Pós-Graduação em Ciência da Computação

Hector Natan Batista Pinheiro

Representações profundas para verificação de locutores independente de texto

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
http://cin.ufpe.br/~posgraduacao

Recife
2020

Hector Natan Batista Pinheiro

Representações profundas para verificação de locutores independente de texto

Tese de Doutorado apresentada ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Doutor em Ciência da Computação.

Área de Concentração: Inteligência Computacional
Orientador: Tsang Ing Ren
Coorientador: George Darmiton da Cunha Cavalcanti

Recife
2020

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

P654r Pinheiro, Hector Natan Batista
Representações profundas para verificação de locutores independente de texto / Hector Natan Batista Pinheiro. – 2020.
160 f.: il., fig., tab.

Orientador: Tsang Ing Ren.
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2020.
Inclui referências e apêndices.

1. Inteligência computacional. 2. Reconhecimento de locutores. I. Ren, Tsang Ing (orientador). II. Título.

006.31 CDD (23. ed.) UFPE - CCEN 2020 - 126

Hector Natan Batista Pinheiro

“Representações profundas para verificação de locutores independente de texto”

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação.

Aprovado em: 28/02/2020.

_____

Orientador: Prof. Dr. Tsang Ing Ren

BANCA EXAMINADORA

_____

Prof. Dr. Carlos Alexandre Barros de Mello
Centro de Informática / UFPE

_____

Prof. Dr. Paulo Salgado Gomes de Mattos Neto
Centro de Informática / UFPE

_____

Prof. Dr. Jugurta Rosa Montalvao Filho
Departamento de Engenharia Elétrica/UFS

_____

Prof. Dr. André Gustavo Adami
Centro de Computação e Tecnologia da Informação/UCS

_____

Prof. Dr. Francisco Madeiro Bernardino Junior
Escola Politécnica de Pernambuco /UPE

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

RESUMO

O desafio no desenvolvimento de sistemas de reconhecimento de locutores consiste em extrair das locuções representações robustas, capazes de distinguir os locutores diante dos mais diversos fatores que podem influenciar na geração dos sinais de voz, como a presença de ruído acústico do ambiente ou as condições físicas do locutor. Este trabalho foca no desenvolvimento de tais representações, levando em consideração a tarefa de verificação independente de texto. Nos últimos anos, diversas abordagens utilizando redes neurais profundas vêm sendo propostas para a geração de representações cada vez mais robustas. Dentre elas, a que mais se destacou consiste nos *x-vectors*, onde uma rede neural supervisionada é treinada para discriminar locuções, inicialmente descritas através de características espectrais de tempo curto. Uma representação vetorial para a locução é gerada através de uma camada de *pooling* que agrega os diversos vetores da locução. A partir dessa camada, a rede neural discrimina locuções inteiras utilizando as classes dos locutores que as produziram. A autenticação é realizada ao decidir se dois *x-vectors* foram produzidos pelo mesmo locutor ou não, através de uma análise probabilística de discriminantes lineares (*Gaussian Probabilistic Linear Discriminant Analysis* – G-PLDA). Neste trabalho, propomos um conjunto de abordagens capazes de melhorar a qualidade das representações baseadas nos *x-vectors*. As abordagens possuem o objetivo de tornar as representações geradas pela rede mais apropriadas para o método de comparação G-PLDA, que, por sua vez, segue a premissa que as representações dos locutores seguem distribuições condicionais e *a priori* gaussianas. Primeiramente, propomos camadas de classificação e *pooling* gaussianos para a geração de representações gaussianas. Em seguida, desenvolvemos um método de regularização variacional para o controle da distribuição *a priori* dos *x-vectors*. A função de regularização minimiza a divergência entre a distribuição das representações geradas e uma determinada distribuição desejada, que no nosso caso é a distribuição normal padronizada. Nessa abordagem, uma amostra da distribuição desejada é apresentada à rede e a função de regularização computa uma medida de divergência não paramétrica entre as amostras. As abordagens propostas foram avaliadas utilizando a base de dados *Fisher English Training*, em um total de oito condições de avaliação, considerando o gênero dos locutores e as durações das locuções de teste. Os métodos foram comparados com a modelagem convencional dos *x-vectors* e outros métodos presentes na literatura para controle do espaço das representações. Nos resultados obtidos, pôde-se observar que as abordagens propostas geram representações mais adequadas à modelagem G-PLDA, proporcionando ganhos de desempenho de, em média, 11,63% e 15,52% nos valores de *Equal Error Rate* (EER) e *Minimum Detection Cost Function* (minDCF), respectivamente.

Palavras-chaves: Reconhecimento de locutores. Verificação de locutores independente de texto. Aprendizado profundo. Redes neurais profundas.

ABSTRACT

The challenge in the development of speaker recognition systems is to extract robust representations from the speech segments, capable of distinguishing the speakers in the presence of factors that can influence the generation of speech signals, such as the presence of environmental acoustic noise or the speakers' physical conditions. This work focuses on the development of such representations, taking into account the task of text-independent verification. In recent years, several approaches using deep neural networks have been proposed for the generation of robust representations. Among them, the one that stood out the most consists of the x-vectors, where a supervised network is trained to discriminate speech segments, initially described through a set of short-term spectral features. A pooling layer aggregates the set of feature vectors to generate a fixed-length representation for the speech segment. From this layer, the network discriminates entire utterances using the classes of the speakers who produced them. The authentication task consists of deciding whether two x-vectors were produced by the same speaker or not, using a Gaussian Probabilistic Linear Discriminant Analysis (G-PLDA) model. In this work, we propose a set of approaches capable of improving the quality of the representations based on x-vectors. The approaches aim to produce representations more appropriate for the G-PLDA backend modeling, which assumes that speakers' representations follow conditional and a priori distributions. Firstly, we propose Gaussian classification and pooling layers for the generation of Gaussian representations. Then, we developed a variational regularization method to control the representations prior distributions. The regularization function minimizes the divergence between the distribution of the generated representations and a given desired distribution, which, in our case, is the standard normal distribution. In this approach, a sample of the desired distribution is presented to the network, and the regularization function computes a non-parametric divergence measure between the samples. We evaluated the proposed approaches using the Fisher English Training database, in a total of eight evaluation conditions, considering the gender of the speakers and the duration of the test utterances. We compared the methods with the conventional x-vectors modeling and other methods previously proposed to control the space of the representations. In the results obtained, we observed that the proposed approaches generate representations better suited to the G-PLDA modelling, providing performance gains of, on average, 11.63% and 15.52% in the values of Equal Error Rate (EER) and Minimum Detection Cost Function (minDCF), respectively.

Keywords: Speaker recognition. Text-independent speaker verification. Deep learning. Deep neural networks.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

2FA	*Two Factor Authentication*
ASR	*Automatic Speech Recognition*
BF	*Bottleneck Feature*
CGM	*Compound Gaussian Model*
CMN	*Cepstral Mean Normalization*
CMS	*Cepstral Mean Subtraction*
CMVN	*Cepstral Mean and Variance Normalization*
DCF	*Detection Cost Function*
DCT	*Discrete Cosine Transform*
DFT	*Discrete Fourier Transform*
DL	*Deep Learning*
DNN	*Deep Neural Networks*
EER	*Equal Error Rate*
ELBO	*Evidence Lower Bound*
EM	*Expectation-Maximization*
FAR	*False Acceptance Error*
FRR	*False Rejection Rate*
FW	*Feature Warping*
G-PLDA	*Gaussian Probabilistic Linear Discriminant Analysis*
GMM	*Gaussian Mixture Model*
HMM	*Hidden Markov Model*
HT-PLDA	*Heavy-tailed Probabilistic Linear Discriminant Analysis*
IDFT	*Inverse Discrete Fourier Transform*
JFA	*Joint Factor Analysis*
KL	Kullback-Leibler
KLT	*Karhunen–Loève Transform*
LDA	*Linear Discriminant Analysis*
LDC	*Linguistic Data Consortium*
LFCC	*Linear Frequency Cepstral Coefficients*
LN	*Length Normalization*

LP	*Linear Prediction*
LTSD	*Long-Term Spectral Divergence*
MAP	*Maximum a posteriori*
MFA	*Multi Factor Authentication*
MFCC	*Mel-Frequency Cepstral Coefficients*
ML	*Maximum likelihood*
MMD	*Maximum Mean Discrepancy*
NIN	*Network-in-Network*
NIST	*National Institute of Standards and Technology*
PCA	*Principal Component Analysis*
PIN	*Personal Identification Number*
PLDA	*Probabilistic Linear Discriminant Analysis*
RASTA	*Relative Spectral*
RBF	*Radial Basis Function*
ReLU	*Rectified Linear Unit*
RKHS	*Reproducing Kernel Hilbert Space*
ROC	*Receiver Operating Characteristic*
SVM	*Support Vector Machine*
TDNN	*Time-delayed Neural Network*
t-SNE	*t-Distributed Stochastic Neighbor Embedding*
UBM	*Universal Background Model*
VAD	*Voice Activity Detector*
VAE	*Variational Autoencoder*
VQ	*Vector Quantization*
WCNN	*Within Class Covariance Normalization*
ZCR	*Zero Crossing Rate*

SUMÁRIO

1 INTRODUÇÃO

Este trabalho foca no desenvolvimento de sistemas de verificação de locutores independente de texto. Essa tarefa consiste em uma modalidade biométrica que possui o objetivo de realizar o processo de identificação pessoal e, em especial, a autenticação de indivíduos através das informações presentes em sua voz. Partindo da definição do processo de identificação pessoal e da sua importância na sociedade, os métodos tradicionais utilizados para esse propósito são listados, assim como os problemas de segurança envolvidos na utilização dos mesmos. É apresentada, então, a abordagem mais bem sucedida para a solução de tais problemas, que consiste na utilização de identificadores biométricos. Após a definição do processo de identificação biométrica e do desenvolvimento de sistemas desse tipo, partimos para o foco deste trabalho: a biometria de voz. Os fundamentos dessa biometria são apresentados, assim como sua importância, aplicações e os desafios enfrentados por sistemas de reconhecimento de locutores. Em seguida, uma visão geral do estado da arte desses sistemas é apresentada. Por fim, são descritos os objetivos deste trabalho, suas contribuições e a organização do restante deste documento.

1.1 IDENTIFICAÇÃO PESSOAL

Entende-se por identificação pessoal o processo de reconhecimento da identidade unicamente atribuída a uma determinada pessoa. Tal processo ocorre cotidianamente e é essencial em quase todos os setores da sociedade. Perguntas como "quem é esse indivíduo?" ou "é essa pessoa quem ela diz ser?" são realizadas diariamente milhões de vezes por instituições governamentais, organizações financeiras, sistemas de saúde, comércio eletrônico ou telecomunicações. Transações financeiras, reivindicação de benefícios sociais, acesso a recursos restritos e compras com cartão de crédito são apenas algumas das diversas operações nas quais a identificação de um indivíduo é necessária.

O processo de identificação pessoal é comumente referenciado, de uma maneira geral, como reconhecimento e, dependendo do contexto, ele pode ocorrer via verificação ou via identificação. No processo de verificação, também chamado de autenticação, o sistema deve comprovar a identidade alegada pelo usuário. Já no processo de identificação, o sistema deve determinar qual dos indivíduos previamente cadastrados é a pessoa sendo reconhecida.

A principal preocupação no processo de identificação pessoal está na sua acurácia, definida através dos possíveis erros cometidos ao atribuir uma identidade a um determinado indivíduo. Em algumas operações, a identificação equivocada de indivíduos pode trazer consequências desastrosas. A atribuição errada da identidade de um usuário ou a incapacidade de uma pessoa se autenticar em um sistema são os exemplos mais comuns. Porém,

existe ainda a tentativa de fraude, onde um indivíduo tenta se passar por outras pessoas a fim de conseguir acesso a dados ou recursos disponíveis apenas a elas. Por exemplo, fraudes aplicadas em transações com cartões de crédito ou em caixas eletrônicos trouxeram um prejuízo de mais de 27 bilhões de dólares no mundo todo (NILSON, 2019), com um aumento de 15% em relação ao ano anterior. Só nos Estados Unidos esse valor ultrapassou os nove bilhões de dólares. Além disso, como as pessoas estão cada vez mais conectadas à Internet, cresce o acesso não presencial a sistemas, aumentando assim a facilidade das ações de impostores. Nos Estados Unidos, entre 2013 e 2018, roubos de acesso a e-mails corporativos resultaram em prejuízos da ordem de 12 bilhões de dólares (FBI, 2018). No Brasil, ocorrem em média 1750 tentativas de fraudes nas transações comerciais *on-line*, por mês, e cerca de três em cada dez dessas tentativas são bem sucedidas (LEXISNEXIS, 2018). Nesse cenário, a necessidade do desenvolvimento de sistemas de identificação pessoal com uma alta acurácia tem se tornado cada vez mais crítica.

Duas são as abordagens tradicionais para identificação pessoal: as baseadas em objeto e as baseadas em conhecimento. Os métodos baseados em objeto utilizam algum artefato físico (chaves, carteiras de identidade, passaportes, carteiras de motorista etc.) pertencente ao indivíduo. Já os métodos baseados em conhecimento usam alguma informação que só o indivíduo deve possuir, como senhas ou números de identificação pessoal (PIN - *Personal Identification Number*). A desvantagem na utilização dos métodos tradicionais provém do fato de eles não utilizarem nenhuma informação inerente ao indivíduo. Por exemplo, objetos como cartões de crédito ou carteiras de identidade podem ser perdidos, roubados ou esquecidos. Além disso, senhas ou PINs podem ser esquecidas pelo indivíduo correspondente ou descobertas por um impostor. Dessa maneira, tais métodos não satisfazem os requisitos de segurança necessários para uma sociedade como a nossa, onde as tentativas de fraude são cada vez mais comuns e os métodos, cada vez mais sofisticados.

1.2 IDENTIFICAÇÃO BIOMÉTRICA

Identificação biométrica diz respeito ao processo de realizar identificação pessoal utilizando características físicas ou comportamentais do indivíduo. O corpo humano possui diversas características que são únicas para cada indivíduo e a utilização delas proporciona uma maior segurança ao processo de identificação, quando comparados aos métodos baseados em objeto ou conhecimento. Tais características são comumente referenciadas como identificadores biométricos ou simplesmente como biometrias. Exemplos de tais identificadores são: face, impressão digital, íris, assinatura e voz.

1.2.1 A biometria ideal

Um identificador biométrico ideal deve ser:

- **universal**: todo indivíduo deve possuí-lo;

- **único**: cada indivíduo deve possuir apenas um identificador, e ele deve ser diferente dos demais;
- **coletável**: fácil coleta e descrição;
- e **permanente**: não se altera com o tempo.

Na prática, é extremamente difícil desenvolver um sistema biométrico que satisfaça todos esses requisitos. Por essa razão, em um sistema biométrico prático, outros requisitos podem ser levados em consideração, tais como:

- **desempenho**: definido em termos de acurácia, velocidade, custo (recursos associados à captura, armazenamento e processamento) e/ou robustez do sistema na presença de fatores que impactam o desempenho do sistema;
- **aceitabilidade**: que diz respeito à disponibilidade das pessoas incorporarem um determinado identificador biométrico em suas vidas cotidianas;
- e **fraudabilidade**: que determina a facilidade de subjugar o sistema a partir de métodos fraudulentos.

1.3 SISTEMAS BIOMÉTRICOS

Um sistema de identificação biométrica realiza o processo de identificação pessoal reconhecendo uma característica fisiológica individual do usuário. Nesse sentido, pode-se afirmar que um sistema biométrico é basicamente um sistema de reconhecimento de padrões associados a essa identidade. Esse processo é realizado através de duas etapas: cadastro e reconhecimento. A Figura 1 apresenta o fluxo genérico de um sistema desse tipo.

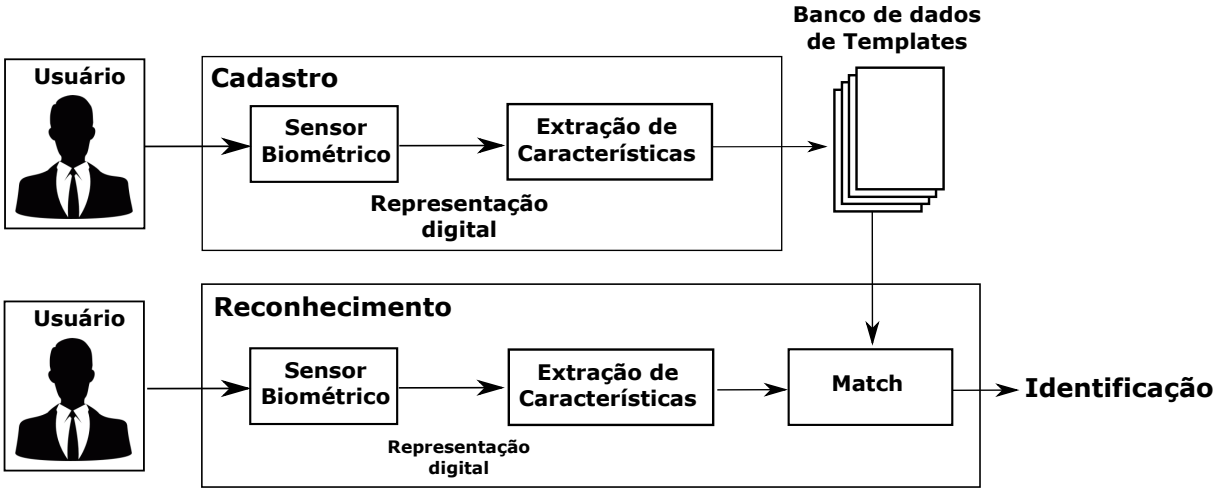

Figura 1 – Etapas de cadastro e reconhecimento de um sistema de identificação biométrica.

No cadastro, a característica biométrica de um indivíduo é capturada por um sensor, que produz uma representação digital da mesma. Em seguida, o módulo de extração de características produz uma representação mais compacta e mais expressiva do sinal, comumente referenciada como *template*. As características extraídas do identificador biométrico variam para cada tipo de biometria, porém, em suma, o objetivo desse módulo é gerar uma representação que facilite o processo de identificação. Para isso, tal representação deve enfatizar as informações que discriminam diferentes indivíduos e suavizar as informações que variam em diferentes capturas de uma mesma pessoa. A etapa de cadastro produz um *template* (ou um conjunto de *templates*) para cada usuário cadastrado no sistema.

Durante a fase de reconhecimento, o mesmo tipo de sensor biométrico é utilizado para capturar a biometria, e o mesmo módulo de extração de características da fase de cadastro é utilizado para gerar um *template*. A identificação é então realizada por meio da comparação entre os *templates* cadastrados e de reconhecimento. Esse processo é referenciado como *match* e estabelece um grau de similaridade entre os diferentes *templates*, proporcionando uma decisão a respeito da identidade do indivíduo.

Assim como em identificação pessoal, os sistemas biométricos podem ser desenvolvidos tanto para o propósito de verificação (autenticação) quanto para a identificação de usuários. Em um sistema de verificação, o usuário submete uma alegação a respeito da sua identidade. Nesse caso deve existir um *template* pré-cadastrado associado a essa identidade. O sistema então compara a característica biométrica capturada com o *template* correspondente à identidade alegada e aceita ou rejeita essa alegação. Já nos sistemas de identificação, o sistema busca em todos os *templates* cadastrados na base de dados por aquele que apresenta maior similaridade com a biometria capturada. Este trabalho se concentra na tarefa de verificação de usuários.

1.3.1 Métricas de desempenho

O desempenho de um determinado sistema biométrico pode ser analisado em termos de acurácia, velocidade, custo e robustez (WAYMAN, 1999). Do ponto de vista de velocidade, podemos considerar como métrica o tempo que o sistema leva para realizar o reconhecimento, que engloba os tempos de captura e de processamento do *match*. Para algumas aplicações, esse tempo pode ser crítico, como em caixas eletrônicos, onde a autenticação do usuário deve ser realizada em poucos segundos. Já sobre os custos associados à operação de um sistema biométrico, diversos são os fatores que devem ser observados. Eles vão desde o custo do sensor propriamente dito (microfones, câmeras, sensores de impressão digital etc) até o custo de armazenamento dos dados cadastrados. Nesse último caso, os tamanhos dos *templates* são a principal preocupação. Quando eles são armazenados em um banco de dados central, acessado através de um canal de comunicação *on-line*, o tamanho de cada um deles pode dificultar o acesso a essas informações devido à quantidade

de banda disponível na rede por onde serão transmitidos.

Sob o ponto de vista de classificação, dois tipos de erros podem ser cometidos por um sistema de verificação. O primeiro deles ocorre quando um usuário genuíno não é capaz se autenticar. Nesse caso, o sistema erroneamente rejeita a alegação submetida pelo usuário a respeito de sua identidade. Esse tipo de erro é geralmente chamado de erro de falsa rejeição. O segundo erro ocorre quando o sistema aceita a alegação realizada por um impostor, autenticando-o quando não deveria. Geralmente esse tipo de erro é chamado de erro de falsa aceitação. As probabilidades de o sistema cometer tais erros são referenciadas como taxa de falsa rejeição (FRR - *False Rejection Rate*), e taxa de falsa aceitação (FAR – *False Acceptance Error*), respectivamente. Na prática, um sistema produz uma medida, referenciada como *score*, que exprime o grau de certeza sobre a decisão. A decisão final é realizada através da comparação do *score* com um determinado limiar de aceitação. Dessa maneira, ao variar o valor desse limiar, o sistema torna-se mais ou menos conservador, isto é, aumenta-se ou diminui-se a dificuldade de um usuário se autenticar. Além disso, os valores de FAR e FRR são definidos para cada ponto de operação e a relação entre as taxas erro é dual: diminuir a probabilidade de ocorrência de um tipo de erro implica no aumento da probabilidade de ocorrência do outro. Portanto, existe um *trade-off* que deve ser levado em consideração. A Figura 2 mostra um exemplo da variação das taxas de erro de um determinado sistema. Essa curva é comumente referenciada como curva ROC (*Receiver Operating Characteristic*[1]) e apresenta de forma explícita o impacto da variação do ponto de operação.

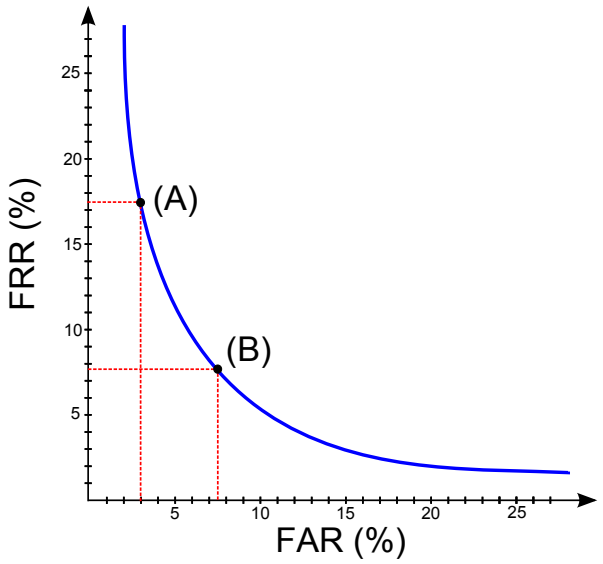

Figura 2 – Taxas de FRR e FAR em diversos pontos de operação de um determinado sistema biométrico. Em (A) apresenta-se um ponto de operação específico, com FAR de aproximadamente 3%, enquanto que em (B), temos o ponto de operação referente ao *Equal Error Rate* (EER).

[1] Mais precisamente, uma curva ROC é comumente expressa utilizando a taxa de verdadeiro positivo $(1 - FRR)$. Aqui apresentamos dessa maneira para evidenciar o ponto de operação referente ao EER.

