pessoa possui características únicas em sua voz. Isso acontece em função de diferenças na forma do trato vocal, tamanho da laringe e outras partes dos órgãos responsáveis pela geração da voz. Além destas diferenças físicas existem diferenças comportamentais que dizem respeito à maneira com que uma pessoa fala, como uma forma particular de acentuação, ritmo, tipo de entonação, vocabulário, entre outros.

Dependendo da aplicação, a área de reconhecimento de locutor pode ser dividida em duas tarefas básicas: verificação e identificação. Em identificação de locutores, o sistema tenta identificar, dado um sinal de voz, qual dos locutores, presentes em um conjunto fechado de locutores registrados, produziu aquela locução. Esse tipo de tarefa também é chamada de identificação de locutores em **conjunto fechado**. Já em verificação de locutores, o objetivo é verificar se uma dada locução foi produzida por um locutor específico, e não por um outro locutor qualquer, chamado de impostor. Uma vez que o conjunto de possíveis impostores não pode ser mensurado e é totalmente desconhecido do sistema, essa tarefa também é conhecida como um problema de **conjunto aberto**. Uma tarefa intermediária é conhecida como **identificação de locutor em conjunto aberto**, que é basicamente um sistema de identificação de locutores, adicionando uma classe chamada de "Nenhum das anteriores", que é utilizada para classificar os locutores desconhecidos do sistema, os impostores.

Além disso, restrições podem ser feitas às locuções utilizadas para o reconhecimento. Quando a locução de treino e a locução de teste devem possuir a mesma palavra ou frase, dizemos que o sistema é **dependente de texto**. Nesse tipo de sistema, o locutor, para ser autenticado, deve dizer uma palavra específica fornecida pelo sistema. Quando não há nenhuma restrição quanto às locuções e o locutor é autenticado ao falar qualquer palavra ou frase, dizemos que o sistema é **independente de texto**.

Uma importante vantagem da biometria de voz é que o sinal de voz pode ser capturado, com qualidade, por praticamente qualquer aparelho eletrônico que contenha um microfone. Com isso, assume-se que atualmente esse processo pode ser realizado em qualquer lugar com um dispositivo móvel como um celular ou tablet, por exemplo. Esta característica faz com que o sistema precise levar em conta ambientes sem controle e desconhecidos que, por consequência, ocasionam em variações nas gravações. Dispositivos móveis de gravação podem ser utilizados em praticamente qualquer ambiente, como escritórios silenciosos, uma cafeteria ou na esquina de uma rua movimentada. Em cada ambiente, variações nas condições acústicas de gravação, além da presença de ruídos de fundo, podem corromper o sinal de voz. Tal fato pode levar a uma variação intra-locutor que venha a reduzir o desempenho do sistema. Além disso, diferentes

tipos de dispositivos usam diferentes microfones, o que também pode ocasionar uma redução no desempenho do sistema [WPH06a].

Dentre as aplicações para sistemas de reconhecimento de locutor destacam-se a autorização de entrada em ambientes, a abertura automática de automóveis, confirmações de identidades em transações telefônicas, entre outras. Um aspecto especial tem sido levado em conta quando os sistemas operam sobre telefones, como em um call center ou quando da realização de transações bancárias: o reconhecimento de locutor é utilizado como um nível a mais de segurança para garantir a identidade do indivíduo. Estes sistemas operam no transcorrer da conversa e de forma abstrata ao usuário. Outra importante aplicação é a área forense. Muitas informações são compartilhadas entre dois indivíduos via telefone, incluindo conversas entre criminosos. Nos últimos anos houve um aumento no interesse da área forense em integrar sistemas biométricos de voz para localizar criminosos. Tais sistemas vêm sendo utilizados como provas em tribunais [Dry07, RRG$^+$03, TAE08].

Para um futuro próximo, projeta-se que alguns serviços de call center sejam trocados por sistemas totalmente automatizados [KZZW07]. O reconhecimento de locutor terá o papel de verificar e confirmar a identidade do usuário. O reconhecimento de fala e o reconhecimento de língua também terão papel fundamental neste tipo de serviço. Um exemplo seria o *reset* de uma senha de forma automática através do telefone. Este tipo de aplicação evidencia que o foco principal de tecnologias para reconhecimento de locutor são aplicações baseadas em telefones e dispositivos móveis.

1.2 Objetivos do trabalho

Os objetivos deste trabalho são:

- Analisar e avaliar o estado da arte em sistemas de verificação de locutores em ambientes independentes de texto;
- Realizar experimentos e validação dos métodos presentes na literatura;
- Propor e analisar uma combinação das técnicas baseadas em GMMs e UBM, a fim de melhorar o desempenho dos sistemas.

1.3 Estrutura do documento

O Capítulo 2 contém um breve histórico e alguns conceitos básicos de sistemas de reconhecimento de locutor. Além disso, nesse capítulo é descrita a arquitetura básica de um sistema de verificação independente de texto e como essa arquitetura é utilizada para realizar a aceitação de uma determinada locuções. Do Capítulo 3 ao Capítulo 5, uma revisão da literatura é feito contemplando as principais técnicas para cada um dos módulos da arquitetura apresentada no Capítulo 2. O Capítulo 6 apresenta os experimentos realizados de modo a comparar as técnicas abordadas. As conclusões deste trabalho são então descritas no Capítulo 7. Além disso, esse

trabalho conta com um Apêndice que possui informações relevantes sobre alguns conceitos matemáticos utilizados pelas técnicas.

CAPÍTULO 2

Sistemas de reconhecimento de locutor

2.1 Histórico

Pesquisas em reconhecimento de pessoas através da voz vêm sendo realizadas já há muito tempo. Um dos primeiros estudos realizados nessa área retomam a década de 1930, onde um estudo realizado por Dr. Francis McGehee, professor de psicologia da Universidade Johns Hopkins (EUA) buscou explorar o quão bem uma pessoa pode identificar pessoas apenas escutando suas vozes [Hol01].

Durante a Segunda Guerra Mundial houve um interesse maior em identificar pessoas pela voz. Com esse intuito, o Bells Labs inventou, em 1941, uma máquina capaz de fazer o espectrograma da voz e esperava-se que esse espectrograma ajudasse a identificar vozes de alemães interceptadas por rádio. Os resultados adquiridos naquela época para identificação de locutores com espectrogramas não foram satisfatórios e os estudos sobre esse tema foram temporariamente abandonados.

Em 1962, Lawrence Kersta, um dos inventores do espectrógrafo de som, publicou na revista Nature o trabalho "Voiceprint Identification" [Ker62]. Nesse trabalho ele defendia a infalibilidade de seu método e relatava taxas de identificação correta de 99%. Contudo, seus métodos e resultados foram considerados controversos e sua aceitação na comunidade científica foi restrita. Apesar das controvérsias, seus estudos são considerados extremamente importantes para o inicio de pesquisas avançadas em reconhecimento de locutor.

Até o início da década de 1960 o processo de reconhecimento de locutor era uma tarefa executada manualmente, através da comparação visual de espectrogramas por um especialista treinado. A primeira técnica de reconhecimento automático de locutor surgiu em 1963, com os estudos de Pruzansky no Bell Labs [Fur04]. Nesse sistema foram utilizados bancos de filtros e dois espectrogramas digitais para medir a similaridade dos sinais de voz. Ainda durante o final da década de 1960 e toda a década de 1970 surgiram vários outros sistemas baseados em analisar a evolução temporal de certos parâmetros da voz, especialmente da frequência fundamental, formantes, intensidade, e coeficientes do preditor linear. Vale salientar que esses sistemas eram dependentes do texto, onde o reconhecimento era feito com base na pronúncia de um texto pré-definido. Uma abordagem também apresentada nessa época foi utilizar a análise das médias dos parâmetros de trechos longos de voz, como em Furui *et al.* em 1972 [FIS72], Furui em 1974 [Fur74] e Markel e Davis em 1979 [MD79], sendo os primeiros métodos independentes de texto.

Nessa época, deve-se enfatizar os trabalhos de Atal em 1974 [Ata74] que demonstraram a superioridade da representação cepstral (cepstrum) frente a diversos outros tipos de parâmetros.

Com o aumento do poder computacional, na década de 1980, as técnicas de reconheci-

mento de locutor ficaram progressivamente mais complexas, proporcionando melhorias de desempenho dos sistemas. Em 1988 Soong *et al.* [SR88], após verificar o sucesso da utilização de técnicas de Quantização Vetorial (*Vector Quantization - VQ*) em sistemas de reconhecimento de padrões, propuseram um sistema baseado nesta técnica para reconhecimento de locutor. Ainda que os experimentos com VQ demonstrassem bons resultados, em geral, havia uma restrição quanto ao tamanho do vocabulário, devido à própria característica da modelagem. Visando suprir este fato surgiram as modelagens probabilísticas.

A partir da década de 1990, houve uma grande popularização dos sistemas baseados em modelagens probabilísticas. Entre eles se destacam os modelos ocultos de Markov (*Hidden Markov Models - HMM*) e os modelos de misturas Gussianas (*Gaussian Mixture Models - GMM*). O HMM é um modelo estatístico baseado em cadeias de Markov que incorporam informações sobre a evolução temporal dos parâmetros, sendo bastante utilizado tanto para o reconhecimento de voz como para o reconhecimento de locutor com dependência de texto. Apesar do HMM obter bons resultados para o reconhecimento com dependência de texto, a informação temporal incorporada nesses modelos não mostrou vantagem nos sistemas independentes de texto. GMM também é um modelo estatístico, mas diferente do HMM, ele não leva em consideração a relação temporal. Assim sendo, o GMM é amplamente utilizado em sistemas de reconhecimento automático de locutor independentes de texto. A utilização do GMM em sistemas de reconhecimento automático de locutor foi introduzida por Reynolds, em 1995, em sua tese de doutorado [Rey95].

2.2 Conceitos básicos

O processo de reconhecimento de um locutor pode ser definido como uma tarefa de classificação onde o objetivo é determinar, dada uma locução (sinal de voz) e um locutor, se a locução foi ou não gerada pelo locutor.

2.2.1 Locução

Uma locução é um determinado trecho de fala produzido por um usuário (locutor). Ela é capturada pelo sistema através de um microfone e o sinal capturado é então transformado em um sinal digital. Após a sua captura e armazenamento, este sinal é processado com o objetivo de extrair informações úteis ao sistema de verificação.

Para o processo de reconhecimento, as características do aparelho fonador que são únicas para cada indivíduo precisam ser extraídas e convenientemente categorizadas. A extração desta informação na maioria dos casos é realizada sobre o cepstrum da componente útil do sinal de voz, conforme será apresentado no Capítulo 4. Os trechos do sinal em que há período de silêncio ou apenas ruído de ambiente devem ser descartados com o emprego de detectores de atividade de voz, conforme detalhado na Seção 3.1. Este processo é importante para o desempenho do sistema [Rey95]. Devido às propriedades estacionárias do sinal de voz para curtos intervalos de tempo, o mesmo deve ser tratado de forma segmentada, ou seja, o sinal deve ser dividido em segmentos sobrepostos com comprimento pré-estabelecido. Desta forma, garante-se que não haverá perda de características significantes ao processo de reconhecimento.

2.2.2 Identificação x Verificação

Dependendo da aplicação, a área de reconhecimento de locutores pode ser dividida em duas classes com tarefas distintas: Identificação e Verificação de locutores.

Na identificação, é apresentada uma determinada locução ao sistema e o mesmo deve determinar a qual locutor pertence esta locução. Nesse caso, todos os locutores foram devidamente registrados no sistema, que por sua vez possui a certeza que aquela locução pertence a algum deles. Uma vez que o sistema opera em um universo fechado de possibilidades, esse tipo de tarefa é comumente chamada de **identificação de locutor em conjunto fechado** (*closed-set speaker identification*).

Na verificação, o individuo afirma ser o locutor que produziu uma determinada locução e o sistema deve confirmar esta identidade. Diferente do processo de identificação, o sistema deve decidir se a locução foi produzida por aquele indivíduo específico, e por nenhum outro. Nesse caso, é impossível que o sistema modele todos os possíveis locutores. Dessa forma, o processo de verificação é um **problema de conjunto aberto** (*open-set problem*).

Existe ainda uma tarefa intermediária, que une as duas complexidades existentes nos processos de identificação e verificação. Quando o sistema realiza a identificação dos locutores, cada um dos locutores é representado por uma classe, e o sistema deve relacionar uma dada locução a uma classe específica. Ao se criar uma classe do tipo "Nenhum dos anteriores", o sistema cria uma classe que representada os locutores que não foram registrados, os impostores. Porém, como no processo de verificação, o sistema não é capaz de conhecer todos os possíveis impostores. Esse fato transforma o problema em um problema de conjunto aberto. Esse tipo de tarefa é chamada de **identificação de locutores em conjunto aberto** (*open-set speaker identification*).

Em termos de aplicações práticas, um sistema de verificação poderia ser usado para confirmar a identidade de um indivíduo durante uma transação bancária realizada por telefone, utilizando a própria voz como uma camada a mais de segurança. Para o caso de identificação, pode-se usar um sistema em que, a partir de uma locução emitida por um criminoso desconhecido, o sistema descubra a identidade deste criminoso.

Neste trabalho é abordara a problemática relacionada aos sistemas de verificação de locutor.

2.2.3 Dependência x Independência de texto

Outra característica dos sistemas de reconhecimento de locutor diz respeito às restrições impostas às locuções utilizadas. Nos sistemas dependentes de texto, o locutor é direcionado a gerar uma locução a partir de um texto pré-estabelecido. Isso faz com que as locuções de teste e as locuções de treino possuam a mesma palavra ou frase. Nos sistemas independentes de texto, nenhuma restrição é imposta à locução. Nesse caso, o sistema deve reconhecer o locutor a partir de qualquer locução produzida por ele.

Em geral, sistemas dependentes de texto tendem a apresentar uma performance melhor em virtude de que se apoiam inicialmente no reconhecimento do texto da locução para então, a partir da divergência entre a locução de teste um modelo selecionado, identificar o locutor [Cam97]. O mesmo não acontece em sistemas independentes de texto, já que não se tem conhecimento prévio sobre as características fonéticas da locução utilizada.

Neste trabalho é abordara a problemática relacionada aos sistemas de verificação de locutor independentes de texto.

2.3 Teste da razão de verossimilhança

Dado um trecho de fala X e um determinado locutor S, o processo de verificação consiste em determinar se X foi produzido por S e não por nenhum outro locutor. Assume-se, porém, que X possua a fala de um único locutor, caracterizando um problema de único locutor. Mais informações sobre problemas com múltiplos locutores pode ser visto em [DRQ00].

Basicamente, a tarefa de verificação pode ser resumida em um teste entre duas hipósteses:

- $H_0 = X$ **foi** produzida por S,
- $H_1 = X$ **não foi** produzida por S.

De acordo com a teoria de decisão estatística [RC08], o teste ótimo que decide entre essas duas hipóteses consiste no teste da razão de verossimilhança, definida por:

$$\frac{p(X|H_0)}{p(X|H_1)} = \begin{cases} \geq \theta, & \text{aceite } H_0, \\ < \theta, & \text{rejeite } H_0. \end{cases} \tag{2.1}$$

onde $p(X|H_i), i = 0, 1$, é a função de densidade de probabilidade para a hipótese H_i calculada para o trecho de fala observado, X. Essa função também é referenciada como a veressimilhança da hipótese H_i, dado o trecho de fala X. Dessa forma, a meta de um sistema de um sistema de verificação de locutor é determinar técnicas para computar a razão entre as verossimilhanças $p(X|H_0)$ e $p(X|H_1)$. Dependendo das técnicas utilizadas essas funções podem ser modelas e aplicadas de forma direta ou indireta no processo de decisão.

2.4 Arquitetura básica de um Sistema de Verificação de Locutores

A Figura 2.1 mostra a arquitetura básica de uma sistema de verificação de locutores independente de texto. Como mencionado na Seção 2.3, a decisão de aceitar ou rejeitar que uma determinada locução X foi produzida ou não por um locutor S se baseia em um teste entre as duas possíveis hipóteses, H_0 e H_1. Para que seja possível o cálculo das verossimilhanças de cada uma dessas hipóteses, dado X, modelos devem ser criados, de modo a estimar as funções de densidade de probabilidade. Os modelos correspondentes às hipóteses H_0 e H_1 são λ_{hip} e $\lambda_{\overline{hip}}$, respectivamente.

2.4.1 Treinamento dos modelos

Na fase de treinamento, locuções são utilizadas de modo que as funções de densidade de probabilidade sejam estimadas. Tais locuções são primeiramente pré-processadas, utilizando técnicas que visam melhorar a qualidade do sinal ou então acenturar informações que sejam úteis no processo de reconhecimento. Esse processo tem como objetivo melhorar o desempenho

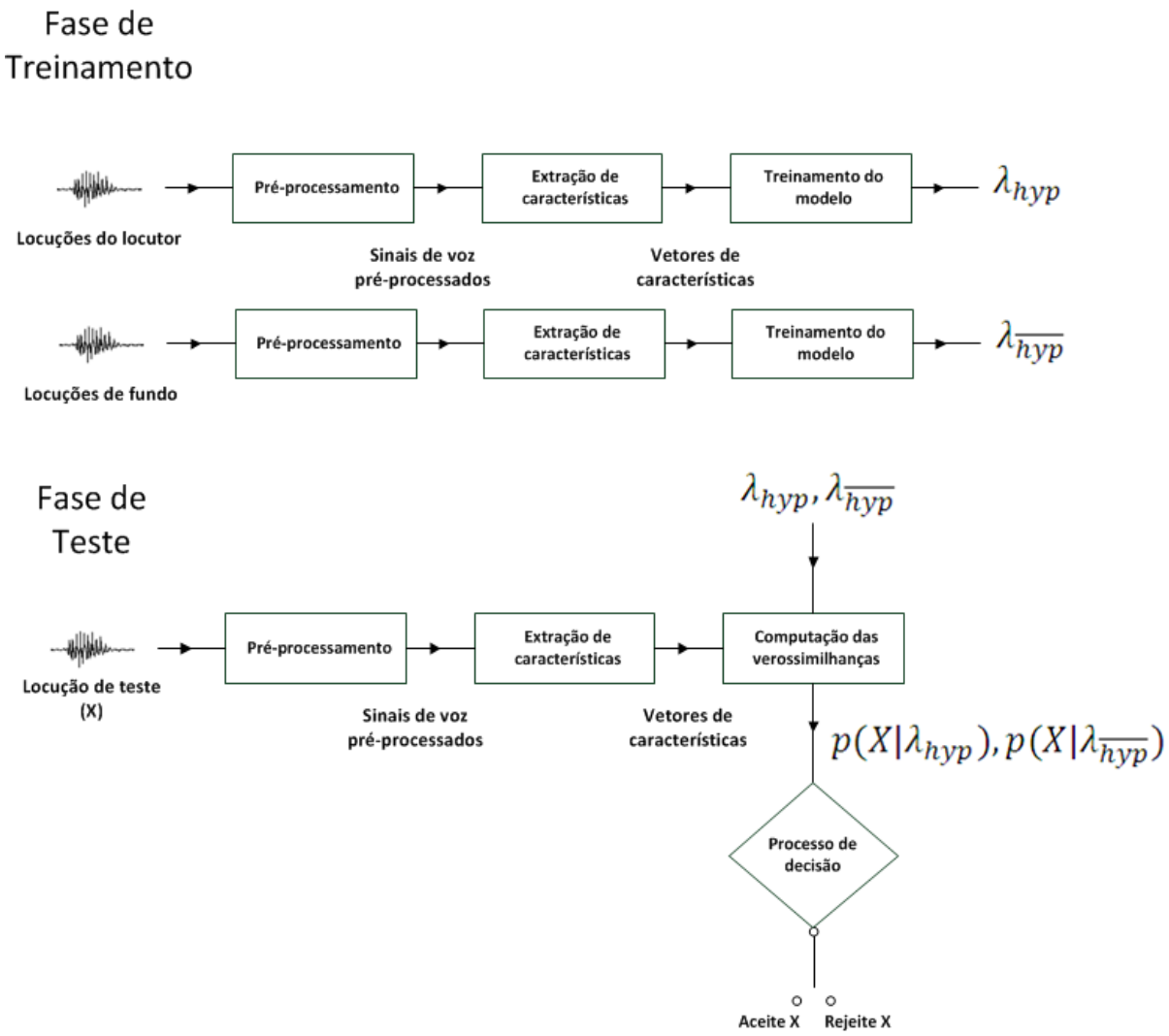

Figura 2.1 Arquitetura básica de um sistema de verificação de locutores independente de texto.

do processo de extração de características. As principais técnicas presentes no módulo de pré-processamente são filtros de eliminação de ruído, filtros para eliminação de distorções causadas pelo canal (microfone) e detectores de atividade de voz, que eliminam as partes do sinal que não possuem voz. Algumas dessas técnicas serão descritas no Capítulo 3. A partir dos sinais de voz pré-processados, o sistema extrai características desse sinal, produzindo vetores de números. Nese processo, o sinal é geralmente dividido em partes, chamados *frames*, que são analisadas, de modo que medidas específicas, as chamadas características (*features*), sejam extraídas delas. Mais detalhes sobre o processo de extração de características será visto no Capítulo 4. Utilizando os vetores de características extraídos das locuções de treinamento, os modelos são então treinados de modo que as suas respectivas funções de probabilidade sejam estimadas.

Como λ_{hip} deve modelar uma função de densidade de probabilidade correspondente ao fato de a locução ser produzida pelo locutor em questão, na fase de treinamento, as locuções desse

locutor específico devem ser utilizadas para estimar essa função.

Por outro lado, $\lambda_{\overline{hip}}$, corresponde a uma função de probabilidade que modela a hipótese que a locução não foi gerada por um determinado locutor. Nesse caso, essa função é estimada utilizando locuções que não pertecem ao locutor em questão. Tais locuções são comumente chamadas de locuções de fundo (*background speeches*). $\lambda_{\overline{hip}}$ também é referenciada como modelo de fundo (*background model*).

Enquanto que λ_{hip} é bem definido e é capaz de estimar bem as locuções de um determinado locutor S, $\lambda_{\overline{hip}}$ não é muito bem definido, uma vez que teoricamente ele deve estimar todo o espaço dos outros locutores que não são S. Diante dessa dificuldade de se estimar $\lambda_{\overline{hip}}$, duas são as principais abordagens que vêm sendo utilizadas.

A primeira abordagem consiste em utilizar um conjunto de modelos de outros locutores. Em diversos contextos, esse conjunto tem sido chamado de **conjuntos de razão de verossimilhança** (*likelihood ratio sets*) [HBP91], ***cohorts*** [RDL$^+$92] ou então de **locutores de fundo** (*background speakers*) [Rey95].

Dado um conjunto de N modelos de locutores de fundo, $\{\lambda_1, \lambda_2, ..., \lambda_N\}$, e um vetor x, a verossimilhança de $\lambda_{\overline{hip}}$, dado x, é definido como:

$$p(\mathbf{x}|\lambda_{\overline{hip}}) = F[p(\mathbf{x}|\lambda_1), p(\mathbf{x}|\lambda_2), ..., p(\mathbf{x}|\lambda_N)], \tag{2.2}$$

onde F é uma função, como média ou máximo, dos valores das verossimilhanças dos modelos de locutores de fundo.