A escolha do ponto de operação depende da aplicação. Por exemplo, em aplicações com alto nível de segurança de acesso, o FAR deve ser muito baixo, uma vez que a prioridade reside na rejeição de impostores. Já em aplicações forenses, por exemplo, o desejo de identificar um criminoso é maior que o inconveniente de examinar um número maior de suspeitos. Nesse caso, o sistema deve operar com um FRR baixo (e, consequentemente, com um FAR mais alto). De modo geral, o desempenho de um sistema biométrico é considerado satisfatório se as taxas de erro associadas ao ponto de operação escolhido forem aceitáveis. Por exemplo, o desenvolvedor de um sistema biométrico pode estar sujeito ao requisito de a taxa de FAR não ser superior a 3%. Nesse caso, o desenvolvedor deve escolher um ponto de operação que torna o sistema satisfatório (ponto de operação (A) da Figura 2, por exemplo).

Apesar de a curva ROC descrever o desempenho do sistema de uma maneira genérica, para a comparação direta entre diferentes sistemas é mais conveniente a utilização de métricas que sumarizam seus desempenhos. Uma das métricas mais utilizadas para esse propósito consiste nas taxas de erro associadas ao ponto de operação onde FAR e FRR são iguais (ponto de operação (B) da Figura 2). Essa taxa de erro é referenciada como EER (*Equal Error Rate*), e apresenta um bom indicativo a respeito do poder discriminatório do sistema, de uma maneira geral e independente de aplicação.

Porém, o EER não descreve a capacidade de se escolher bons pontos de operação para o sistema. Isto é, ele não consegue informar se é possível calibrar o sistema de maneira a satisfazer algum requisito de desempenho específico. Além disso, a aplicação prática do sistema ocorre em um contexto que possui características importantes que devem ser consideradas para a escolha do ponto de operação. Em um sistema de verificação, a proporção entre as tentativas autênticas (verdadeiros positivos) e fraudulentas (verdadeiros negativos) são tipicamente diferentes daquelas utilizadas para a criação da curva ROC e cálculo do EER. Além disso, dependendo da aplicação, um tipo de erro pode ser mais grave do que outro, e um maior custo deve ser associado a ele. Através da combinação dessas características com as taxas de erro em um ponto de operação, temos o chamado custo de detecção (DODDINGTON et al., 2000):

$$C_{det} = C_{FR} \times FRR \times P_{pos} + C_{FA} \times FAR \times (1 - P_{pos}), \tag{1.1}$$

onde os valores de FRR e FAR são ponderados levando em consideração seus custos C_{FR} e C_{FA}, respectivamente, e a proporção esperada de verdadeiros positivos P_{pos}.

Como para cada ponto de operação temos um valor de custo de detecção, podemos produzir uma curva, como a ROC. Essa curva é referenciada como função de custo de detecção (DCF - *Detection Cost Function*). Para sumarizar a curva em uma única medida, geralmente é escolhido o ponto de operação ótimo e a medida é definida pelo valor mínimo da função (minDCF). Diferentemente do EER, tal medida é parametrizada pelos custos associadas a cada um dos erros e a proporção dos testes positivos, que definem o contexto da aplicação do sistema.

1.4 RECONHECIMENTO DE LOCUTORES

A vida em sociedade só é possível através da comunicação entre os indivíduos. A linguagem, nas formas escrita, falada ou gesticulada, sustenta todos os aspectos das interações humanas. Na linguagem falada, indivíduos se comunicam entre si por meio do aparato vocal humano e os sinais acústicos produzidos não apenas são capazes de definir a mensagem sendo transmitida como também inclui características individuais do locutor (RABINER; JUANG, 1993; BEIGI, 2011).

A produção da voz humana envolve uma combinação de características físicas e comportamentais que são únicas para cada pessoa. Dentre as características fisiológicas, o tamanho e formato do trato vocal desempenha grande importância na caracterização do locutor. Refere-se a trato vocal o conjunto de órgãos de produção de voz situados acima das cordas vocais, e inclui a faringe e as cavidades oral e nasal (CAMPBELL, 1997). Além disso, existem outros aspectos (culturais ou comportamentais) da produção da voz que também podem ser úteis para discriminação de locutores. Exemplos disso incluem velocidade da fala, efeitos de prosódia e sotaque. Tais características podem ser observadas a partir da movimentação dos lábios, da mandíbula, da língua, do véu palatino e da laringe, e podem variar com o tempo devido à idade ou às condições de saúde do indivíduo.

As tecnologias que utilizam a voz humana para reconhecer, identificar ou autenticar um indivíduo são referenciadas como sistemas de reconhecimento de locutores (BEIGI, 2011). Em suma, reconhecimento de locutores é uma modalidade biométrica que se propõe a realizar o processo de identificação pessoal a partir das informações presentes unicamente na voz do indivíduo. Similarmente aos demais sistemas biométricos, existem dois tipos de tarefas: verificação e identificação de locutores. Neste trabalho, nosso foco está na tarefa de verificação de locutores. São apresentadas a um sistema automático de verificação de locutores uma locução, cuja identidade de quem a produziu é desconhecida, e uma alegação acerca dessa identidade. O sistema então decide se a locução pertence à identidade alegada ou não. Essa tarefa também é referenciada como autenticação de locutores, autenticação de voz ou verificação de voz. Devido aos diferentes tipos de abordagens utilizadas nessa tarefa, as informações relacionadas aos locutores, que são extraídas durante a fase de cadastramento e que são utilizadas para a descrição vetorial dos indivíduos, é comumente referenciada como modelo do usuário. Diferentemente do termo *template*, utilizado para os sistemas biométricos, os usuários do sistemas são geralmente descritos através de modelos probabilísticos, capazes de descrever a distribuição das características extraídas (REYNOLDS; QUATIERI; DUNN, 2000).

Na fase de reconhecimento, o modelo do locutor correspondente à alegação é utilizado para avaliar a locução de teste e realizar a decisão (Figura 3). Como pode ser observado, a verificação é um processo que envolve a comparação apenas com o modelo correspondente à alegação, de modo que, mesmo aumentando a quantidade de locutores cadastrados no sistema, o tempo de teste, em geral, permanece constante. Diferentemente da tarefa de

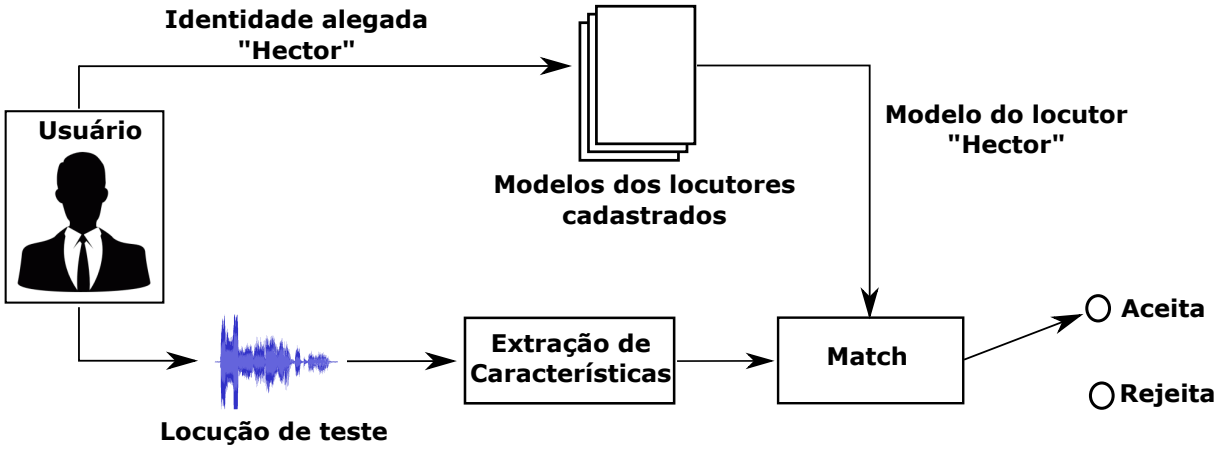

Figura 3 – Processo de verificação (autenticação) de um determinado locutor.

identificação de locutores, em verificação não há a premissa de que a locução de teste foi produzida por um locutor já cadastrado. Isso significa que o sistema deve ser capaz de diferenciar um determinado locutor de qualquer outro (sem restrições). Por essa razão, esse problema é comumente referenciado como reconhecimento de conjunto aberto.

1.4.1 Dependência e independência de texto

Os sistemas de reconhecimento de locutores podem ser dependentes e independentes de texto (BEIGI, 2011). Em sistemas dependentes de texto, o locutor deve pronunciar uma palavra ou frase específica. Nesse caso, o modelo do locutor é construído utilizando locuções com o mesmo conteúdo fonético daquele encontrado nas locuções de teste. Mesmo que o indivíduo esteja cadastrado no sistema, ele não deverá ser reconhecido se a frase dita for diferente daquela utilizada no cadastro. Como sistemas desse tipo operam sob um contexto fonético fixo, eles são capazes de apresentar melhores desempenhos. Porém, tal abordagem limita a gama de aplicações para esse tipo de sistema.

Em sistemas independentes de texto, o reconhecimento ocorre sem qualquer tipo de restrição a respeito do que está sendo dito. Sistemas desse tipo extraem características que modelam apenas o trato vocal do locutor, sem utilizar qualquer suposição a respeito do conjunto fonético que pode ser apresentado. Para isso, tais sistemas devem levar em consideração a comum variabilidade fonética das locuções utilizadas no cadastro e no reconhecimento. Este trabalho concentra-se na tarefa de verificação de locutores independente de texto.

1.4.2 Aplicações

De imediato, podemos verificar que a biometria de voz é adequada para aplicações atreladas aos serviços de telefonia. Estamos cada vez mais próximos da integração de sistemas de reconhecimento de locutor e de fala em serviços telefônicos, com o propósito de au-

xiliar a interação com o usuário ou até mesmo de substituir o papel do operador nesses serviços. Em 2013, a Barclays Wealth, um dos maiores bancos privados do mundo, anunciou a implantação de um sistema de reconhecimento de voz nos serviços disponibilizados via telefone, que seria capaz de autenticar seus clientes após trinta segundos de conversa sem restrição (WARMAN, 2013). Após a implantação, 93% dos clientes avaliaram o sistema com nota máxima levando em consideração aspectos de velocidade, facilidade e segurança. Mais recentemente, em 2016, o HSBC e o banco digital First Direct implantaram sistemas de autenticação utilizando voz para 15 milhões de usuários no Reino Unido (KOLLEWE, 2016).

Outra área onde a biometria de voz se sobressai consiste no controle de acesso remoto, onde o usuário pode se autenticar estando em qualquer lugar. Exemplos desse tipo de sistema são os sistemas de controle de acesso remoto que funcionam através da Internet. Como o sensor biométrico deve estar localizado junto ao usuário, existe o custo de implantação associado ao sensor utilizado. No caso da biometria de voz, tal sensor é um microfone, que atualmente está acoplado à maioria dos dispositivos que utilizamos diariamente, o que facilita a sua adoção. Nos últimos anos, sistemas têm adotado o uso de biometrias (principalmente face e voz) como uma camada a mais de segurança durante a autenticação tradicional utilizando senhas. Esse tipo de abordagem é chamada de autenticação de multifator (MFA - *Multi Factor Authentication*) e a mais popular é a autenticação de dois fatores (2FA - *Two Factor Authentication*), onde o sistema envia uma determinada informação (via SMS, por exemplo) para o dispositivo móvel do usuário. Nesse cenário, tem-se popularizado o uso das biometrias como uma alternativa ao SMS, devido à violabilidade dos dispositivos. A popularização desse tipo de abordagem também se deve à disponibilidade de serviços de autenticação prontos-para-uso de fácil integração aos sistemas, tais como as soluções da Nuance (NUANCE, 2019) e a da Microsoft (MICROSOFT, 2019).

Além disso, uma das mais importantes aplicações da tecnologia de reconhecimento de locutores está na área forense. Durante muitos anos, houve bastante interesse no uso da biometria de voz por parte de juristas, advogados e detetives, com o objetivo de decidir se uma determinada gravação de voz foi produzida por um determinado suspeito ou não. Historicamente, o uso desse tipo de metodologia para a geração de provas a serem consideradas em uma corte jurídica foi bastante controverso (BONASTRE et al., 2015). Em (KERSTA, 1962), ocorreu a introdução do termo "*voiceprint*", que levou a comunidade a ter a impressão errônea de que indivíduos pudessem ser identificados através de uma simples análise visual do espectro do sinal de voz, que possuiria padrões visuais claros e fáceis de serem identificados, como, por exemplo, acontece com impressões digitais. A partir desse momento, essa foi a abordagem adotada por "especialistas", que desenvolveram técnicas para realizar a comparação visual dos espectros de duas locuções e decidir se elas foram produzidas pela mesma pessoa. Apesar da falta de comprovação científica

sobre a abordagem e dos danos causados em casos jurídicos (BOË, 2000), a comunidade acadêmica se desassociou completamente desse tipo de prática apenas em 2007 (IAFPA, 2013). Mais recentemente, o descrédito com relação ao uso da biometria de voz na área forense tem diminuído graças ao crescente aumento de desempenho alcançado pelos métodos de reconhecimento automático de locutores. Além disso, iniciativas para a avaliação sistemática da acurácia e robustez desses métodos têm se tornado um catalisador para o aumento da credibilidade desses sistemas. Neste sentido, a NIST (*National Institute of Standards and Technology*) tem desempenhado um importante papel na organização dessas avaliações (NIST, 1996). O uso de técnicas de reconhecimento automático de locutores na área forense ainda é tratado com cautela pela comunidade (BONASTRE et al., 2003; CAMPBELL et al., 2009). Porém, muito esforço vem sendo empregado com o objetivo de adequar os sistemas automáticos de reconhecimento de locutores ao arcabouço forense (AJILI et al., 2016; DRYGAJLO; HARAKSIM, 2017; AL-ALI; SENADJI; NAIK, 2017; SOLEWICZ; JESSEN; VLOED, 2017).

1.4.3 Desafios

O contexto de captura de uma determinada locução é referenciado como sessão. Em uma determinada sessão, podem-se encontrar diversos fatores que podem influenciar no processo de geração do sinal de voz e, consequentemente, aumentar a variabilidade das características acústicas do sinal. Alguns desses fatores são (FURUI, 1997; BEIGI, 2011; TOGNERI; PULLELLA, 2011):

1. O tipo de microfone utilizado.
   Diferentes tipos de microfones geram diferentes distorções no sinal.

2. A distância do locutor para o microfone.
   Para a geração de áudios com qualidade satisfatória, alguns microfones requerem que essa distância seja pequena. Além disso, quanto maior a distância, maior será a influência de outras fontes sonoras presentes no local.

3. Ruído acústico proveniente do ambiente onde o áudio foi gravado.
   Além da adição de outros sinais acústicos provenientes de outras fontes (ruído de fundo), em lugares barulhentos, o ser humano naturalmente aumenta a intensidade de sua voz e, ao forçar seu aparato vocal, provoca distorções arbitrárias no sinal sendo gerado. Esse fenômeno é referenciado como Efeito Lombard (JR et al., 1989; JUNQUA, 1993).

4. As condições físicas do locutor.
   Situações onde o locutor está doente, sob estresse ou emotivo, geralmente provocam distorções no sinal de voz.

5. Qualidade do canal de comunicação por onde o áudio é transmitido.
   Ruído aditivo proveniente do canal (devido à interferência eletromagnética) ou então perda das informações transmitidas.

6. Possíveis transformações no sinal de voz.
   Dependendo da aplicação, o sinal pode ser convertido para diferentes formatos de mídia. Na utilização de métodos de compressão de dados, por exemplo, é bastante comum haver perda de informações.

Para as principais aplicações em reconhecimento de locutores, é desejável que não se imponham restrições às sessões de captura dos sinais de voz. Dessa maneira, o maior desafio enfrentado no desenvolvimento desses sistemas está na inconsistência dos sinais de voz proveniente dos diferentes cenários onde o sistema pode operar. Tal problema é referenciado como incompatibilidade de sessão. Cada um dos itens listados anteriormente pode ser chamado de fator (ou fonte) de incompatibilidade.

O processo de atenuar algum tipo de incompatibilidade é referenciado na literatura como compensação. Inúmeras técnicas têm sido propostas para lidar com tais incompatibilidades, geralmente tentando atenuar algum tipo de incompatibilidade específica. Durante as últimas décadas, o foco da maioria das técnicas de compensação propostas consistia em atenuar distorções causadas pelo tipo de microfone utilizado ou pelo canal de comunicação por onde o sinal era transmitido (GISH et al., 1985; ORTEGA-GARCÍA; GONZÁLEZ-RODRÍGUEZ, 1996; REYNOLDS, 2003; SOLOMONOFF; CAMPBELL; BOARDMAN, 2005). Esse tipo de incompatibilidade é chamado de incompatibilidade de canal. O grande interesse nesse tipo de compensação se devia à disponibilidade de sinais de voz gravados utilizando telefone e pela necessidade do desenvolvimento de sistemas para aplicações de telefonia.

Outro tipo de compensação que também recebeu bastante atenção são aquelas destinadas às distorções causadas pelo ruído acústico do ambiente onde a locução é gerada. Esse tipo de incompatibilidade é referenciado como incompatibilidade de fundo ou incompatibilidade de ambiente (MING; STEWART; VASEGHI, 2005; TOGNERI; PULLELLA, 2011). Nesse tipo de incompatibilidade são considerados não somente outras fontes sonoras presentes no ambiente como também os efeitos de reverberação em ambientes fechados (GONZÁLEZ-RODRÍGUEZ et al., 1996; PEER; RAFAELY; ZIGEL, 2008; GARCIA-ROMERO; ZHOU; ESPY-WILSON, 2012).

Observou-se também a deficiência dos métodos de reconhecimento na realização das tarefas quando locuções de duração curtas são utilizadas (VOGT; LUSTRI; SRIDHARAN, 2008; KANAGASUNDARAM et al., 2011; KANAGASUNDARAM et al., 2012). Nesse cenário, a disponibilidade das características que caracterizam os locutores é escassa, o que pode dificultar tanto o cadastramento quanto o reconhecimento dos locutores. A capacidade

de realizar o reconhecimento utilizando locuções curtas possibilita a requisição de menos amostras aos usuários, possibilitando que o reconhecimento ocorra de maneira rápida.

Muitas técnicas de compensação foram desenvolvidas durante os anos, e a maioria delas busca suavizar algum tipo de incompatibilidade específica. As principais delas são descritas no Capítulo 2. Apesar de muitas abordagens diferentes terem sido desenvolvidas, poucas delas são utilizadas atualmente. Com o desenvolvimento de técnicas de aprendizagem de máquina mais sofisticadas, os maiores avanços na robustez dos sistemas surgiram a partir de técnicas que são capazes de aprender representações robustas utilizando amostras de dados de diversos (geralmente milhares) locutores. Tais técnicas têm o objetivo de gerar representações que são capazes de discriminar os locutores e também capazes de lidar com a variabilidade de contextos onde as locuções são produzidas.

1.5 VISÃO GERAL DO ESTADO DA ARTE

Nos sistemas de verificação de locutores modernos, o processo de decisão é definido por duas etapas principais. Na primeira etapa, uma locução de duração arbitrária é mapeada para uma representação de dimensão fixa. Para uma determinada locução, é atribuído um vetor de características de dimensão d, $\boldsymbol{x} \in \mathbb{R}^d$. O objetivo principal dessa etapa consiste na produção de representações que sejam capazes de discriminar diferentes locutores. Além disso, também é desejável que as representações produzidas por um mesmo locutor não possuam uma alta variabilidade para locuções diferentes (possivelmente produzidas em contextos diferentes, na presença de fatores de incompatibilidade). A segunda etapa consiste então da decisão sobre a identidade alegada comparando a representação correspondente à locução de teste com as representações extraídas das locuções de cadastro do usuário em questão. Isso é realizado empregando um método que decide se dois vetores $\boldsymbol{x_i}$ e $\boldsymbol{x_j}$ foram gerados pelo mesmo locutor ou não.

Através dos anos, a abordagem mais bem sucedida consistia na representação conhecida como vetor-identidade ou *i-vector* (*identity vector*) (DEHAK et al., 2011). Nessa abordagem, primeiramente são extraídas das locuções características de tempo curto, como os Coeficientes Mel-cepstrais (*Mel-Frequency Cepstral Coefficients* - MFCCs), de segmentos de voz com duração entre 20-30ms. Um modelo probabilístico chamado de Modelo Universal de Fundo (*Universal Background Model* - UBM) é então estimado utilizando vetores MFCC provenientes de diversos locutores. O UBM consiste de um Modelo de Misturas Gaussianas (*Gaussian Mixture Model* - GMM) com tipicamente milhares de componentes.

Para uma determinada locução, os vetores MFCCs são utilizados para calcular as estatísticas de ordem zero e de primeira ordem com relação ao UBM. Apesar de possuir dimensão altíssima, essas estatísticas definem uma representação de tamanho fixo para uma locução inicialmente descrita por uma quantidade variável de vetores MFCCs. O *i-vector* é definido pelo resultado de um mapeamento entre essas estatísticas para um espaço de

dimensão baixa (tipicamente centenas de características). Essa projeção consiste de uma decomposição fatorial não supervisionada do espaço original e preserva informações importantes para a discriminação de diferentes locuções. Apesar de os *i-vectors* poderem ser efetivamente comparados entre si através da similaridade do cosseno, na prática técnicas de pós-processamento como normalização de comprimento e análise de discriminantes lineares (*Linear Discriminant Analysis* - LDA) são empregados para aumentar o poder discriminante dos *i-vectors*.

Por fim, a decisão final do sistema é realizada através de um modelo discriminativo, a análise probabilística de discriminantes lineares (*Probabilistic Linear Discriminant Analysis* - PLDA) que realiza um teste estatístico sobre a hipótese de que dois *i-vectors* foram gerados pelo mesmo locutor (KENNY, 2010; PRINCE; ELDER, 2007). Em termos mais precisos, um caso especial da PLDA é utilizado, onde a componente correspondente ao espaço dos locutores e a componente relativa às variabilidades intra-locutores são assumidas seguirem distribuições normais. Esse modelo é conhecido como PLDA gaussiano (*Gaussian Probabilistic Linear Discriminant Analysis* - G-PLDA).

Mais recentemente, devido ao sucesso alcançado por métodos de aprendizado profundo (*Deep Learning* - DL) em outros problemas envolvendo sinais de voz, como reconhecimento automático de fala (*Automatic Speech Recognition* - ASR), modelos baseados em redes neurais profundas (*Deep Neural Networks* - DNNs) têm sido aplicados para reconhecimento de locutores. Em um primeiro momento, ganhos de desempenho foram observados ao incorporar informações fonéticas extraídas de DNNs treinadas para ASR. As componentes fonéticas ou eram incorporadas às estatísticas do UBM no processo de decomposição para a geração dos *i-vectors* (LEI et al., 2014a; KENNY et al., 2014) ou eram diretamente combinadas com os *i-vectors* para geração de um único vetor de características (MCLAREN; LEI; FERRER, 2015). Apesar do avanço proporcionado por essas novas informações, essa abordagem requer a disponibilidade das informações fonéticas presentes nas locuções (transcrições dos áudios), o que aumenta significativamente a complexidade no desenvolvimento dos sistemas.

A abordagem que surgiu em seguida se tornou a mais popular e mais bem sucedida, que consiste no uso de DNNs supervisionadas para aprender representações a partir da classificação das locuções com respeito aos locutores. Nesse tipo de abordagem, o modelo é treinado para realizar a tarefa de classificação e depois é utilizado para geração das representações das locuções utilizando as saídas de uma determinada camada escondida[2]. Os primeiros avanços surgiram em verificação de locutores dependente de texto (VARIANI et al., 2014), onde um modelo DNN foi treinado para classificar características de tempo curto das locuções. Após o treinamento, a camada de classificação é retirada e o modelo gera uma representação para cada segmento de tempo curto e uma única representação

[2] Neste trabalho, referenciamos as representações geradas por uma DNN como "representações profundas".

para a locução inteira é definida pelo vetor médio dos segmentos. Essa representação é referenciada como *d-vector* e apresentou bons desempenhos quando comparado com o *i-vector* utilizando a similaridade do cosseno.

Em seguida, um sistema fim-a-fim foi proposto em (HEIGOLD et al., 2016), onde um modelo DNN é treinado utilizando pares de locuções para decidir se elas foram produzidas pelo mesmo locutor ou não. Esse modelo foi então adaptado para verificação de locutores independente de texto em (SNYDER et al., 2016). As entradas dos modelos também são definidas por características de tempo curto e, por essa razão, descritas em um espaço de dimensão variável. Porém, foi introduzida uma camada escondida de *pooling* temporal, cuja saída possui tamanho fixo para a locução inteira. Essa representação intermediária é processada pelo restante da rede, que realiza a decisão para os pares de vetores. Após o treinamento, o modelo é diretamente utilizado para comparar as locuções de cadastro e de teste, realizando assim a decisão do sistema. Essa abordagem fim-a-fim, combinado com o *pooling* temporal, resultou em um avanço em termos de desempenho, apresentando em alguns casos resultados melhores que aqueles alcançados pelos *i-vectors*. Porém, para essa abordagem, existe a dificuldade proveniente do fato de o modelo ser treinado utilizando pares de locuções. Para enfrentar as incompatibilidades resultantes da diferença de contexto de treinamento e teste do sistema, uma grande quantidade de dados é necessária.

Snyder *et al.* propuseram, então, o retorno à abordagem anterior, dividindo a modelagem em duas partes: (i) utilizar um modelo DNN similar para aprender as representações dos locutores e (ii) decidir se duas representações distintas pertencem ao mesmo locutor utilizando a modelagem G-PLDA (SNYDER et al., 2017). Essa proposta se aproveita do poder de geração de boas representações dos modelos DNNs sem lidar com os problemas envolvidos na decisão sobre pares de locuções. A entrada do modelo em questão consiste dos MFCCs, que são processados pelas chamadas camadas em nível de janela (*frame-level layers*), mantendo assim a dimensionalidade variável da representação. Uma camada de *pooling* estatístico é então utilizada para agregar as representações das janelas, computando os vetores de média e desvio-padrão das representações temporais produzidas pela camada anterior, produzindo assim um vetor intermediário de dimensão fixa para cada locução. Tais vetores são processados pelas camadas restantes, chamadas de camadas em nível de sequência (*sequence-level layers*), até a camada de saída que define as probabilidades associadas à classificação dos locutores. A representação aprendida, chamada de *x-vector*, é extraída de uma das camadas de sequência, posteriores ao *pooling*, possuindo dimensão fixa independentemente da duração da locução. Em (SNYDER et al., 2018), os autores apresentaram a robustez dos *x-vectors* em relação aos *i-vectors*, especialmente com o aumento dos dados de treinamento (*data augmentation*), alcançando um desempenho consideravelmente superior.

Similarmente aos *i-vectors*, métodos de pós-processamento, tais como o LDA e normalização de comprimento (*Length Normalization* - LN), são empregados e a decisão

do sistema é realizada através da modelagem G-PLDA. Porém, como demonstrado em (ZHANG; LI; WANG, 2019), enquanto que a etapa de pós-processamento dos *i-vectors* possui o objetivo tornar as representações mais discriminantes, para os *x-vectors* o objetivo dessa etapa é o de regularização, dado que os *x-vectors* já são naturalmente discriminativos. Isso ocorre porque a modelagem G-PLDA assume que as probabilidades condicionais e *a priori* dos vetores dos locutores seguem uma distribuição normal. Como os *x-vectors* são treinados apenas com o objetivo de distinguir locuções de diferentes locutores, nenhum controle é imposto sobre as distribuições dos vetores. Os métodos de pós-processamento então mapeiam os vetores para um espaço mais adequado ao G-PLDA, o que resulta em um ganho de desempenho. Mesmo com esse ganho de desempenho, tais métodos não são completamente adequados, uma vez que eles não impõem restrições às distribuições geradas.

Algumas abordagens foram propostas para a geração de *x-vectors* mais adequados à modelagem G-PLDA. Mais especificamente, essas abordagens possuem o objetivo de fazer com que as representações sigam distribuições normais. Em (LI et al., 2019), os autores propuseram a adição de um termo de regularização à função de custo da DNN que minimiza a norma L2 entre os *x-vectors* e o peso da camada de saída ($\boldsymbol{\theta_S}$), associado ao locutor correspondente, S. Os autores argumentam que, utilizando um fator de importância suficientemente grande à regularização, $\boldsymbol{\theta_S}$ converge para o *x-vector* médio associado ao locutor e a distribuição dos *x-vectors* converge para $N(\boldsymbol{\theta_S}, \mathbf{I})$. De fato, a abordagem se mostrou eficaz para ambas as representações *d-vectors* e *x-vectors*. Porém, diferentes fatores de importância foram atribuídos para a função de regularização e todos os resultados apresentados foram alcançados ao empregar LDA às novas representações, o que dificulta a análise do efeito da regularização nas distribuições das representações.

Já em (ZHANG; LI; WANG, 2019), um segundo modelo DNN foi proposto para projetar *x-vectors* já treinados em um novo espaço mais compacto e com distribuição controlada. O modelo consiste de um Auto-codificador Variacional (*Variational Autoencoder* - VAE) (KINGMA; WELLING, 2013), que, assim como os auto-codificadores (*autoencoders*), é um modelo não-supervisionado (ou auto-supervisionado) treinado para reconstruir a própria entrada, gerando para isso uma representação intermediária, geralmente mais compacta que a original. Além disso, VAEs possuem uma função de regularização variacional definida pela divergência entre as distribuições das variáveis intermediárias e uma distribuição paramétrica desejada (nesse caso, uma distribuição gaussiana padrão).

Essa abordagem foi proposta como uma alternativa aos métodos de pós-processamentos das representações, uma vez que tanto regulariza o espaço dos *x-vectors* como também realiza uma redução de dimensionalidade. Os resultados apresentaram um ganho de desempenho em relação aos demais métodos, porém, não ao ponto de justificar o treinamento de um segundo modelo apenas para esse propósito. Os autores então apresentaram mais tarde uma utilidade mais apropriada para esse segundo modelo: o de adaptação de do-

mínio (WANG; LI; WANG, 2019). Nesse cenário, a abordagem é utilizada para projetar *x-vectors* extraídos de um modelo treinado em um determinado domínio (utilizando locuções telefônicas, por exemplo) em um espaço mais adequado para o domínio onde a tarefa ocorrerá (cadastramento e teste utilizando locuções gravadas com microfones, por exemplo).

Em suma, a abordagem adotada para a geração dos *x-vectors* mostrou-se bastante apropriada para verificação de locutores independente de texto. O modelo DNN é capaz de produzir representações robustas em relação aos *i-vectors* e, além disso, a realização da tarefa utilizando a modelagem G-PLDA é ainda a mais bem sucedida. Entretanto, apesar do seu poder discriminativo, o fato de não haver imposições sobre o espaço das representações acarreta na geração de distribuições que não são completamente apropriadas para a utilização do G-PLDA.

1.6 OBJETIVOS

Diante do exposto, o objetivo principal deste trabalho é o desenvolvimento de abordagens para a geração de representações de melhor qualidade baseadas nos *x-vectors*. Consideramos a qualidade das representações das locuções através da adequação do espaço à modelagem G-PLDA, que, por sua vez, assume que um determinado *x-vector*, $\boldsymbol{x}$, segue uma distribuição cuja decomposição em fatores é descrita por:

$$\boldsymbol{x} = \boldsymbol{x_m} + \boldsymbol{\Phi\beta} + \boldsymbol{\epsilon_r}, \tag{1.2}$$

onde $\boldsymbol{x_m}$ é o *x-vector* médio da população, $\boldsymbol{\Phi}$ é uma matriz retangular que mapeia o *x-vector* para a componente correspondente ao locutor $\boldsymbol{\beta}$ e $\boldsymbol{\epsilon_r}$ é a componente redisual, correspondente às variabilidades intra-locutor. Além disso, o G-PLDA assume que $\boldsymbol{\beta}$ segue uma distribuição normal padronizada e que $\boldsymbol{\epsilon_r}$ segue uma distribuição normal com média zero e matriz de covariância $\boldsymbol{\Sigma}$. Dessa maneira, a adequação à modelagem pode ser descrita através das distribuições associadas aos vetores produzidos pelos locutores.

Dessa maneira, podemos definir algumas características desejadas para as representações geradas através das locuções. Dado o conjunto de locutores $S = \{S_i\}$, i = 1, 2, ..., observados durante a fase de treinamento, e os conjuntos de representações das locuções produzidas por eles $X = \{X_i\}$, i = 1, 2, ..., idealmente temos que:

(i) $\boldsymbol{x} \in X$ segue uma distribuição normal;

(ii) para $\boldsymbol{x} \in X_i$, $P(\boldsymbol{x}|S_i) > 0$;

(iii) para $\boldsymbol{x} \notin X_i$, $P(\boldsymbol{x}|S_i) = 0$;

(iv) para $\boldsymbol{x} \in X_i$, $p(\boldsymbol{x}|S_i)$ é uma distribuição normal.

Além disso, estamos aqui assumindo que os vetores $\boldsymbol{x} \in X$ possuem dimensionalidade suficiente para que as condições expostas sejam satisfeitas. Enquanto que as condições (ii) e (iii) definem a capacidade de discriminação entre diferentes locutores, a condição (iv) determina a distribuição dos vetores gerados por um mesmo locutor. As disposições das representações geradas pelos locutores são definidas pelas distribuições normais correspondentes, com médias e desvios-padrão definidas de maneira que a distinção entre diferentes locutores possa ser realizada através das verossimilhanças. Já a condição (i) resulta de algumas premissas da modelagem G-PLDA, que considera que o espaço dos vetores pode ser decomposto em duas componentes estatisticamente independentes e que seguem distribuições normais. Pode-se observar que o treinamento dos *x-vectors* leva em consideração apenas as condições (ii) e (iii), uma vez que a DNN é treinada com o objetivo apenas de aumentar a discriminação entre os locutores.

O foco deste trabalho consiste no desenvolvimento de métodos para treinar um modelo DNN observando todas as condições simultaneamente, ou seja, métodos que possuem o objetivo de controlar as distribuições do espaço de representações gerado sem perder o poder discriminativo já alcançado pelas técnicas atuais. Diante disso, como objetivos específicos, podemos citar:

- Desenvolvimento de um método que proporcione a geração de um espaço de representações onde os vetores dos locutores se distribuam seguindo uma distribuição normal;
- Desenvolvimento de um método para restringir a distribuição *a priori* das representações geradas pelo modelo DNN;
- Combinação dos métodos desenvolvidos mantendo a capacidade discriminatória do modelo;
- Analisar o ganho de desempenho alcançado pelos métodos, utilizando a modelagem G-PLDA para a realização da tarefa de verificação.

Neste trabalho, tais objetivos são alcançados a partir do desenvolvimento de abordagens para o controle das distribuições *a priori* e condicionadas aos locutores dos vetores gerados a partir de um modelo DNN baseado naquele utilizado para a extração dos *x-vectors*. Diferentes abordagens são desenvolvidas para cada um dos tipos de controle e diferentes hipóteses são associadas a elas. Para o controle das distribuições condicionadas aos locutores, levantamos a hipótese de que isso pode ser alcançado ao mudar a maneira como as informações são codificadas pela rede neural. Em seguida, apresentamos a hipótese de que, ao controlar a distribuição *a priori* dos vetores, estamos indiretamente realizando uma imposição sobre a componente independente de locutor assumida pela modelagem G-PLDA (componente $\boldsymbol{\epsilon_r}$ da Equação 1.2). Tais hipóteses estão expostas em detalhes na Seção 3.1.

1.7 CONTRIBUIÇÕES

As contribuições deste trabalho podem ser divididas em duas partes. A primeira delas diz respeito ao desenvolvimento de técnicas capazes de gerar um espaço de representações para as locuções onde os vetores associados a um determinado locutor seguem uma distribuição normal. Para isso, seguimos a abordagem de restringir os espaços gerados em algumas camadas da rede para concentrarem informações em distribuições gaussianas. Nesse contexto, duas foram as contribuições deste trabalho:

- Utilização de uma camada de classificação gaussiana, composta por nós definidos por funções de base radial (*Radial Basis Functions* - RBFs), modelando assim as distribuições das representações dos locutores, na última camada, como gaussianas;
- Desenvolvimento de uma camada de *pooling* gaussiano, a partir de uma camada composta por RBFs. Tal camada modela o espaço das representações temporais através de um modelo de mistura de RBFs. A agregação das representações temporais é realizada através da computação das estatísticas de ordem zero e primeira ordem, levando em consideração as adequações dos vetores a cada uma das componentes de mistura.

Ao utilizar ambas as camadas durante o treinamento da rede, um controle sobre o espaço das representações é realizado tanto sobre as representações temporais geradas pela primeira parte da rede quanto no espaço final onde a discriminação entre os locutores ocorre.

Já a segunda parte das contribuições diz respeito ao controle sobre distribuição *a priori* das representações profundas geradas pelo modelo. Nesse contexto, a contribuição deste trabalho está no desenvolvimento de um termo de regularização para a função de custo da rede seguindo uma abordagem variacional, que minimiza a divergência entre a distribuição das representações geradas e uma distribuição *a priori* desejada. A função de divergência utilizada foi a Máxima Divergência Média (*Maximum Mean Discrepancy* - MMD), que define um teste de hipóteses não paramétrico entre as amostras das distribuições. Nessa abordagem, uma amostra da distribuição desejada é apresentada à rede como uma nova entrada. Dentre as vantagens da técnica proposta podemos citar:

(i) Assim como em (ZHANG; LI; WANG, 2019), a distribuição dos vetores é controlada através de uma abordagem variacional[3]. Porém, ao invés de treinar um segundo modelo para a regularização, o termo variacional é adicionado ao modelo convencional, regularizando os vetores durante o treinamento de um único modelo.

(ii) Uma vez que o MMD é uma função de divergência não paramétrica, nenhum parâmetro de distribuição é inferido pela rede e nenhum esquema de amostragem é

[3] Os autores realizaram a regularização dos *x-vectors* através de um modelo VAE (Seção 1.5).

necessário (como ocorre em um VAE). A comparação entre as distribuições é realizada de maneira direta, utilizando uma amostra da distribuição desejada e os vetores gerados pela rede.

(iii) Como a amostra da distribuição desejada é apresentada como uma entrada do modelo, ele se torna flexível com respeito à distribuição desejada. Nesse sentido, diferentes distribuições podem ser experimentadas facilmente.

As duas abordagens propostas neste trabalho foram desenvolvidas com o intuito de controlar aspectos distintos do espaço de representações. Além disso, elas foram desenvolvidas para que possam ser utilizadas juntas, realizando assim um controle conjunto da distribuição *a priori* dos *x-vectors* e das distribuições condicionadas aos locutores. As abordagens propostas foram avaliadas e comparadas com as abordagens presentes na literatura. Em termos de desempenho, todas apresentaram ganhos de desempenho em relação à modelagem convencional. A combinação entre as abordagens alcançou ganhos significativos e superiores às demais técnicas presentes na literatura. Além da análise dos desempenhos dos sistemas, também realizamos uma análise visual do espaço gerado através das representações, onde ficou evidenciado o ganho de qualidade nas distribuições dos vetores com a utilização das abordagens propostas.

1.8 ORGANIZAÇÃO DO DOCUMENTO

O próximo capítulo descreve as principais técnicas propostas para o desenvolvimento de sistemas de verificação de locutores independente de texto. No Capítulo 3, são apresentadas as técnicas propostas neste trabalho e as hipóteses levantadas para o desenvolvimento das mesmas. Os experimentos realizados neste trabalho são descritos no Capítulo 4. Os experimentos foram divididos em etapas que possuem objetivos específicos. Além de descrever a base de dados utilizada, esse capítulo explicita, analisa e compara os desempenhos apresentados pelas técnicas descritas nos Capítulos 2 e 3. Finalmente, as conclusões e considerações finais deste trabalho são apresentadas no Capítulo 5.

2 VERIFICAÇÃO DE LOCUTORES INDEPENDENTE DE TEXTO

Este capítulo apresenta a literatura sobre sistemas de verificação de locutores independente de texto. Partindo da definição do problema, a arquitetura clássica de um sistema desse tipo é mostrada. Em seguida são apresentadas as principais técnicas utilizadas em cada uma das etapas que compõem os sistemas. A Seção 2.2 se concentra nas técnicas de pré-processamento das locuções, enquanto que a Seção 2.3 foca nos métodos de extração de características, dando uma atenção maior aos MFCCs. Na Seção 2.4, apresentamos os métodos desenvolvidos para a modelagem dos locutores, desde o método GMM-UBM, passando pela abordagem GMM-SVM, até a utilização dos chamados *i-vectors*. Já na Seção 2.5, apresentamos os principais métodos desenvolvidos seguindo a abordagem de aprendizagem profunda (DL), até a abordagem considerada atualmente o estado da arte, que consiste das representações conhecidas como *x-vectors*. Por fim, descrevemos, na Seção 2.6, os métodos utilizados para normalização dos *scores* do sistema.

2.1 DEFINIÇÃO

Em verificação de locutores, a partir de uma determinada locução e de uma alegação a respeito da identidade da pessoa que a produziu, a tarefa consiste em decidir se a alegação está correta ou não. Basicamente, a tarefa consiste em inferir se uma locução foi produzida por um locutor específico ou não. Dessa maneira, dado uma locução X e um determinado locutor S, a tarefa do sistema pode ser definida por um teste de hipóteses entre:

$$H_0 = X \textbf{ foi} \text{ produzida por S.} \tag{2.1}$$

$$H_1 = X \textbf{ não foi} \text{ produzida por S.} \tag{2.2}$$

Além disso, como estamos tratando de sistemas independentes de texto, tal decisão é realizada a despeito do conteúdo fonético presente na locução. Isto é, a decisão deve ser realizada seja qual for o conjunto de palavras pronunciadas em X.

Na abordagem clássica, as hipóteses nula e alternativa (Equações 2.1 e 2.2, respectivamente) são modeladas de maneira explícita, de modo que seja possível o cálculo de suas verossimilhanças ($p(X|H_0)$ e $p(X|H_1)$, respectivamente) com respeito à locução de teste. Nesse caso, o teste ótimo de decisão consiste no teste de razão das verossimilhanças:

$$\frac{p(X|H_0)}{p(X|H_1)} = \begin{cases} \geq \theta, & \text{não rejeite } H_0, \\ < \theta, & \text{rejeite } H_0, \end{cases} \tag{2.3}$$

onde $\theta \in \mathbb{R}$ é um limiar de rejeição.

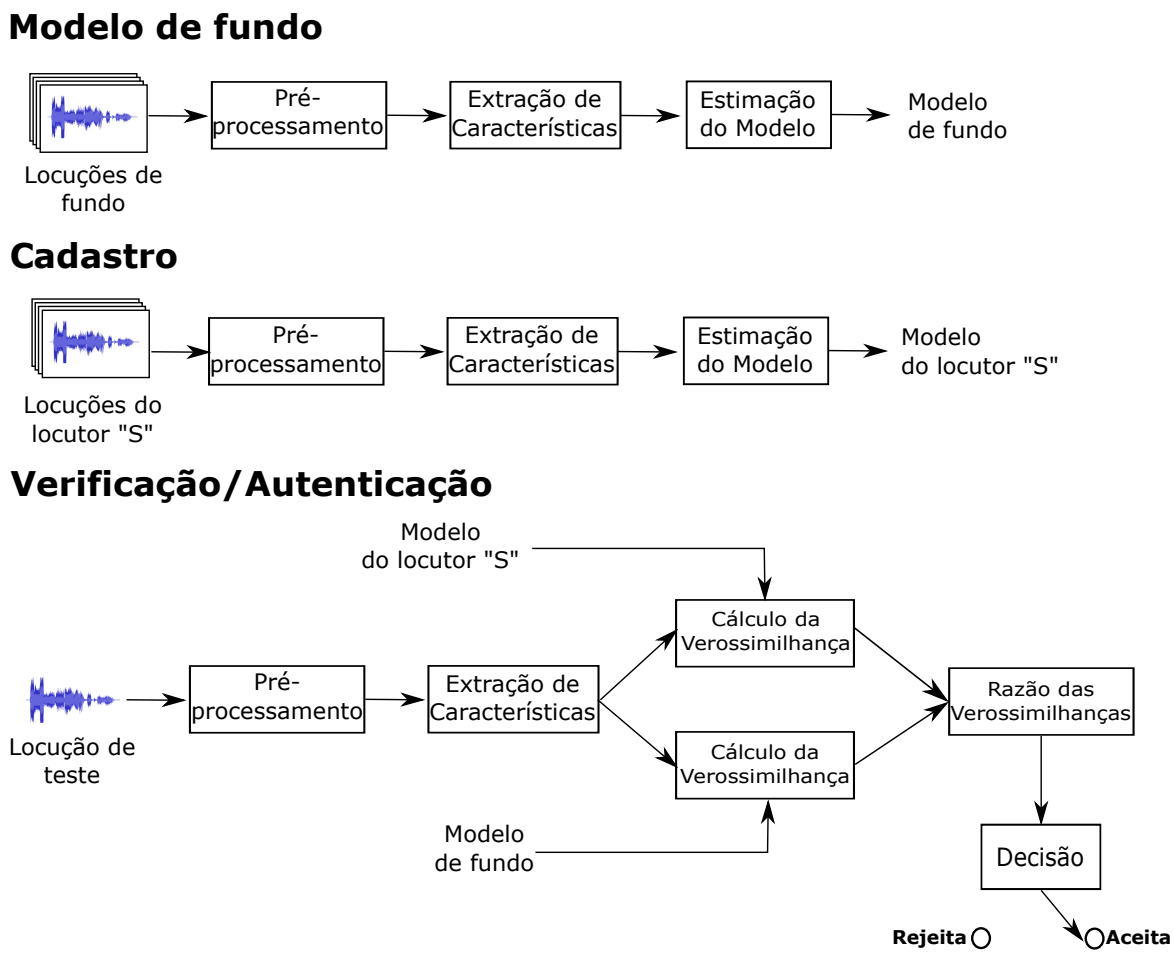

Figura 4 – Arquitetura clássica de um sistema de verificação de locutores independente de texto.

Mais precisamente, esse teste é ótimo apenas para o caso onde as distribuições de probabilidade são conhecidas, o que raramente ocorre. Dessa maneira, os sistemas realizam estimações a respeito das distribuições das hipóteses e utilizam diferentes abordagens matemáticas para realizar a comparação das verossimilhanças associadas a elas.

A Figura 4 mostra a arquitetura clássica de um sistema de verificação de locutores independente de texto. A modelagem da hipótese alternativa geralmente ocorre antes de qualquer cadastro ser realizado. O modelo correspondente a essa hipótese é chamado de modelo de fundo (*background model*) (ou modelo independente do locutor) e é estimado utilizando locuções provenientes de diversos locutores (locuções de fundo). Na fase de cadastro, ocorre a estimação do modelo associado à hipótese nula. Tal modelo é referenciado como modelo do locutor e é estimado utilizando locuções de treino produzidas pelo locutor sendo cadastrado. Na fase de verificação, as verossimilhanças das hipóteses são computadas utilizando uma determinada locução de teste. Tais verossimilhanças são então comparadas para a realização da decisão final.

Como visto no capítulo anterior, as locuções são mapeadas para um espaço de descri-

ção, definido por um vetor (ou um conjunto de vetores) de características. O mapeamento de uma determinada locução para esse espaço é realizado através da etapa de extração de características. Possivelmente, o sinal digital da locução pode ser inicialmente processado a fim de que certas informações presentes nele possam ser enfatizadas (ou mitigadas). Essa função é desempenhada pela etapa de pré-processamento.

2.2 PRÉ-PROCESSAMENTO

A primeira etapa do sistema consiste no pré-processamento do sinal de voz. Nesta etapa, são aplicadas técnicas que visam melhorar a qualidade do sinal ou acentuar determinadas características importantes para o processo de extração de características. Em um sistema de reconhecimento de locutores, duas são as principais operações realizadas nessa etapa: a pré-ênfase e a detecção de voz.