A seleção, o tamanho ou a combinação dos locutores de fundo ainda é assunto de pesquisa. Mas no geral, constatou-se que para obter o melhor desempenho dessa abordagem, os modelos de background devem ser estimados utilizando locuções apenas de outros locutores e não do locutor em questão. Esse fato é desvantajoso em sistemas onde há um grande número de locutores, uma vez que cada um deles deverá possuir o seu próprio modelo de background.

A segunda abordagem mais utilizada consiste em criar apenas um modelo de fundo, que é estimado utilizando trechos de fala de vários locutores diferentes. Esse único modelo é geralmente chamado de **modelo de fundo universal** (*Universal Background Model - UBM*). Pesquisas nesse contexto estudam a seleção e composição dos locutores e dos trechos de fala para estimar o modelo. A grande vantagem dessa abordagem é que apenas um modelo de fundo é estimado e utilizado na fase de teste para todos os locutores.

É possível, ainda, a combinação de vários modelos de fundo para estimar $\lambda_{\overline{hip}}$, porém, a utilização de um único modelo de fundo (referenciado a partir de agora como UBM) é a abordagem que predomina em sistemas de verificação de locutores independentes de texto [RC08].

2.4.2 Fase de teste

Como mencionado na Seção 2.3, o processo de decisão é baseado em um *Score*, calculado a partir da razão das verossimilhanças dos modelos das hipóteses H_0 e H_1 (λ_{hip} e $\lambda_{\overline{hip}}$), como é mostrado na Equação 2.1. Portanto, dado uma locução de teste X, deve-se calcular o valor das verossimilhanças, $p(X|\lambda_{hip})$ e $p(X|\lambda_{\overline{hip}})$.

Um detalhe importante é que o processo de extração de características de X produz um conjunto de vetores $\{x_1, x_2, ..., x_T\}$, uma vez que primeiramente esse sinal é dividido em *frames*.

Então, define-se $p(X|\lambda)$ como o produto das verossimilhanças dos vetores:

$$p(X|\lambda) = \prod_{i=1}^{T} p(x_i|\lambda). \tag{2.3}$$

Além disso, é bastante comum a utilização da verossimilhança normalizada na escala logarítmica:

$$log[p(X|\lambda)] = \frac{1}{T} \sum_{i=1}^{T} log[p(x_i|\lambda)]. \tag{2.4}$$

Dessa maneira a razão das verossimilhanças se torna uma subtração:

$$Score(X) = log[p(X|\lambda_{hip})] - log[p(X|\lambda_{\overline{hip}})]. \tag{2.5}$$

Dependendo da técnica, o *Score* pode ser utilizado diretamente ou indiretamente no processo de decisão de aceitar ou rejeitar X. A abordagem mais simples consiste em aceitar se o valor ultrapassar um determinado limiar. Outras abordagens mais elaboradas presentes na literatura serão vistas mais adiante.

CAPÍTULO 3

Pré-processamento

A primeira etapa do sistema que lida com as locuções consiste em um pré-processamento dos sinais. Nesta etapa são aplicadas técnicas e algoritmos que visam melhorar a qualidade do sinal ou acentuar determinadas características importantes para o processo final de reconhecimento. O principal objetivo com isso é otimizar o processo posterior de extração de características e, como consequência, obter um desempenho superior no sistema como um todo. Nesta etapa podem-se destacar filtros de eliminação de ruído, filtros para eliminação de distorções causadas pelo canal (microfone) e detectores de atividade de voz, que eliminam as partes do sinal que não possuem voz. Algumas das principais técnicas serão descritas a seguir.

3.1 Detector de atividade de voz (Voice Activity Detector - VAD)

Técnicas que detectam a atividade de voz, mais conhecido como VADs, sao utilizadas nesse tipo de sistema para descartar as partes dos sinal que não possuem voz. Essas partes geralmente possuem silêncio ou ruído de fundo e degradam o desempenho do sistema, uma vez que produzirão vetores que não possuem informação da voz do locutor.

3.1.1 VAD baseado em energia e taxa de cruzamento pelo zero

Muitos dos algoritmos de VAD são baseados em medidas de energia. Estes algoritmos têm a vantagem de serem simples e de não levarem em conta as características específicas do ruído. Entretanto, algoritmos baseados na energia são sensíveis ao ruído e, portanto, variações no mesmo podem reduzir seu desempenho [Rab74]. Tais algoritmos têm conseguido bons desempenhos em ambientes com pouco ruído, principalmente quando a energia dos segmentos de voz é significativamente maior do que os segmentos que possuem somente ruído.

Dado um sinal digital de voz x com N amostras, a energia desse sinal é definida como a soma das magnitudes de cada uma das amostras [Rab74]:

$$E(\mathbf{x}) = \sum_{n=1}^{N} |\mathbf{x}[n]|. \tag{3.1}$$

Primeiramente o sinal é dividido em *frames*, geralmente de 20ms, e a energia é calculada em cada um deles. Na Figura 3.1 é possível visualizar a distribuição da energia de um sinal de voz extraído da base de dados utilizada neste trabalho. A linha horizontal delimita um limiar de separação para a extração da componente útil. As amostras acima do limiar indicam uma componente útil da locução.

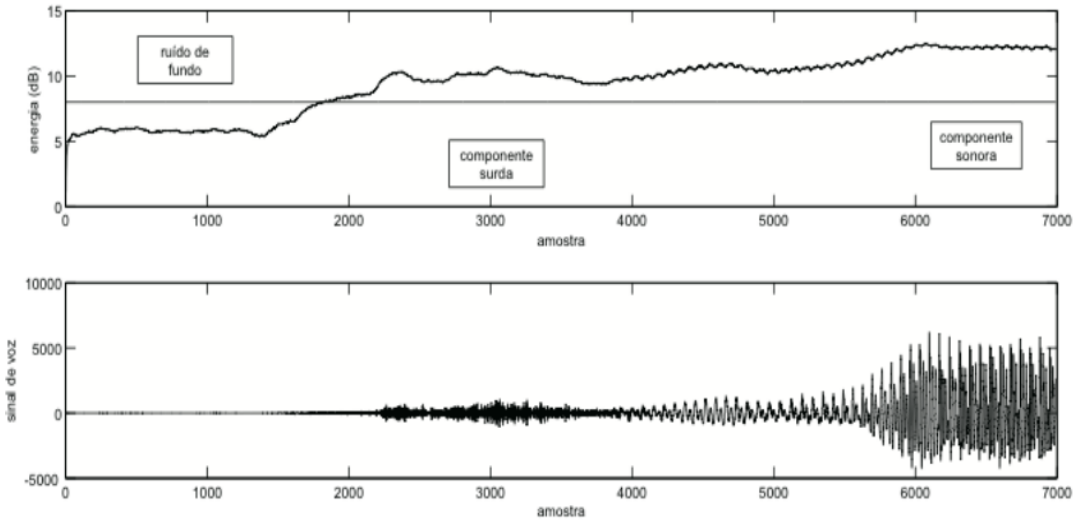

Figura 3.1 Distribuição da energia (figura superior) de um sinal de voz (figura inferior).

A técnica de cruzamento pelo zero consiste em verificar a frequência de alternância entre sinais positivos e negativos dentro da janela de um determinado sinal. O número de cruzamentos pelo zero em um sinal de voz encontra-se em um intervalo fixo de valores. Por exemplo, para uma janela de 10 ms, o número de cruzamentos fica, hipoteticamente, entre 10 e 20 vezes. O número de cruzamentos no ruído é aleatório e imprevisível. Esta propriedade permite formular uma regra de decisão que é independente da energia do sinal. O objetivo da técnica é extrair do sinal as partes que não correspondem a componente útil, ou seja, partes do sinal onde apenas o ruído de fundo está presente. Isso é possível em função de que estes tipos de sinais apresentam um elevado grau de aleatoriedade, ocasionando que suas taxas de cruzamento pelo zero sejam mais elevadas quando comparadas com as janelas com a componente útil.

Na Figura 3.2 é possível comparar a taxa de cruzamento pelo zero (figura superior) com as correspondentes partes de um sinal de voz real, gravado em um estúdio (figura inferior). Percebe-se que a taxa de cruzamento é significativamente maior nos pontos do sinal onde existe apenas a presença de ruído de fundo e da componente surda. Nos pontos da componente sonora, a taxa é menor. Para tanto, uma linha horizontal foi traçada na figura superior para servir como referencia para discriminar a componente sonora das demais. Ou seja, quando um ponto no diagrama de cruzamento por zero está acima deste limiar, tem-se a predominância da componente surda ou ruído de fundo.

Uma constatação que se faz a partir da análise das Figuras 3.1 e 3.2 é que não é possível destacar o ruído de fundo da componente surda do sinal de voz, uma vez que ambos apresentam uma alta taxa de cruzamento pelo zero. Portanto, o uso exclusivo desta técnica não garante que o sinal será segmentado adequadamente. Por isso, uma combinação entre as técnicas de energia e cruzamento por zero foi proposta.

Inicialmente o sinal de entrada é dividido em janelas. A energia de cada janela é calculada e se sua energia não for superior ao limiar, é calculado o número de cruzamento pelo zero para a mesma. Caso o número de cruzamentos esteja compreendido no intervalo [10, 20] a janela é

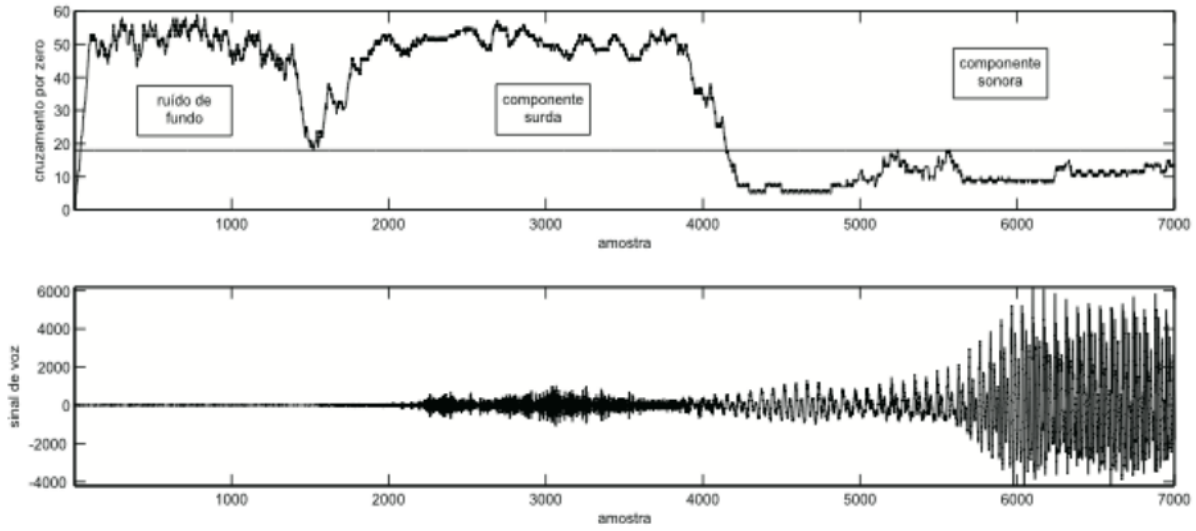

Figura 3.2 Taxa de cruzamento pelo zero de um sinal. A figura superior exibe as taxas no decorrer do sinal, enquanto que a inferior exibe o sinal analisado.

classificada como voz, do contrário, é descartada pois não possui voz. A escolha deste intervalo foi baseada nos estudos desenvolvidos por Yamamoto *et al.* em [YJRK06].

3.1.2 VAD baseado na análise biespectral

Apesar de na literatura prevalecer métodos de detecção de atividade de voz baseados em energia e taxa de cruzamento pelo zero, recentemente uma técnica baseada na análise do biespectro do sinal foi proposta por Dou *et al.* [DWFQ10]. O objetivo dos autores foi desenvolver uma metodologia para classificar de forma precisa as diferentes regiões de um sinal de áudio de acordo com a presença de voz, discriminando eficientemente, e independentemente de ruídos, situações de locução e silêncio. Com esta finalidade, foi desenvolvido um método que, a partir do cálculo do biespectro de um sinal, extrai as características relevantes do mesmo e as analisa.

Antes de definir o biespectro, é necessário compreender a noção de cumulantes de uma sequência $x[n]$, podendo assumir diferentes ordens, como pode ser visto a seguir:

$$C_{1x} = E\{x[n]\}, \tag{3.2}$$

$$C_{2x}(k) = E\{x^*[n]x[n+k]\}, \tag{3.3}$$

$$C_{3x}(k,1) = E\{x^*[n]x[n+k]x[n+1]\}, \tag{3.4}$$

onde $E\{\}$ é o operador de esperança.

Nota-se que o cumulante de ordem 1 de um sinal representa sua média; e o de ordem 2, a sequência de autocovariância do mesmo. Por sua vez, o poliespectro de ordem k é definido como a Transformada de Fourier dos cumulantes de mesma ordem, correspondendo a uma função de $k-1$ frequências.

Como o biespectro é uma função de duas frequências, sua fórmula equivale ao poliespectro de ordem 3:

$$B_x(w_1, w_2) = \sum_{\tau_1=-\infty}^{\infty} \sum_{\tau_2=-\infty}^{\infty} C_{3x}(\tau_1, \tau_2) e^{-j(w_1\tau_1 + w_2\tau_2)}, \tag{3.5}$$

onde os coeficientes de C_{3x} representam os cumulantes de terceira ordem do sinal de entrada $x[n]$.

Geralmente, utiliza-se a abordagem direta para estimar o biespectro de um sinal, método também adotado pelos autores e que possui essencialmente quatro passos:

- Divisão do sinal em K segmentos $\{x^{(1)}, x^{(2)}, ..., x^{(K)}\}$ de M amostras cada, onde é aceitável a existência de sobreposição entre os segmentos;
- Cálculo dos coeficientes da Transformada Discreta de Fourier (*Discrete Fourier Transform - DFT*) para cada um dos K segmentos:

$$X^{(k)}(\lambda) = \frac{1}{M} \sum_{n=0}^{M-1} x^{(k)}[n] e^{\frac{-j2\pi n\lambda}{M}}, \tag{3.6}$$

  onde $\lambda = 0, 1, ..., \frac{M}{2}$ e $k = 1, 2, ..., K$.

- Computação da correlação tripla da transformada de cada segmento, dada por:

$$\widehat{b_k}(\lambda_1, \lambda_2) = \frac{1}{\triangle_0^2} \sum_{i_1=-L_1}^{L_1} \sum_{i_2=-L_1}^{L_1} X^{(k)}(\lambda_1 + i_1) X^{(k)}(\lambda_2 + i_2) X^{(k)}(-\lambda_1 - \lambda_2 - i_1 - i_2), \tag{3.7}$$

$$\triangle_0^2 = \frac{f_s}{N_0}, \tag{3.8}$$

  onde f_s é a taxa de amostragem de $x[n]$, $0 \leq \lambda_2 \leq \lambda_1$, $\lambda_1 + \lambda_2 \leq f_s/2$ e $k = 1, 2, ..., K$. Além disso, a seleção de N_0 e L_1 deve satisfazer:

$$M = \frac{2L_1 + 1}{N_0}. \tag{3.9}$$

- Estimativa do biespectro através da média da correlação tripla de cada um dos K segmentos:

$$\widehat{B}_D(\omega_1, \omega_2) = \frac{1}{K} \sum_{k=1}^{K} \widehat{b_k}(\omega_1, \omega_2), \tag{3.10}$$

  onde $\omega_1 = \frac{2\pi f_s}{N_0}\lambda_1$ e $\omega_2 = \frac{2\pi f_s}{N_0}\lambda_2$.

O resultado da abordagem direta de estimativa do biespectro é um gráfico tridimensional, onde as duas primeiras dimensões representam as frequências independentes ω_1 e ω_2 e a terceira, a magnitude do biespectro. Como exemplo, pode-se visualizar o biespectro na Figura 3.3, representado em curvas de nível. O intervalo de ambas as frequências encontram-se normalizados e limitados pela frequência de Nyquist (igual à metade da frequência de amostragem, normalizada em 1).

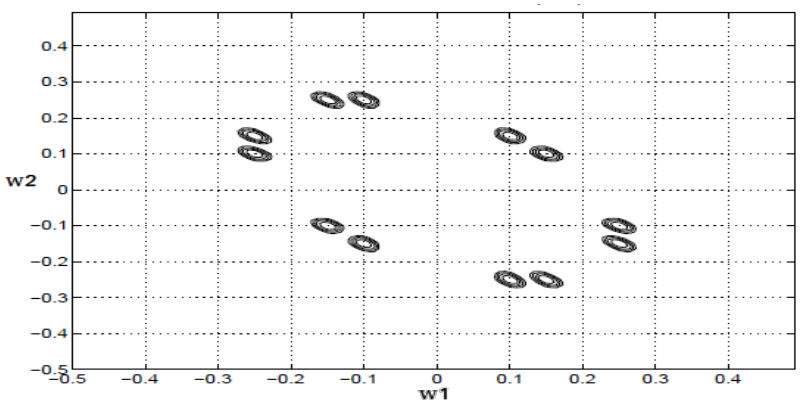

Figura 3.3 Biespectro gerado pelo método de estimação direta.

No método proposto, a análise biespectral é realizada apenas para valores iguais de ω_1 e ω_2, ou seja, observa-se apenas o corte diagonal do biespectro tridimensional, como indicado na fórmula abaixo:

$$\widehat{B}_{diagonal} = \widehat{B}_D(\omega_1, \omega_2)|_{\omega_1=\omega_2} \tag{3.11}$$

Os autores fundamentaram a etapa de extração de características do biespectro a partir da análise de quatro sinais de voz, consistindo da combinação de dois fatores: sinal com voz surda (ou não) e presença (ou não) de ruído. O ruído desses sinais é devido ao ambiente interno de um automóvel e possui SNR (Signal-to-Noise Ratio) de 0 dB. O corte diagonal de cada um dos quatro biespectros pode ser visto nas Figuras 3.4 a 3.7.

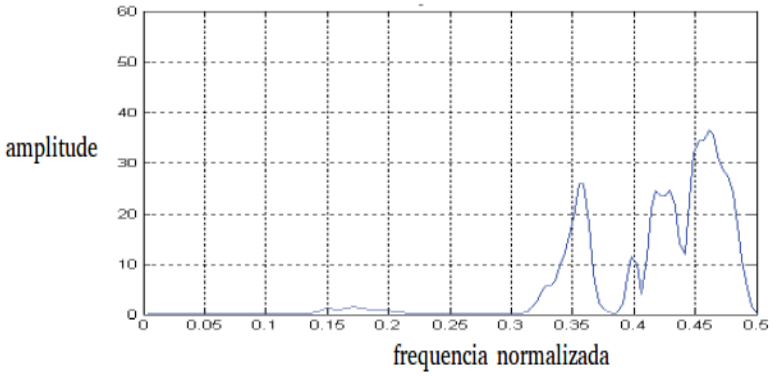

Figura 3.4 Corte diagonal do biespectro do sinal de voz sem ruído.

A análise desses biespectros levou os autores a identificar os intervalos de frequência (para cada uma das duas modalidades de voz, surda e útil, que demonstraram pouca ou nenhuma variação de amplitude entre os casos com e sem ruído. Foi realizada, finalmente, a interseção desses dois grupos de intervalos, resultando em dois segmentos de frequência tomados como cruciais para o processo de detecção de voz: [0.125; 0.25] e [0.3125; 0.375].

Sendo E_l a energia calculada no primeiro intervalo e E_h a energia calculada no segundo, os autores decidiram utilizar essas duas medidas, em conjunto com um limiar, para classificar um

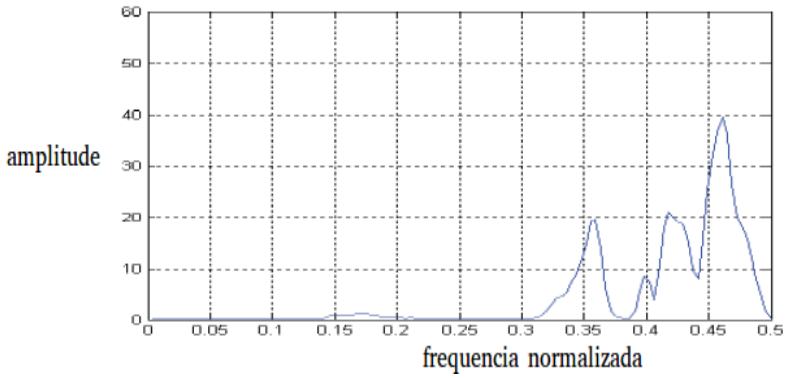

Figura 3.5 Corte diagonal do biespectro do sinal de voz com ruído.

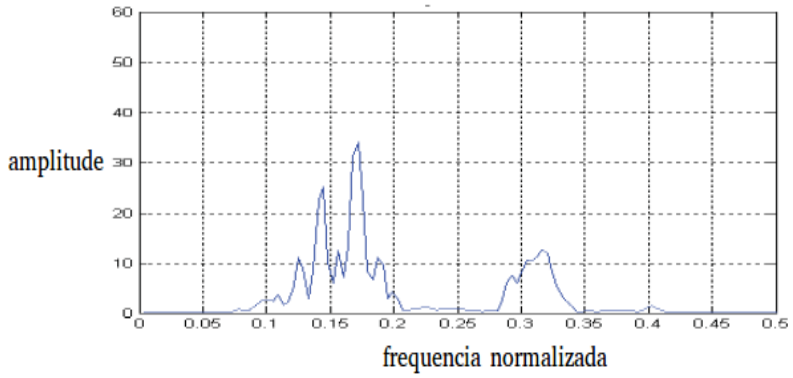

Figura 3.6 Corte diagonal do biespectro do sinal de voz surda sem ruído.

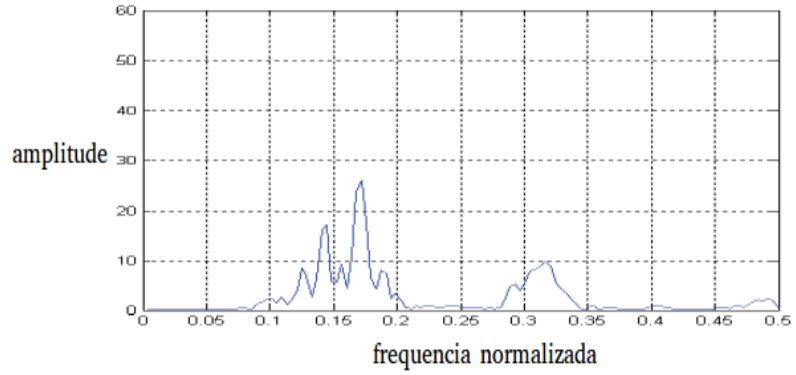

Figura 3.7 Corte diagonal do biespectro do sinal de voz surda com ruído.

trecho de áudio como sendo de voz ou silêncio:

$$E_l = \sum_{i=0.125}^{0.25} |\widehat{B}_{diagonal}(i)|^2 \tag{3.12}$$

$$E_h = \sum_{i=0.3125}^{0.375} |\widehat{B}_{diagonal}(i)|^2 \tag{3.13}$$

Finalmente, o algoritmo de VAD é descrito pelos autores como possuindo os seguintes passos:

- Modifica-se do sinal de áudio x[n], subtraindo-o por sua média e normalizando sua amplitude através do desvio padrão, como indicado abaixo:

$$\widehat{x}[n] = \frac{x[n] - E\{x[n]\}}{std(x[n])} \tag{3.14}$$

- Divide-se do sinal em frames de 20 ms, e computa-se o biespectro através do método direto para cada um dos frames;
- Para cada segmento, calcula-se o valor de E_l e E_h, as energias do biespectro nos intervalos tomados como característicos da voz humana, usando as Equações 3.12 e 3.13;
- Aplica-se limiares sobre E_l e E_h e se ambas as energias estiverem acima dos limiares, o frame é considerado de voz; caso contrário, será considerado de silêncio.