2.2.1 Pré-ênfase

Nos sinais de voz, ocorre uma queda acentuada na potência do sinal nas altas frequências. Aproximadamente 80% da potência total de um sinal de voz está concentrada em frequências abaixo de 1KHz (entre 1-8KHz, a cada dobro de frequência (uma oitava) há um decaimento de 12dB de potência)(DENG; O'SHAUGHNESSY, 2003). Mesmo com baixa potência, a percepção auditiva humana é capaz de captar bem tais faixas de frequência através de um mecanismo de retorno do cérebro que as amplifica. A pré-ênfase do sinal se propõe a algo parecido, enfatizando as altas frequências de modo que seja possível a utilização das características importantes presentes nelas. Um método bastante comum consiste na utilização de um filtro diferenciador (passa-alta) de primeira ordem. Para um sinal de voz amostrado no tempo $s[n]$, o processo de pré-ênfase é definido por

$$\tilde{s}[n] = s[n] - \alpha \times s[n-1], \tag{2.4}$$

onde $\tilde{s}[n]$ é o sinal temporal resultante e α é o coeficiente do filtro. Valores comumente utilizados para α estão entre $0,95$ e $0,97$ (VERGIN; O'SHAUGHNESSY, 1995). Ao se aplicar tal filtro, as potências absolutas são reduzidas, mas melhor distribuídas no espectro.

2.2.2 Detecção de atividade de voz

Técnicas que detectam a atividade da voz (*Voice Activity Detectors* - VADs) são utilizadas para descartar as partes do sinal que não possuem voz. Essas partes geralmente possuem silêncio ou ruído de fundo e degradam o desempenho do sistema, uma vez que não possuem informação útil a respeito do locutor que produziu o sinal. Eles são particularmente importantes em sistemas de reconhecimento de locutores quando as locuções utilizadas possuem longos períodos de silêncio. Apesar de parecer um problema simples de classificação binária, o desenvolvimento de detectores que funcionam bem para os mais

diversos ambientes é bastante desafiador, especialmente quando o ambiente apresenta alto nível de ruído.

Geralmente, as técnicas dividem o sinal de voz em janelas de tamanhos curtos (10-30ms) e realizam a decisão em cada janela de maneira independente. Uma solução simples que possui desempenho satisfatório para locuções telefônicas utiliza a energia e a taxa de passagem pelo zero (*Zero Crossing Rate* - ZCR) do sinal na janela para realizar a detecção. Limiares são utilizados para identificar se um determinado segmento do sinal possui voz ou não. Tais limiares são dinâmicos e se adaptam de acordo com as estimativas temporais do ruído presente no sinal (REYNOLDS; ROSE; SMITH, 1992).

Segmentos que possuem energia abaixo de um determinado limiar são considerados segmentos de silêncio. Se um determinado segmento não é descartado pela análise da energia, ele é então analisado pelo ZCR, que é definido como

$$ZCR = \frac{1}{N} \sum_{n=0}^{N-1} \chi(h[n] \times h[n-1]), \tag{2.5}$$

onde $h[\cdot]$ é um segmento de tamanho N do sinal e $\chi(\cdot)$ é uma função definida como

$$\chi(x) = \begin{cases} 1, & \text{se } x < 0, \\ 0, & \text{se } x \geq 0. \end{cases} \tag{2.6}$$

Essa medida é geralmente utilizada para descartar partes do sinal que possuem apenas ruído de fundo. Nesse caso, o ZCR apresenta valores tipicamente superiores àqueles apresentados pelos segmentos que possuem apenas voz, cuja periodicidade é limitada e definida através de bandas de frequências conhecidas. Exemplos de técnicas que utilizam tais medidas podem ser vistos em (HARSHA, 2004) e em (REYNOLDS, 1992).

Outras abordagens foram propostas como alternativas para a utilização dos limiares de decisão. Um exemplo desse tipo de alternativa segue uma abordagem estatística, onde a decisão é realizada a partir de um teste de razão de verossimilhanças (SOHN; KIM; SUNG, 1999; HARSHA, 2004). Outras abordagens se baseiam na transformada Wavelet (STADTSCHNITZER; PHAM; CHIEN, 2008) e na periodicidade dos segmentos do sinal (HAUTAMÄKI et al., 2007). Além disso, algumas aplicações requerem um processamento dos sinais de voz em tempo real, de maneira que a classificação dos segmentos deve ser realizada localmente, sem analisar o sinal completo. Um exemplo desse tipo de VAD é o chamado método de divergência espectral de longo prazo (*Long-Term Spectral Divergence* - LTSD). O método LTSD e outras técnicas de VAD existentes na literatura são descritos mais apropriadamente em (RAMIREZ et al., 2004).

2.3 EXTRAÇÃO DE CARACTERÍSTICAS

Muitas características podem ser extraídas de um determinado sinal de voz, porém para o desenvolvimento de sistemas de reconhecimento de locutores foca-se nas que são im-

portantes para a distinção entre diferentes locutores. Nesse contexto a característica ideal deve (ROSE, 2003; WOLF, 1972):

- possuir alta variabilidade entre locutores e baixa variabilidade em um mesmo locutor;
- ser robusta quanto a distorções (Seção 1.4.3);
- ocorrer naturalmente e frequentemente na fala;
- ser fácil de medir em um sinal de voz;
- ser difícil de fraudar (imitar);
- e não ser afetada pelas condições de saúde do locutor ou pelas variações a longo prazo de sua voz.

Na prática, é bem improvável encontrar algum conjunto de características que possua tais atributos simultaneamente. Dependendo da aplicação, é necessária a flexibilização de alguns desses requisitos.

Existem diferentes maneiras de se categorizarem os tipos de características que podem ser extraídas de um sinal de voz. De uma maneira geral, elas podem ser divididas entre características de baixo e de alto nível. As características de baixo nível focam na extração de informações que refletem o aparato vocal humano. Por outro lado, as características de alto nível se concentram nos aspectos comportamentais do locutor, como os aspectos linguísticos, que podem ser aprendidos ou mudados com o passar do tempo.

De maneira geral, as características de baixo nível são mais fáceis de extrair, uma vez que são calculadas a partir de pequenas partes do sinal de voz, o que possibilita suas utilizações em aplicações de tempo real (KINNUNEN; LI, 2010). Por outro lado, elas são facilmente distorcidas por ruídos provenientes do ambiente ou do canal de comunicação utilizado. As características de alto nível, por sua vez, são mais robustas quanto a esses tipos de ruídos (ADAMI, 2005). Porém, além de serem mais difíceis de extrair, elas geralmente necessitam de uma grande quantidade de dados e de técnicas sofisticadas para serem modeladas (CAMPBELL et al., 2007).

Sob o ponto de vista físico, uma taxonomia mais elaborada (KINNUNEN; LI, 2010) divide as características em (i) espectrais de tempo curto, (ii) de fonte de voz, (iii) espectro-temporais, (iv) prosódicas e (v) de alto nível (ver Figura 5). As características de fontes de voz e espectrais de tempo curto focam nas informações fisiológicas do locutor, enquanto que as de alto nível extraem informações comportamentais do indivíduo. As características prosódicas e espectro temporais medem informações de ambas as fontes.

Como o nome sugere, as características espectrais de tempo curto são calculadas a partir de segmentos curtos do sinal, com duração entre 20 e 30 milissegundos. Tais ca-

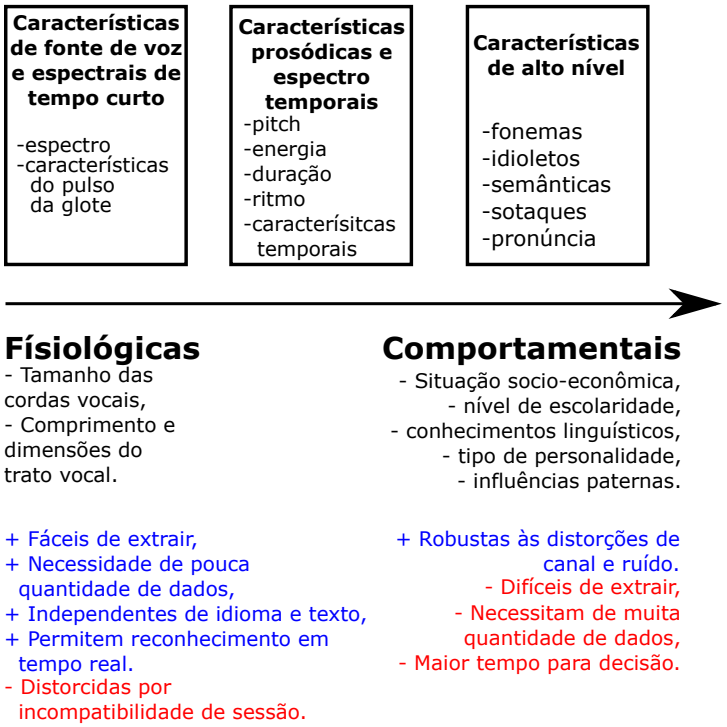

Figura 5 – Categorização sob o ponto de vista físico dos tipos de características que podem ser extraídas do sinal de voz. Imagem inspirada em (KINNUNEN; LI, 2010).

racterísticas são utilizadas como descritores dos envelopes espectrais de tempo curto, que estão associados acusticamente ao timbre e às ressonâncias produzidas no trato vocal.

As características de fonte de voz extraem medidas do sinal de excitação da glote, como o formato do pulso, frequência fundamental e a taxa da vibração. Apesar de apresentarem razoável poder de discriminação entre locutores distintos, tais características da glote não são diretamente mensuráveis, em função da filtragem que ocorre no trato vocal. Essas medidas são comumente calculadas utilizando técnicas de filtragem inversa após a extração das características espectrais de tempo curto, medindo, assim, a influência do trato vocal no sinal de voz (KINNUNEN; ALKU, 2009). Várias abordagens podem ser utilizadas para realizar a filtragem inversa. Alguns exemplos incluem redes neurais (PRASANNA; GUPTA; YEGNANARAYANA, 2006), modelos probabilísticos paramétricos (PLUMPE; QUATIERI; REYNOLDS, 1999) e coeficientes cepstrais (GUDNASON; BROOKES, 2008). Tais características não são tão discriminativas quanto às espectrais de tempo curto, porém, estudos mostram que combinações entre elas podem aumentar o desempenho do sistema (ZHENG; LEE; CHING, 2007).

Características espectro-temporais calculam medidas temporais das características espectrais de tempo curto, como velocidade e aceleração. Elas não extraem informações específicas do trato vocal, mas analisam o comportamento temporal do sinal produzido por ele. De maneira geral, tais medidas são anexadas ao conjunto final de características para aumentar o desempenho do sistema (FURUI, 1981).

As características prosódicas dizem respeito àquelas que analisam aspectos globais da locução, como estilo de fala, padrões de entonação, velocidade de fala (em termos de palavras), ritmo, emoções, duração (estatísticas de pausas entre as palavras), entre outros. Evidentemente, tais características devem analisar partes mais longas da locução. Características baseadas na frequência fundamental e nas distribuições de energia do sinal são alguns exemplos dessas medidas. Mais detalhes sobre esse tipo de característica podem ser vistos em (SHRIBERG et al., 2005; ADAMI, 2005; ADAMI et al., 2003).

Apesar de as características de baixo nível permitirem o desenvolvimento de sistemas com desempenho satisfatório para uma vasta gama de aplicações, elas ignoram características léxicas do locutor. Tais características léxicas são os principais exemplos de características de alto nível. O estudo a respeito da utilização desse tipo de características começou em (DODDINGTON et al., 2001), com a análise dos idioletos, que são variações de uma língua manifestadas por padrões de escolha de palavras, gramáticas, frases ou metáforas. A ideia por trás da utilização desse tipo de característica está na descrição de uma locução a partir de um conjunto de *tokens*, que caracterizam idioletos específicos. Seus padrões de ocorrência seriam, então, utilizados para diferenciar locutores. Alguns exemplos de *tokens* seriam palavras, fonemas ou gírias. Com a disponibilidade de técnicas que possam identificar idioletos (como algumas técnicas utilizadas para reconhecimento de fala), a incorporação desse tipo de característica como complemento às de baixo nível melhora, em alguns contextos, os desempenhos dos sistemas de reconhecimento de locutores (CAMPBELL; REYNOLDS; DUNN, 2003).

A escolha das características a serem extraídas do sinal de voz depende da aplicação, dos recursos computacionais e da quantidade de dados disponíveis tanto para a fase de cadastramento quanto para a de reconhecimento. Porém, muitos estudos comparativos revelaram que as características fisiológicas são as mais efetivas para o reconhecimento de locutores. Particularmente, aquelas características baseadas no espectro da voz e no *pitch*[1] (REYNOLDS, 1992; WOLF, 1972; SAMBUR, 1975).

Segundo a teoria da produção de fala (FLANAGAN, 1972), o sinal de voz é produzido a partir da corrente de ar que atravessa as cordas vocais e passa pela glote, produzindo ressonantes no trato vocal e nas cavidades oral e nasal. Se por um lado o *pitch* é extremamente afetado por fatores não-fisiológicos (como estado emocional, por exemplo) (DODDINGTON, 1985), por outro o espectro da voz reflete a estrutura anatômica do trato vocal e das cavidades e se mostrou mais bem sucedido na extração de informações provindas de atributos únicos do locutor.

A seguir são descritas as técnicas de extração de características espectrais de tempo curto mais conhecidas. Em especial, descrevemos os chamados Coeficientes Mel-cepstrais (*Mel-Frequency Cepstral Coefficientss* - MFCCs), que compõem o conjunto de características mais abrangentemente utilizado para o reconhecimento de locutores. Além disso, são

[1] *Pitch* é uma medida de percepção associada à frequência de vibração das cordas vocais.

descritos os chamados coeficientes dinâmicos, que são características espectro-temporais geralmente anexadas aos coeficientes MFCC para a formação do vetor de características final. Mais detalhes sobre os outros tipos de características podem ser encontrados em (KINNUNEN; LI, 2010).

2.3.1 Extratores espectrais de tempo curto

Como mencionado anteriormente, as características espectrais de tempo curto extraem características do chamado envelope espectral, que possui informações a respeito das propriedades ressonantes do trato vocal. Tais medidas são extraídas de janelas curtas do sinal de voz, com comprimentos entre 20 e 30 milissegundos. Tais segmentos devem ser curtos porque o sinal de voz é assumido estacionário, possibilitando uma descrição apropriada do espectro (RABINER; JUANG, 1993).

A primeira fase da extração tem o objetivo de produzir essas janelas. Primeiramente, segmentos de tamanho fixo são extraídos do sinal periodicamente. Na literatura, é bastante comum a extração de janelas de 20 ms de comprimento a cada 10 ms, o que faz com que segmentos consecutivos possuam sobreposição de 10 ms. Após o particionamento do sinal, faz-se necessária a operação de janelamento, que é utilizada em processamento de sinais discretos quando apenas uma porção do sinal deve ser processada. A análise espectral de um determinado segmento é afetada pelas descontinuidades produzidas nas bordas. Para suavizar tais efeitos, o sinal segmentado é multiplicado por uma função de janela (ou função de suavização). Qualquer função utilizada no desenvolvimento de filtros digitais pode ser utilizada, porém, a janela de Hamming é a mais utilizada em processamento de voz. Para um segmento do sinal $s[n]$ com N pontos no tempo, a janela de Hamming $w[n]$ com N pontos é utilizada. Ela é definida pela função

$$w[n] = 0,54 - 0,46\cos(\frac{2\pi n}{N-1}). \tag{2.7}$$

A Figura 6 mostra a janela de Hamming com 100 pontos. A multiplicação no domínio do tempo possui um efeito convolucional no domínio da frequência. A multiplicação do sinal segmentando pela janela de Hamming suaviza as distorções existentes no seu espectro.

Após o processo de janelamento, ao aplicar a Transformada Discreta de Fourier (*Discrete Fourier Transform* - DFT), gera-se o chamado espectro de tempo curto. O espectro é um sinal complexo com partes reais (magnitude) e imaginárias (fase). Devido ao fato de o aparato auditório humano não ser sensível às informações de fase, geralmente ela é desprezada. Além disso, é comum a análise ser realizada através do espectro de potência na escala logarítmica, em especial, na escala decibel (dB). Apresentamos, nas próximas seções, as vantagens de se trabalhar com o sinal na escala logarítmica. Porém, seu uso também provém da tentativa de replicar características da audição humana, que por sua vez interpreta as amplitudes dos sons em uma escala logarítmica. A Figura 7 apresenta o espectro extraído de um segmento de voz e a sua representação na escala logarítmica.

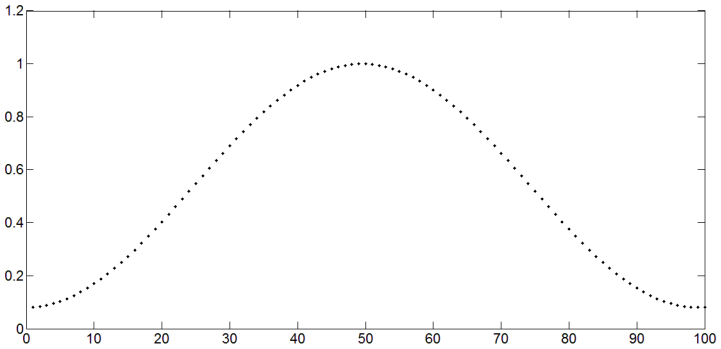

Figura 6 – Janela de Hamming de tamanho 100.

O próximo passo consiste na produção de um vetor de características para cada um dos segmentos produzidos. Tais características devem possuir medidas (explícitas ou não) a respeito da distribuição espectral do sinal contido no segmento. Um detalhe importante sobre a extração das características é que ela ocorre em cada segmento. Isto é, cada segmento produz um vetor de características. Uma determinada locução produz, portanto, um conjunto de vetores de características. Para a produção desses vetores de características, são detalhadas as duas abordagens: a predição linear (*Linear Prediction* - LP) e a extração de coeficientes Mel-cepstrais.

2.3.2 Método de predição linear

Na predição linear (MAKHOUL, 1975; MAMMONE; ZHANG; RAMACHANDRAN, 1996), cada amostra do sinal presente no segmento $s[n]$ é modelada como uma combinação linear das amostras anteriores:

$$\tilde{s}[n] = \sum_{k=1}^{p} a_k s[n-k], \tag{2.8}$$

onde $\tilde{s}[n]$ é o sinal predito, os a_k, para $k = 1, ..., p$, são os coeficientes de predição e p é a ordem do preditor, isto é, a quantidade de coeficientes. O sinal de erro da predição é definido como

$$e[n] = s[n] - \tilde{s}[n], \tag{2.9}$$

e é referenciado como erro residual.

Os coeficientes a_k são determinados pela minimização do erro residual. Para esse propósito, é comum a utilização do chamado algoritmo de Levinson-Durbin (HARRINGTON; CASSIDY, 1999; HUANG et al., 2001; RABINER; JUANG, 1993). Os coeficientes, em si, raramente são utilizados como características e geralmente eles são processados com o propósito de extrair características mais robustas. Exemplos desse tipo de processamento incluem os coeficientes cepstrais de predição linear (*Linear Predictive Cepstral Coefficients* - LPCC) (HUANG et al., 2001) e a técnica de extração de predição linear perceptual (*Perceptual Linear Prediction* - PLP), que incorpora fatores da percepção humana na análise espectral dos coeficientes produzidos (HERMANSKY, 1990).

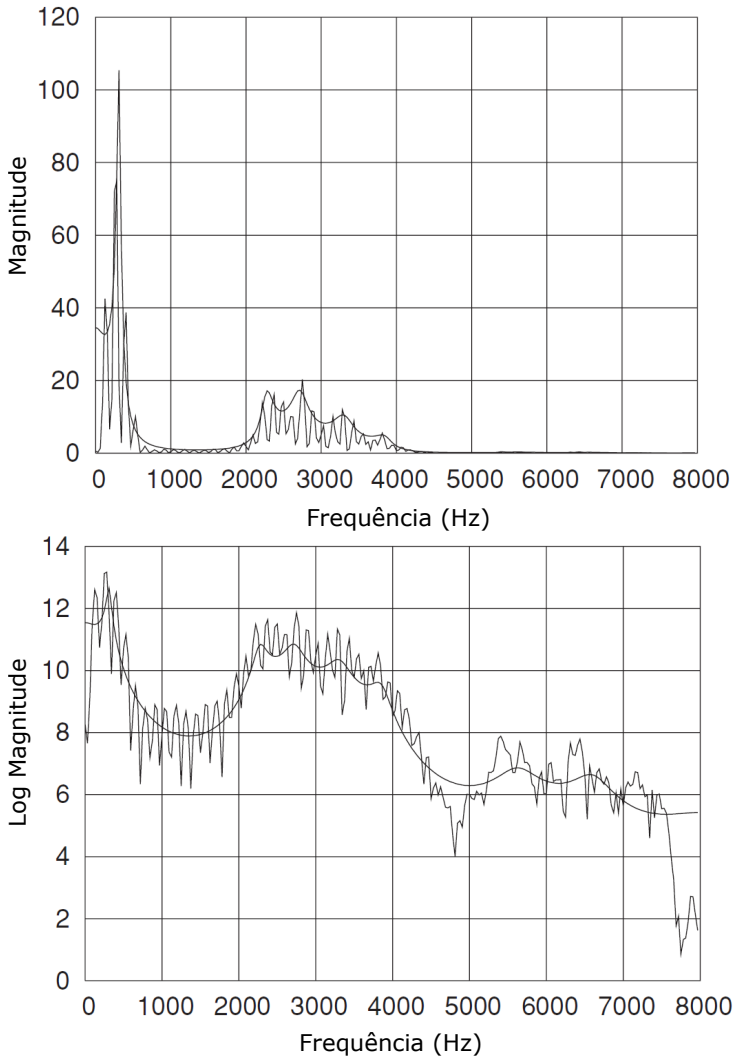

Figura 7 – Exemplos de sinais de espectro de um segmento de voz. A imagem do topo apresenta as magnitudes das frequências, enquanto que a imagem de baixo apresenta o logaritmo das magnitudes. Em ambos os gráficos uma estimativa do envelope espectral é apresentada como uma curva contínua. Gráficos adaptados de (TAYLOR, 2009).

2.3.3 Análise cepstral do sinal de voz

Uma das representações mais importantes na análise de sinais de voz é o chamado *cepstrum*, que é definido pela aplicação da DFT inversa (*Inverse Discrete Fourier Transform* - IDFT) ao logaritmo da magnitude do espectro de um sinal. Dado um sinal discreto $s[n]$, o *cepstrum* desse sinal, $c[n]$, é definido como

$$c[n] = \mathcal{F}^{-1}\{\log|\mathcal{F}\{s[n]\}|\}, \tag{2.10}$$

onde $\mathcal{F}$ e $\mathcal{F}^{-1}$ são a DFT e a IDFT, respectivamente. A Figura 8 apresenta o *cepstrum*[2] do sinal apresentado na Figura 7.

[2] O termo *cepstrum* foi criado para enfatizar o domínio inverso do espectro (mas diferente do tempo). Sua origem vem do termo *spectrum*, invertendo a ordem das primeiras quatro letras.

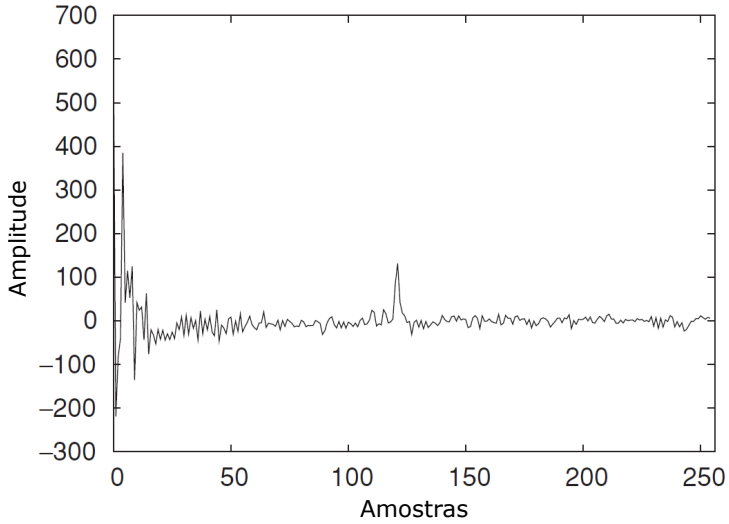

Figura 8 – Exemplo do *cepstrum* do sinal de voz apresentado na Figura 7. Gráfico adaptado de (TAYLOR, 2009).

O *cepstrum* de um sinal possibilita a análise de algumas características presentes no espectro do sinal. O sinal de magnitude do espectro apresenta harmônicos em intervalos igualmente espaçados, com um decaimento expressivo com o aumento da frequência. Como visto na Figura 7, a aplicação do logaritmo à magnitude suaviza esse decaimento e comprime as variações bruscas de amplitude, enfatizando a periodicidade dos harmônicos. Ao observar esse novo espectro como um sinal temporal, aplicar a IDFT produz um novo espaço de frequências onde é possível separar as componentes relativas às mudanças bruscas de amplitude e a componente periódica do espectro. Dessa maneira, diferentes resoluções do espectro podem ser geradas ao considerar apenas uma determinada quantidade dos primeiros coeficientes de $c[n]$ e aplicar a DFT para retornar ao espaço espectral[3]. A Figura 9 apresenta esse processo para diferentes valores de coeficientes sendo mantidos durante a reconstrução do espectro. É possível notar que com uma quantidade reduzida de coeficientes já é suficiente para realizar uma estimativa razoável do envelope espectral do sinal.

Uma outra motivação para o uso da análise cepstral surge ao levarmos em consideração o modelo matemático mais abrangentemente utilizado para modelar a produção do sinal de voz. Nesse modelo, um sinal é originado de uma determinada fonte (como a glote para sons vogais) e é filtrado pelas componentes ressonantes do trato vocal e das radiações sonoras que ocorrem nos lábios (FANT, 1970). Tais filtros são considerados lineares e, pela análise de tempo curto, estáticos (invariantes de tempo). Nesse contexto, podemos descrever um determinado sinal de voz como o resultado de uma convolução no tempo dessas três componentes:

$$s[n] = u[n] * v[n] * r[n], \tag{2.11}$$

[3] O processo de filtragem no domínio cepstral é definido como *liftering*, que tem sua origem no termo *filtering*, invertendo as primeiras quatro letras.

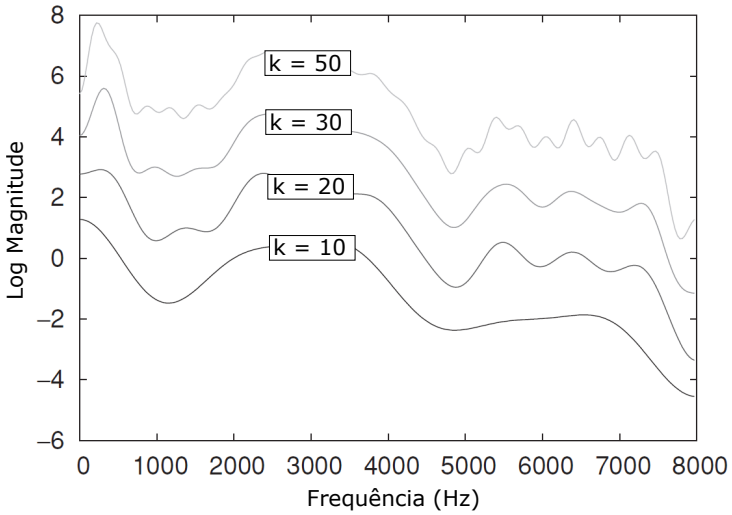

Figura 9 – Exemplo de diferentes resoluções para o logaritmo da magnitude do espectro de um sinal após o processo de *liftering* do *cepstrum* e do retorno ao espaço espectral através da DFT. Em cada caso, apenas os K primeiros coeficientes foram mantidos. Pode-se notar um aumento de precisão do espectro com o aumento do valor de K. Imagem adaptada de (TAYLOR, 2009).

onde $u[n]$ é o sinal proveniente da fonte e $v[n]$ e $r[n]$ os filtros correspondentes ao trato vocal e à radiação sonora dos lábios, respectivamente. Considerando $v[n]$ e $r[n]$ como um único filtro, $v'[n]$, e aplicando a DFT ao sinal, o espectro do sinal, no domínio das frequências, é descrito como a multiplicação dos espectros:

$$S(e^{jw}) = U(e^{jw})V'(e^{jw}). \tag{2.12}$$

Ao considerar o logaritmo da magnitude do espectro, as componentes são desacopladas, surgindo uma relação aditiva entre elas:

$$\log(|S(e^{jw})|) = \log(|U(e^{jw})|) + \log(|V'(e^{jw})|). \tag{2.13}$$

Por fim, para ao computar o *cepstrum* do sinal através da IDFT, o sinal resultante apresenta, no domínio original, as componentes separadamente:

$$\hat{s}[n] = \hat{u}[n] + \hat{v}'[n]. \tag{2.14}$$

O *cepstrum* se torna bastante útil nesse contexto porque ele separa as duas componentes: a da fonte (coeficientes mais altos) e a do trato vocal (coeficientes mais baixos), possibilitando uma análise mais apurada de cada uma delas ao considerar diferentes regiões dos componentes.

2.3.4 Coeficientes Mel-cepstrais

Os coeficientes Mel-cepstrais (MFCCs) (DAVIS; MERMELSTEIN, 1980) foram introduzidos na década de 1980 e formam o conjunto de características mais popular em processamento

de áudio e voz. Inicialmente, eles foram propostos para o problema de reconhecimento de fala e então foram aplicados para o reconhecimento de locutores. Mesmo com estudos posteriores de inúmeros extratores de características, o bom desempenho dos MFCCs faz com que sejam utilizados até hoje. Tal desempenho é resultante de um conjunto de aspectos dos MFCCs, como é visto mais adiante, porém, há a conjectura de que o bom desempenho se deve ao fato de tais coeficientes simularem a percepção auditiva humana. Apesar de uma das etapas do cálculo dos coeficientes ser fortemente inspirada na percepção auditiva humana, tal conjectura nunca foi provada ou refutada. O diagrama de blocos para o cálculo dos MFCCs, para um determinado segmento de voz, é apresentado na Figura 10.

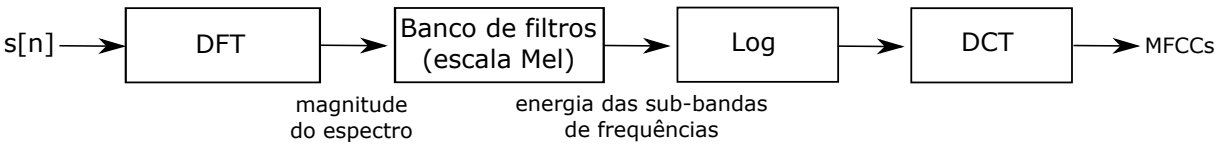

Figura 10 – Diagrama de blocos para o cálculo dos MFCCs.

O objetivo principal do método consiste na extração de uma representação robusta para o envelope espectral, extraindo assim informações das componentes ressonantes do trato vocal. Os MFCCs são extraídos a partir de uma combinação entre duas abordagens: **análise de banco de filtros** (*filter-bank analysis*) e **análise cepstral**.

O primeiro estágio consiste no processamento da magnitude do espectro através de um banco de filtros, que é um conjunto de filtros passa-faixa centrados em diferentes frequências. Cada filtro desse banco, então, captura informações de uma banda de frequência específica. Os filtros são desenvolvidos sob algum formato específico, geralmente triangular, e são definidos pela frequência do centro do filtro e pela largura. As informações das bandas são medidas sob a forma de energias dos sinais resultantes da multiplicação da magnitude do espectro com a função de peso do filtro. Tal processo suaviza as grandes variações de amplitude geradas pelos harmônicos presentes na banda de frequência. Após a passagem desses filtros, produz-se uma representação de como as energias variam com a frequência, que está intrinsecamente ligada às componentes do trato vocal e pouco relacionada com as informações de fonte (*pitch*).

Além disso, as frequências de centro e larguras dos filtros são escolhidas de maneira que o banco simule a percepção auditiva humana. Estudos realizados por Stevens, Volkmann e Newman, no final da década de 1930, resultaram na definição da medida Mel[4] como uma medida de *pitch*. Tal estudo permitiu a análise da capacidade perceptiva de um indivíduo de diferenciar tons de diferentes frequências. Os dados foram publicados em 1937 (STEVENS; VOLKMANN; NEWMAN, 1937) e em 1940 (STEVENS; VOLKMANN, 1940), e

[4] Abreviação de *melody*, que significa melodia.

posteriormente analisados por O'Shaughnessy, que produziu uma equação que relaciona a capacidade perceptual em função de um tom de frequência específica (O'SHAUGHNESSY, 1987). Essa relação é dada pela equação:

$$Mel(f) = \frac{1000}{\ln(1 + \frac{1000}{700})} \ln(1 + \frac{f}{700}) = 1127 \ln(1 + \frac{f}{700}), \tag{2.15}$$

onde f é a frequência do tom, em Hz, e $Mel(f)$ é a medida de *pitch* na escala mel.

A escala Mel permitiu uma análise do quão bem a percepção auditiva humana atua sobre o espectro audível[5]. A maneira mais usual para o desenvolvimento do banco consiste em definir filtros triangulares igualmente espaçados na escala Mel com larguras definidas pelas frequências adjacentes (ver Figura 11).

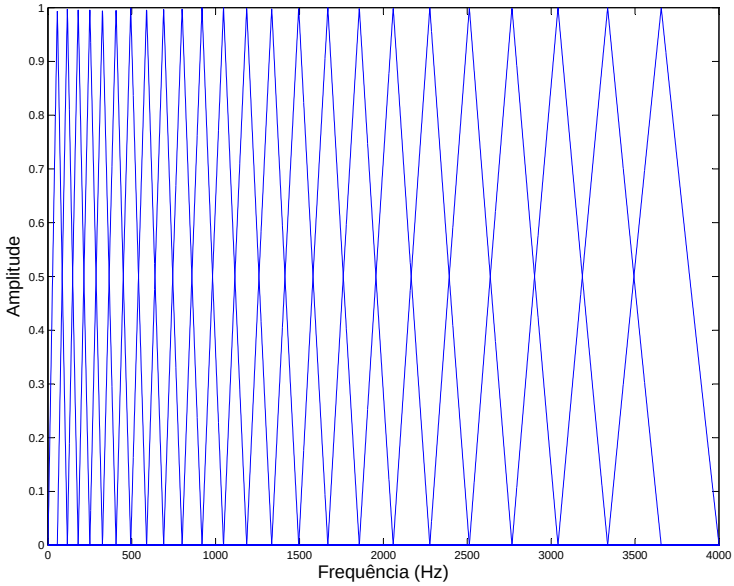

Figura 11 – Banco de filtros com 24 filtros triangulares igualmente espaçados na escala Mel com larguras definidas pelos filtros adjacentes. Nesse caso, o banco de filtros foi construído para operar sobre áudios com frequência de amostragem de 8KHz.

Como o nome sugere, os MFCCs são coeficientes cepstrais. Para isso, aplica-se o logaritmo às energias resultantes da passagem do sinal pelo banco de filtros:

$$L[k] = \log_{10}(S_k), \qquad k = 1, ..., K, \tag{2.16}$$

[5] Apesar disso, existe a crença de que a escala Mel não é apropriada para caracterização da voz de mulheres (ZHOU et al., 2011). Isso porque sinais de voz feminina possuem mais componentes de alta frequência. Como a escala Mel possui baixa resolução nas bandas mais altas, ela se torna inapropriada. Geralmente é considerada uma alternativa onde o banco de filtro é construído seguindo uma escala linear. Nesse caso, os coeficientes são referenciados como Coeficientes Cepstrais de Frequência Linear (*Linear Frequency Cepstral Coefficientss* - LFCCs).

onde S_k é a energia resultante do k-ésimo filtro e K é quantidade de filtros presentes no banco. Além das vantagens em se trabalhar com o logaritmo da magnitude do espectro, mencionadas anteriormente, essa operação também tenta refletir aspectos do aparato perceptivo humano sobre a medida de intensidade sonora[6].

Ao invés de utilizar a IDFT como na análise cepstral convencional, a transformação do sinal das energias para o espaço cepstral é realizada através da Transformada Discreta do Cosseno (*Discrete Cosine Transform* - DCT):

$$c[n] = \sum_{k=0}^{K-1} L[k] \cos\left(\frac{\pi}{K}(k + \tfrac{1}{2})\ n\right), \qquad n = 1, ..., N, \tag{2.17}$$

onde $c[n]$ é o n-ésimo coeficiente MFCC e N é a quantidade de coeficientes extraídos. Para $n = 0$, a componente do cosseno da equação toma o valor 1 e $c[0]$ se torna igual ao logaritmo da energia média do sinal. Como a energia média não possui informação discriminativa e nem tão pouco reflete o aparato vocal do locutor, é comum a mesma não ser extraída. Portanto, quando é afirmada a extração de N coeficientes, o coeficiente $c[0]$ não é levado em consideração. Quando $c[0]$ é utilizado, é comum afirmar a extração de N coeficientes MFCCs anexados à energia. Nesse caso, os vetores de características possuem dimensão $N + 1$.

Para o cálculo da DCT[7] de um sinal discreto de tamanho K, primeiramente tal sinal é espelhado e deslocado ($n' = n + 1/2$) para a geração de um sinal de tamanho $2K$ simétrico. A DCT do sinal original é definido pela DFT desse sinal simétrico, e, consequentemente, possui componentes imaginárias nulas. Por essa razão, é comum imaginar essa transformada como "uma versão real" da DFT. Porém, a justificativa para utilização da DCT provém da sua capacidade de descorrelação do sinal das energias, que está fortemente relacionada à Transformada de Karhunen–Loève (*Karhunen–Loève Transform* - KLT), que é a base para a Análise dos Componentes Principais (*Principal Component Analysis* - PCA), técnica amplamente utilizada para projetar dados de alta dimensão em um espaço de base ortogonal. Mais precisamente, para sinais descritos por um modelo autoregressivo de primeira ordem:

$$x[n] = \alpha x[n-1] + z[n], \quad 0 \leq \alpha \leq 1, \tag{2.18}$$

onde α é o coeficiente de correlação e z um ruído branco, a DCT é assintoticamente equivalente à KLT quando $\alpha \to 1$ (RAO; YIP, 2014). Apesar de a DCT muitas vezes estar associada à descorrelação dos coeficientes, na prática, a condição necessária para isso ocorrer ($\alpha = 1$) pode não ser alcançada e alguma correlação pode ser observada. Porém, apesar de a descorrelação não ser garantida, a DCT possui a capacidade de descrever o espectro em um espaço mais limitado e a comparação entre espectros pode ser realizada diretamente através dos coeficientes.

[6] Intensidade sonora (*loudness*) define um conceito em psicoacústica que relaciona medidas de intensidade (pressão na cóclea) e *pitch*. A utilização da função logarítmica simula a escala decibel (dB).

[7] Para ser exato, a DCT-II (Equação 2.17).

Como é visto mais adiante, a tendência dos MFCCs em descorrelacionar as energias das bandas do espectro proporciona uma simplificação dos modelos que utilizam essa representação para descrição das locuções, onde a independência dos coeficientes é assumida. Exemplos disso são funções de densidade de probabilidade sendo modeladas com matrizes de covariância diagonal (Seção 2.4.1).

2.3.5 Extratores espectro-temporais

Como mencionado na seção anterior, além da utilização de características espectrais de tempo curto, é comum também a utilização de suas medidas temporais, como velocidade e aceleração. Essas medidas são chamadas de características espectro-temporais e geralmente são anexadas às características de tempo curto para a formação do vetor final de características. A seguir, descrevemos o cálculo dessas medidas para os coeficientes MFCC.

2.3.5.1 Coeficientes MFCC dinâmicos

As características cepstrais capturam a distribuição espectral de um segmento específico do sinal de voz. Porém, muita informação discriminativa pode ser extraída levando em consideração a mudança temporal dessa distribuição. Para capturar esse tipo de informação, geralmente as derivadas no tempo são calculadas. Sob esse ponto de vista, um determinado coeficiente possui uma trajetória temporal, uma vez que os coeficientes são extraídos de janelas de tempo fixas. Cada coeficiente $c[n]$ pode ser observado, portanto, como pertencente a um sinal temporal, $c_n[t]$. Porém, as amostras temporais de um determinado coeficiente cepstral não possuem uma forma analítica e o cálculo de sua derivada pode ser apenas aproximado por uma diferença finita.

A diferença finita de primeira ordem é definida como

$$d_n[t] = c_n[t+1] - c_n[t]. \tag{2.19}$$

Porém, esse tipo de estimação da derivada é bastante sensível a distorções presentes nos coeficientes. Dessa maneira, Furui propôs a utilização de um ajuste ortogonal polinomial da trajetória temporal de um determinado coeficiente cepstral utilizando uma janela de tamanho finito (FURUI, 1981).

O termo constante do polinômio ortogonal é dado por

$$\widehat{c}_n[t] = \frac{\sum_{k=-K}^{K} h_k c_n[t+k]}{\sum_{k=-K}^{K} h_k}, \tag{2.20}$$

onde h_k é uma janela simétrica de tamanho $2K+1$. Geralmente, uma janela retangular é utilizada. Além disso, o valor mais utilizado é $K = 2$.

O coeficiente de primeira ordem do polinômio ortogonal, denotado por $\Delta c_n[t]$, é, então, definido como

$$\frac{dc_n[t]}{dt} \approx \Delta c_n = \frac{\sum_{k=-K}^{K} k h_k c_n[t+k]}{\sum_{k=-K}^{K} h_k k^2}. \tag{2.21}$$

Coeficientes polinomiais ortogonais de ordens maiores podem ser derivados de forma semelhante. 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. Por outro lado, Furui mostrou que a utilização dos coeficientes de primeira ordem para caracterizar a dinâmica dos coeficientes cepstrais é geralmente suficiente. Porém, quando os coeficientes MFCC são utilizados para reconhecimento de locutores, geralmente coeficientes dinâmicos de primeira e segunda ordem são utilizados.

2.3.6 Técnicas de compensação de características

Grande parte dos esforços desenvolvidos para aumentar a robustez dos sistemas de reconhecimento de locutores quanto a ruído foram desenvolvidos sobre os módulos de extração de características. Além de prover uma discriminação entre diferentes locutores, o conjunto de características ideal não deve ser distorcido por informações presentes no sinal que não dizem respeito ao locutor que está produzindo a locução (como ruído de ambiente, por exemplo). Dessa maneira, o desenvolvimento de um conjunto de características robusto a ruídos possui a vantagem de poder ser utilizado de maneira geral para diversos tipos de distorções.

Devido ao poder de discriminação proporcionado pelas características espectrais de tempo curto (especialmente pelas características cepstrais), as principais técnicas de compensação de características foram propostas para esse tipo de extratores. Muitas delas foram primeiramente propostas para reconhecimento de fala e não de locutores. Por essa razão, os trabalhos que as propõem realizam tais compensações sobre a família de características baseada em predição linear, basicamente sobre coeficientes LP, LPCC ou PLP. Porém, no desenvolvimento de sistemas de reconhecimento de locutores, tais técnicas começaram a ser utilizadas sobre os coeficientes MFCC. Na prática, o processo de extração desses coeficientes atualmente é realizado incorporando uma ou mais técnicas de compensação de ruído. As principais técnicas encontradas na literatura são descritas a seguir.

2.3.6.1 Subtração de média cepstral

Subtração de média cepstral (*Cepstral Mean Subtraction* - CMS) (ATAL, 1974; FURUI, 1981) – ou normalização de média cepstral (*Cepstral Mean Normalization* - CMN) - foi uma das primeiras técnicas propostas e é amplamente utilizada até hoje. Geralmente ela é integrada aos extratores de características cepstrais e é utilizada para supressão dos efeitos causados por filtragens lineares (como as distorções de canal) e baseia-se na natureza aditiva desse tipo de distorção no domínio espectral logarítmico.

Para cada um dos coeficientes cepstrais extraídos de uma determinada janela do sinal de voz, subtrai-se o valor da média temporal desse coeficiente. Novamente, observemos os coeficientes como um sinal temporal, $c_n[t]$. O novo valor desse coeficiente, $\widehat{c}_n[t]$, é definido

como

$$\widehat{c}_n[t] = c_n[t] - \overline{c}_n, \tag{2.22}$$

onde $\overline{c}_n[t]$ é a média temporal daquele coeficiente levando em consideração toda a locução:

$$\overline{c}_n = \frac{1}{T}\sum_{t=1}^{T} c_n[t], \tag{2.23}$$

onde T é a quantidade de coeficientes presentes na locução.

Outro método geralmente empregado consiste na normalização dos coeficientes através da divisão da Equação 2.22 pelo desvio-padrão dos valores do coeficiente no tempo. Essa técnica é geralmente referenciada como normalização de média e variância cepstral (*Cepstral Mean and Variance Normalization* - CMVN). Esse tipo de normalização geralmente é empregado no cálculo dos coeficientes de predição (LPs). Como a variância dos dados era equalizada, a busca dos coeficientes era mais eficiente (FURUI, 1981). Já no contexto dos coeficientes MFCCs, CMS e CMN geralmente são utilizadas antes do cálculo dos coeficientes dinâmicos (Seção 2.3.5.1). A Figura 12 mostra a adição desses módulos no processo de extração dos coeficientes MFCC.

Avaliações mostraram que a remoção da média cepstral melhora a robustez do sistema para incompatibilidades de canal (REYNOLDS, 1994; VUUREN, 1996; FURUI, 1981). Porém, uma queda de desempenho é observada quando locuções com pouca distorção são utilizadas. Isso ocorre porque tais métodos assumem que a média cepstral do sinal de voz puro é zero. Geralmente, é necessária certa variabilidade fonética na locução para que essa suposição possa ser feita.

Tais métodos também assumem que o efeito de canal é constante em toda a locução. Para locuções longas, as distorções provocadas pelo canal podem mudar com o tempo e a compensação desses métodos pode não ser suficiente. Nesse contexto, a alternativa geralmente utilizada consiste na subtração (ou normalização) dos coeficientes utilizando médias (e desvios-padrão) calculados em janelas de tamanho fixo, ao invés de calculados sobre toda a locução (VIIKKI; LAURILA, 1998). Tipicamente, tais janelas possuem duração de 3 a 5 segundos.

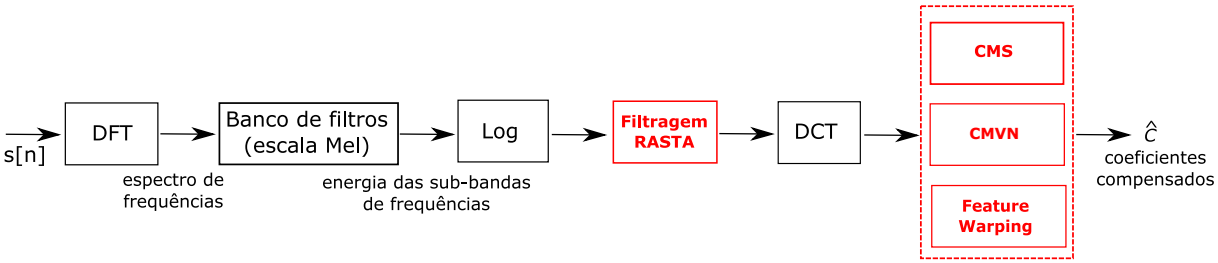

Figura 12 – Métodos de compensação de características, em vermelho, que são adicionados ao processo de extração de coeficientes cepstrais. Enquanto a técnica RASTA opera sobre as energias dos bancos de filtros, as técnicas CMS, CMVN e *Feature Warping* operam sobre os coeficientes.