Na prática, os limiares utilizados para comparar os valores de E_l e E_h devem ser estimados de forma empírica. Alguns exemplos são mostrados nas Figuras 3.8, 3.9 e 3.10. Essas três figuras mostram o resultado da detecção (a função possui valor um quando a atividade de voz é detectada). Na Figura 3.8, o sinal foi gravado em um ambiente livre de ruídos externos (em uma sala de escritório silenciosa). Já no exemplo mostrado na Figura 3.9, o sinal já possui algum ruído, uma vez que foi gravado no *hall* de entrada de um prédio. E no sinal da Figura 3.10, podemos perceber a presença de muito ruído. Esse sinal foi gravado no cruzamento de duas ruas com tráfego de carros.

Podemos perceber o bom desempenho da detecção em todos os três casos.

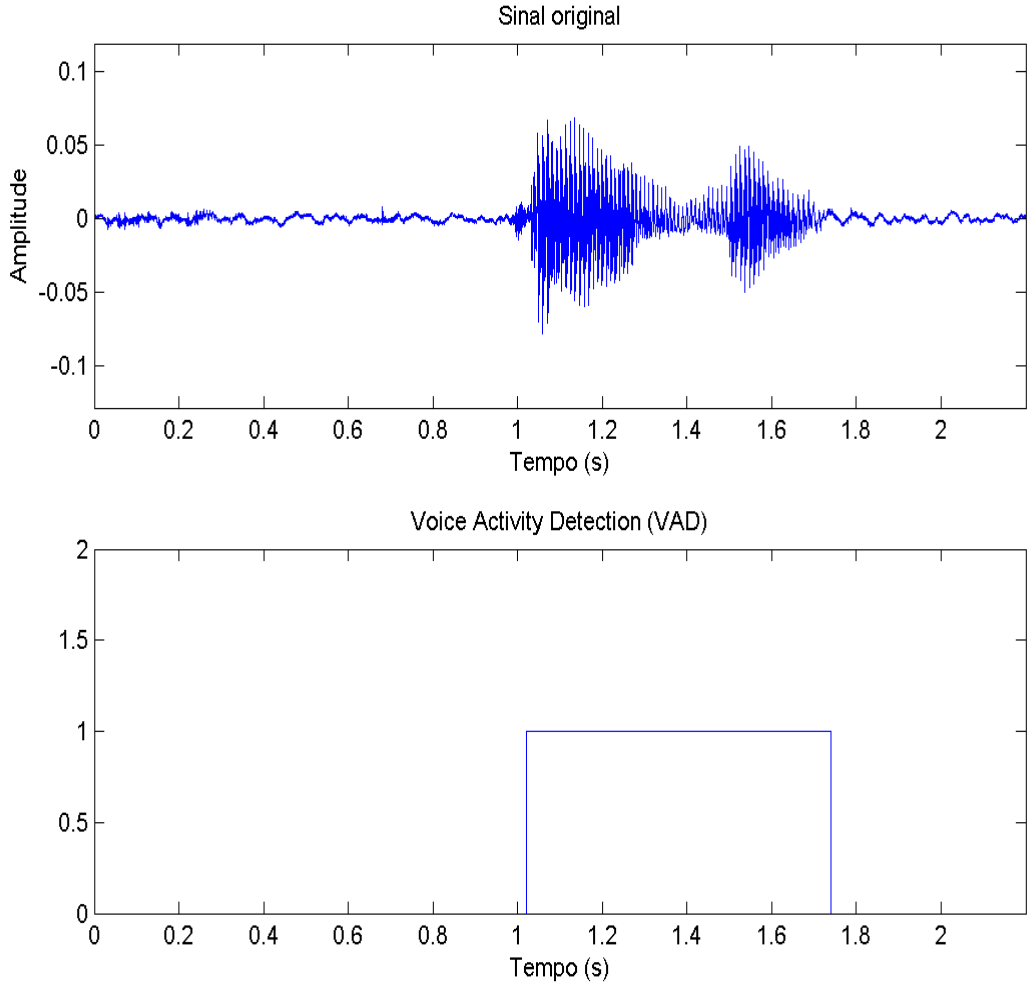

Figura 3.8 Exemplo da utilização do VAD baseado na análise do biespectro em um sinal de voz sem ruído.

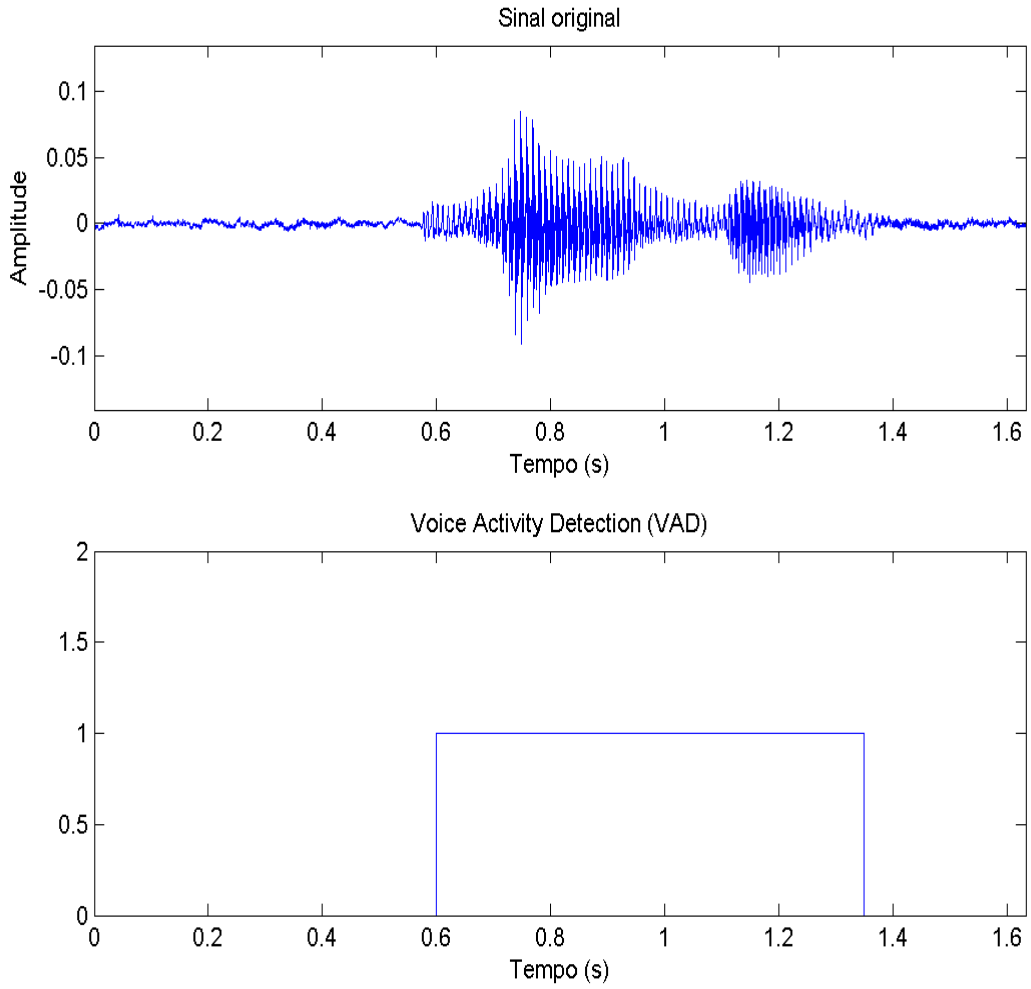

Figura 3.9 Exemplo da utilização do VAD baseado na análise do biespectro em um sinal de voz com pouco ruído.

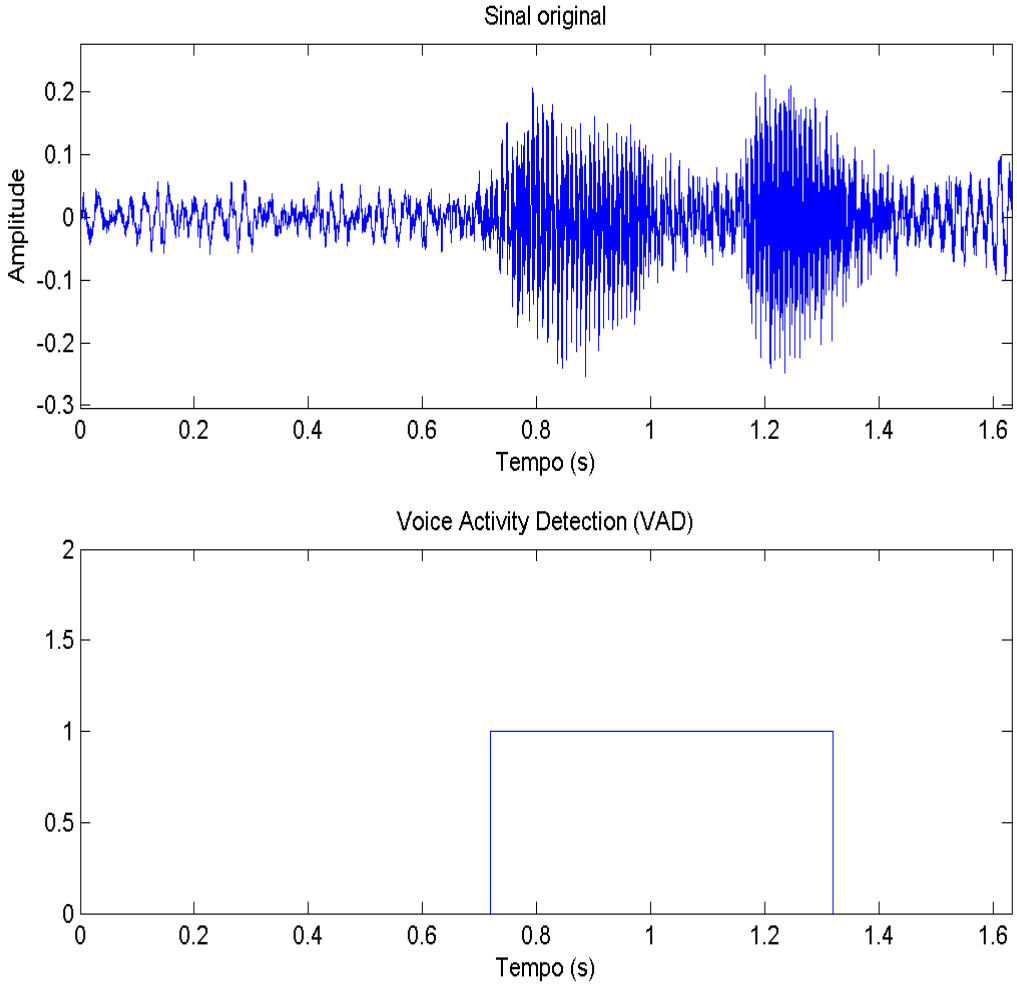

Figura 3.10 Exemplo da utilização do VAD baseado na análise do biespectro em um sinal de voz com muito ruído.

CAPÍTULO 4

Extração de Características

O processo de extração de características é responsável por transformar o sinal de voz em vetores que contenham as características únicas do trato vocal do usuário e que tornem redundantes as informações que são comuns a todos os usuários.

O sinal de voz possui inúmeras características que são relevantes para o processo de discriminação entre um usuário e outro. Segundo Wolf em [Wol72], uma característica ideal deve:

- possuir uma alta variação entre os locutores e uma baixa variação intra-locutor;
- ser robusta quando da presença de ruídos e distorções;
- ocorrer frequentemente e naturalmente durante a fala;
- ser fácil de medir e extrair do sinal de voz;
- ser difícil de ser produzida artificialmente;
- não ser afetada por questões de saúde do locutor ou por variações de longo tempo ocorridas na voz do mesmo.

Existem diferentes formas de categorizar as características, como exibido na Figura 4.1. Baseando-se no trato vocal e em aspectos comportamentais, elas podem ser divididas em (i) espectrais de tempo curto, (ii) espectro-temporais, (iii) prosódicas e (iv) de alto nível. As características espectrais de tempo curto são computadas sobre intervalos de, em geral, de 10 a 30 ms do sinal. Elas são referenciadas como descritores do envelope espectral da voz, que é composto por propriedades supralaríngeas do trato vocal, como o timbre. As características prosódicas e espectro-temporais definem-se no sinal ao longo do tempo, como ritmo e entonação. As de alto nível, representam características em nível de conversação, como maneiras diferentes de falar uma mesma palavra, sotaques, entre outros.

4.1 Coeficientes Cepstrais da Escala Mel

Na área de processamento de voz, a representação mais utilizada para representar um sinal consiste nos coeficientes cepstrais extraídos do domínio espectral na escala Mel (*Mel-Frequency Cepstral Coefficients, MFCC*), que geralmente são associados às suas derivadas no tempo e ao logarítimo da energia do sinal. Uma razão para a completa aceitação desse tipo de representação está no bom desempenho em praticamente todos os sistemas que utilizam sinais de voz, como identificação, classificação e reconhecimento.

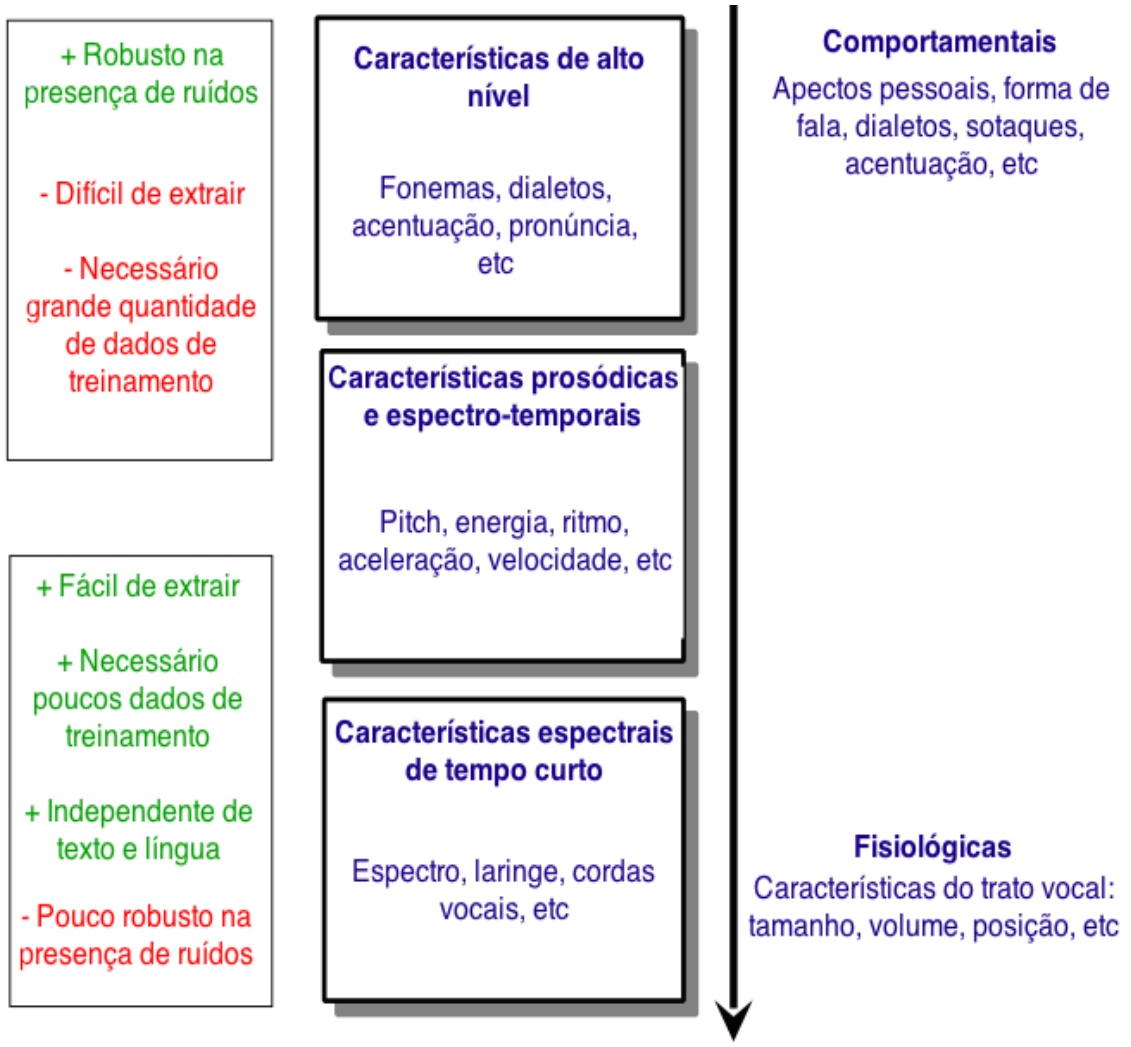

Figura 4.1 Resumo dos tipos de características que podem ser extraídos de um sinal de voz. A figura apresenta também as vantagens e desvantagens de cada uma delas, partindo das características comportamentais até as fisiológicas.

A técnica de extração dos MFCCs foi proposta na década de 1980 por Davis *et al.* em [DM80] para o uso em reconhecimento de fala e posteriormente começou a ser utilizado também para o reconhecimento de locutores. Mesmo com várias técnicas alternativas, como os estudos razoavelmente difundidos realizados sobre os centroides espectrais [KZZW07], MFCC continua sendo a técnica mais utilizada em função de ser a que obtém melhor desempenho. Esse bom desempenho é geralmente associado ao fato dessa representação concentrar características perceptualmente relevantes [DM80].

O ouvido humano identifica as frequências do espectro de forma não-linear. Em um experimento perceptual conduzido por Stevens *et al.* [SVN37], investigou-se a relação entre a altura escutada pelo ouvinte e a frequência do tom emitido. Foi pedido a ouvintes para definirem escalas de alturas equidistantes de acordo com suas impressões subjetivas. Primeiro atribui-se

o valor de 1000 mels para a altura relacionada ao tom de 1000 Hz. Para um tom que soava, na média, duas vezes mais alto que o tom de 1000 Hz, atribui-se o valor de 2000 mels, enquanto que atribuiu-se o valor de 500 mels ao tom que soaria, na média, a metade da altura do tom de 1000 Hz. Baseado nos resultados desse experimento, a escala Mel foi definida como:

$$f_{mel} = 2595 mel * log_{10}(1 + \frac{f}{700}) \tag{4.1}$$

onde f é a frequência em Hz.

Na maioria das aplicações de voz, características cepstrais são utilizadas. A análise do *Cepstrum* [BHTR63, OW04] é tipicamente usado quando os efeitos da fonte do sinal e uma funçao de transferência (possivelmente variante no tempo) de um sistema devem ser separados.

O chamado *Cepstrum* de frequência na escala Mel (*Mel-Frequency Cepstrum - MFC*) é o espectro de potência de tempo curto de um sinal de voz baseado na transformada linear do cosseno do logarítimo do espectro de potência na escala Mel.

A Figura 4.2 mostra o diagrama de blocos do processo de extração dos coeficientes MFCC. O processo de extração é realizado a partir dos seguintes passos:

- Primeiramente, na fase de pré-processamento, um filtro passa-alta é aplicado ao sinal de voz normalizado. Usualmente, utiliza-se o seguinte filtro:

$$H(z) = 1 + (-0.95 * z^{-1}) \tag{4.2}$$

- O sinal de voz pré-processado é então dividido em blocos. Geralmente, os blocos são formados de maneira que possuam de 10 a 30 ms e possuam um *overlap* de 10 ms. O mais usual é gerar blocos de 20 ms a cada 10 ms.

- Para manter a continuidade entre os blocos, técnicas de janelamento são utilizadas, sendo a principal delas a utilização da janela de Hamming (ver Apêndice A).

- Aplica-se, então, a transforma de Fourier para levar cada um dos sinais presentes nos blocos para o domínio da frequência. Usualmente, o algoritmo da Transformada Rápida de Fourier (*Fast Fourier Transform - FFT*) é utilizado, com tamanho 512.

- Nesse ponto do processo, um banco de filtros na escala Mel é utilizado. Esse banco foi produzido a partir dos experimentos conduzidos por Stevens *et al.*, como mencionado anteriormente. Esse banco possui um total de 24 filtros, geralmente triangulares, centrados em uma certa frequência e possuindo um certo comprimento de banda. Esses valores estão descritos na Tabela 4.1.

- O próximo passo consiste na extração do *Cepstrum* na escala Mel, que se resume em aplicar um certo número de filtros, L, do banco apresentado acima.

  Dado $S[k]$ o espectro de frequência do sinal e M_i a função do filtro i, o *Cepstrum* de $S[k]$ é definido como:

$$\widetilde{S}[i] = \sum_{k=-\infty}^{\infty} |S[k]| M_i[k] \tag{4.3}$$

  onde $i = 0, 1, ..., L-1$.

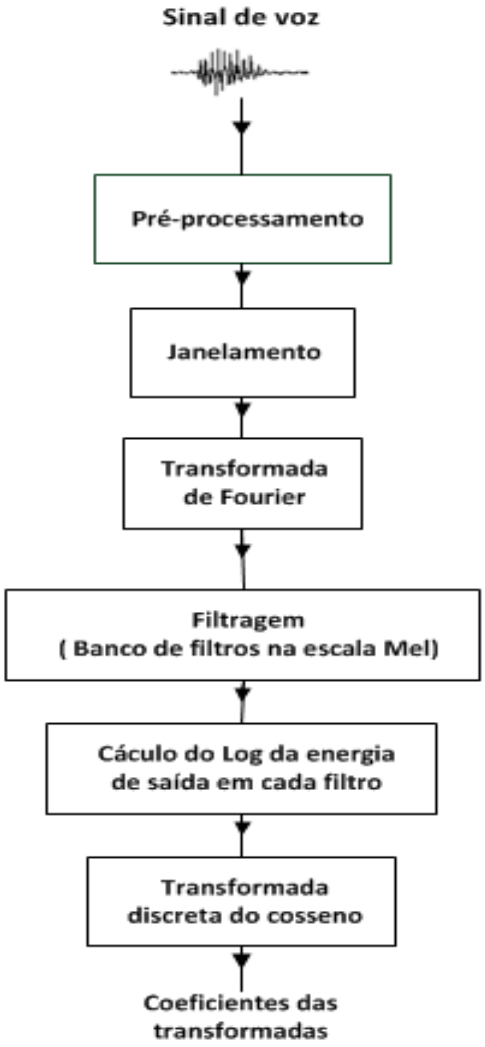

Figura 4.2 Diagrama de blocos do processo de extração dos coeficientes MFCC de um sinal de voz.