2.3.6.2 Filtragem RASTA

A filtragem espectral relativa (HERMANSKY et al., 1992; HERMANSKY; MORGAN, 1994), RASTA (*Relative Spectral*), é um dos métodos mais utilizados na prática. A ideia básica do método consiste na supressão das pequenas variações temporais das bandas de frequência do sinal. Mais precisamente, o método consiste da aplicação de um filtro passa-faixa sobre o domínio espectral de tempo curto. O filtro é aplicado sobre cada uma das características espectrais e suprime as variações dessas características que estão fora do intervalo de variação tipicamente encontrado em sinais de voz.

Inicialmente, a técnica foi proposta para reconhecimento de fala e integrada ao extrator de características PLP (Seção 2.3.2). Porém, atualmente ele é amplamente utilizado na extração dos coeficientes MFCC e o filtro é geralmente aplicado às trajetórias temporais do logaritmo da resposta de cada um dos filtros presentes no banco de filtros (ver Figura 12). Assumindo uma taxa de amostragem de 100 Hz, o filtro possui a seguinte função de transferência:

$$H(z) = 0,1z^4 \times \frac{2 + z^{-1} - z^{-3} - 2z^{-4}}{1 - 0,94z^{-1}}. \tag{2.24}$$

A frequência de corte da componente passa-alta encontra-se em 0,26Hz, mas também atinge resposta nula em 28,9Hz e novamente em 50Hz (ver Figura 13). Espera-se que a componente passa-alta do filtro suavize as distorções convolucionais[8] introduzidas pelo canal, enquanto que a componente passa-baixa suavize algumas variações bruscas do espectro de tempo curto.

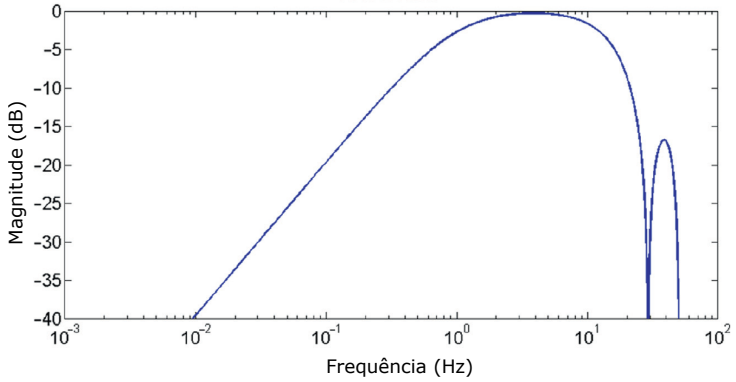

Figura 13 – Respostas em frequência do filtro RASTA para uma frequência de amostragem de 100Hz. Imagem adaptada de (LI et al., 2015).

A filtragem RASTA possui um tempo constante de integração, o que provoca um atraso quando o filtro é aplicado. Para o pólo utilizado, 0,94, o atraso é de aproximadamente 160ms. No processo de extração dos coeficientes MFCC, isso significa o descarte das primeiras janelas. Avaliações mostraram desempenhos similares na compensação de canal entre a filtragem RASTA e o método CMS, porém, a filtragem RASTA geralmente

[8] Distorções resultantes da convolução entre o sinal de voz e um sinal de ruído.

apresenta menor queda de desempenho quando sinais de voz com pouco ruído são utilizados (REYNOLDS, 1994; VUUREN, 1996). Apesar disso, é comum encontrar na literatura sistemas que utilizam ambas as técnicas.

2.3.6.3 Deformação de características

Após a consolidação e utilização em massa dos métodos de filtragem e normalização para compensação de canal, no final dos anos 90 surgiram algumas técnicas que se propuseram a suavizar melhor as pequenas distorções provenientes do ruído aditivo. Como mencionado anteriormente, tais ruídos provocam distorções não lineares no domínio log-espectral. O método de deformação de características (*Feature Warping* - FW) (PELECANOS; SRIDHARAN, 2001) é a principal técnica proposta para essa finalidade. A ideia do método é realizar transformações a cada uma das características cepstrais a fim de mapeá-las para uma distribuição de probabilidade específica. Os autores mostraram a aplicação dessa técnica diretamente na extração dos coeficientes MFCC (Figura 12), que mapeia cada um dos coeficientes para uma representação mais robusta. Esse mapeamento é geralmente realizado antes do cálculo dos coeficientes dinâmicos, similarmente aos métodos CMS e CMN.

Suponha novamente a trajetória temporal dos coeficientes, $c_n[t]$, para o n-ésimo coeficiente. Como mapeamento de cada um dos coeficientes leva em consideração apenas o seu valor no tempo, vamos referenciar $c_n[t]$ como c_t, por simplicidade. Para o mapeamento de c_t, o método primeiramente separa os coeficientes presentes em uma janela temporal (de tipicamente três segundos) centrada em c_t. Suponha que a janela temporal represente um total de T coeficientes. A separação dos coeficientes da janela produz um conjunto de coeficientes cepstrais, C, tal que:

$$C = \{c_{t-T/2}, ..., c_{t-1}, c_t, c_{t+1}, ..., c_{t-1+T/2}\}. \tag{2.25}$$

O conjunto C é então ordenado e associa-se ao coeficiente c_t um *rank* de acordo com a posição onde ele se encontra no conjunto ordenado. O *rank* é definido de modo que ao coeficiente de maior valor seja atribuído o *rank* um, enquanto que ao de menor valor o *rank* T.

Supondo que os valores iniciais dos coeficientes seguem uma distribuição com função de densidade de probabilidade $y(c)$ (que é desconhecida *a priori*), o novo valor do coeficiente central, $\widehat{c}_t$, é calculado considerando que os novos valores dos coeficientes seguem uma distribuição específica com função de densidade $h(c)$, que é fixada. Esse procedimento é bastante similar àquele empregado no processo de equalização de histogramas de imagens, onde a função de densidade final, $h(c)$, é definida pela função de densidade da distribuição uniforme. O valor do *rank* do coeficiente central é utilizado para realizar o seu mapeamento, preservando sua posição relativa em relação aos outros coeficientes.

Isso é realizado a partir da função de densidade acumulada:

$$\int_{\infty}^{c_t} y(c)dc = \int_{\infty}^{\widehat{c_t}} h(c)dc. \tag{2.26}$$

Intuitivamente, o coeficiente de maior valor deveria ser mapeado para um valor cuja probabilidade acumulada, em $h(c)$, seja 1, enquanto que o menor valor para um valor cuja probabilidade acumulada seja nula. Em distribuições contínuas que se estendem por toda a reta, tais valores são alcançados apenas em $\pm\infty$. Por essa razão, os autores fixaram o valor mínimo que pode ser alcançado pela probabilidade acumulada em $\frac{1}{2N}$ e o valor máximo em $1 - \frac{1}{2N}$.

Utilizando o valor do *rank* do coeficiente, R, o novo valor do coeficiente é calculado resolvendo a seguinte equação:

$$\frac{T + \frac{1}{2} - R}{T} = \int_{c=-\infty}^{\widehat{c_t}} h(c)dc. \tag{2.27}$$

É bastante provável que a distribuição ideal para $h(c)$ seja mutimodal, porém os autores, por simplicidade, utilizaram a distribuição normal univariada com média zero e desvio-padrão unitário:

$$h(c) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-c^2}{2}\right). \tag{2.28}$$

Em (PELECANOS; SRIDHARAN, 2001), os resultados mostrados validam o ganho de desempenho quando há incompatibilidade de canal nos sinais. Por outro lado, o desempenho encontrado equiparou-se àqueles resultantes da utilização das técnicas CMS e CMN.

Em (XIANG et al., 2002), uma técnica chamada de Gaussianização de tempo curto (*Short-time Gaussianization*) foi proposta. A ideia consiste em realizar a mesma deformação das características, porém, para uma distribuição multimodal. A distribuição utilizada consiste do chamado Modelo Gaussiano composto (*Compound Gaussian Model* - CGM). Os resultados não reportaram nenhum ganho significativo em relação à técnica original, que ainda é a mais utilizada das duas.

2.3.6.4 Combinações de métodos de compensação de características

Como mencionado anteriormente, as técnicas de compensação mais bem sucedidas são, na prática, aplicadas ao processo de extração de características cepstrais de tempo curto. Além disso, elas atuam em diferentes etapas do cálculo dos coeficientes que compõem o vetor final de características que descrevem o sinal de voz utilizado no reconhecimento. A Figura 12 mostra em que etapas essas técnicas são aplicadas. Dessa maneira, a combinação de métodos de compensação diferentes pode ser realizada com o intuito de aumentar a robustez do conjunto final de características. Por essa razão, não é incomum encontrar na literatura sistemas que utilizam mais de uma técnica. Por exemplo, em (REYNOLDS; QUATIERI; DUNN, 2000) e em (CAMPBELL; STURIM; REYNOLDS, 2006), a filtragem RASTA é utilizada em conjunto com o método CMS.

2.4 MODELAGEM DOS LOCUTORES

Esta seção tem por objetivo a descrição dos principais métodos propostos para a criação dos modelos dos locutores. Tais métodos devem utilizar os vetores extraídos das locuções para criar modelos matemáticos para realização da decisão sobre as hipóteses nula e alternativa (Equações 2.1 e 2.2, respectivamente).

Durante os anos, as abordagens mais bem sucedidas consistiam na criação de modelos probabilísticos que seguem uma determinada distribuição de probabilidade. Mais precisamente, elas utilizam os chamados Modelos de Misturas Gaussianas (*Gaussian Mixture Models* - GMMs). Outro fator preponderante consiste na modelagem da hipótese alternativa através do chamado Modelo Universal de Fundo (UBM). A primeira técnica a ser descrita consiste da modelagem GMM-UBM. A segunda abordagem é baseada nas chamadas Máquinas de Vetores Suporte (*Support Vector Machines* - SVMs) e realiza a decisão utilizando os denominados supervetores GMM. Essa técnica é denominada de GMM-SVM. Já a terceira técnica (e considerada a mais bem sucedida nesse tipo de abordagem) realiza uma modelagem através da análise de fatores (*factor analysis*) no espaço dos supervetores GMM, criando uma nova representação para as locuções, chamada de vetor-identidade (*i-vector*). Tais conceitos e técnicas são descritos a seguir.

2.4.1 Modelos de Misturas Gaussianas

Em um sistema de verificação de locutores baseado no teste da razão das verossimilhanças (Equação 2.3), as hipóteses nula e alternativa devem ser modeladas de alguma maneira. Isto é, modelos matemáticos devem ser estimados e associados às hipóteses, a fim de que seja possível a discriminação entre elas a partir da locução de teste. A maneira como a discriminação entre as hipóteses é realizada é o que define o tipo de modelagem imposta às hipóteses. A abordagem mais direta consiste na estimação de distribuições de probabilidade que são utilizadas para o cálculo explícito das verossimilhanças associadas às hipóteses H_0 e H_1 para a uma determinada locução de teste X. A escolha da distribuição de probabilidade dos modelos claramente depende tanto das características extraídas das locuções quanto das especificações do sistema. Para sistemas de verificação independente de texto, onde não há nenhum conhecimento prévio a respeito do que será dito pelo locutor, as funções de verossimilhança mais bem sucedidas tem sido aquelas definidas por Modelos de Misturas Gaussianas (GMMs).

GMMs foram primeiramente propostos para reconhecimento de locutores por Douglas Reynolds em 1995 (REYNOLDS; ROSE, 1995; REYNOLDS, 1995) e desde então vêm sendo utilizados na modelagem das características dos locutores em sistemas de verificação/identificação independentes de texto. O sucesso da utilização de GMMs para a modelagem das características de um determinado locutor se deve ao fato de serem capazes de modelar funções de densidade de probabilidade arbitrárias. Além disso, a estimação de seus parâ-

metros é realizada por técnicas bem fundamentadas sob o ponto de vista estatístico. Outra vantagem está no baixo custo computacional associado ao cálculo das probabilidades.

A função de densidade de probabilidade de um GMM é definida pela soma ponderada de distribuições normais multivariadas. Portanto, para um GMM, λ, com M componentes de dimensão D, sua função de densidade de probabilidade é definida por

$$p(\boldsymbol{x}|\lambda) = \sum_{m=1}^{M} \omega_i N(\boldsymbol{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \tag{2.29}$$

onde $\boldsymbol{x}$ é um vetor de dimensão D e ω_i, $i = 1, ..., M$, são os pesos associados a cada uma das componentes e satisfazem $\sum_{i=1}^{M} \omega_i = 1$, $\boldsymbol{\mu}_i \in \mathbb{R}^{D\times 1}$ e $\boldsymbol{\Sigma}_i \in \mathbb{R}^{D\times D}$ são o vetor de média e a matriz de covariância da componente i, respectivamente. As funções $N(\boldsymbol{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ são as densidades de probabilidade correspondentes a cada componente e seguem a função de densidade de probabilidade normal:

$$N(\boldsymbol{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}_i|^{1/2}} \exp\Big[-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_i)\Big]. \tag{2.30}$$

Geralmente, matrizes de covariância diagonais são utilizadas,

$$\boldsymbol{\Sigma}_i = \text{diag}(\sigma_1^2, \sigma_2^2, ..., \sigma_D^2), \tag{2.31}$$

ao invés de matrizes de covariância completas. Isso é uma aproximação usualmente justificada pelo fato de os extratores de características geralmente produzirem características com baixa correlação matemática, como é o caso dos coeficientes MFCCs. Além de diminuir consideravelmente a quantidade de parâmetros a serem estimados, há muita redução no custo computacional. Em termos mais precisos, mostrou-se que a utilização de matrizes de covariância diagonal apresenta resultados semelhantes aos apresentados com a utilização de matrizes completas (REYNOLDS; QUATIERI; DUNN, 2000; REYNOLDS; ROSE, 1995).

Um GMM é definido, portanto, pelos pesos, vetores de média e matrizes de covariância associados a cada uma de suas M componentes:

$$\lambda = \{\omega_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}, \qquad i = 1, ..., M. \tag{2.32}$$

Tais parâmetros são estimados via maximização de verossimilhança (*Maximum likelihood* - ML) dos dados disponíveis para a caracterização de cada uma das hipóteses (Figura 4). Para um GMM, a definição desse problema pode ser sumarizada como a seguir: dado um conjunto de vetores $X = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$, definir λ (Equação 2.32), de modo a maximizar a verossimilhança de λ com respeito a X. Basicamente, espera-se encontrar λ de modo que o GMM seja capaz de modelar bem os vetores utilizados na estimação. Além disso, também se espera certa capacidade de generalização, isto é, que o modelo seja capaz de apresentar altas verossimilhanças para vetores similares aos que foram apresentados no momento da estimação.

Como mencionado anteriormente, uma das razões do sucesso dos GMMs é a existência de um paradigma poderoso e versátil para estimação de parâmetros. Tal paradigma consiste do algoritmo de Maximização de Expectativa (*Expectation-Maximization* - EM) (DEMPSTER; LAIRD; RUBIN, 1977; BILMES et al., 1998). O algoritmo EM é a técnica mais utilizada para estimar parâmetros de distribuições de probabilidades maximizando a verossimilhança das distribuições com respeito a um conjunto de dados observados. Ele é um algoritmo iterativo e garante uma convergência monotônica a cada iteração. Uma descrição detalhada do algoritmo EM é apresentada no Apêndice A.

2.4.2 Modelo Universal de Fundo

Como visto anteriormente, os sistemas de verificação de locutores mais bem sucedidos modelam as hipóteses nula e alternativa explicitamente através de distribuições de probabilidade. O modelo responsável pela hipótese nula deve modelar a função de densidade de probabilidade correspondente ao fato de a locução ser produzida pelo locutor em questão, S. Portanto, na fase de treinamento, as locuções desse locutor específico devem ser utilizadas para estimar essa função. Esse modelo é bem definido, no sentido que o universo que deve ser modelado é bem conhecido. Por outro lado, o modelo responsável pela hipótese alternativa não é bem definido, uma vez que, teoricamente, ele deve estimar todo o espaço dos locutores que não são S. Dessa maneira, tal modelo, denominado de modelo de fundo, deve ser genérico o bastante para modelar uma grande variabilidade de locutores. Diante dessa dificuldade, duas são as principais abordagens utilizadas na sua estimação.

A primeira abordagem consiste em utilizar um conjunto de modelos de diferentes locutores. Em diversos contextos, esse conjunto tem sido chamado de conjuntos de razão de verossimilhança (*likelihood ratio sets*) (HIGGINS; BAHLER; PORTER, 1991), *cohorts* (ROSENBERG et al., 1992) ou então de locutores de fundo (*background speakers*) (REYNOLDS, 1995). Dado um conjunto de N modelos de locutores de fundo, $\{\lambda_1, \lambda_2, ..., \lambda_N\}$, e um vetor $\boldsymbol{x}$, a verossimilhança da hipótese alternativa (H_1), com respeito a $\boldsymbol{x}$, é definida como:

$$p(\boldsymbol{x}|H_1) = F[p(\boldsymbol{x}|\lambda_1), p(\boldsymbol{x}|\lambda_2), ..., p(\boldsymbol{x}|\lambda_N)], \tag{2.33}$$

onde F é uma função de combinação (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 não se mostraram definitivas em (REYNOLDS, 1995). Mas, no geral, constatou-se que para obter o melhor desempenho dessa abordagem, os modelos de fundo 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 fundo.

A segunda abordagem (e a mais utilizada) consiste em criar apenas um modelo de fundo, independente de locutor, que é estimado utilizando locuções de vários locutores diferentes. Esse modelo único é geralmente chamado de Modelo Universal de Fundo (UBM)

e representa a distribuição dos vetores de características independente do locutor que as produziu. Geralmente, dezenas ou até centenas de horas de gravações são utilizadas para gerar o UBM, que consiste basicamente de um GMM cujos parâmetros são estimados utilizando o algoritmo EM. Por causa da grande quantidade de dados, geralmente um número elevado de misturas é utilizado, como 1024 ou 2048. É possível, ainda, a combinação de vários UBMs para a produção de um único modelo. Na prática, uma abordagem bastante utilizada consiste em construir um UBM para cada gênero (masculino e feminino) utilizando locuções produzidas por locutores de cada gênero. Os dois UBMs são, então, combinados para a produção de um único modelo UBM. Essa combinação é realizada da maneira mais simples: para dois UBMs com M e N componentes, cria-se o modelo final com as $M + N$ componentes e utiliza-se a metade dos valores dos pesos das distribuições.

2.4.3 Modelagem GMM-UBM

Em 1995, Douglas Reynolds e Richard Rose demonstraram a superioridade dos GMMs para identificação de locutores independente de texto (REYNOLDS; ROSE, 1995), comparando com modelos propostos anteriormente, como quantização vetorial (*Vector Quantization* - VQ) (SOONG et al., 1987), classificador gaussiano unimodal (GISH et al., 1985) e redes neurais com função de base radial (*Radial Basis Function* - RBF) (OGLESBY; MASON, 1991).

No mesmo ano, em (REYNOLDS, 1995), Reynolds apresentou a utilização de GMMs em conjunto com UBMs para verificação de locutores independente de texto. Foi nesse trabalho que ocorreu a consolidação da necessidade de se modelar a hipótese alternativa quando a tarefa é de verificação. Um importante detalhe sobre esse trabalho é que os modelos dos locutores e o UBM eram estimados de maneira independente, utilizando o algoritmo EM. Se por um lado locuções de diferentes locutores foram utilizadas para estimar os parâmetros do UBM, um conjunto menor de locuções produzidas pelo locutor S foi utilizado para estimar os parâmetros do modelo de S (λ_S).

No caso ideal, quando o teste da razão de verossimilhança (Equação 2.3) é utilizada para realizar a rejeição ou não de uma determinada locução de teste X, as probabilidades produzidas pelos modelos devem se complementar ($p(X|\lambda_S) + p(X|UBM) = 1$). Claramente tal característica não ocorre na prática, uma vez que apenas uma pequena amostra do universo é disponível para a estimação. Porém, o fato de os modelos serem estimados de maneira independente inibe a existência de uma correlação entre as funções de densidade de probabilidade associadas a eles. Nesse contexto, em 2000, Reynolds *et al.* propuseram um método que estabelece uma forte correlação entre os modelos do locutor e do UBM. Nesse método, descrito em (REYNOLDS; QUATIERI; DUNN, 2000), o modelo do locutor é produzido através da adaptação dos parâmetros do UBM. Essa adaptação é realizada utilizando locuções produzidas pelo locutor em questão e uma forma de adaptação Bayesiana (GAUVAIN; LEE, 1994; DUDA; HART, 1973). Tal adaptação também é chamada de

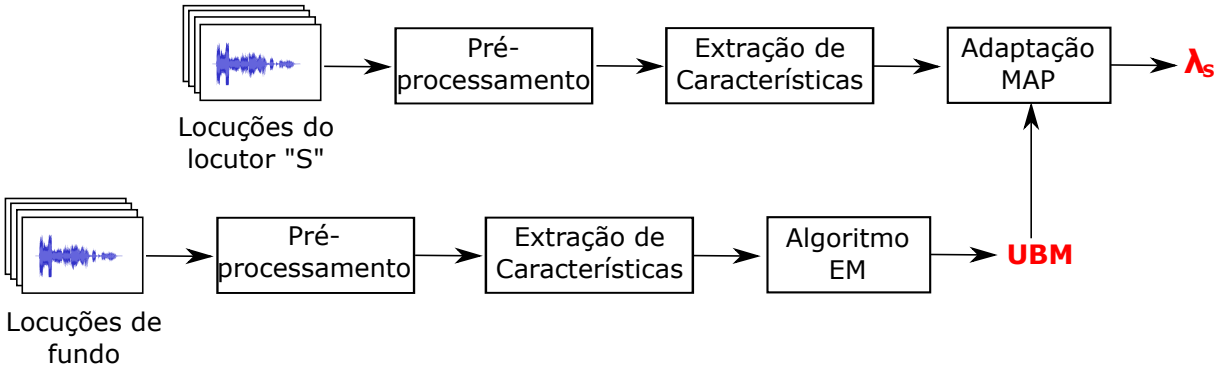

Figura 14 – Estimação dos modelos dos locutores na modelagem GMM-UBM.

aprendizagem Bayesiana e de estimação de máximo *a posteriori* (*Maximum a posteriori* - MAP). A arquitetura da modelagem GMM-UBM está ilustrada na Figura 14.

Diferente do método de estimação do modelo 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 as distribuições (componentes) presentes no UBM e no modelo do locutor. Fato que não só leva a um melhor desempenho do sistema como proporciona um método de teste mais eficiente, do ponto de vista computacional (como será visto mais adiante).

Como o algoritmo EM, a adaptação MAP 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 parâmetros antigos do UBM utilizando os chamados coeficientes de mistura dependentes dos dados. Esses coeficientes de mistura são produzidos de modo que as componentes da mistura que apresentem maior probabilidade *a posteriori* utilizem mais as estatísticas extraídas para a adaptação dos parâmetros.

Dado um UBM com matrizes de covariância diagonais e um conjunto de amostras de treinamento de um determinado locutor, $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T\}$, primeiro determina-se a representatividade das distribuições do UBM com relação às amostras. Isto é, para cada componente i do UBM, calcula-se a probabilidade *a posteriori* em relação a cada uma das amostras:

$$Pr(\lambda_i|\boldsymbol{x}_t) = \frac{\omega_i p(\boldsymbol{x}_t|\lambda_i)}{\sum_{j=1}^{M} \omega_j p(\boldsymbol{x}_t|\lambda_j)} \tag{2.34}$$

onde $p(\boldsymbol{x}_t|\lambda_i)$ é calculado utilizando a Equação 2.30 utilizando os parâmetros da componente i.