- O último passo, enfim, consiste na extração dos coeficientes do *Cepstrum*, utilizando a Transformada Discreta do Cosseno:

$$c_j = \sqrt{\frac{2}{p}} \sum_{i=1}^{p} log(\widetilde{S}[i]).cos(\frac{\pi j}{p}(i-0.5)), \tag{4.4}$$

  onde p é o número de coeficientes e $j = 1, ..., p$.

4.2 Energia

Além das informações espectrais, a energia ou itensidade do sinal é geralmente incluída no vetor de características. Essa intensidade é estimada a partir do logarítimo da energia do sinal

Tabela 4.1 Especificação do banco de filtros do MFCC, na escala Mel.

Filtro	Frequência (Hz)	Comprimento da banda (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
21	4595	639
22	5278	734
23	6063	843
24	6964	969

presente no bloco:

$$E = log \sum_{n=1}^{N} [s[n]]^2. \tag{4.5}$$

4.3 Transformações do vetor de características

As análises espectrais até agora descritas produzem um vetor de características para cada bloco do sinal. Porém, o sistema deve possuir uma certa robustez quanto às possíveis distorções ocasionadas pelo ambiente onde o sistema será utilizado.

4.3.1 Normalização de canal

Normalização de canal é o processo de compensar as possíveis distorções causadas no sistema que produziu o sinal. Distorções podem ser causadas pelo ambiente (outros sons ou efeitos

sonoros de fundo) ou durante a transmissão do sinal.

Existem dois principais métodos para reduzir essa distorção. Em **normalização ou subtração cepstral de média** (*Cepstral Mean Normalization or Subtraction - CMN or CMS*) [Fur81], cada vetor de características é subtraído do vetor de média:

$$c_i = c_i - \frac{1}{N}\sum_{k=1}^{N} c_{ik} \tag{4.6}$$

onde c_{ik} é o i-ésimo coeficiente do k-ésimo bloco.

Essa operação reduz o impacto das distorções estacionárias e das que possuem pequenas distorções variantes no tempo.

4.3.2 Filtragem RASTA

A chamada **filtragem relativa ao espectro** (*Relative Spectral - RASTA*) foi proposto inicialmente como pré-processamento da técnica de extração de características PLP (Perceptual Linear Prediction) [HMBK92, HM94]. A filtragem RASTA aplica filtros passa-faixa nos domínios cepstrais ou espectrais logarítmicos e possui a seguinte função de transferência:

$$T(z) = 0.1z^4 * \frac{2 + z^{-1} - z^{-3} - 2z^{-4}}{1 - 0.98z^{-1}} \tag{4.7}$$

Distorções lineares são causadas, por exemplo, pelo canal de comunicação ou pela utilização de um microfone diferente do utilizado para registrar um determinado locutor. Essas distorções aparecem como constantes no domínio espectral logarítmico. É esperado que a parte passa-alta do filtro alivie os efeitos dos ruídos convolucionais introduzidos pelo canal. Já a parte passa-baixa do filtro ajuda a suavizar algumas das mudanças bruscas presentes nas características espectrais extraídas de frames consecutivos.

4.3.3 Coeficientes delta e de aceleração

As características cepstrais capturam a distribuição espectral durante um bloco específico. Porém, muita informação do sinal de voz está em como essas distribuições mudam com o tempo, ou seja, está na sua dinâmica. Para capturar esse tipo de informação, geralmente as derivadas no tempo são estimadas. Porém, uma série de amostras temporais de um determinado coeficiente cepstral, $c_m(t)$, geralmente não possui uma forma analítica e o cálculo de sua derivada, $\delta c_m/\delta t$, pode ser apenas aproximada por uma diferença finita.

A diferença finita de primeira ordem é definida como:

$$d_m[t] = c_m[t+1] - c_m[t]. \tag{4.8}$$

Porém, esse tipo de estimação da derivada é intrínsicamente ruidosa. Dessa maneira, Furui [Fur81] propôs a utilização de um *fit* ortogonal polinomial da trajetória temporal de um determinado coeficiente cepstral utilizando uma janela de tamanho finito.

O termo constante do polinômio ortogonal é dado por:

$$\widehat{c}_m(t) = \frac{\sum_{k=-K}^{K} h_k c_m(t+k)}{\sum_{k=-K}^{K} h_k}, \tag{4.9}$$

onde h_k é uma janela, geralmente simétrica, de tamanho $2K+1$. O coeficiente de primeira ordem do polinômio ortogonal, denotado por Δc_m, é definido como:

$$\frac{\delta c_m(t)}{\delta t} \approx \Delta c_m = \frac{\sum_{k=-K}^{K} k h_k c_m(t+k)}{\sum_{k=-K}^{K} h_k k^2}. \tag{4.10}$$

Coeficientes polinomiais ortogonais de ordens maiores podem ser derivados de forma semelhante [Bev69]. Por outro lado, Furui [Fur81] mostrou que a utilização dos coeficientes de primeira ordem para caracterizar a dinâmica dos coeficientes cepstral é geralmente adequada.

A mesma equação pode ser utilizada para computar os coeficientes de aceleração (ou de segunda ordem), basta que seja aplicada aos coeficientes delta de primeira ordem.

CAPÍTULO 5

Modelos dos Locutores

Uma importante etapa na implementação de um sistema de reconhecimento de locutor é a escolha da função de densidade de probabilidade responsável pelo cálculo da verrosimilhança das hipóteses apresentadas na Seção 2.3. O processo de decisão de uma determinada locução, como mostrado na Equação 2.1, necessita de dois modelos, que representam os casos em que a locução foi produzida por um determinado locutor ou não. Como vimos na Seção 2.4.1, esses modelos são referenciados como λ_{hip} e $\lambda_{\overline{hip}}$, respectivamente.

Apesar de muitos trabalhos se concentrarem na escolha das características extraídas do sinal de voz, os últimos grandes avanços se deram em pesquisas relacionadas em encontrar métodos de aprendizagem para estimar os modelos. Tais modelos podem ser vistos como funções de densidade de probabilidade e os métodos de aprendizagem são utilizados para estimar os parâmetros dessas funções de modo a maximizar a verossimilhança dos modelos, dados os vetores extraídos no processo anterior. A escolha desta função é altamente dependente das características escolhidas para representar o usuário e da forma de utilização das mesmas [Rey95]. Nesta seção são discutidas as formas de modelagem comumente utilizadas em sistemas independentes de texto.

5.1 Modelos de Misturas Gaussianas

Os modelos de misturas gaussianas (*Gaussian Mixture Models - GMM*) foram primeiramente utilizados em reconhecimento de locutor em 1995, no trabalho realizado por Reynolds em [Rey95]. Desde então esta técnica vem sendo utilizada como referência para a modelagem dos locutores e se mostrou a mais bem-sucedida função de verossimilhança em sistemas independentes de texto.

GMM representa de forma geral a dependência das características da voz associadas ao locutor, em conjunto com a capacidade de modelar densidades de probabilidades desconhecidas, especificamente a distribuição dos vetores de características extraídos de uma locução.

Segundo Reynolds *et al.* em [DRQ00] as vantagens de se usar GMM como técnica de modelagem para sistemas de reconhecimento de locutor independentes de texto são o baixo custo computacional, a já bem fundamentada teoria estatística do modelo e, principalmente, o fato de que não são sensíveis aos aspectos temporais de um sinal de voz, representando com precisão apenas os aspectos característicos ao locutor. Essa última característica é importante para sistemas que não possuem dependência de texto, entretanto a mesma é uma desvantagem quando da dependência do mesmo. Em geral, sistemas que dependem do texto costumam utilizar técnicas que também modelem os aspectos temporais do sinal de voz, como Modelos

Escondidos de Markov (*Hidden Markov Models - HMM*).

GMM nada mais é que uma combinação linear de um número finito, M, de distribuições Normais. Dado um vetor $\mathbf{x}$ de dimensão D, sua função de densidade é dado por:

$$p(\mathbf{x}|\lambda) = \sum_{i=1}^{M} \omega_i N(\mathbf{x}; \mu_i, \Sigma_i), \tag{5.1}$$

onde μ_i e Σ_i são o vetor de média e a matrix de covariância da distribuição i, respectivamente. ω_i representa o peso da distribuição i na modelagem global e os pesos satisfazem $\sum_{i=1}^{M} \omega_i = 1$.

$N(\mathbf{x}; \mu_i, \Sigma_i)$ é a distribuição Normal multivariada e é definida como:

$$N(\mathbf{x}; \mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu_i)^T \Sigma_i^{-1}(\mathbf{x}-\mu_i)}. \tag{5.2}$$

A matriz de covariância pode ser representada de forma completa ou somente através de sua diagonal [DRQ00]. Comumente utiliza-se somente o vetor diagonal da matriz de covariância, especialmente por razões de desempenho computacional. Além disso, estimar a matriz completa com precisão requer uma quantidade de dados significativamente maior.

De maneira geral, um modelo GMM pode ser completamente definido pelos seus pesos, seus vetores de média e suas matrizes de covariância: $\lambda = \{\omega_i, \mu_i, \Sigma_i\}$, para $i = 1, ..., M$.

Portanto, com o objetivo de criar um modelo para o cálculo da semelhança quanto às locuções de um determinado locutor, a metodologia empregada consiste em utilizar locuções desse locutor, extrair os vetores e estimar os parâmetros do GMM utilizando esses vetores. Da mesma forma, para estimar os parâmetros do modelo de fundo universal (UBM), utiliza-se os vetores extraídos das locuções de fundo.

O método mais empregado para estimar os parâmetros de um GMM a partir de um conjunto de vetores é a utilização de um algoritmo que maximize a verossimilhança do modelo quanto aos vetores utilizado para treinamento. Esse processo é geralmente referenciado como treinamento do modelo. O algoritmo mais utilizado é o chamado "Maximização de Expectativa" (*Expectation-Maximization - EM*). Mais detalhes sobre esse algoritmo pode ser visto no Apêndice B.

5.1.1 Processo de verificação utilizando GMMs

Uma vez que tanto o modelo do locutor (λ_{SPK}) quanto o modelo de fundo universal (λ_{UBM}) foram estimados, o teste de um vetor pode ser realizado com o chamado teste da razão das verossimilhanças, como descrito na Equação 2.1. Cada um dos GMMs provê uma função de densidade de probabilidade que é utilizada para calcular a verossimilhança do vetor à hipótese que o modelo representa. O processo de teste de uma locução segue o procedimento descrito na Seção 2.4.2.

Uma visão geral da arquitetura de treino e teste pode ser visualizada na Figura 2.1.

5.2 Modelos de Misturas Gaussianas Adaptativas

Como visto na seção anterior, a abordagem de utilizar GMMs como funções de verossimilhança obteve bastante aceitação e têm sido o método mais utilizado para esse propósito. Além disso, verificou-se que a utilização de um modelo de fundo universal (UBM) era um método bem apropriado para modelar a hipótese que a locução não foi produzida pelo locutor em questão. Em [Rey95], que foi o trabalho que consolidou essa metodologia, tanto o GMM do locutor quanto o GMM do UBM eram treinados de forma independente, pelo algoritmo EM (Apêndice B).

Quando utilizamos o teste da razão de verossimilhança (ver Seção 2.3) para realizar a aceitação ou não de uma determinada locução de teste, X, podemos verificar que os valores das probabilidades produzidas pelos modelos, $p(X|\lambda_{SPK})$ e $p(X|\lambda_{UBM})$ devem possuir, no caso ideal, uma relação intrínseca de complementação. Isto é, espera-se que, se os parâmetros dos modelos foram estimados de tal forma que eles representem bem as hipóteses, e quanto maior for $p(X|\lambda_{SPK})$, menor deve ser $p(X|\lambda_{UBM})$ e vice-versa. Dessa forma, o ideal seria que os modelos do locutor e do UBM fossem tão íntrinsecamente relacionados que essa relação ocorresse sempre.

Pensando nisso, Reynolds *et al.* [RQD00] propuseram uma método que estabelece uma relação forte entre os modelos do locutor e do UBM. Nesse método, o modelo do locutor é produzindo através da adaptação dos parâmetros do UBM. Essa adaptação é realizada utilizando locuções do locutor em questão e uma forma de adaptação Bayesiana [GL94, DH73]. A Figura 5.1 mostra a arquitetura desse método.

Diferente do método convencional de treinamento do modelo de locutor, via maximização da verossimilhança, independentemente do UBM, o método de adaptação se concentra em adaptar o UBM para produzir o modelo do locutor. Isso gera um casamento forte entre o UBM e o modelo do locutor, que não só leva a um melhor desempenho como proporciona um método de teste mais eficiente, do ponto de vista computacional [RQD00].

Como o algoritmo EM (Apêndice B), a adaptação também é um processo iterativo com dois passos. O primeiro passo é idêntico ao passo de expectativa do algoritmo EM, onde estatísticas são extraídas dos dados de treinamento do locutor para cada uma das distribuições que compõem o UBM. No segundo passo, essas estatísticas são combinadas com os parâmetos antigos do UBM utilizando os chamados coeficientes de mixtura dependentes dos dados. Esse coeficientes de mixtura são produzidos de modo que distribuições que possuem grande número de amostras dependam mais das estastísicas extraídas no processo de estimação de seus parâmetros, enquanto que as distribuições que possuem poucas amostras não sofram tanta influência dessas estatísticas.

O proceso de adaptação é descrito a seguir.

Dado um UBM e um conjunto de amostras de treinamento de um determinado locutor, $X = \{x_1, x_2, ..., x_T\}$, primeiro determina-se o quão alinhado as distribuições do UBM estão com relação às amostras. Isto é, para cada distribuição i do UBM, calcula-se a probabilidade *a posteriori* de cada uma das amostras:

$$Pr(\lambda_i|x_t) = \frac{\omega_i p(x_t|\lambda_i)}{\sum_{j=1}^{M} \omega_j (x_t|\lambda_j)} \tag{5.3}$$

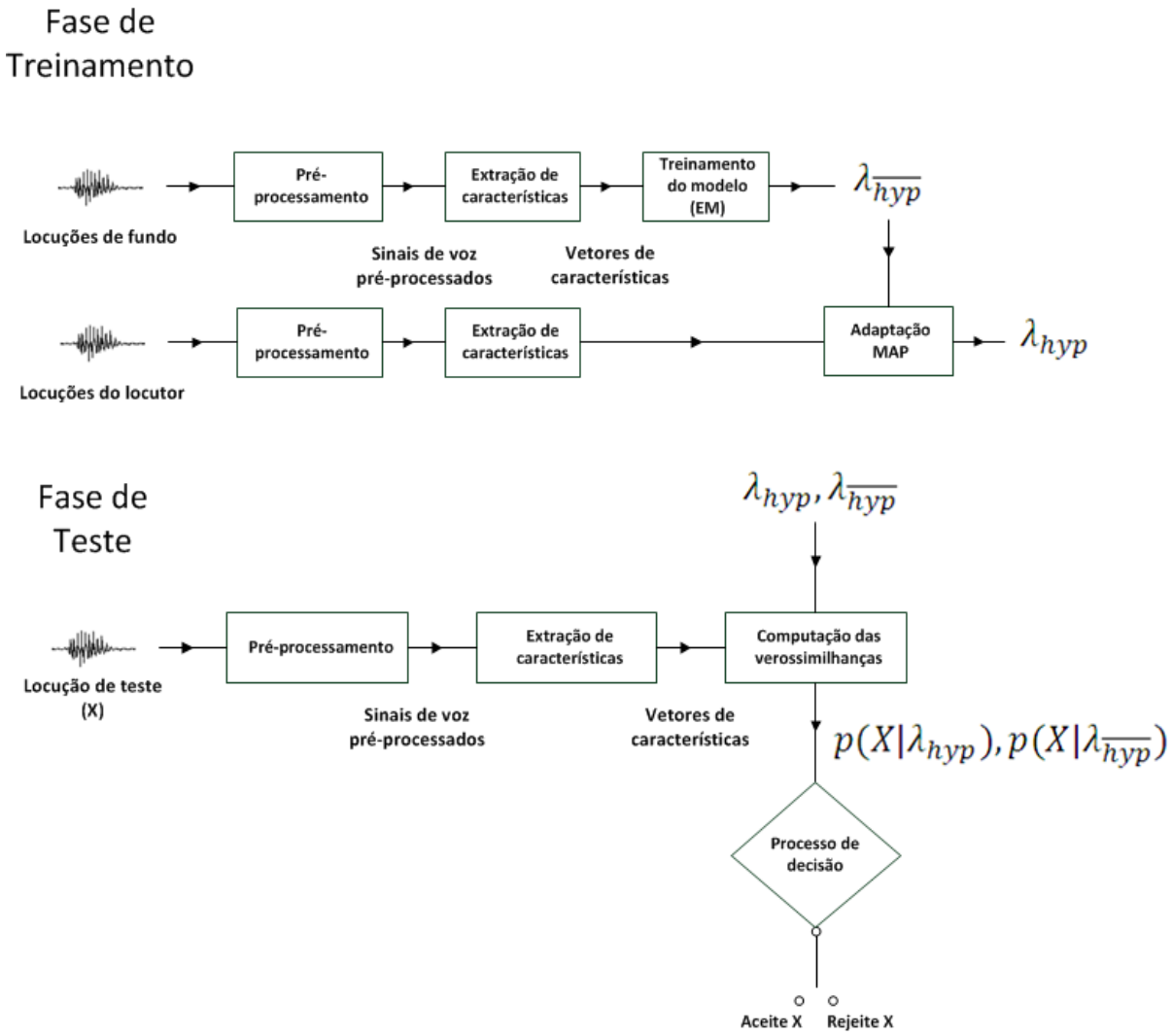

Figura 5.1 Arquitetura de um sistema baseado em modelos de misturas Gaussianas adaptativas.

onde $p(x_t|\lambda_i)$ é calculado utilizando a Equação 5.2 utilizando os parâmetros da distribuição i.

$Pr(\lambda_i|x_t)$ é então utilizada para extrair as estatísticas necessárias para os pesos, as médias e as variâncias:

$$n_i = \sum_{t=1}^{T} Pr(\lambda_i|x_t), \tag{5.4}$$

$$E_i(\mathbf{x}) = \frac{1}{n_i} \sum_{t=1}^{T} Pr(\lambda_i|x_t) x_t, \tag{5.5}$$

$$E_i(x^2) = \frac{1}{n_i} \sum_{t=1}^{T} Pr(\lambda_i|x_t) x_t^2. \tag{5.6}$$

Finalmente, essas estatísticas extraídas das amostras de treinamento são utilizadas para modificar os parâmetros correntes de cada uma das distribuições do UBM, segundo as seguintes

equações:

$$\widehat{\omega}_i = [\alpha_i^w / T + (1 - \alpha_i^w)\omega_i]\gamma, \tag{5.7}$$

$$\widehat{\mu}_i = \alpha_i^m E_i(\mathbf{x}) + (1 - \alpha_i^m)\mu i, \tag{5.8}$$

$$\widehat{\sigma}_i{}^2 = \alpha_i^v E_i(x^2) + (1 - \alpha_i^v)(\sigma_i^2 + \mu_i^2) - \widehat{\mu}_i{}^2. \tag{5.9}$$

Os coeficientes que controlam o balanço entre a estimativa corrente e as novas estimativas dos parâmetros são $\{\alpha_i^w, \alpha_i^m e \alpha_i^v\}$ para os pesos, as médias e as variâncias, respectivamente. O fator de escala γ é calculado sobre todas as distribuições de modo que os pesos somem 1.

Para cada distribuição e cada parâmetro, um coeficiente de adaptação dependente de dado, $\alpha_i^p, p \in \{w, m, v\}$, é utilizado. Ele é definido como:

$$\alpha_i^p = \frac{n_i}{n_i + r^p}, \tag{5.10}$$

onde r^p é o fator de relevância para o parâmetro p.

A atualização dos parâmetros descritos pelas Equações 5.7 - 5.9 podem ser derivadas do método de estimação *maximum a posteriori* (MAP) para GMMs [GL94].

A utilização de coeficientes de adaptação que são dependentes das amostras de treinamento permite que as adaptações dos parâmetros dependam das misturas. Se uma mistura possui uma contagem probabilística baixa, n_i, dos dados, então $\alpha_i^p \to 0$ fazendo com que a adaptação não dependa das estatísticas extraídas. Por outro lado, se n_i for alto, $\alpha_i^p \to 1$, fazendo com que as adaptações utilizem essas estatísticas. O fator de escala é um meio de controlar a importância dessas estatísticas para a atualização dos parâmetros. Esse método é, então, robusto para os casos em que o conjunto de amostras de treinamento é limitado. Reynolds *et al.* [RQD00] utilizaram um único fator de escala para os parâmetro $\alpha_i^w = \alpha_i^m = \alpha_i^v = n_i/n_i + r$, com fator de fator de relevância igual a 16.

5.2.1 Teste de razão de verossimilhanças para GMMs adaptativas

A teste da razão do log das verossimilhanças para um conjunto de amostras X é definido como:

$$\Lambda(X) = \log p(X|\lambda_{SPK}) - \log p(X|\lambda_{UBM}). \tag{5.11}$$

Porém, o fato de o modelo do locutor λ_{SPK} ter sido gerado pela adaptação do modelo UBM, λ_{UBM}, permite que um método mais rápido do cálculo do *Score* seja realizado, do que meramente calcular a verossimilhança dos dois GMMs. Esse método mais rápido é baseado em dois efeitos da adaptação MAP. O primeiro deles é que, quando um GMM é composto por várias distribuições, a probabilidade de um vetor é concentrado em poucas distribuições. Isso ocorre porque, geralmente, um GMM cobre um grande espaço vetorial e um vetor de teste estará próximo de apenas algumas distribuições. Portanto, a verossimilhança pode ser muito bem aproximada ao calcular as probabilidades apenas das C distribuições mais próximas do vetor. Essas distribuições podem ser encontradas ao calcular as probabilidades individuais de cada uma delas. O segundo fato é que o modelo do locutor retêm uma relação bem próxima

com o modelo UBM. Então um vetor que é próximo de uma distribuições específica no UBM, também será próxima dessa mesma distribuição, no modelo do locutor.

Utilizando esses dois fatos, um método rápido de cálculo do *Score* pode ser realizado:

- Para cada amostra de X, determine as C distribuições do UBM que levam aos maiores *Scores*;
- Compute a verossimilhança do UBM utilizando apenas essas C distribuições;
- Compute a verossimilhança do modelo do locutor utilizando essas mesmas C distribuições;
- O *Score* é definido como a razão das verossimilhanças calculadas acima.

Para um UBM com M distribuições, esse método requer $M+C$ cálculos de distribuições de probabilidade Gaussianas para cada vetor. A sua eficiência se mostra clara ao comparar com as $2M$ computações do método convencional. Em [RQD00], o valor de C foi setado para 5.

Apesar de ser possível a atualização de todos os parâmetros das distribuições, Reynolds *et al.* mostraram que o melhor desempenho surge quando apenas as médias das distribuições são atualizadas [RQD00].