Tais probabilidades, $Pr(\lambda_i|x_t)$, são então utilizadas para o cálculo das estatísticas para cada uma das distribuições:

$$n_i = \sum_{t=1}^{T} Pr(\lambda_i|\boldsymbol{x}_t), \tag{2.35}$$

$$E_i(\boldsymbol{x}) = \frac{1}{n_i}\sum_{t=1}^{T} Pr(\lambda_i|\boldsymbol{x}_t)\boldsymbol{x}_t, \tag{2.36}$$

$$E_i(\mathbf{x}^2) = \frac{1}{n_i}\sum_{t=1}^{T} Pr(\lambda_i|\boldsymbol{x}_t)\boldsymbol{x}_t^2. \tag{2.37}$$

Por fim, essas estatísticas extraídas das amostras de treinamento são utilizadas para adaptar os parâmetros correntes de cada uma das distribuições do UBM, seguindo as seguintes equações:

$$\widehat{\omega_i} = [\alpha_i^{\omega}/T + (1-\alpha_i^{\omega})\omega_i]\gamma, \tag{2.38}$$

$$\widehat{\boldsymbol{\mu}_i} = \alpha_i^{\boldsymbol{\mu}} E_i(\boldsymbol{x}) + (1-\alpha_i^{\boldsymbol{\mu}})\boldsymbol{\mu}_i, \tag{2.39}$$

$$\widehat{\sigma_i}^2 = \alpha_i^{\sigma} E_i(\boldsymbol{x}^2) + (1-\alpha_i^{\sigma})(\sigma_i^2 + \boldsymbol{\mu}_i^2) - \widehat{\boldsymbol{\mu}_i}^2, \tag{2.40}$$

onde p_i são os parâmetros correntes e $\widehat{p_i}$ são os parâmetros adaptados na iteração, para $p \in \{\omega, \boldsymbol{\mu}, \sigma\}$.

Os coeficientes que controlam o balanço entre a estimativa corrente e as novas estimativas dos parâmetros são α_i^{ω},α_i^{μ} e α_i^{σ} para os pesos, as médias e as variâncias, respectivamente. O fator de escala γ é utilizado em todas as distribuições de modo que os pesos somem um. A atualização dos parâmetros descrita pelas Equações 2.38 - 2.40 é derivada do método de maximização *a posteriori* (MAP) para GMMs com matrizes de covariância diagonais (GAUVAIN; LEE, 1994). Porém, para cada distribuição e cada parâmetro, um coeficiente de adaptação, α_i^p, $p \in \{\omega, \mu, \sigma\}$, é adicionado às equações. Ele é definido como

$$\alpha_i^p = \frac{n_i}{n_i + r^p}, \tag{2.41}$$

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

A utilização dos coeficientes de adaptação suprime ou evidencia as atualizações das misturas dependendo da sua contagem *a posteriori* em relação às amostras, n_i. Se uma mistura apresenta valor de n_i baixo então $\alpha_i^p \to 0$ fazendo com que a adaptação seja suprimida. Por outro lado, se n_i for alto ($\alpha_i^p \to 1$), a adaptação evidencia as estatísticas calculadas. Reynolds *et al.* utilizaram um único fator de escala para os parâmetros:

$$\alpha_i^{\omega} = \alpha_i^{\mu} = \alpha_i^{\sigma} = \frac{n_i}{n_i + r}, \tag{2.42}$$

com fatores de relevância iguais a 16.

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 adaptadas. Isto é, o UBM e λ_S apresentam misturas com mesmos pesos e matrizes de variância. Nesse sentido, o processo de adaptação das médias das misturas do UBM se assemelha ao método VQ, onde os centros das gaussianas correspondem

aos vetores que são atraídos pelas observações presentes em X. Nesse caso a atualização dos vetores é adaptativa, de maneira que apenas as observações próximas aos vetores influenciam em suas atualizações.

Estimados o UBM e o modelo do locutor, λ_S, na fase de treinamento, a decisão a respeito de uma determinada locução, X, é realizada a partir do teste de razão das verossimilhanças. A arquitetura da fase de teste, na modelagem GMM-UBM, segue a mesma definida na Figura 4. Porém, nesse caso, o logaritmo das verossimilhanças é calculado. Portanto, para uma determinada locução de teste descrita a partir do conjunto de vetores extraídos, $X = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_T\}$, decide-se que X foi produzida por S se

$$\log(p(X|\lambda_S)) - \log(p(X|UBM)) \geq \theta, \tag{2.43}$$

onde θ é um limiar de rejeição e as verossimilhanças dos modelos com respeito a X são calculadas assumindo independência dos vetores:

$$p(X|\lambda) = \prod_{i=1}^{T} p(\boldsymbol{x}_i|\lambda), \tag{2.44}$$

onde λ é o modelo em questão (λ_S ou UBM) e $p(\boldsymbol{x}_i|\lambda)$ é a verossimilhança do modelo em relação ao vetor $\boldsymbol{x}_i$ (Equação 2.29). Além disso, calcula-se uma média das verossimilhanças dos vetores, de modo que o cálculo do *score* não seja impactado pela duração da locução. Dessa maneira, o logaritmo da verossimilhança de um modelo com respeito a X pode ser definido como

$$\log(p(X|\lambda)) = \frac{1}{T}\sum_{i=1}^{T} \log(p(\boldsymbol{x}_i|\lambda)). \tag{2.45}$$

Porém, o fato de o modelo do locutor λ_S ter sido gerado pela adaptação do modelo UBM permite que um método mais eficiente do cálculo do *score* seja realizado. Essa eficiência se dá sob o ponto de vista computacional e permite que as verossimilhanças sejam aproximadas levando em consideração apenas um subconjunto das misturas. Esse método de cálculo do *score* se apoia sobre dois fatos. O primeiro deles é que, geralmente, a verossimilhança de um determinado GMM com relação a um vetor se concentra em poucas misturas. Portanto, a verossimilhança pode ser muito bem aproximada ao calcular as probabilidades apenas das C distribuições mais próximas do vetor, por exemplo. O segundo fato diz respeito a um efeito importante da adaptação MAP realizada no UBM para geração do modelo do locutor. Como visto na seção anterior, os coeficientes de adaptação (Equação 2.41) controlam a adaptação dos parâmetros em função da probabilidade *a posteriori* apresentada pela mistura. Isso faz com que apenas algumas misturas sofram fortes adaptações, de maneira que o modelo do locutor e o UBM apresentem muitas misturas similares. Por essa razão, espera-se que um vetor próximo a uma determinada mistura no UBM também esteja próximo da mesma mistura adaptada no modelo do locutor.

Aliando esses dois fatos, o método proposto para o cálculo do *score* realiza o cálculo das verossimilhanças utilizando apenas as C misturas correspondentes àquelas que apresentam

maior probabilidade *a posteriori* em relação ao UBM (Equação 2.34). Para um UBM com M distribuições, esse método requer $M + C$ cálculos de verossimilhanças para cada vetor presente em X. A sua eficiência se mostra clara ao comparar com as $2M$ computações quando as verossimilhanças são calculadas levando em consideração todas as misturas. Em (REYNOLDS; QUATIERI; DUNN, 2000), Reynolds *et al.* fixaram o valor de C para 5, que é uma quantidade bastante pequena, se comparada com as 2048 misturas presentes no UBM.

2.4.4 Modelagem GMM-SVM

Em aprendizagem supervisionada, duas são as principais abordagens utilizadas para classificação: utilização de modelos generalizados ou de classificadores discriminativos. Na modelagem generalizada, padrões de cada uma das classes são utilizados para a geração de um único modelo matemático responsável pelo reconhecimento daquela classe específica. Um exemplo desse tipo modelo é uma distribuição de probabilidade, e os vetores são vistos como variáveis aleatórias que seguem essa distribuição. Já na modelagem discriminativa, os padrões das diversas classes são utilizados em conjunto para a criação de um modelo matemático responsável pela discriminação das classes. Esse modelo é comumente referenciado como classificador. Como pode ser observado, a modelagem GMM-UBM pertence à primeira família, uma vez que temos a estimação de modelos separados para cada uma das hipóteses nula e alternativa. Apesar de ter havido tentativas anteriores na utilização de classificadores discriminativos para reconhecimento de locutores, a modelagem GMM-SVM foi a primeira a conseguir resultados comparáveis (e em alguns contextos até melhores) que a modelagem puramente generativa.

A modelagem GMM-SVM faz uso das chamadas Máquinas de Vetores Suporte (*Support Vector Machines* - SVM), que são classificadores binários baseados em *kernel*. O Apêndice B descreve detalhadamente o funcionamento de uma SVM. Devido à boa capacidade de generalização das SVMs, sua adoção é capaz de proporcionar desempenho comparável ou até superior à modelagem GMM-UBM utilizando uma quantidade menor de amostras de treinamento (CAMPBELL et al., 2006). Porém, como veremos a seguir, os modelos generativos utilizados na modelagem GMM-UBM não são completamente desprezados na modelagem GMM-SVM.

Para a devida utilização de SVMs em verificação de locutores, duas considerações devem ser analisadas. A primeira delas, e provavelmente a mais importante, diz respeito ao espaço de descrição onde a classificação ocorrerá. A SVM é um classificador que realiza a separação das classes através das observações descritas através de vetores com dimensão fixa. Dessa maneira, o fato de uma locução ser descrita por uma sequência de vetores, cuja quantidade de vetores depende da duração da locução, alguma abordagem deve ser empregada para treinar um SVM capaz de separar locuções inteiras. A segunda consideração consiste no fato de os vetores produzidos pela extração das características das locuções de

treino, como os MFCCs, não serem linearmente separáveis (SCHMIDT; GISH, 1996). Para a utilização de SVMs, funções de *kernel* apropriadas devem ser utilizadas para mapear os vetores para uma dimensão maior onde as classes são linearmente separáveis e possam ser separadas por um hiperplano.

A utilização de SVMs para reconhecimento de locutores foi iniciada por Schmidt e Gish, em 1996 (SCHMIDT; GISH, 1996). Nesse trabalho, SVMs foram utilizadas para identificação de locutores e os classificadores foram treinados utilizando os próprios vetores de características estáticos extraídos das locuções. Para a decisão, foi utilizada a soma dos *scores* produzidos por cada um dos vetores da locução de teste como *score* final do sistema. Os autores utilizaram o *kernel* polinomial. Os resultados mostraram que o desempenho do sistema decai com o aumento da quantidade de vetores utilizados para treinamento. Quando os dados são inseparáveis, a quantidade de vetores suporte cresce com a quantidade de dados, e o sistema perde o poder de generalização. Essa foi a primeira evidência a respeito da necessidade de se utilizar funções de *kernel* mais apropriadas para o problema.

Em 2000, Wan e Campbell propuseram o uso de SVMs para verificação de locutores independente de texto (WAN; CAMPBELL, 2000). Nesse trabalho, o *kernel* polinomial também foi utilizado, produzindo classificadores polinomiais, bastante similar a um trabalho anterior de Campbell (CAMPBELL; ASSALEH, 1999). A decisão de uma determinada locução foi realizada ao calcular a média aritmética dos *scores* produzidos por cada um dos vetores que a compõem. A maior contribuição foi observada na fase de treinamento das SVMs. Para cada locutor, uma SVM foi treinada utilizando locuções do locutor em questão, S, e locuções de diversos outros locutores como modelagem dos impostores. Para o problema de verificação, existe uma grande quantidade de vetores da classe negativa, e como observado anteriormente, SVMs tendem a generalizar menos quando muitos vetores são utilizados. Como abordagem inicial, o método de quantização vetorial (VQ) foi utilizado para diminuir a quantidade de vetores de treinamento e eliminar *outliers*.

Em 2002, Wan e Renals avaliaram a utilização de diferentes funções de *kernel* para o problema de verificação e identificação de locutores independente de texto (WAN; RENALS, 2002). Além da avaliação utilizando SVMs treinadas com vetores MFCCs, também foi avaliada a utilização de funções de *kernel* que operam sobre o espaço dos *scores* produzidos por GMMs. Tais funções foram chamadas de *kernels* dinâmicos (ou sequenciais). As SVMs foram então treinadas a partir das probabilidades produzidas pelos GMMs utilizando os vetores extraídos das locuções. Para cada locução, calculou-se o logaritmo da verossimilhança dos modelos em relação à locução, utilizando a Equação 2.45. Para o problema de identificação, o espaço dos *scores* era definido pelos *scores* calculados utilizando cada um dos modelos dos locutores cadastrados. Para verificação, o espaço era definido pelos *scores* produzidos pelo modelo do locutor em questão e pelo UBM. Desempenhos comparáveis à modelagem GMM-UBM foram alcançados utilizando ambas as abordagens estática e dinâmica.

No caso da abordagem dinâmica, os autores propuseram uma variação do *kernel* de Fisher (JAAKKOLA; HAUSSLER, 1999) foi utilizada. Para verificação de locutores, a transformação aplicada aos vetores leva em consideração o logaritmo da razão das verossimilhanças e produz um vetor de tamanho fixo. Esse vetor é produzido pela transformação:

$$U_\theta(X) = \bigtriangledown_\theta \log \frac{P(X|\lambda_S)}{P(X|UBM)}, \tag{2.46}$$

onde θ são os parâmetros dos modelos λ_S e UBM, isto é, os parâmetros dos GMMs (vetores de média, matrizes diagonais de variância e pesos das distribuições dos dois modelos). O vetor produzido é definido, portanto, como o vetor derivada do logaritmo da razão das verossimilhanças com respeito aos parâmetros de um GMM. Levando em consideração duas locuções, X e Y, o *kernel* de Fisher é definido como

$$k(X, Y) = U_\theta(X)^t F^{-1} U_\theta(Y), \tag{2.47}$$

onde F é matriz de informação de Fisher e é definida como

$$F = \mathbb{E}_{X,Y}\Big[U_\theta(X)U_\theta(Y)^t\Big]. \tag{2.48}$$

Ainda sobre *kernels* sequenciais, em 2005, Wan e Renals propuseram melhorias na utilização do *kernel* de Fisher utilizando normalização esférica (WAN; RENALS, 2005). Nesse trabalho, SVMs ainda eram utilizados sobre o espaço dos *scores* produzidos pelo sistema GMM-UBM. Pela primeira vez, a utilização de SVMs melhoraram consideravelmente os resultados obtidos pelo sistema GMM-UBM. Nos experimentos apresentados, uma melhora de 34% foi observada.

Porém, foi em 2006 que Campbell, Sturim e Reynolds propuseram uma nova abordagem que utiliza SVMs para a verificação de locutores independente de texto que se tornou a abordagem padrão GMM-SVM (CAMPBELL; STURIM; REYNOLDS, 2006). Nela, os autores definiram os chamados supervetores[9] GMM (*GMM supervectors*), que foram utilizados como descritores de locuções. SVMs então operam sobre esse espaço utilizando funções de *kernel* apropriadas para diferenciar tais supervetores. Na modelagem GMM-SVM proposta por Campbell *et al.*, cada locução é descrita por um vetor estático de dimensão altíssima. Esse vetor é denominado supervetor GMM e é produzido utilizando uma abordagem definida pela modelagem GMM-UBM. A Figura 15 mostra como um supervetor é produzido a partir de uma determinada locução. O UBM sofre a adaptação MAP definida por Reynolds *et al.* em (REYNOLDS; QUATIERI; DUNN, 2000) (e utilizada na modelagem GMM-UBM) utilizando os vetores extraídos da locução. Nessa adaptação, apenas as médias são alteradas. Tal procedimento foi descrito na Seção 2.4.3. A adaptação produz um modelo GMM com o mesmo número de misturas que o UBM. Além

[9] Esse termo geralmente é empregado quando há uma combinação (mais comumente, concatenação) de vetores para a criação de um vetor de mais alta dimensão que possua informação de todos os vetores combinados.

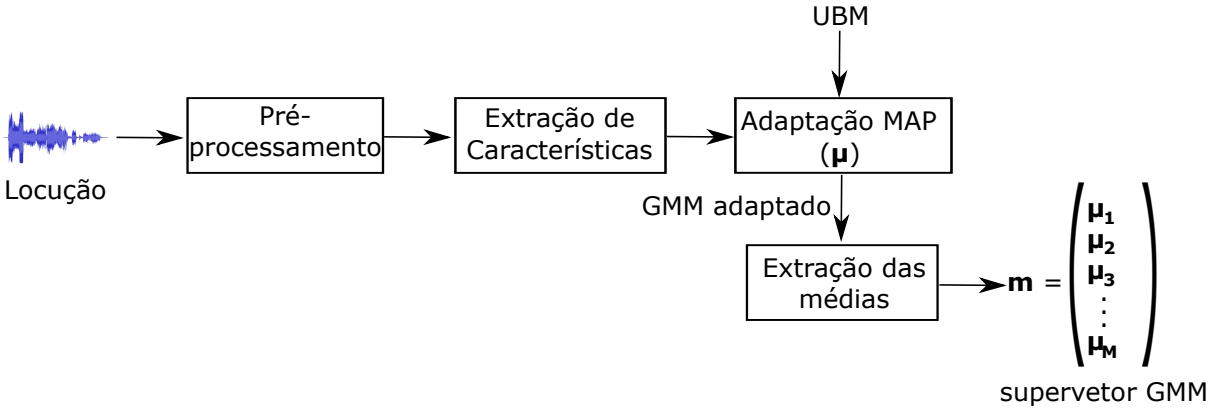

Figura 15 – Geração de um supervetor GMM a partir de uma determinada locução utilizando o UBM e a adaptação MAP das médias.

disso, como apenas as médias foram adaptadas, os pesos e as matrizes de variância das distribuições são iguais àqueles presentes no UBM. O supervetor GMM é então definido pela concatenação dos vetores das médias das misturas do modelo adaptado.

Como mencionado anteriormente, em (REYNOLDS; QUATIERI; DUNN, 2000), os autores descobriram que, na modelagem GMM-UBM, melhores resultados são obtidos quando apenas as médias são adaptadas na geração dos modelos dos locutores a partir da adaptação do UBM. Nesse contexto, pode-se observar que o supervetor, em conjunto com os parâmetros não adaptados do UBM) carrega toda a informação necessária para definição do modelo treinado apenas com aquela locução. Produz-se, então, uma descrição que possui todo o poder de reconhecimento que um GMM, adaptado para aquela locução, possui. Nota-se também que tal descrição possui dimensão altíssima. Por exemplo, se o UBM estimado anteriormente possuir 2048 misturas, e os vetores de características possuírem dimensão 57 (se o conjunto de características extraídas for formado por 19 coeficientes MFCC e os coeficientes delta de primeira e segunda ordem, por exemplo), tais supervetores possuirão dimensão 116736 (2048×57).

O supervetor GMM pode ser visto como um mapeamento entre a locução e um vetor de alta dimensão. Tal conceito se encaixa bem com a ideia de SVMs com *kernel* sequencial. A ideia básica desse tipo de SVM é comparar duas locuções, X e Y, diretamente através de uma função de *kernel* $k(X, Y)$. Além disso, é conveniente (pelas razões expostas no Apêndice B) que tal função seja escrita como o produto interno dos mapeamentos realizados nas locuções:

$$k(X, Y) = \phi(X) \cdot \phi(Y). \tag{2.49}$$

O mapeamento realizado pela geração do supervetor GMM pode ser visto, então, como uma parte do mapeamento $\phi(X)$. Por exemplo, para o caso simplório da função de *kernel* linear[10], o mapeamento $\phi(X)$ consiste exatamente do supervetor GMM. Porém, os autores foram motivados a utilizar funções de *kernel* mais apropriadas para o problema.

[10] O *kernel* linear consiste do *kernel* polinomial (Equação B.10) de ordem 1. Isto é, $k(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y}$.

Como os supervetores GMM descrevem as locuções utilizando modelos GMMs, os *kernels* propostos consideram medidas de distância entre GMMs. O que é bastante apropriado, uma vez que existe uma relação intrínseca entre produto interno e medida de distância. Os autores propuseram duas funções de *kernel*, que são descritas a seguir.

- *Kernel* linear para supervetores GMM

  Essa função de *kernel* é baseada na divergência KL (Kullback-Leibler), que é uma medida de diferença assimétrica entre distribuições de probabilidade. Suponha dois GMMs, em $\mathbb{R}^D$, gerados pela adaptação MAP do UBM utilizando as locuções A e B, g_a e g_b. A divergência KL entre eles é definida como

  $$D(g_a \parallel g_b) = \int_{\mathbb{R}^D} g_a(\mathbf{x}) \log \frac{g_a(\mathbf{x})}{g_b(\mathbf{x})} d\mathbf{x}. \tag{2.50}$$

  Ao invés de utilizar a divergência diretamente como função de *kernel*, os autores consideraram uma aproximação. A ideia é limitar a divergência a partir da desigualdade logaritmo-soma (DO, 2003). Para GMMs com M misturas (como definidos na Equação 2.29):

  $$D(g_a \parallel g_b) \leq \sum_{i=1}^{M} \omega_i D(N_{a,i} \parallel N_{b,i}), \tag{2.51}$$

  onde $N_{a,i}$ e $N_{b,i}$ são as distribuições normais da i-ésima mistura para g_a e g_b, respectivamente. Os parâmetros das distribuições são definidos pelos vetores de média, $\boldsymbol{\mu}_{a,i}$ e $\boldsymbol{\mu}_{b,i}$, e as matrizes de covariância $\boldsymbol{\Sigma}_i$, que nesse caso são iguais para ambos os GMMs, uma vez que foram produzidos pela adaptação MAP das médias do UBM. O mesmo vale para os pesos ω_i. Considerando matrizes de variância diagonais, a aproximação realizada pode ser escrita na seguinte fórmula fechada:

  $$d(g_a, g_b) = \frac{1}{2} \sum_{i=1}^{M} \omega_i (\boldsymbol{\mu}_{a,i} - \boldsymbol{\mu}_{b,i})^T \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{\mu}_{a,i} - \boldsymbol{\mu}_{b,i}). \tag{2.52}$$

  A desigualdade final é, portanto,

  $$0 \leq D(g_a \parallel g_b) \leq d(g_a, g_b). \tag{2.53}$$

  Pode-se, finalmente, escrever $d(g_a, g_b)$ em função de produtos internos a fim de produzir a função de *kernel* final[11]:

  $$k(A, B) = \sum_{i=1}^{M} \omega_i (\boldsymbol{\mu}_{a,i})^T \boldsymbol{\Sigma}_i (\boldsymbol{\mu}_{b,i}) \tag{2.54}$$

  $$= \sum_{i=1}^{M} (\sqrt{\omega_i} \boldsymbol{\Sigma}_i^{(1/2)} \boldsymbol{\mu}_{a,i}) \cdot (\sqrt{\omega_i} \boldsymbol{\Sigma}_i^{(1/2)} \boldsymbol{\mu}_{b,i}) \tag{2.55}$$

[11] A conversão da distância em produtos internos (e vice-versa) é realizada através da identidade polar: $\boldsymbol{u} \cdot \boldsymbol{v} = \frac{1}{2}(\parallel \boldsymbol{u} \parallel^2 + \parallel \boldsymbol{v} \parallel^2 - \parallel \boldsymbol{u} - \boldsymbol{v} \parallel^2)$ (CONWAY, 2010).

Pode-se observar que a função resultante é escrita em função dos supervetores GMM ($\boldsymbol{\mu}_{a,i}$ e $\boldsymbol{\mu}_{b,i}$, $i = 1, ..., M$) e dos parâmetros não adaptados do UBM. Dessa maneira, ela pode ser escrita como função dos supervetores, $k(\boldsymbol{\mu}_a, \boldsymbol{\mu}_b)$, e a expansão realizada (para a i-ésima componente do supervetor) é definida como

$$\phi(\boldsymbol{\mu}_i) = \sqrt{\omega_i}\boldsymbol{\Sigma}_i^{(1/2)}\boldsymbol{\mu}_i, \tag{2.56}$$

de modo que o supervetor projetado é definido pela concatenação das suas componentes projetadas.