5.3 Combinação de máquinas de vetores suporte e supervetores de GMM

Como vimos na seção anterior, a contrução dos modelos dos locutores utilizando o UBM e a adaptação MAP leva a um casamento entre a modelagem dos locutores e dos impostores. Essa relação entre os dois modelos propicia um melhor desempenho no teste da vazão das verossimilhanças, o que leva a um melhor desempenho. Além disso, constatou que o melhor desempenho é alcançado quando apenas as médias são atualizadas no processo de adaptação MAP. Visando esses resultados, Campbell *et al.* propuseram a combinação de máquinas de vetores suporte (*Support Vector Machines (SVMs)*) com vetores extraídos do modelo produzido pela adaptação MAP [CSR06].

SVMs são classificadores binários que vêm ganhando bastante atenção na área de reconhecimento de padrões. Uma descrição mais detalhada do SVM pode ser vista no Apêndice C. A grosso modo, um SVM identifica, entre os padrões de treinamento, os chamados vetores suporte e o processo de classificação é realizado com base na distância da amostra de teste aos vetores suporte. Essa distância é calculada utilizando uma função de Kernel, $K(\cdot,\cdot)$, que consiste em um produto interno entre os dois vetores. A função de Kernel utilizada pelo SVM depende do problema em questão, apesar de ser possível encontrar funções que são utilizadas em diversos problemas de classificação.

Em [CSR06], Campbell *et al.* propuseram a utilização de SVMs no espaço de vetores produzidos pelas médias do modelo do locutor. Esses vetores são referenciados como supervetores de um modelo GMM (*GMM supervectors*). Além disso, eles propuseram duas funções de Kernel adequadas para o cálculo de distância entre dois modelos GMM distintos. Mais detalhes sobre essa técnica são apresentados a seguir.

5.3.1 Supervetores de um modelo GMM

Como mencionado anteriormente, um classificador SVM é treinado utilizando vetores produzidos pelas locuções de treino. Esses vetores são chamados supervetores de um modelo GMM. A Figura 5.2 mostra o processo de extração de um supervetor a partir de uma locução qualquer.

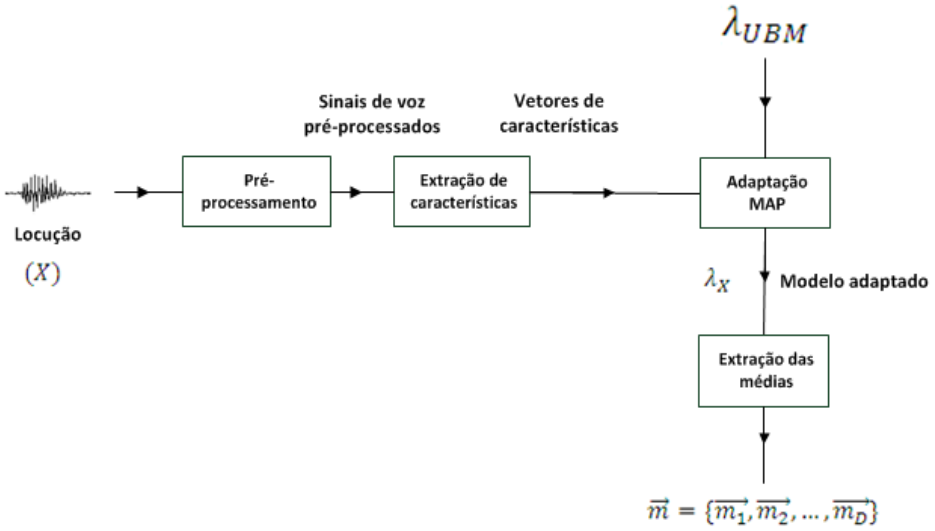

Figura 5.2 Extração dos supervetores de um modelo GMM. A partir de uma locução, um UBM é adaptado, via adaptação MAP. As médias das distribuições do modelo resultante são concatenados para a produção do supervetor.

A partir de um modelo de fundo já treinado, λ_{UBM}, e o conjunto de caracterísitcas extraídas de uma dada locução, X, segue a adaptação dos parâmetros do UBM, via adaptação MAP. Como mencionado anteriormente, apenas as médias são atualizadas. Esse processo resulta em um modelo com o mesmo número de distribuições de λ_{UBM}. Se o número de distribuições é D, extrai-se então as D médias do modelo produzido. Finalmente, esses vetores de média são concatenados de modo a produzir o supervetor referente à locução X.

5.3.2 Combinando SVM e os Supervetores

A Figura 5.3 mostra as fases de treinamento e teste do sistema em questão. Nessa técnica, a primeira coisa a se fazer é treinar um modelo de fundo, λ_{UBM}, utilizando o algoritmo EM. A próxima fase do treinamento consiste em treinar o classificador SVM para cada um dos locutores. Para isso, temos que utilizar dois conjuntos de dados, um que possua as locuções do locutor e outro conjunto que contenha locuções de outros locutores, a fim de produzir supervetores que representem o locutor e os impostores.

Para cada uma das locuções de cada um dos conjuntos, os supervetores correspondentes são extraídos seguindo o procedimento definido na seção anterior. Esses supervetores formarão o conjunto de vetores de treino para o processo de treinamento do classificador SVM. Esse processo de treinamento utiliza os *labels* +1 e -1 para os vetores produzidos utilizando as locuções do locutor e as locuções dos impostores, respectivamente. O processo de otimização é realizado

utilizando uma função de Kernel específica. Os detalhes das funções de Kernel propostas pelos autores são mostrados na próxima seção.

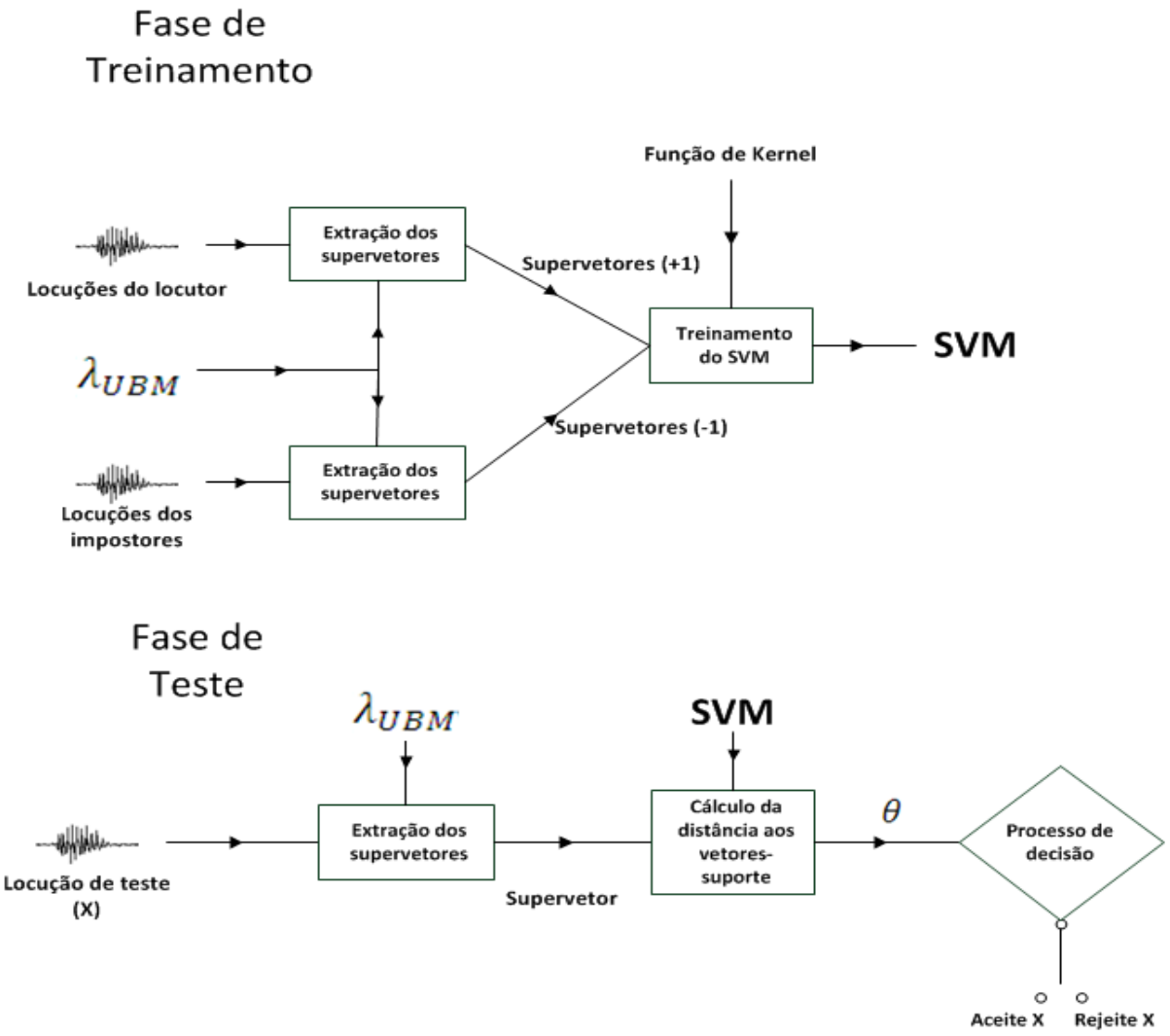

Figura 5.3 Arquitetura do sistema que combina o classificador SVM com os supervetores produzidos pelas locuções.

No processo de teste, um supervetor é produzido utilizando o modelo de fundo e a locução de teste X. Calcula-se então, a distância ponderada do vetor produzido com os vetores suporte definidos no processo de treinamento do SVM. No Apêndice C, esse valor é definido como o valor da função $f(x)$. Esse valor é analisado de modo a fazer a aceitação ou não da locução de entrada. Similarmente às técnicas anteriores, geralmente compara-se esse valor a um limiar e aceita a locução se o valor for superior ao limiar.

5.3.3 Funções de Kernel para supervetores de modelos GMM

A ideia básica de se utilizar uma função de Kernel é comparar duas locuções, loc_A e loc_B, utilizando os modelos adaptados referentes às locuções. Uma vez que apenas as médias são adaptadas, calcular uma distância entre os vetores de médias é uma maneira de calcular a distância entre os modelos e, consequentemente, a distância entre as locuções. Dessa maneira, a extração dos supervetores pode ser vista como um mapeamento entre a locução e um espaço de dimensão mais alta. Esse conceito se encaixa bem à ideia por trás de um Kenel de um SVM. A ideia se resume a comparar duas locuções utilizando uma função de Kernel de forma direta.

Os autores propuseram duas funções de Kernel, que são definidos a seguir.

5.3.3.1 Kernel Linear de um GMM supervector

Suponha que tenhamos duas locuções, loc_A e loc_B, e os dois modelos GMMs, g_A e g_B, produzidos utilizando a adaptação MAP a partir de um modelo de fundo. Uma distância natural entre as duas locuções consiste na divergência de Kullback-Leibler (KL) [HO07]:

$$D(g_A||g_B) = \int_{R^n} g_A(x)\log(\frac{g_A(x)}{g_B(x)})dx. \tag{5.12}$$

Porém, a divergência KL não satisfaz as condições de Mercer e o treinamento de um SVM utilizando esse kernel é problemático.

Ao invés de utilizar a divergência diretamente, os autores consideraram uma aproximação. A ideia é limitar a divergência utilizando a desigualdade da soma logarítimica [Do03]:

$$D(g_A||g_B) \leq \sum_{i=1}^{N} \omega_i D(N(\cdot;m_i^A,\Sigma_i)||N(\cdot;m_i^B,\Sigma_i)), \tag{5.13}$$

onde m^A e m^B são os supervetores das médias adaptadas.

Assumindo as matrizes de covariância como matrizes diagonais, a aproximação em (5.13) pode ser calculada na seguinte forma fechada:

$$d(m^A,m^B) = \frac{1}{2}\sum_{i=1}^{N} \omega_i (m_i^A - m_i^B)^T \Sigma_i^{-1}(m_i^A - m_i^B). \tag{5.14}$$

A desigualdade final é:

$$0 \leq D(g_A||g_B) \leq d(m^A,m^B), \tag{5.15}$$

de modo que se a distância entre m^A e m^B é pequena, então a divergência entre os respectivos GMMs também é.

A medida de distância em (5.14) tem a propriedade de ser simétrica e têm sido utilizada em métodos de clusterização de locutores [BBBG04]. A partir dessa distância, define-se a função de Kernel encontrando o produto interno correspondente (converte-se produtos interno em funções de distância e vice-versa via identidade polar [Con90]).

A função de Kernel resultante é:

$$K(loc_A,loc_B) = \sum_{i=1}^{N} \omega_i (m_i^A)^T \Sigma_i^{-1} m_i^B. \tag{5.16}$$

Uma vez que a função de Kernel em (5.16) é linear, ela satisfaz as condições de Mercer [NJ00] e pode ser utilizada diretamente.

5.3.3.2 Kernel proveniente do produto interno (L^2) de GMMs

Essa segunda função de kernel foi motivada pelas funções de produto interno no espaço das funções. Dados os dois modelos, anteriormente definidos, g_A e g_B, o produto interno padrão no espaço das funções é definido como:

$$K(loc_A, loc_B) = \int_{R^n} g_A(x) g_B(x) dx. \tag{5.17}$$

A forma fechada encontrada para a função de Kernel em (5.17) é:

$$K(loc_A, loc_B) = \sum_{i=1}^{N} \sum_{j=1}^{N} \omega_i \omega_j N(m_i^A - m_j^B; \mathbf{0}; \Sigma_i + \Sigma_j), \tag{5.18}$$

onde $\mathbf{0}$ é o vetor de zeros. Uma vez que todos os termos dos somatórios em (5.18) são fuções de Kernel, então a soma das funções de Kernel também é uma função de Kernel.

Uma aproximação muito conveniente para (5.18) consiste em assumir que médias de distribuições distintas são afastadas entre si, de modo que os termos onde $i \neq j$ são pequenos. Dessa forma, a função de Kernel resultante é:

$$K(loc_A, loc_B) = \sum_{i=1}^{N} \omega_i^2 N(m_i^A - m_i^B; \mathbf{0}; 2\Sigma_i), \tag{5.19}$$

CAPÍTULO 6

Experimentos

Nesse capítulo descrevo os experimentos realizados nesse trabalho. Eles contemplam avaliações das técnicas presentes desde o módulo de pré-processamento e extração de características até a modelagem dos locutores e o teste das locuções. Primeiramente, é feita uma descrição da base de dados utilizada. Depois, passamos para os resultados utilizando as técnicas de pré-processamento e extração de características dos sinais de voz. Além disso, as técnicas baseadas em GMM-UBM. Por fim, uma combinação dessas técnicas é proposta e avaliada.

6.1 Base de dados

A base de dados utilizada nos experimentos deste trabalho é conhecida como **MIT Mobile Device Speaker Verification Corpus (MIT-MDSCV)** [WPH06b]. Ela é um *corpus* para avaliação de sistemas biométricos de voz desenvolvida com objetivo de representar, da melhor forma possível, uma aplicação biométrica de alta mobilidade. Desta forma, todas as locuções dos usuários foram gravadas em aparelhos móveis, como celulares e PDAs, de diversas marcas e modelos.

A fim de captar a variabilidade esperada das condições ambientais e acústicas inerentes ao uso de um dispositivo móvel, foram variadas as condições ambientais e as condições do microfone durante a coleta de dados. Para cada sessão, os dados foram coletados em três locais diferentes (um escritório silencioso, um hall de entrada barulhento e um cruzamento de rua movimentado), bem como com dois microfones diferentes (o microfone interno embutido do dispositivo e um fone de ouvido auricular externo) resultando em seis condições experimentais distintas.

Em cada sessão de coleta de dados, o usuário recitou uma lista de nomes (sabores de sorvetes) que foram exibidos na tela do dispositivo. Usuários inscritos recitaram duas listas de frases, que eram idênticas, diferindo apenas na localização das frases com os sabores dos sorvetes nas listas. A primeira lista de frases foi lida na sessão de coleta de dados inicial dos usuários inscritos, enquanto que a segunda lista de frases foi lida num dia posterior à sessão inicial.

A base possui 48 usuários, sendo 22 do sexo feminino e 26 do sexo masculino. Em cada uma das duas sessões, 54 locuções foram gravadas (18 locuções em cada um dos ambientes, das quais 9 foram gravadas utilizando cada microfone) com uma duração média de 1,8 segundos cada. Além disso, para 40 usuários, 54 locuções de impostores foram gravadas, igualmente distribuídas uniformemente nos 6 cenários possíveis (ambiente e microfones de gravação).

Temos portanto, um total de três sessões: duas com gravações de locutores registrados e

Tabela 6.1 Divisão da base de dados em conjuntos de Treinamento e Teste.

Sessão	Treinamento	Teste
Locutores registrados I	x	
Locutores registrados II		x
Impostores		x

uma com gravações de impostores.

Nos experimentos que se seguem, a primeira sessão foi utilizada para o treinamento dos modelos dos locutores, enquanto que as duas sessões seguintes (uma com locuções dos mesmos locutores e outra com gravações de impostores) foram utilizadas no processo de teste. A Tabela 6.1 resume a divisão da base de dados.

6.2 Análise dos desempenhos dos sistemas

Em sistemas de verificação de locutores, o método mais utilizado para comparar as técnicas consiste na análises das chamadas curvas DET (*Detection EERor Tradeoff*)[MDK$^+$97]. Uma vez que o problema de verificação é um problema de classificação binário, pode-se calcular as taxas de verdadeiro positivo e falso positivo, comuns na área de Aprendizagem de Máquina (AM). Nessa área, cada ponto de operação do sistema gera um par de taxas de verdadeiro e falso positivo (fp,vp). Os pontos de operação extremos equivalem ao sistema operar sobre as taxas (0,1) e (1,0), ou seja, o sistema aceita qualquer entrada ou rejeita qualquer entrada, respectivamente.

Ao se variar os mais variados pontos de operação do sistema, podemos criar uma curva entre esses dois pontos. Ao plotar essa curva, temos a chamada curva ROC, abrangentemente utilizada em AM.

No caso das curvas DET, a curva é bastante similar e também é gerada ao se variar os pontos de operação do sistema entre esses dois pontos extremos. Porém, duas são as taxas de EERo plotadas no gráfico: a taxa de falso positivo e a taxa de falsa rejeição (1-tp). Isso faz com que as curvas possuam uma aparência distinta das curvas ROC, indo do ponto (0,1) até o ponto (1,0). Além disso, é comum a normalização dos eixos pelos seus desvios-padrão correspondentes, ou então, a plotagem na escala logarítimica. Isso faz com que as curvas possuam um aspecto mais linear, fato que os pesquisadores da área acham mais conveniente para o problema em questão.

Nesse trabalho, plotaremos as curvas DET mais naturais, isto é, sem alguma normalização e na escala convencional.

Além disso, nesse tipo de problema, às vezes é interessante observar um ponto de operação específico alcançado pelo sistema. Esse ponto corresponde ao ponto em que as taxas de falso positivo e falsa rejeição possuam o mesmo valor. Essa taxa comum em ambos os erros é chamada de Taxa de Erro Igual (*Equal Error Rate - EER*). Tal taxa também será considerada para a comparação das técnicas nos experimentos que seguem.

Tabela 6.2 EERs geradas pelo sistema para cada um dos conjuntos de características.

Conjunto de características	EER
19 MFCCs	11.85
19 MFCCs + 19 CD	8.06
19 MFCCs + 19 CD2	6.99

6.3 Experimentos com conjuntos de características

Com o objetivo de formar a melhor combinação de características para ser utilizada na análise de desempenho das técnicas,

Os primeiros experimentos consistiram na utilização de diferentes conjuntos de características extraídas dos sinais de voz. Esse experimento tem como objetivo identificar qual conjunto de características leva a um melhor desempenho do sistema como um todo. Três foram os conjuntos experimentados:

- (I): 19 coeficientes MFCCs (19 MFCCs);
- (II): 19 coeficientes MFCCs + 19 coeficientes delta de primeira ordem extraídos dos coeficientes MFCCs (19 MFCCs + 19 CDs);
- (II): 19 coeficientes MFCCs + 19 coeficientes delta de primeira ordem extraídos dos coeficientes MFCCs + 19 coeficientes delta de segunda ordem extraídos dos coeficientes MFCCs (19 MFCCs + 19 CDs + 19 CD2s).

Os coeficientes MFCCs foram extraídos seguindo o método descrito na Seção 4.1. Os coeficientes delta de primeira ordem foram extraídos utilizando a Equação 4.10 com $K = 4$. Por fim, os coeficientes delta de segunda ordem foram extraídos utilizando a mesma equação, porém com valor de $K = 1$.

Segue a metodologia utilizada para o teste de cada um desses conjuntos. Para cada locutor, treinou-se um modelo GMM com 32 distribuições e um modelo UBM com a mesma quantidade de distribuições. Ambos os treinos foram feitos utilizando o algoritmo EM. Para cada locução de teste, calculou-se a razão de verossimilhança utilizando a Equação 2.1. O valor de θ foi variado, de modo a produzir uma curva DET. Analisando as curvas DET geradas para cada um desses conjuntos identificamos o EER.

A Figura 6.3 mostra as curvas DET geradas pelos conjuntos de características. Similarmente, a Tabela 6.3 mostra os valores de EER para cada um dos conjuntos.

A partir da Figura 6.3 e da Tabela 6.3, podemos observar que a utilização de ambos os coeficientes de primeira e segunda ordem melhoram o desempenho do sistema. Após o resultado desse experimento, decidiu-se a utilização do conjunto (III) de características para os experimentos seguintes.

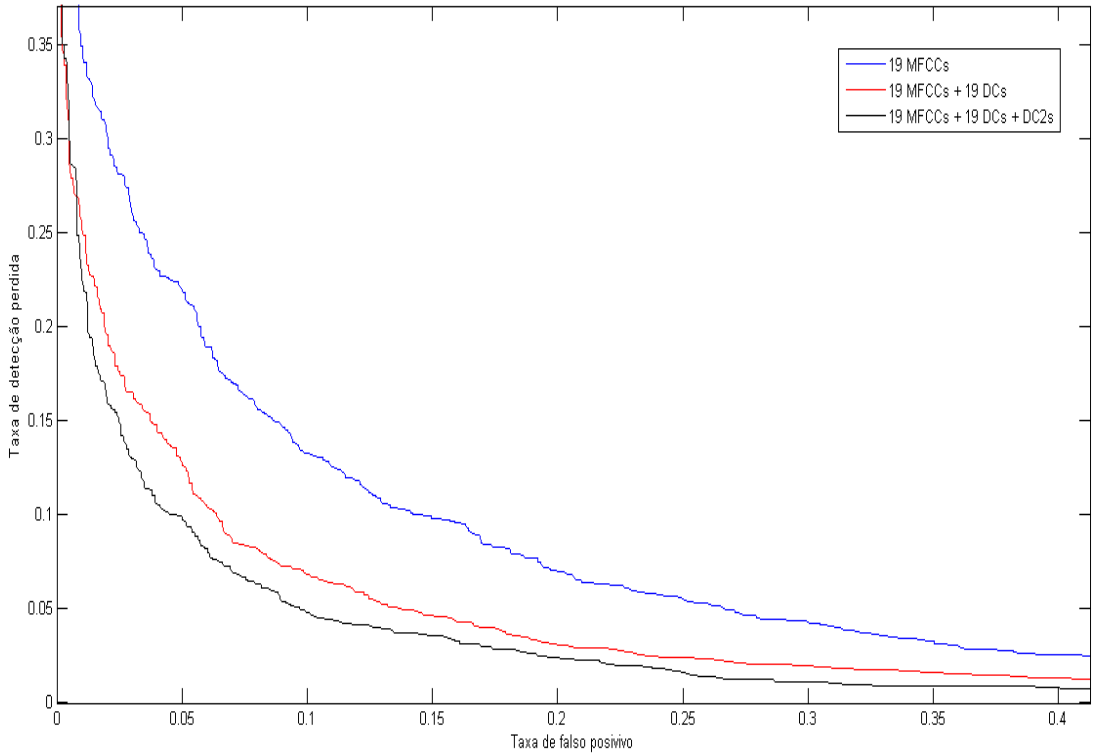

Figura 6.1 Curvas DET geradas pelo sistema GMM-UBM com 32 distribuições estimadas pelo algoritmo EM para cada um dos conjuntos de características.

6.4 Experimentos com técnicas de pré-processamento

Dado o vetor de características, devemos analisar as técnicas de pré-processamento que de fato melhoram o desempenho do sistema. Utilizando o mesmo sistema do experimento anterior, três combinações de técnicas de pré-processamento foram analisadas:

- (IV): Características + CMS;
- (V): Características + filtragem RASTA;
- (VI): Características + CMS + filtragem RASTA.

Ambas as técnicas de Subtração de Média Cepstral (CMS) e filtragem RASTA são descritas nas Seções 4.3.1 e 4.3.2, respectivamente.

As curvas DET geradas pelo sistema para cada uma das três combinações de técnicas são mostradas na Figura 6.2. Além disso os valores de EER correspondentes estão mostrados na Tabela 6.4.

Analisando os resultados, podemos inferir que a utilização da técnica CMS melhora o desempenho do sistema, enquanto que a filtragem RASTA não.

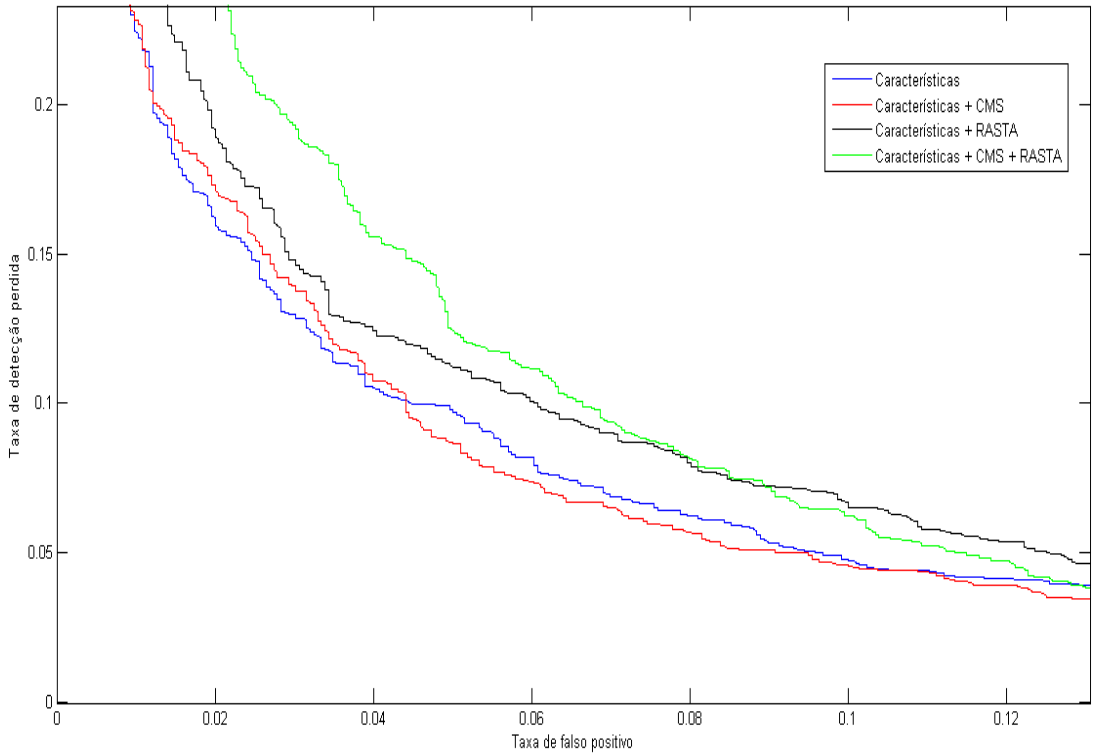

Figura 6.2 Curvas DET geradas pelo sistema para cada uma das combinações de técnicas de pré-processamento.

A partir dos últimos resultados, pudemos identificar o conjunto de vetores extraídos dos sinais de voz de modo que levou a um melhor desempenho do sistema. Portanto, nos próximos experimentos, o conjunto (IV) de características foi extraído dos sinais a fim de produzir uma representação matemática das locuções.

6.5 Experimentos com técnicas de modelagem de locutores

Nessa última fase dos experimentos, dois sistemas baseados em GMM-UBM foram considerados. Ambos os sistemas criam um modelo GMM para cada locutor e outro modelo GMM para o UBM. Ambos os modelos dos locutores e do UBM possuem o mesmo número de distribuições. Além disso, para ambos os sistemas, um único UBM foi treinado utilizando todas as locuções de todos os locutores registrados. Isto é, 2592 (54*48) locuções foram utilizadas para a estimação do UBM, utilizando o algoritmo EM.

A diferença entre os sistemas consiste na forma como os modelos dos locutores foram estimados. No primeiro sistema, todos os modelos foram treinados de forma independente, utilizando o algoritmo EM e as 54 locuções do conjunto de treinamento, conforme descrito na

Tabela 6.3 EERs geradas pelo sistema para cada uma das combinações de técnicas de pré-processamento.

Combinação	EER
Características	6.99
Características + CMS	6.71
Características + RASTA	8.02
Características + CMS + RASTA	8.08

Seção 2.4.1. A partir de agora, referenciaremos esse sistema como GMM-UBM. Já no segundo sistema a técnica empregada consiste no emprego dos modelos de misturas Gaussianas adaptativas, descrito na Seção 5.2. Nessa técnica, os modelos dos locutores são gerados adaptando-se os parâmetros do UBM, utilizando as locuções do conjunto de treinamento. Iremos referenciar esse sistema como Adap-GMM-UBM.

O número de distribuições dos modelos foram variados em potências de 2, indo desde 16 distribuições até 256.

Em ambos os sistemas, o teste de uma locução foi realizada pelo Teste da Razão de Verossimilhança, descrito na Seção 2.3. Além disso, variamos o valor de θ a fim de produzir uma curva DET para cada experimento utilizando cada uma das técnicas.

Como mencionado na Seção 6.1, as locuções presentes na base de dados provém de três ambientes distintos:

- Ambiente 1: escritório silencioso;
- Ambiente 2: *hall* de entrada de um prédio;
- Ambiente 3: cruzamento entre ruas movimentadas.

Para obter resultados mais precisos quanto à variabilidade dos ambientes, realizamos testes que contemplam todas as combinações de treinamento e teste. Ou seja, temos, por exemplo, o resultado da técnica quando o sistema é treinado com amostras geradas no Ambiente 1 (pouco ruído) e testada no Ambiente 3 (muito ruído). Desse modo, podemos analisar a robustez dos sistemas quanto à variabilidade do sistema. Além disso, para se ter uma ideia geral do desempenho, também foram conduzidos experimentos onde o sistema é treinado com amostras de todos os ambientes e testado com amostras de todos os ambientes.

Seguem os resultados das técnicas.

6.6 Resultados da técnica GMM-UBM

Nos experimentos conduzidos por essa técnica, não foi possível a estimação dos modelos quando mais de 64 distribuições eram utilizadas. Nesses casos, as matrizes de covariância das GMMs não podiam ser utilizadas pois possuíam variâncias muito pequenas. Isso geralmente ocorre quando mais distribuições que o necessário são utilizadas para cobrir os dados de treinamento. Isso faz com que algumas distribuições sejam capazes de cobrir apenas algumas

Tabela 6.4 EERs geradas pelo sistema GMM-UBM (16 distribuições) para cada um dos cenários de treinamento/teste são descritos.

EER(%)	**Ambiente 1**	**Ambiente 2**	**Ambiente 3**	**Todos os ambientes**
Ambiente 1	13.76	24.31	25.69	20.98
Ambiente 2	15.72	12.37	19.88	16.81
Ambiente 3	20.96	21.27	14.58	19.44
Todos os ambientes	7.89	7.95	8.72	8.02

Tabela 6.5 EERs geradas pelo sistema GMM-UBM (32 distribuições) para cada um dos cenários de treinamento/teste são descritos.

EER(%)	**Ambiente 1**	**Ambiente 2**	**Ambiente 3**	**Todos os ambientes**
Ambiente 1	13.89	25.25	26.26	22.05
Ambiente 2	16.92	12.75	20.83	17.91
Ambiente 3	22.35	19.19	16.92	19.68
Todos os ambientes	6.38	5.56	8.15	6.57

amostras pontuais, perdendo o poder de generalização do modelo. Portanto, para essa técnica, mostraremos os resultados para quando 16, 32 e 64 distribuições foram utilizadas nos modelos.

As Figuras D.1-D.3 mostram as curvas DET geradas pelo sistema quando os modelos possuem 16, 32 e 64 distribuições, respectivamente, e estão situadas no Apêndice D. Em cada uma delas, pode-se analisar o desempenho do sistema em todos os casos combinando os ambientes de treinamento e teste. Além disso nas Tabelas 6.4-6.6 estão as taxas de EER para cada um dos testes realizados.

Podemos observar pelas figuras e tabelas que o melhor desempenho do sistema é alcançado quando 32 distribuições são utilizadas nos modelos. Aqui, considera-se o desempenho no cenário onde o sistema é treinado por amostras de todos os ambientes e testado com amostras também de todos os ambientes.

Além disso, podemos observar que quando o sistema é treinado em um determinado ambiente, o melhor desempenho é alcançado quando o sistema é testado no mesmo ambiente. Ou seja, quando há variabilidade do ambiente onde se encontra o sistema, o desempenho dele é degradado. Podemos observar que esse fato é intrínseco a todas as configurações.

Tabela 6.6 EERs geradas pelo sistema GMM-UBM (64 distribuições) para cada um dos cenários de treinamento/teste são descritos.

EER(%)	**Ambiente 1**	**Ambiente 2**	**Ambiente 3**	**Todos os ambientes**
Ambiente 1	17.36	27.21	29.86	24.81
Ambiente 2	19.44	16.23	22.22	19.91
Ambiente 3	26.83	24.43	19.83	24.62
Todos os ambientes	6.50	7.00	8.90	7.55

Tabela 6.7 EERs geradas pelo sistema Adap-GMM-UBM (16 distribuições) para cada um dos cenários de treinamento/teste são descritos.

EER(%)	**Ambiente 1**	**Ambiente 2**	**Ambiente 3**	**Todos os ambientes**
Ambiente 1	12.50	27.65	27.21	22.77
Ambiente 2	15.53	12.63	18.56	15.61
Ambiente 3	20.14	20.71	13.89	18.56
Todos os ambientes	9.16	10.86	12.06	11.01

Tabela 6.8 EERs geradas pelo sistema Adap-GMM-UBM (32 distribuições) para cada um dos cenários de treinamento/teste são descritos.

EER(%)	**Ambiente 1**	**Ambiente 2**	**Ambiente 3**	**Todos os ambientes**
Ambiente 1	9.98	25.38	24.43	20.75
Ambiente 2	14.02	9.47	15.15	13.13
Ambiente 3	18.31	16.54	13.63	16.01
Todos os ambientes	5.12	7.64	8.77	7.26

6.7 Resultados da técnica Adap-GMM-UBM

Como mencionado anteriormente, variamos o número de distribuições utilizadas nos modelos com números potência de 2. Diferente da técnica anterior, nenhum problema foi encontrado ao estimas os modelos com 16, 32, 64, 128 e 256 distribuições. Isso ocorreu porque nessa técnica, o único modelo a ser estimado pelo algoritmo EM é o UBM, que por sua vez possui um grande número de locuções. Portanto, aumentar o número de distribuições não acarretaria em uma má estimação dos parâmetros, como no sistema anterior. Porém, pudemos observar em experimentos preliminares que quando aumentamos o número de distribuições para mais de 512, o mesmo problema do sistema anterior pode ocorrer.

As Figuras E.1-E.5 mostram as curvas DET geradas pelo sistema em cada um dos cenários de experimento quando 16, 32, 64, 128 e 256 distribuições são utilizadas nos modelos, respectivamente, e estão situadas no Apêndice E. As taxas de EER correspondentes são mostradas nas Tabelas 6.7-6.11.

Podemos observar que o melhor desempenho é alcançado quando 128 distribuições são utilizadas nos modelos. Além disso, similarmente ao sistema GMM-UBM, quando o sistema é treinado em um ambiente específico, há degradação do desempenho quando o sistema é testado

Tabela 6.9 EERs geradas pelo sistema Adap-GMM-UBM (64 distribuições) para cada um dos cenários de treinamento/teste são descritos.

EER(%)	**Ambiente 1**	**Ambiente 2**	**Ambiente 3**	**Todos os ambientes**
Ambiente 1	10.16	25	24.18	19.74
Ambiente 2	14.97	9.85	14.33	13.53
Ambiente 3	15.97	14.5	14.90	15.21
Todos os ambientes	3.85	7.00	7.1991	6.00

Tabela 6.10 EERs geradas pelo sistema Adap-GMM-UBM (128 distribuições) para cada um dos cenários de treinamento/teste são descritos.

EER(%)	**Ambiente 1**	**Ambiente 2**	**Ambiente 3**	**Todos os ambientes**
Ambiente 1	10.86	25.95	23.30	20.47
Ambiente 2	17.62	10.29	15.41	15.05
Ambiente 3	17.23	17.36	18.18	17.72
Todos os ambientes	3.34	5.56	6.00	4.97

Tabela 6.11 EERs geradas pelo sistema Adap-GMM-UBM (256 distribuições) para cada um dos cenários de treinamento/teste são descritos.

EER(%)	**Ambiente 1**	**Ambiente 2**	**Ambiente 3**	**Todos os ambientes**
Ambiente 1	12.50	28.60	26.26	22.45
Ambiente 2	20.90	13.89	19.70	17.59
Ambiente 3	23.61	19.32	22.35	21.72
Todos os ambientes	3.60	4.86	6.76	5.01

em algum outro ambiente com diferentes níveis de ruído.

CAPÍTULO 7

Conclusão

Este trabalho apresentou uma descrição do estado da arte em sistemas de verificação de locutores independentes de texto. Foram analisadas técnicas que vão desde o pré-processamento do sinal de voz até a modelagem dos locutores e do teste final de uma locução.

Na parte de pré-processamento, pudemos analisar os chamadas detectores de atividade de voz. Duas técnicas foram descritas, uma baseada em energia e taxa de cruzamento pelo zero e outra baseada na análise biespectral. Quanto às características extraídas dos sinais de voz, descrevemos os chamados Coeficientes Cepstrais da Escala MEL (MFCC), além de transformações realizadas com coeficientes, entre elas a Subtração de Média Cepstral, a filtragem RASTA e os coeficientes delta de primeira e segunda ordem. A partir de 2000, as técnicas de modelagem de locutores mais utilizadas foram as estatísticas, em especial os modelos de misturas Gaussianas (GMMs) e a utilização do chamada modelo de fundo (UBM). Aqui, pudemos descrever as últimas técnicas propostas para a modelagem dos locutores, que são baseadas em modelos GMM para a modelagem dos locutores e do UBM. Além disso, duas outras técnicas foram abordadas, uma que utiliza o conceito de GMMs adaptativas e outra que combina as GMMs adaptativas com Máquinas de Vetores Suporte (SVMs). As técnicas descritas foram implementadas e testadas em uma base de dados construída para resolver esse problema específico. A base MIT-MDSCV foi gravada a partir de dispositivos móveis e em vários ambientes, desde escritórios silenciosos até avenidas movimentadas. Os experimentos e resultados apresentados nesse trabalho identificam o melhor conjunto de características e de técnicas de pré-processamento que leva a um melhor desempenho dos sistemas. Além disso, os experimentos foram conduzidos de modo a ser possível a análise das mais diversas combinações de ambientes utilizados para o treinamento e teste dos sistemas. Observou-se que os desempenhos dos sistemas são degradados quando o sistema opera em ambientes diferentes do ambiente onde o sistema foi treinado. Esse resultado mostra que muitas das técnicas do estado da arte ainda não conseguem uma boa robustez nesse sentido. Sendo necessário, portanto, algum tipo de trabalho direcionado nesse sentido. Os estudos realizados sobre o estado da arte proporcionaram uma boa revisão da literatura e foram essenciais para o aprofundamento do aluno nesse problema.

APÊNDICE A

Janela de Hamming

Dado um sinal $s[n], n = 1, ..., N$, o sinal após a aplicação da janela da Hamming é $s[n] * w[n]$, onde $*$ é o operador de convolução e $w[n]$ é a janela de Hamming definida por:

$$w[n]_\alpha = (1 - \alpha) - \alpha cos[\frac{2\pi n}{N-1}] \tag{A.1}$$

onde $0 \leq n \leq N-1$. O valor de α determina a forma da curva da janela, como mostra a Figura A.1, e N é o tamanho da janela.

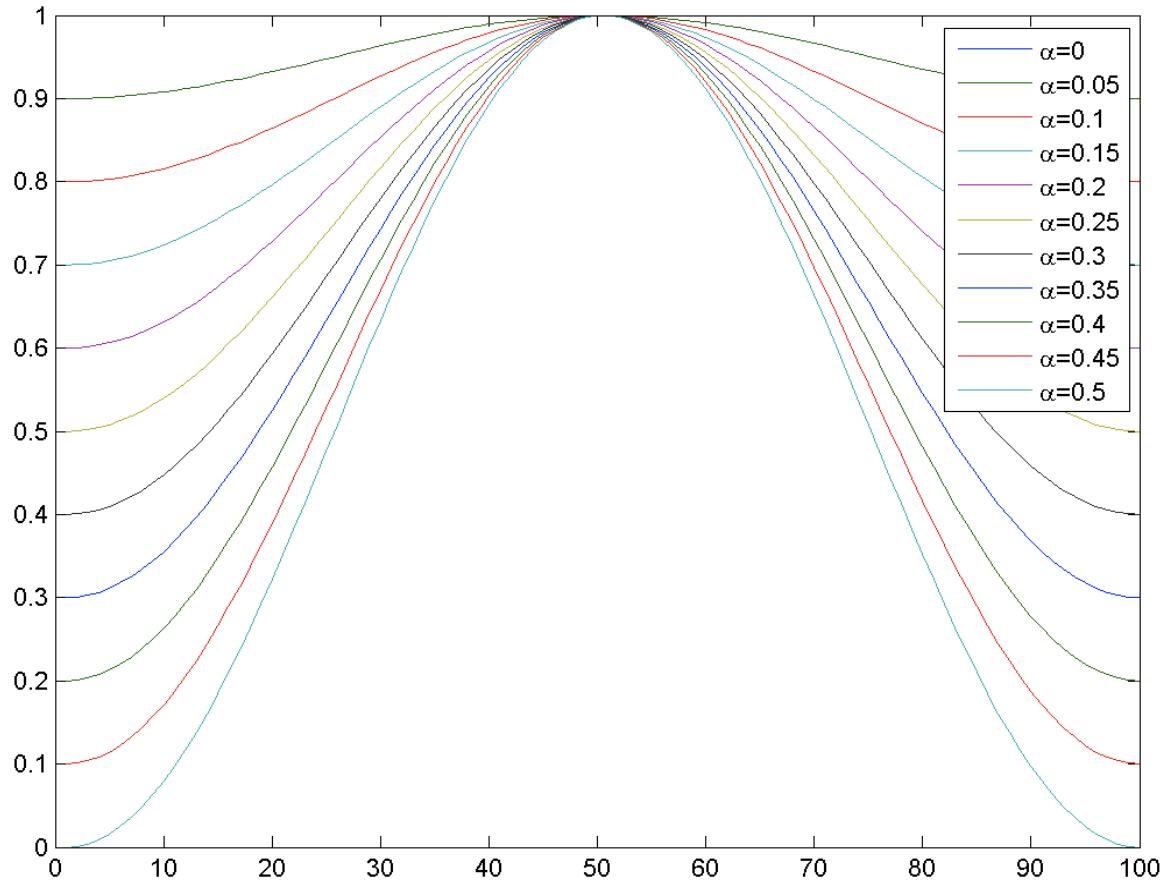

Figura A.1 Variação da curva de uma janela de Hamming de acordo com o parâmetro de variação (α).

APÊNDICE B

Algoritmo de Maximização de Expectativa

O algoritmo de Maximização de Expectativa (*Expectation-Maximization - EM*) foi formalizado e intitulado em um artigo clássico de 1977 por Arthur Dempster, Nan Laird e Donald Rubin [DLR77]. Apesar disso, os autores deixaram claro que o algoritmo já havia sido utilizado antes por outros autores.

O algoritmo EM é largamente utilizado para estimar parâmetros de funções de distribuição de probabilidade que maximizam a verossimilhança de um certo conjunto de dados. Geralmente é utilizado quando as equações não podem ser solucionadas de forma direta [Moo96]. De maneira prática, busca-se encontrar um conjunto de parâmetros, λ, utilizando um conjunto de amostras observadas, X, de modo que a verossimilhança de λ dado X seja a maior possível. Por essa razão, é comum dizer que o algoritmo EM é um método de estimação do máximo da verossimilhança (*Maximum Likelihood Estimation - ML Estimation*).

O algoritmo EM funciona de forma iterativa. A cada iteração ele utiliza o modelo atual, λ, e o conjunto de amostras, X, para produzir um novo modelo λ' de modo que a verossimilhança de λ' seja maior que a de λ. Esse processo é repetido até que a verossimilhança seja estabilizada.

A ideia por trás do algoritmo é envolver o conjunto de amostras observadas, X, e o conjunto de parâmetros desconhecidos, λ, com as chamadas variáveis latentes, que são encaradas como amostras que faltam ao conjunto X. A cada variável conhecida de X é atribuída uma variável latente, que possui a informação de qual distribuição a variável conhecida provém. Sabemos que λ possui os parâmetros de cada uma das distribuições que compõem a mistura de distribuições final. Assume-se que o número de distribuições é conhecido *a priori*. Para cada amostra de X associa-se uma variável latente que indica a qual distribuição a amostra pertence.

Encontrar a solução que maximiza a verossimilhança do modelo requer tomar as derivadas da função de verossimilhança com respeito às variáveis desconhecidas, isto é, os parâmetros das distribuições e as variáveis latentes, e simultaneamente, resolver as equações produzidas.

A Figura B.1 mostra uma visão geral do algoritmo. Dado um conjunto de dados observados, X, um conjunto de variáveis latentes ou valores desconhecidos, Z, e um conjunto de parâmetros desconhecidos, θ, associados a uma função de verossimilhança $L(\theta;X,Z) = p(X,Z|\theta)$, a estimativa do máximo de verossimilhança (*Maximum Likelihood Estimate - MLE*) dos parâmetros desconhecidos é dado pelo verossimilhança marginal dos dados observados:

$$L(\theta;X) = p(X|\theta) = \sum_{Z} p(X,Z|\theta). \tag{B.1}$$

O algoritmo procura encontrar o MLE iterativamente aplicando os seguintes dois passos:

- **Passo de expectativa (Passo E)**: calcular o valor esperado do logarítimo da verossimilhança com respeito à distribuição condicional de Z e X com respeito ao valor corrente

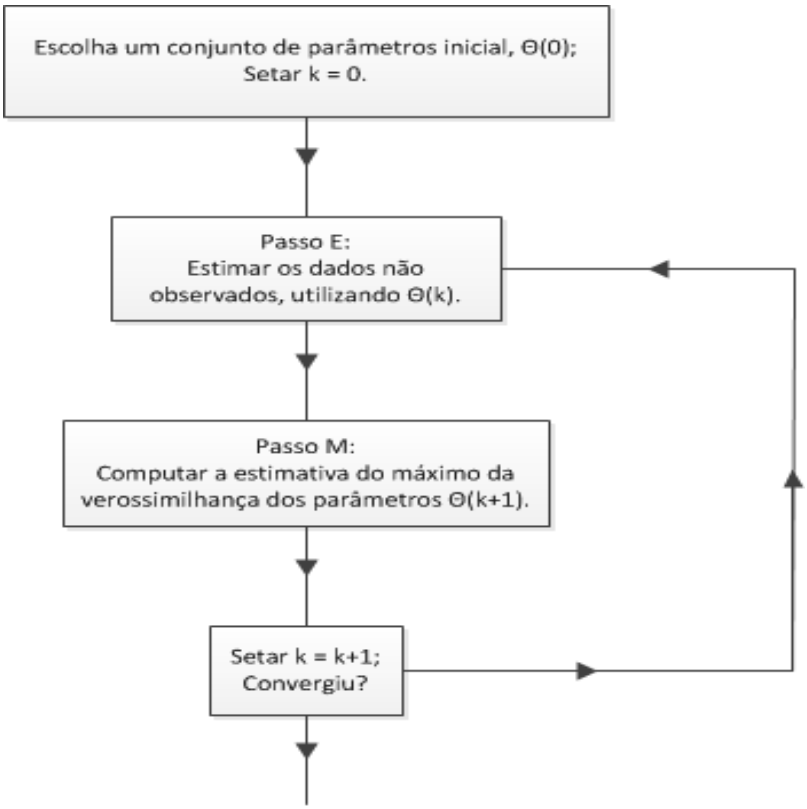

Figura B.1 Visão geral do algoritmo EM. Os passos E e M são alternados até que a estimativa dos parâmetros convirja.

dos parâmetros $\theta^{(t)}$:

$$Q(\theta|\theta^{(t)}) = E_{Z|X,\theta^{(t)}}[\log L(\theta;X,Z)]. \tag{B.2}$$

- **Passo de maximização (Passo M)**: encontrar os parâmetros que maximizam $Q(\theta|\theta^{(t)}$:

$$\theta^{(t+1)} = \underset{\theta}{\arg\max} \quad Q(\theta|\theta^{(t)}) \tag{B.3}$$

B.1 EM aplicado a um Modelo de Misturas Gaussianas

Dado um conjunto de amostras conhecidas, X, deseja-se estimar os parâmetros μ_i, Σ_i e $P(\omega_i)$ de cada uma das distribuições que compõem o modelo final λ. Aqui, podemos enxegar as variáveis ω_i como as variáveis latentes desconhecidas. E stimar $P(\omega_i)$ significa, então, estimar o peso final da distribuição i.

Suponha $X = \{x_1, ..., x_T\}$ e que tenhamos C distribuições no modelo, cujos parâmetros são referenciados como θ, então:

$$p(x_k|\theta) = \sum_{j=1}^{C} p(x_k|\omega_j, \theta_j)P(\omega_j). \tag{B.4}$$

Por definição, a verossimilhança do modelo, com relação às N amostras de X é:

$$p(X|\theta) = \prod_{k=1}^{N} p(x_k|\theta), \tag{B.5}$$

e a estimativa do máximo da verossimilhança, $\theta^{'}$, é o valor de θ que maximiza $p(X|\theta)$.

Se assumirmos que $p(X|\theta)$ é uma função diferenciável em θ, então podemos derivar as condições necessárias para $\theta^{'}$. Definimos então L como sendo o logarítimo da verossimilhança e $\nabla_{\theta_i} L$ o gradiente de L com respeito a θ_i, então:

$$L = \sum_{k=1}^{N} \log p(x_k|\theta) \tag{B.6}$$

e

$$\nabla_{\theta_i} L = \sum_{k=1}^{N} \frac{1}{p(x_k|\theta)} \nabla_{\theta_i} [\sum_{j=1}^{C} p(x_k|\omega_j, \theta_j) P(\omega_j)]. \tag{B.7}$$

Se assumirmos que os parâmetros de duas distribuições diferentes, θ_i e θ_j são independentes e se introduzimos a probabilidade *a posteriori*,

$$P(\omega_i|x_k, \theta) = \frac{p(x_k|\omega_i, \theta_i) P(\omega_i)}{p(x_k|\theta)}, \tag{B.8}$$

podemos observar que o gradiente do logarítimo da verossimilhança com respeito aos parâmetros pode ser escrito como:

$$\nabla_{\theta_i} L = \sum_{k=1}^{N} P(\omega_i|x_k, \theta) \nabla_{\theta_i} [\log p(x_k|\omega_i, \theta_i)]. \tag{B.9}$$

Uma vez que o gradiente deve desaparecer em θ_i que maximiza L, a estimativa do máximo de verossimilhança, $\theta_i^{'}$, deve satisfazer a condição:

$$\sum_{k=1}^{N} P(\omega_i|x_k, \theta^{'}) \nabla_{\theta_i} [\log p(x_k|\omega_i, \theta_i^{'})] = 0, \tag{B.10}$$

$$i = 1, ..., C. \tag{B.11}$$

As regras de atualização dos parâmetros do modelo, em cada etapa do passo de maximização, são dadas pela solução da equação acima:

$$P(\omega_i)^{'} = \frac{1}{N} \sum_{k=1}^{N} \widehat{P}(\omega_i|x_k, \theta), \tag{B.12}$$

$$\mu_i^{'} = \frac{\sum_{k=1}^{N} \widehat{P}(\omega_i|x_k, \theta) x_k}{\sum_{k=1}^{N} \widehat{P}(\omega_i|x_k, \theta)}, \tag{B.13}$$

$$\Sigma_i^{'} = \frac{\sum_{k=1}^{N} \widehat{P}(\omega_i|x_k, \theta)(x_k - \mu_i)(x_k - \mu_i)^T}{\sum_{k=1}^{N} \widehat{P}(\omega_i|x_k, \theta)}, \tag{B.14}$$

onde

$$\widehat{P}(\omega_i|x_k, \theta) = \frac{p(x_k|\omega_i, \theta_i) P(\omega_i)}{\sum_{j=1}^{C} p(x_k|\omega_j, \theta_j) P(\omega_j)}. \tag{B.15}$$

APÊNDICE C

Máquinas de vetores suporte

Máquinas de vetores suporte (*Support Vector Machines - SVM*) [NJ00] constituem uma técnica de aprendizado que vem recebendo cada vez mais atenção da comunidade de Aprendizagem de Máquina (AM). As SVMs são baseadas na teoria do aprendizado estatístico, que estabelece princípios a serem seguidos com o objetivo de obter classificadores com grande poder de generalização.

Um SVM é um classificador binário construído a partir de somas de uma função de kernel $K(\cdot,\cdot)$, de modo que:

$$f(x) = \sum_{i=1}^{L} \alpha_i t_i K(x, x_i) + d, \tag{C.1}$$

onde t_i são as saídas ideais, d é uma constante produzida pelo processo de aprendizagem e $\sum_{i=1}^{L} \alpha_i t_i = 0, \alpha_i > 0$. Os vetores x_i são os chamados vetores suporte e são definidos a partir do conjunto de treinamento, através de um processo de otimização. As saídas ideias constituem de -1 ou 1, dependendo da classe a qual pertence o vetor suporte associada à saída. O processo de classificação ocorre ao comparar o valor de $f(x)$ a um limiar.

A escolha dos vetores suporte e da constante constrói hiperplanos de decisão. Essa escolha é realizada através de um algoritmo de otimização da separabilidade dos dados de treinamento.

A função de kernel $K(\cdot,\cdot)$ é restrita a possuir certas condições, as chamadas condições de Mercer, de modo que pode ser escrita como:

$$K(x,y) = b(x)^T b(y), \tag{C.2}$$

onde $b(x)$ é um mapeamento entre o espaço de entrada para um espaço possivelmente de dimensão infinita. As condições de Mercer garantem o conceito apropriado de margem [NJ00] e a otimização do SVM é bem definida.

A condição de otimzação é baseada no conceito da margem ótima. Para um conjunto de dados separáveis, o sistema gera um hiperplano no espaço de alta dimensão de modo que possua a margem ótima. As amostras do conjunto de treino que se encontram nas fronteiras dos hiperplanos são definidas como os vetores suporte. O foco do processo de treinamento do SVM é, então, modelar as fronteiras entre as duas classes que separam os dados de treino.

APÊNDICE D

Curvas DET geradas nos experimentos com o sistema GMM-UBM

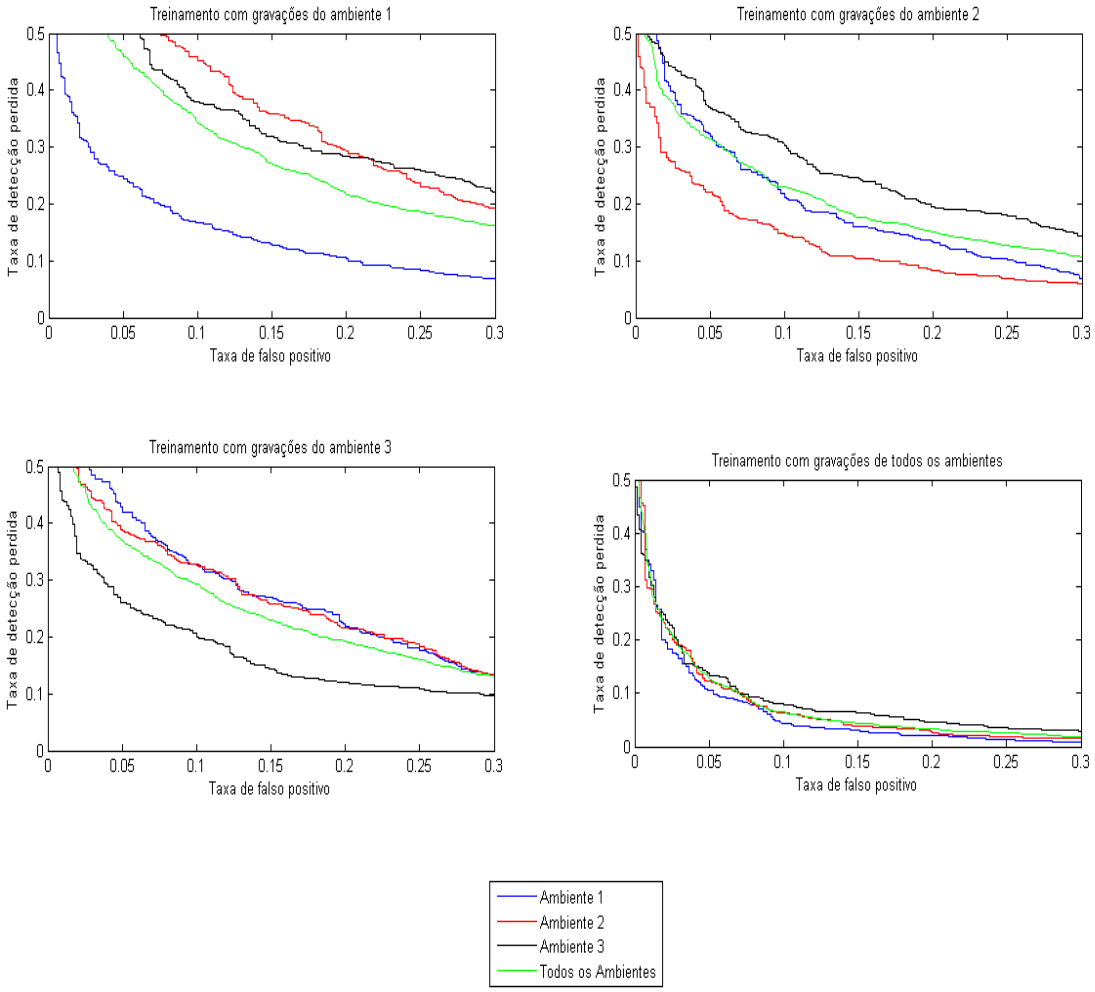

Figura D.1 Curvas DET geradas pelo sistema GMM-UBM para modelos GMM com 16 distribuições.

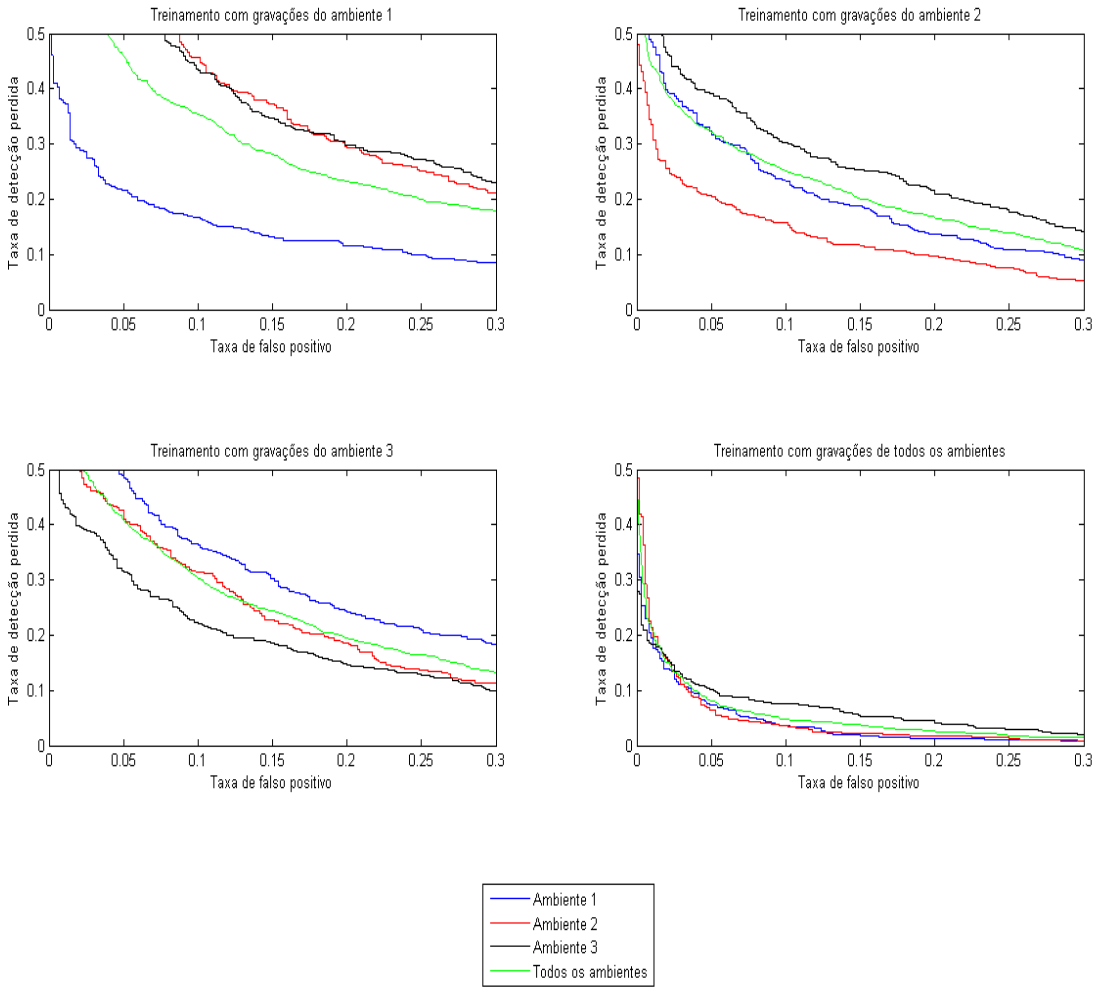

Figura D.2 Curvas DET geradas pelo sistema GMM-UBM para modelos GMM com 32 distribuições.

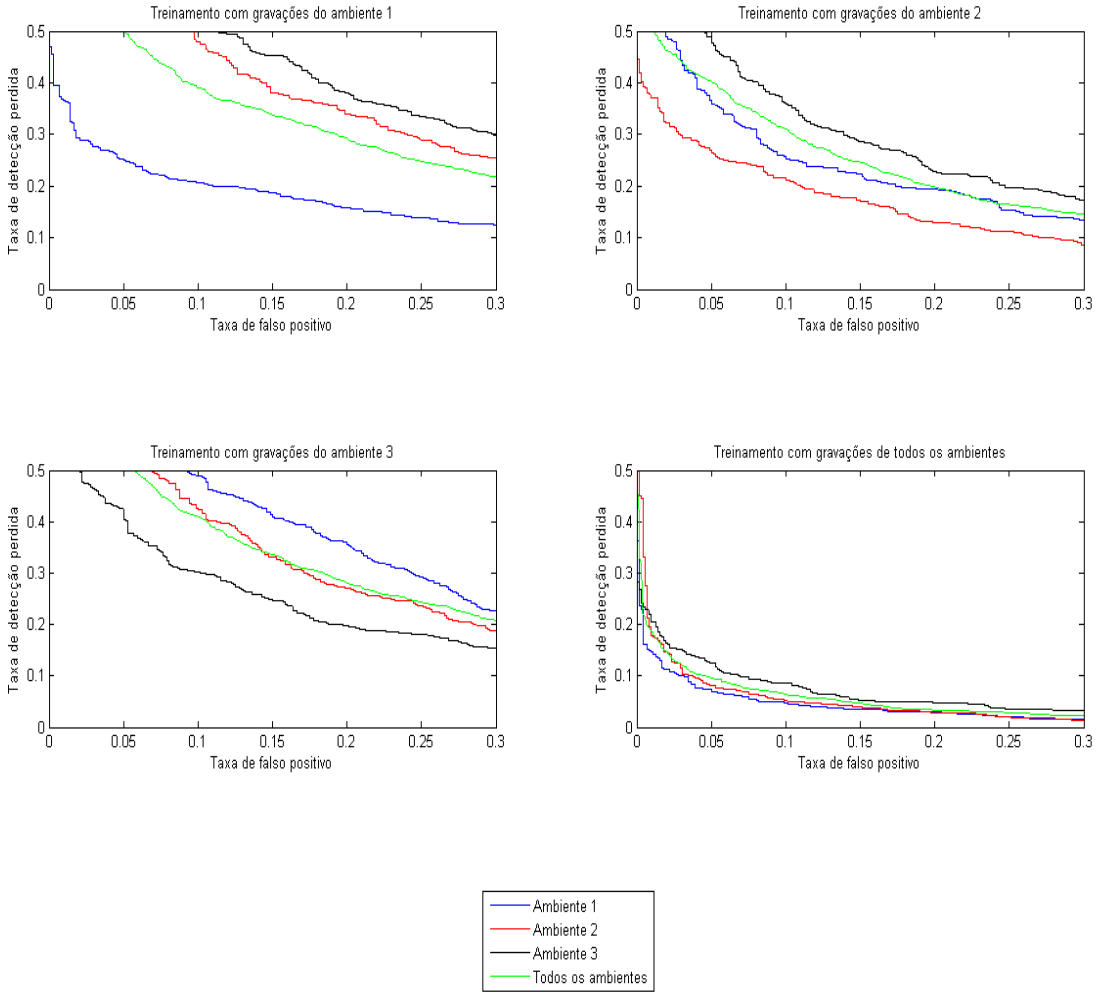

Figura D.3 Curvas DET geradas pelo sistema GMM-UBM para modelos GMM com 64 distribuições.

APÊNDICE E

Curvas DET geradas nos experimentos com o sistema Adap-GMM-UBM

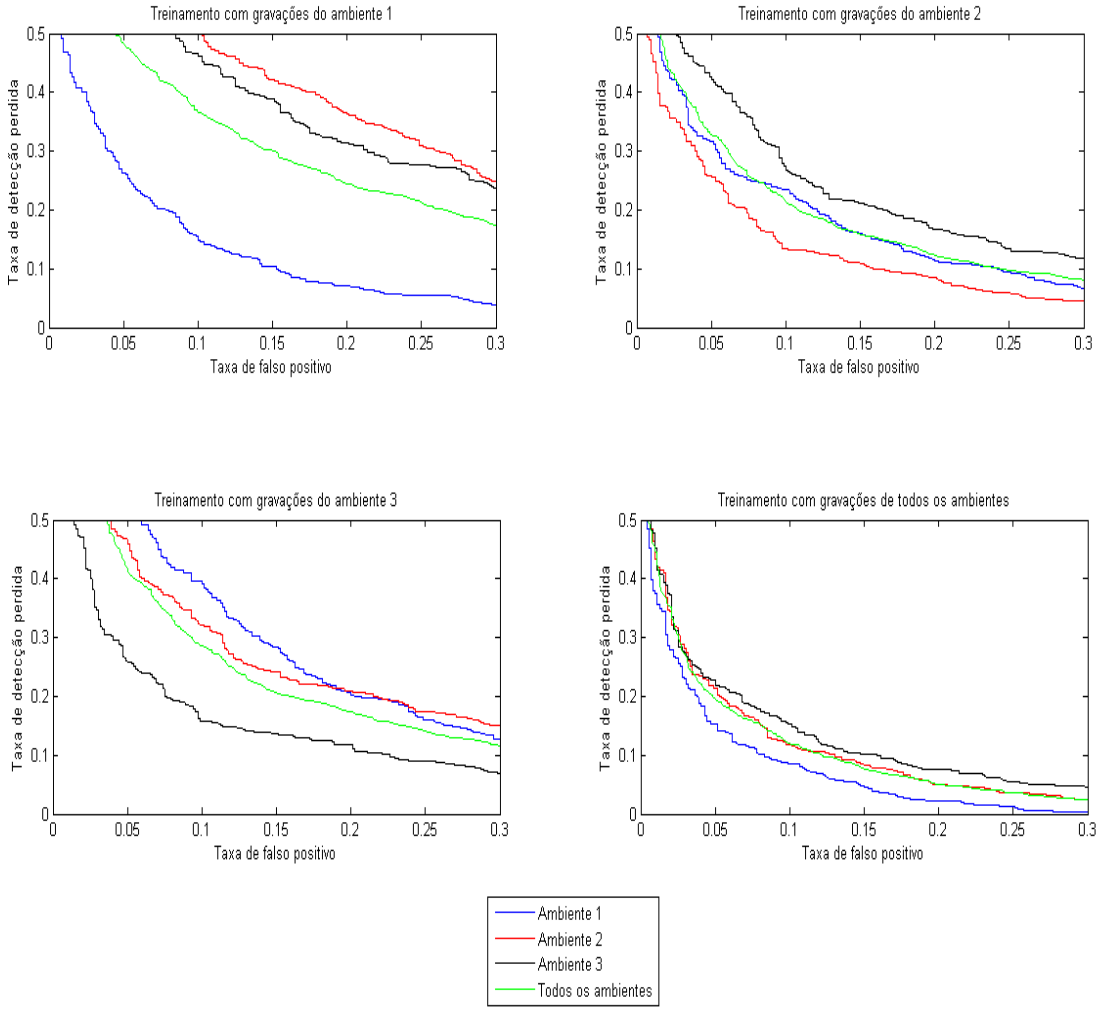

Figura E.1 Curvas DET geradas pelo sistema Adap-GMM-UBM para modelos GMM com 16 distribuições.

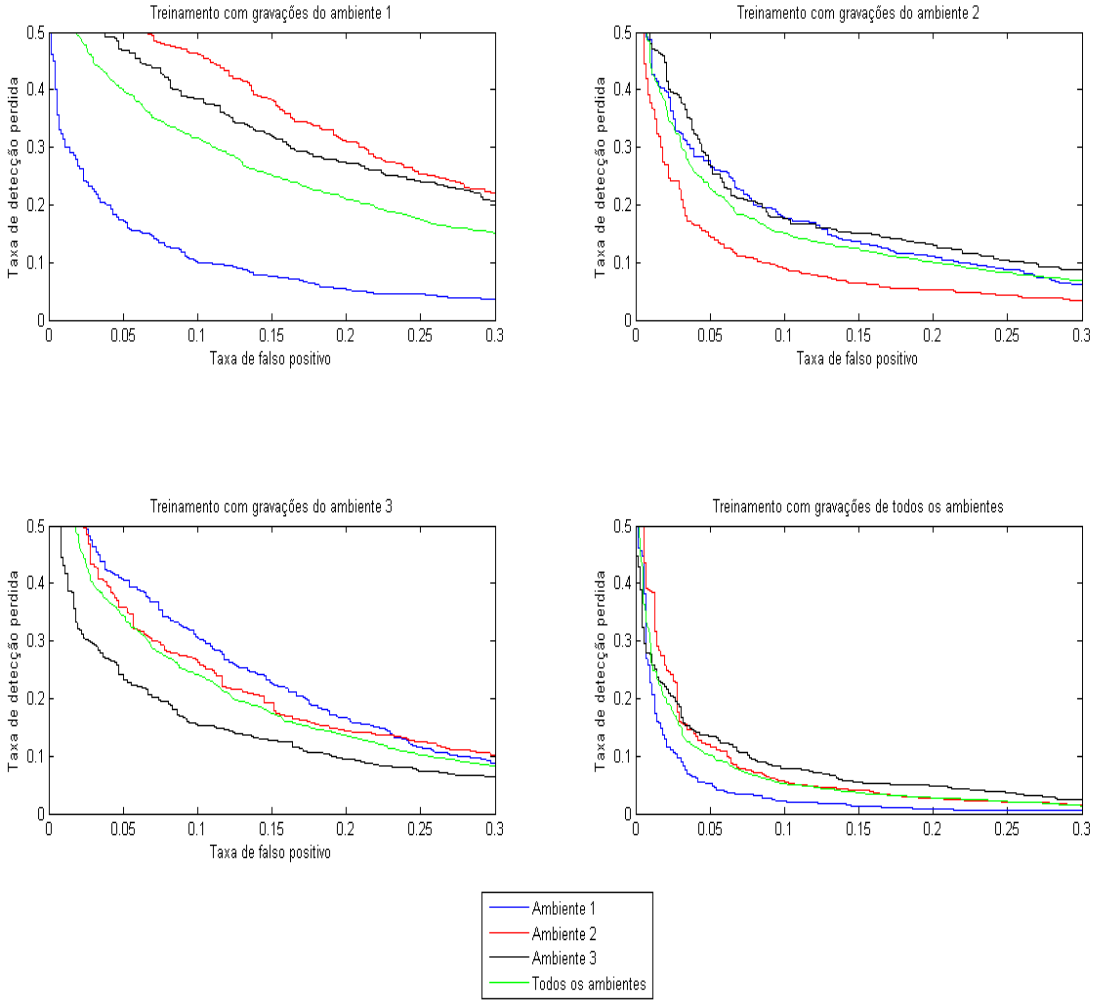

Figura E.2 Curvas DET geradas pelo sistema Adap-GMM-UBM para modelos GMM com 32 distribuições.

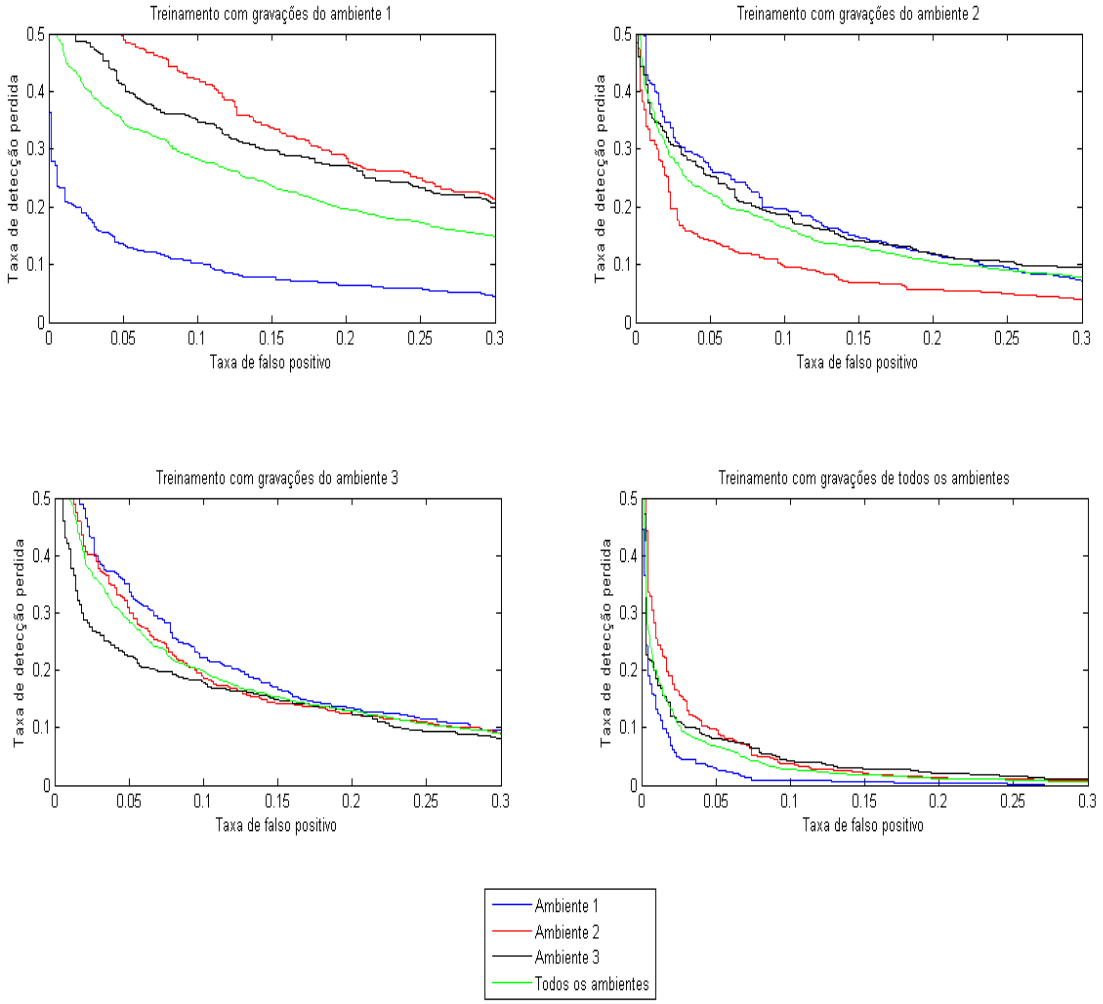

Figura E.3 Curvas DET geradas pelo sistema Adap-GMM-UBM para modelos GMM com 64 distribuições.

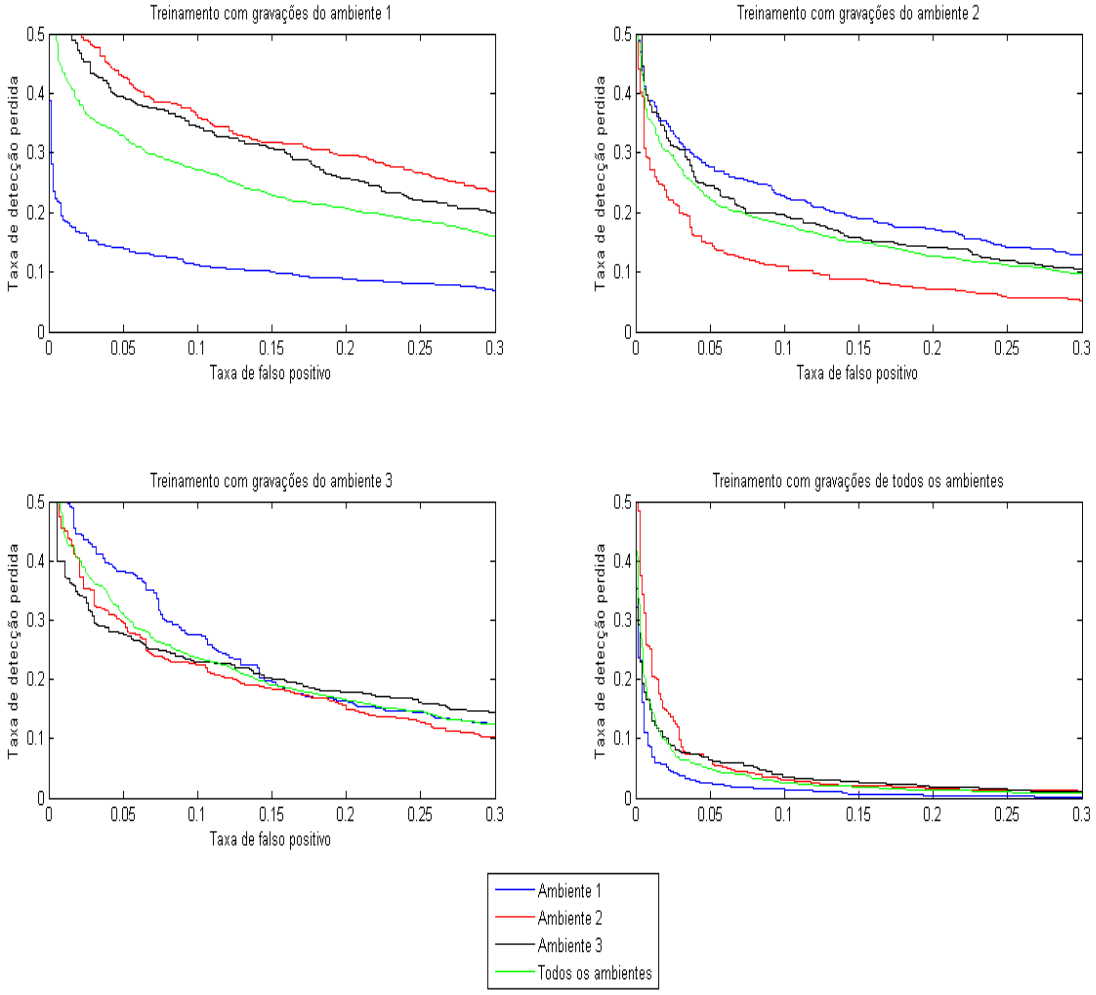

Figura E.4 Curvas DET geradas pelo sistema Adap-GMM-UBM para modelos GMM com 128 distribuições.

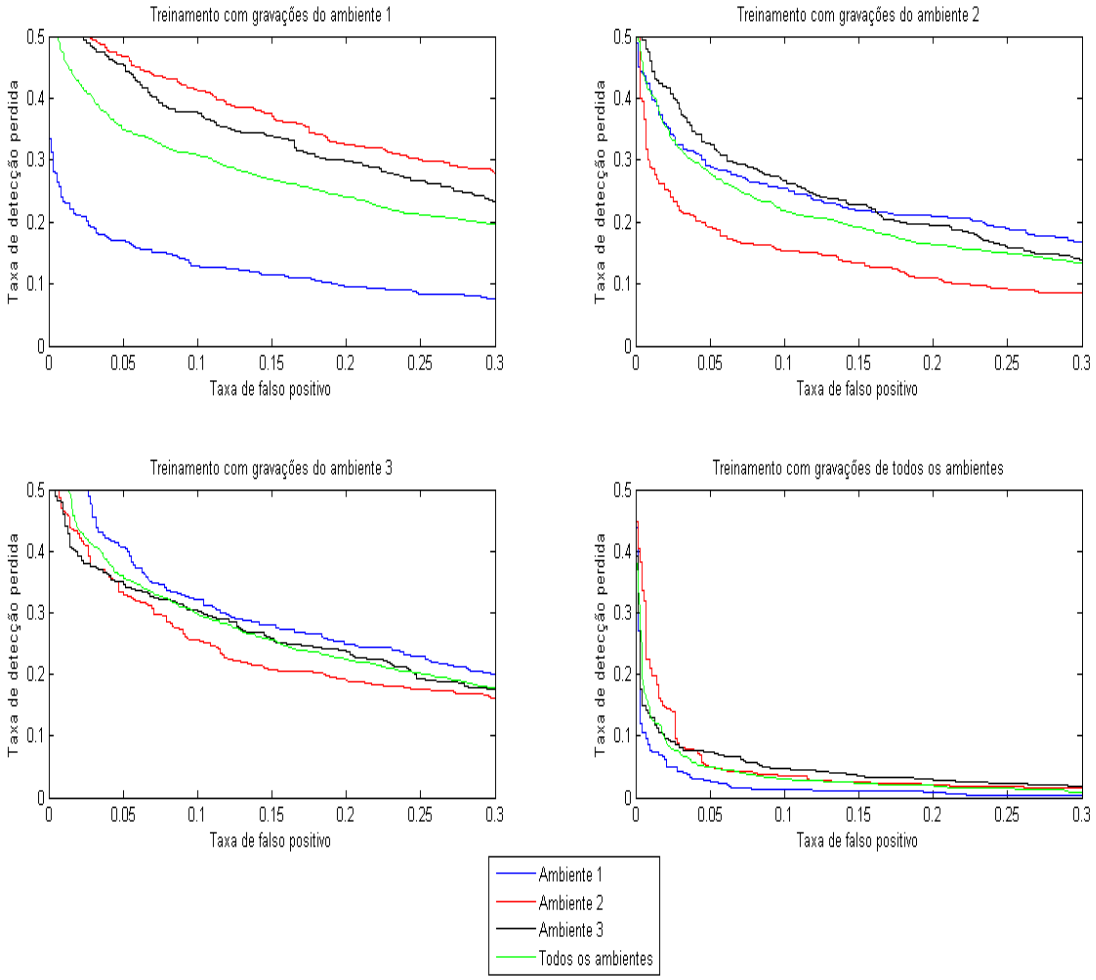

Figura E.5 Curvas DET geradas pelo sistema Adap-GMM-UBM para modelos GMM com 256 distribuições.

Referências Bibliográficas

[Ata74] B. S. Atal. Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. In *Journal of the Acoustical Society of America*, volume 55, pages 1304–1322, 1974.

[BBBG04] M. Ben, M. Betser, F. Bimbot, and G. Gravier. Speaker diarization using bottom-up clustering based on a parameter-derived distance between adapted gmms. In *Proc. ICSLP*, volume 2004, 2004.

[Bev69] P. R. Bevington. *Data Reducrion and Error Analysis for Physical Sciences*. McGraw-Hill, New York, 1969.

[BHTR63] B. P. Bogert, M. J. R. Healy, J. W. Tukey, and M. Rosenblatt. The quefrency alanysis of time series for echoes: Cepstrum, pseudo-autocovariance, cross-cepstrum, and saphe cracking. In *Time Series Analysis*, pages 209–243, 1963.

[Cam97] J. P. Campbell. Speaker recognition: a tutorial. In *Proceedings of the IEEE*, volume 85, pages 1437–1462, 1997.

[Con90] J. B. Conway. *Functional Analysis*. Springer-Verlag, New York, 1990.

[CSR06] W. M. Campbell, D. E. Sturim, and D. A. Reynolds. Support vector machines using GMM supervectors for speaker verification. In *IEEE Signal Processing Letters*, volume 13, pages 308–311, may 2006.

[DH73] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. volume 39, pages 1–38, 1977.

[DM80] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. In *IEEE Transactions on Acoustics, Speech and Signal Processing*, volume 28, pages 357–366, 1980.

[Do03] M. N. Do. Fast approximation of kullback-leibler distance for dependence trees and hidden markov models. In *IEEE Signal Processing Letters*, volume 10, pages 115–118, 2003.

[DRQ00] R. B. Dunn, D. A. Reynolds, and T. F. Quatieri. Approaches to speaker detection and tracking in multi-speaker audio. In *Digital Signal Processing*, volume 10, pages 93–112, 2000.

[Dry07] A. Drygajlo. Forensic automatic speaker recognition. In *Signal Processing Magazine, IEEE*, volume 24, pages 132–135, march 2007.

[DWFQ10] H-J. Dou, Z-Y. Wu, Y. Feng, and Y-Z. Qian. Voice activity detection based on the bispectrum. In *IEEE 10th International Conference on Signal Processing*, pages 502–505, 2010.

[FIS72] S. Furui, F. Itakura, and S. Saito. Talker recognition by long-time averaged speech spectrum. In *Electronics Communications of Japan*, volume 55A, pages 54–61, 1972.

[FR04] N. B. Fairweather and S. Rogerson. Biometric identification. In *Journal of information, communication and ethics in society*, volume 2, pages 3–8, 2004.

[Fur74] S. Furui. An analysis of long-term variation of feature parameters of speech and its application to talker recognition. In *Transactions IECE*, volume 57A, pages 880–887, 1974.

[Fur81] S. Furui. Cepstral analysis technique for automatic speaker verification. volume 29, pages 254–272, 1981.

[Fur04] S. Furui. 50 years of progress in speech and speaker recognition research. In *Journal of the Acoustical Society of America*, volume 116, pages 2497–2498, 2004.

[GL94] J. L. Gauvain and C.-H. Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. In *IEEE Transactions on Speech and Audio Processing*, volume 2, pages 291–298, apr 1994.

[HBP91] A. L. Higgins, L. Bahler, and J. Porter. Speaker verification using randomized phrase prompting. In *Digital Signal Processing*, volume 1, pages 89–106, 1991.

[HM94] H. Hermansky and N. Morgan. Rasta processing of speech. In *IEEE Transactions on Speech and Audio Processing*, volume 2, pages 578–589, 1994.

[HMBK92] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn. Rasta-plp speech analysis technique. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 121–124, 1992.

[HO07] J. R. Hershey and P. A. Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages IV–317–IV–320, 2007.

[Hol01] H. Hollien. *Forensic Voice Identification*. Academic Press, 2001.

[Ker62] L. Kersta. Voiceprint identification. In *Nature*, volume 196, pages 1253–1257, 1962.

[KZZW07] T. Kinnunen, B. Zhang, J. Zhu, and Y. Wang. Speaker verification with adaptive spectral subband centroids. In *Proceedings of the 2007 international conference on Advances in Biometrics*, pages 58–66, 2007.

[MD79] J. Markel and S. Davis. Text-independent speaker recognition from a large linguistically unconstrained time-spaced data base. In *IEEE Transactions on Acoustics, Speech, and Signal Processing*, volume ASSP-27, pages 74–82, 1979.

[MDK+97] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The DET curve in assessment of detection task performance. In *Proceedings of the European Conference on Speech Communication and Technology*, page 1895â1898, 1997.

[Moo96] T. K. Moon. The expectation-maximization algorithm. volume 13, pages 47–60, 1996.

[NJ00] C. Nello and S.-T. John. *Support Vector Machines*. Cambridge Univ. Press, Cambridge, U.K., 2000.

[OW04] A. V. Oppenheim and R. W.Schafer. From frequency to quefrency: a history of the cepstrum. volume 21, pages 95–106, 2004.

[Rab74] L. R. Rabiner. Algorithm for determining the endpoints of isolated utterances. In *Journal of the Acoustical Society of America*, volume 56, page S31, 1974.

[RC08] D. A. Reynolds and W. M. Campbell. Text-independent speaker recognition. In *Springer Handbook of Speech Processing*. Springer, 2008.

[RDL+92] A. E. Rosenberg, J. DeLong, C-H. Lee, B-H. Juang, and F. K. Soong. The use of cohort normalized scores for speaker verification. In *International Conference Speech Language Processing*, pages 599–602, 1992.

[Rey95] D. A. Reynolds. Speaker identification and verification using gaussian mixture speaker models. In *Speech Communication*, volume 17, pages 91–108, 1995.

[RJ93] L. R. Rabiner and B. H. Juang. *Fundamentals of speech recognition*. PTR Prentice Hall, 1993.

[RQD00] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted gaussian mixture models. In *Digital Signal Processing*, volume 10, pages 19–41, 2000.

[RRG+03] J. G. Rodriguez, D. G. Romero, M. G. Gomar, D. R. Castro, and J. O. Garcia. Robust likelihood ratio estimation in bayesian forensic speaker recognition. In *8th European Conference on Speech Communication and Technology*, pages 693–696, 2003.

[SR88] F. K. Soong and A. E. Rosenberg. On the use of instantaneous and transitional spectral information in speaker recognition. In *IEEE Transactions on Acoustics, Speech and Signal Processing*, volume 36, pages 871–879, 1988.

[SVN37] S. S. Stevens, J. Volkmann, and E. B. Newman. A scale for the measurement of the psychological magnitude pitch. In *Journal of the Acoustical Society of America*, number 8, page 185â190, 1937.

[TAE08] T. Thiruvaran, E. Ambikairajah, and J. Epps. FM features for forensic speaker recognition. In *European Conference on Speech Communication and Technology*, pages 1497–1500, September 2008.

[Wol72] J. J. Wolf. Efficient acoustic parameters for speaker recognition. In *Journal of the Acoustical Society of America*, volume 51, pages 2044–2056, 1972.

[WPH06a] R. H. Woo, A. Park, and T. J. Hazen. The MIT mobile device speaker verification corpus: Data collection and preliminary experiments. In *IEEE Odyssey 2006: The Speaker and Language Recognition Workshop, 2006*, pages 1–6, 2006.

[WPH06b] R. H. Woo, A. Park, and T. J. Hazen. The mit mobile device speaker verification corpus: Data collection and preliminary experiments. In *IEEE Odyssey 2006: The Speaker and Language Recognition Workshop*, pages 1–6, 2006.

[YJRK06] K. Yamamoto, F. Jabloun, K. Reinhard, and A. Kawamura. Robust endpoint detection for speech recognition based on discriminative feature extraction. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 805–808, 2006.