Como mencionado no Apêndice B, SVMs realizam a otimização quadrática no espaço projetado por ϕ sem de fato computar tais projeções. Isso é realizado implicitamente, no espaço original, pela utilização da função de *kernel*. Porém, uma característica bastante útil dessa função de *kernel* é que uma técnica de compactação de modelo (CAMPBELL, 2002) pode ser utilizada, de maneira que a classificação de um determinado vetor (Equação B.9) pode ser realizada da seguinte maneira:

$$f(\mathbf{x}) = \sum_i \alpha_i y_i (k(\mathbf{x}, \mathbf{x}_i)) + b \tag{2.57}$$

$$= \sum_i \alpha_i y_i (\phi(\mathbf{x}) \cdot \phi(\mathbf{x}_i)) + b \tag{2.58}$$

$$= \left(\sum_i \alpha_i y_i \phi(\mathbf{x}_i) \right)^T \phi(\mathbf{x}) + b. \tag{2.59}$$

Isto significa que a classificação pode ser realizada a partir de um único produto interno entre a projeção do supervetor GMM com uma combinação das projeções dos vetores suporte da SVM.

- *Kernel* de produto interno L^2

  Essa função de *kernel* é baseada no produto interno no espaço de funções. Suponha, novamente, dois GMMs gerados pela adaptação MAP do UBM utilizando as locuções A e B, g_a e g_b, com dimensão $\mathbb{R}^D$. O produto interno padrão no espaço de funções é definido como

  $$k(g_a, g_b) = \int_{\mathbb{R}^D} g_a(\mathbf{x}) g_b(\mathbf{x}) d\mathbf{x}. \tag{2.60}$$

  Uma fórmula fechada para a integral acima pode ser encontrada. Utilizando a notação anterior dos GMMs com M misturas, obtém-se:

  $$k(g_a, g_b) = \sum_{i=1}^{M} \sum_{j=1}^{M} \omega_i \omega_j \int_{\mathbb{R}^D} N(\mathbf{x}; \boldsymbol{\mu}_{a,i}, \boldsymbol{\Sigma}_i) N(\mathbf{x}; \boldsymbol{\mu}_{b,j}, \boldsymbol{\Sigma}_j) d\mathbf{x} \tag{2.61}$$

  $$= \sum_{i=1}^{M} \sum_{j=1}^{M} \omega_i \omega_j N(\boldsymbol{\mu}_{a,i} - \boldsymbol{\mu}_{b,j}; \mathbf{0}, \boldsymbol{\Sigma}_i + \Sigma_j), \tag{2.62}$$

onde $\mathbf{0}$ é o vetor nulo e $N(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ é a distribuição normal multivariada. Os autores ainda utilizaram uma aproximação computacionalmente conveniente ao assumir que as médias de diferentes misturas são distantes. Tal suposição faz com que os termos $i \neq j$ sejam pequenos, de maneira que possam ser ignorados. A função de *kernel* resultante utilizada é:

$$k(g_a, g_b) = \sum_{i=1}^{M} \omega_i^2 N(\boldsymbol{\mu}_{a,i} - \boldsymbol{\mu}_{b,i}; \mathbf{0}; 2\boldsymbol{\Sigma}_i). \tag{2.63}$$

Apesar da proposta de ambas as funções de *kernel*, os experimentos realizados por Campbell *et al.* mostraram a superioridade do *kernel* linear para supervetores GMM. Esse *kernel* se tornou a função padrão utilizada na modelagem GMM-SVM. De fato, em trabalhos posteriores (CAMPBELL et al., 2006; STURIM et al., 2009; KINNUNEN; LI, 2010), apenas essa função é levada em consideração.

2.4.5 Modelagem via *i-vectors*

Como visto anteriormente, na abordagem GMM-UBM, os modelos dos locutores são produzidos através da adaptação MAP das médias do UBM utilizando vetores extraídos de locuções provenientes dos locutores. Logo após, vimos que as médias adaptadas são capazes de produzir uma descrição para o modelo resultante da adaptação utilizando apenas uma locução, e que isso pode ser utilizado como uma representação para a locução, chamada de supervetor GMM. Se imaginarmos diferentes locuções de um mesmo locutor, a variabilidade de sessão provoca uma variabilidade nos supervetores correspondentes a elas. Dessa maneira, se diferentes locuções produzem diferentes modelos, o modelo do locutor, na abordagem GMM-UBM, é inerentemente afetado pelas variabilidades de sessão.

Nesse contexto, em 2005, Kenny *et al.* propuseram uma modelagem baseada na análise fatorial (*factor analysis*) para decompor o espaço dos parâmetros dos GMMs em um subespaço latente referente às informações do locutor e um subespaço residual (KENNY; BOULIANNE; DUMOUCHEL, 2005; KENNY, 2005). O intuito era utilizar os supervetores produzidos pelas locuções para determinar um subespaço das médias dos modelos dos locutores (espaço dos *eigenvoices*) sem a influência dos fatores de sessão, permitindo assim a criação de modelos mais robustos. Os métodos desenvolvidos para essa decomposição formaram o ferramental necessário para o desenvolvimento de uma abordagem de decomposição mais geral, que foi denominada de análise fatorial conjunta (*Joint Factor Analysis* - JFA) (KENNY et al., 2005; KENNY et al., 2007b; KENNY et al., 2007a; KENNY et al., 2008).

Suponha que o espaço dos supervetores é definido através da adaptação de um UBM com C componentes de mistura e vetores de dimensão D. Na abordagem JFA, um supervetor $\boldsymbol{M}$, de dimensão $CD = C \times D$, é descrito pela soma de dois supervetores: $\boldsymbol{s}$, referente ao locutor, e $\boldsymbol{c}$, correspondente à sessão:

$$\boldsymbol{M} = \boldsymbol{s} + \boldsymbol{c}. \tag{2.64}$$

Assume-se que $\boldsymbol{s}$ e $\boldsymbol{c}$ são estatisticamente independentes e seguem distribuições normais. Além disso, assume-se que $\boldsymbol{s}$ segue distribuição cuja decomposição em variáveis latentes é definida por

$$\boldsymbol{s} = \boldsymbol{m} + \boldsymbol{V}\boldsymbol{y} + \boldsymbol{R}\boldsymbol{z}, \tag{2.65}$$

onde $\boldsymbol{m}$ é um supervetor de dimensão CD, geralmente definido como o supervetor médio da população, $\boldsymbol{V}$ é uma matriz retangular de *rank* reduzido[12], $\boldsymbol{R}$ é uma matriz diagonal e $\boldsymbol{y}$ e $\boldsymbol{z}$ seguem a distribuição normal padronizada $N(0, \boldsymbol{I})$ correspondente a cada dimensionalidade. O supervetor $\boldsymbol{m}$ é o supervetor médio, independente de locutor e de sessão, e geralmente é extraído do UBM. Enquanto $\boldsymbol{V}$ é a matriz de *eigenvoice* que define a projeção para o subespaço referente ao locutor, $\boldsymbol{R}$ define a projeção para o subespaço residual. Similarmente, a componente de sessão segue uma decomposição da forma

$$\boldsymbol{c} = \boldsymbol{U}\boldsymbol{x}, \tag{2.66}$$

onde $\boldsymbol{U}$ é uma matriz retangular de *rank* reduzido e $\boldsymbol{x}$ segue uma distribuição normal padronizada. $\boldsymbol{U}$ é chamada de matriz de *eigenchannel*[13] e define o subespaço de sessão (ou canal). $\boldsymbol{y}$, $\boldsymbol{z}$ e $\boldsymbol{x}$ são os fatores do locutor, residual e de sessão, respectivamente. A abordagem utiliza supervetores produzidos por diversos locutores para estimar as matrizes $\boldsymbol{V}$, $\boldsymbol{R}$ e $\boldsymbol{U}$. Para essas estimativas os supervetores GMM são descritos através das estatísticas de Baum–Welch (Equações 2.35, 2.36 e 2.37), extraídas utilizando o UBM (KENNY; BOULIANNE; DUMOUCHEL, 2005; KENNY et al., 2005; KENNY et al., 2007b). Uma vez estimadas, para a geração do modelo de um determinado locutor S, as matrizes são utilizadas para calcular os fatores $\boldsymbol{y}_S, \boldsymbol{z}_S, \boldsymbol{x}_S$ correspondentes ao supervetor extraído das locuções de cadastro de S. As componentes $\boldsymbol{y}_S$ e $\boldsymbol{z}_S$ são utilizadas para geração do modelo do locutor sem a influência do subespaço referente à componente de sessão ($\boldsymbol{M} - \boldsymbol{U}\boldsymbol{x}$), que é então utilizado para a verificação de uma locução de teste. A decisão, em si, geralmente é realizada através do teste de razão das verossimilhanças entre o modelo do locutor e do UBM (como na abordagem GMM-UBM), mas diferentes métodos para produção do *score* do sistema foram propostos (KENNY et al., 2008; GLEMBEK et al., 2009).

Em 2009, Dehak *et al.* propuseram a incorporação da decomposição JFA dos supervetores na abordagem GMM-SVM (DEHAK et al., 2009). Os fatores referentes ao locutor ($\boldsymbol{y}_S, \boldsymbol{z}_S$) foram utilizados como novas representações das locuções para o treinamento dos SVMs. Duas importantes descobertas foram realizadas pelos autores. A primeira delas é que, ao utilizar os fatores como representações, bons resultados são alcançados utilizando funções de *kernel* mais simples, como o *kernel* linear e o do cosseno. Nesse momento, as locuções não são mais descritas sob a forma de distribuições e a comparação entre os diferentes locutores pode ser realizada através da comparação direta entre os vetores $\boldsymbol{y}_S$. A

[12] $rank(\boldsymbol{V}) \ll CD$.

[13] O termo "canal" passou a ser utilizado, de uma maneira mais genérica, para denominar sessão. Nesse caso, as componentes de sessão, responsáveis pela variabilidade intra-locutor, são denominadas, também, como componentes de canal.

segunda descoberta consiste no fato de, ao invés de realizar a decomposição completa do JFA (Equação 2.64), apenas a decomposição dos *eigenvoices* foi realizada (Equação 2.65). A compensação de sessão foi realizada pós-processando os vetores das locuções, através do método de Normalização de Covariância Intraclasse (*Within Class Covariance Normalization* - WCNN), que, basicamente, define uma matriz de transformação para os dados baseado na variabilidade dos vetores dos locutores em relação ao vetor-médio de cada locutor (HATCH; KAJAREKAR; STOLCKE, 2006). Isto é, ao invés de empregar um método complexo de decomposição para remover - do espaço dos parâmetros das distribuições - a variabilidade de sessão ($\boldsymbol{c}$), realiza-se a decomposição apenas para as componentes do locutor ($\boldsymbol{s}$) e aplica-se a compensação nesse novo espaço, de dimensão reduzida, utilizando métodos mais simples, como o WCNN.

Em (DEHAK, 2009), Dehak mostrou que as componentes de locutor e de sessão não são estatisticamente independentes e que a componente de canal também possui informações de locutor. Baseado nisso, Dehak *et al.* propôs uma simplificação da decomposição para os supervetores que, ao invés de definir dois subespaços independentes, define apenas um subespaço contendo simultaneamente as informações de locutor e de sessão (DEHAK et al., 2009). Esse espaço foi denominado de espaço de variabilidade total (*total variability space*), e possui o objetivo de criar uma representação compacta da locução, agregando todo tipo de informação responsável pela variabilidade entre os supervetores. Neste caso, o supervetor definido na Equação 2.64 é reescrito sob a forma

$$\boldsymbol{M} = \boldsymbol{m} + \boldsymbol{T}\boldsymbol{w}, \tag{2.67}$$

onde $\boldsymbol{T}$ é uma matriz retangular de *rank* reduzido e $\boldsymbol{w}$ segue uma distribuição normal padronizada. $\boldsymbol{T}$ é chamada de matriz de variabilidade total e $\boldsymbol{w}$ é o fator de variabilidade total. O método utilizado para estimação de $\boldsymbol{T}$ é o mesmo utilizado para a estimação da matriz de *eigenvoices* (matriz V da Equação 2.65). Esse método é descrito no Apêndice C. A única diferença é que, para a estimação da matriz de variabilidade total, considera-se que cada locução foi produzida por um locutor diferente. Dessa maneira, tal decomposição pode ser vista como simples análise de fatores que permite a projeção de uma determinada locução para um espaço de variabilidade total de baixa dimensão. Uma vez estimada a matriz, assim como na modelagem JFA, a componente $\boldsymbol{w}$ é definida através das estatísticas de Baum–Welch do UBM. Na fase de cadastro, o fator correspondente às locuções do locutor é computada e armazenada como *template*. Durante a fase de verificação, a representação referente à locução de teste é comparada com a do locutor correspondente à alegação.

Dehak *et al.* mostraram que, utilizando os fatores de variabilidade total como representação para as locuções e aplicando técnicas de pós-processamento para compensação das variabilidades de sessão, métodos de comparação mais simples podem ser empregados. Em (DEHAK et al., 2009), os fatores foram comparados através da similaridade do cosseno:

$$score(\boldsymbol{w_S}, \boldsymbol{w_t}) = \frac{\boldsymbol{w_S} \cdot \boldsymbol{w_t}}{\| \boldsymbol{w_S} \| \| \boldsymbol{w_t} \|}, \tag{2.68}$$

onde $score(\boldsymbol{w_S}, \boldsymbol{w_t})$ define *score* do sistema ao comparar o *i-vector* de teste $\boldsymbol{w_t}$ com o *i-vector* correspondente ao locutor S, $\boldsymbol{w_S}$. Esse *score* é comparado com um limiar de aceitação para a decisão final. Ao pós-processar os fatores com técnicas como WCNN e LDA, melhores resultados foram alcançados quando comparados com a abordagem GMM-SVM. Em (DEHAK et al., 2011), os autores denominaram os fatores de variabilidade total como "vetores-identidade", ou *i-vectors*. Com experimentos mais extensos, eles mostraram novamente a robustez dos *i-vectors* em relação ao GMM-SVM. Além disso, melhores resultados foram observados ao utilizar o LDA como método de pós-processamento para aumentar o poder de discriminação dos *i-vectors*.

Em (KENNY, 2010), Kenny *et al.* propuseram a utilização de modelos generativos a fim de realizar a processo de compensação no espaço dos *i-vectors* através da modelagem das variabilidades de sessão. A modelagem utilizada foi a análise de discriminação linear probabilística (*Probabilistic Linear Discriminant Analysis* - PLDA). Essa técnica foi originalmente proposta para reconhecimento facial (PRINCE; ELDER, 2007) e mais tarde adaptada para reconhecimento de locutores (KENNY, 2010; BURGET et al., 2011). A modelagem inicial é chamada de PLDA Gaussiana (*Gaussian* PLDA – G-PLDA) e assume que um determinado *i-vector* pode ser decomposto em suas componentes específicas de locutor e de sessão:

$$\boldsymbol{w} = \boldsymbol{w_m} + \boldsymbol{\Phi\beta} + \boldsymbol{\Gamma\alpha} + \boldsymbol{\epsilon}_r. \tag{2.69}$$

onde $\boldsymbol{w_m}$ é o *i-vector* médio da população, $\boldsymbol{\Phi}$ é a matriz de *eigenvoices* e $\boldsymbol{\beta}$ é a componente latente do locutor, $\boldsymbol{\Gamma}$ é a matriz de *eigenchannels* e $\boldsymbol{\alpha}$ é a componente latente de sessão, e $\boldsymbol{\epsilon}$ é a componente residual. A decomposição é muito similar àquela utilizada na modelagem JFA. Todas as componentes são assumidas estatisticamente independentes e $\boldsymbol{\beta}$ e $\boldsymbol{\alpha}$ seguem distribuições normais padronizadas. Na modelagem G-PLDA convencional, assume-se que componente residual $\boldsymbol{\epsilon_r}$ segue uma distribuição normal com média zero e matriz de covariância diagonal $\boldsymbol{\Sigma}$. Porém, como a dimensionalidade dos *i-vectors* é reduzida, Kenny *et al.* propuseram uma simplificação do modelo utilizando uma matriz de covariância completa para a componente residual e a remoção das componentes de *eigenchannels*, $\boldsymbol{\Gamma\alpha}$. Essa simplificação foi adotada desde então (KENNY, 2010; BURGET et al., 2011; GARCIA-ROMERO; ESPY-WILSON, 2011). Nesse caso a decomposição é da forma

$$\boldsymbol{w} = \boldsymbol{w_m} + \boldsymbol{\Phi\beta} + \boldsymbol{\epsilon}_r, \tag{2.70}$$

onde $\boldsymbol{\epsilon_r}$ segue uma distribuição normal com média zero e matriz de covariância completa $\boldsymbol{\Sigma}$. O método para estimação dos parâmetros, utilizando EM, é muito similar ao utilizado na abordagem JFA (PRINCE; ELDER, 2007). Antes da estimação, os *i-vectors* são centralizados ($\boldsymbol{w} - \boldsymbol{w_m}$) utilizando o *i-vector* médio da população de treinamento. Dessa maneira, o objetivo é estimar a matriz de *eigenvoices* $\boldsymbol{\Phi}$ e a matriz de covariância residual $\boldsymbol{\Sigma}$.

Uma das grandes vantagens da modelagem G-PLDA é que, além de realizar uma compensação sobre as variabilidades de sessão, ela também pode ser utilizada para a comparação entre dois *i-vectors* e, assim, realizar a tarefa de verificação. Dados os dois *i-vectors* envolvidos na tarefa de verificação: $\boldsymbol{w_S}$, correspondente ao locutor alegado e $\boldsymbol{w_t}$, referente à locução de teste, considere o teste de razão das verossimilhanças sobre duas hipóteses H_s e H_d:

$$\text{score} = \log \frac{p(\boldsymbol{w_S}, \boldsymbol{w_t}|H_s)}{p(\boldsymbol{w_S}|H_d)p(\boldsymbol{w_t}|H_d)}, \tag{2.71}$$

onde

$$\begin{aligned} H_s&: \quad \boldsymbol{\beta_S} = \boldsymbol{\beta_t}, \\ H_d&: \quad \boldsymbol{\beta_S} \neq \boldsymbol{\beta_t}, \end{aligned} \tag{2.72}$$

e $\boldsymbol{\beta_S}$ e $\boldsymbol{\beta_t}$ são as componentes de locutor correspondentes à decomposição de $\boldsymbol{w_S}$ e $\boldsymbol{w_t}$, respectivamente. Tal abordagem basicamente define um teste sobre as hipóteses dos *i-vectors* terem sido ou não produzidas pelo mesmo locutor, compartilhando assim, a mesma componente $\boldsymbol{\beta}$ da Equação 2.70. Para o caso gaussiano (G-PLDA), a razão das verossimilhanças da Equação 2.71 possui uma solução fechada, definida por:

$$\begin{aligned} \text{score} = \quad & \log\ N\left(\begin{bmatrix} \hat{\boldsymbol{w}}_{\boldsymbol{S}} \\ \hat{\boldsymbol{w}}_{\boldsymbol{t}} \end{bmatrix}; \mathbf{0}, \begin{bmatrix} \boldsymbol{\Sigma_A} & \boldsymbol{\Sigma_L} \\ \boldsymbol{\Sigma_L} & \boldsymbol{\Sigma_A} \end{bmatrix}\right) \\ & \text{- } \log\ N\left(\begin{bmatrix} \hat{\boldsymbol{w}}_{\boldsymbol{S}} \\ \hat{\boldsymbol{w}}_{\boldsymbol{t}} \end{bmatrix}; \mathbf{0}, \begin{bmatrix} \boldsymbol{\Sigma_A} & 0 \\ 0 & \boldsymbol{\Sigma_L} \end{bmatrix}\right), \end{aligned} \tag{2.73}$$

onde $\hat{\boldsymbol{w}}_{\boldsymbol{S}}$ e $\hat{\boldsymbol{t}}_{\boldsymbol{S}}$ são os *i-vectors* centralizados $\boldsymbol{w_S} - \boldsymbol{w_m}$ e $\boldsymbol{w_t} - \boldsymbol{w_m}$, $\boldsymbol{\Sigma_L}$ é a matriz de covariância relativa à componente do locutor e $\boldsymbol{\Sigma_A}$ a matriz de covariância total:

$$\boldsymbol{\Sigma_A} = \boldsymbol{\Phi}\boldsymbol{\Phi}^{\boldsymbol{t}} + \boldsymbol{\Sigma}, \tag{2.74}$$

$$\boldsymbol{\Sigma_L} = \boldsymbol{\Phi}\boldsymbol{\Phi}^{\boldsymbol{t}}. \tag{2.75}$$

Expandindo, temos:

$$score = \boldsymbol{w_S}^t \boldsymbol{Q} \boldsymbol{w_S} + \boldsymbol{w_t}^t \boldsymbol{Q} \boldsymbol{w_t} + 2\boldsymbol{w_S}^t \boldsymbol{P} \boldsymbol{w_t}, \tag{2.76}$$

onde:

$$\boldsymbol{Q} = \boldsymbol{\Sigma_A}^{-1} - \left(\boldsymbol{\Sigma_A} - \boldsymbol{\Sigma_L}\boldsymbol{\Sigma_A}^{-1}\boldsymbol{\Sigma_L}\right)^{-1}, \tag{2.77}$$

$$\boldsymbol{P} = \boldsymbol{\Sigma_A}^{-1}\boldsymbol{\Sigma_L} - \left(\boldsymbol{\Sigma_A} - \boldsymbol{\Sigma_L}\boldsymbol{\Sigma_A}^{-1}\boldsymbol{\Sigma_L}\right)^{-1}. \tag{2.78}$$

Em (KENNY, 2010) Kenny *et al.* ainda propuseram uma mudança na modelagem padrão, onde, ao invés de assumir distribuições normais para as variáveis latentes, estas

assumem uma distribuição $t-Student$. Os autores realizaram uma análise sobre os graus de liberdade das distribuições e propuseram alternativas para o cálculo da razão das verossimilhanças da Equação 2.71, uma vez que a mesma não possui uma solução fechada como no caso onde se assumem distribuições normais[14]. Essa proposta é referenciada como PLDA de cauda pesada (*Heavy-tailed* PLDA – HT-PLDA). Apesar da modelagem HT-PLDA ter apresentado ganhos de desempenho significativos quando comparada com a modelagem G-PLDA, Garcia-Romero *et al.* mostraram que ao se utilizar algum método de pós-processamento (LDA, por exemplo) seguido de normalização de comprimento (divisão pela norma do vetor) antes da modelagem G-PLDA, resultados semelhantes à modelagem HT-PLDA são alcançados (GARCIA-ROMERO; ESPY-WILSON, 2011). Isso se deve principalmente ao fato de que o processo de normalização realiza uma transformação para um espaço mais próximo do gaussiano, isto é, retira parte do aspecto de cauda longa das distribuições dos *eigenvoices* (LYU; SIMONCELLI, 2009). Pela simplicidade e eficiência computacional, quando comparado com o método HT-PLDA, a abordagem mais utilizada consiste na modelagem G-PLDA com *i-vectors* pós-processados e normalizados.

2.5 VERIFICAÇÃO DE LOCUTORES E APRENDIZAGEM PROFUNDA

A década de 2010 foi marcada pelo avanço e consolidação de novas abordagens para reconhecimento de locutores, baseadas em aprendizagem profunda e, mais especificamente, em redes neurais profundas (DNNs). Esse avanço foi impulsionado principalmente pelos resultados obtidos por métodos utilizando DNNs em outros problemas envolvendo sinais de voz, como reconhecimento automático de fala (ASR) (HINTON et al., 2012; DENG; HINTON; KINGSBURY, 2013) e identificação de idioma (MATEJKA et al., 2014; SONG et al., 2013; FERRER et al., 2015).

2.5.1 Uso de DNNs treinadas para ASR

Em ASR, DNNs foram introduzidos ao processo de modelagem acústica, desempenhando o papel, até então realizado com GMMs, de computar as probabilidades *a posteriori* das classes fonéticas. Acoplado à modelagem sequencial utilizando Modelos Escondidos de Markov (*Hidden Markov Models* - HMMs), ganhos de desempenho de até 30% foram observados em relação ao convencional sistema GMM-HMM (VESELỲ et al., 2013). Nesse tipo de sistema, uma DNN é utilizada para inferir o conteúdo fonético de janelas extraídas das locuções. A abordagem consiste em um modelo de classificação cuja entrada é composta por um conjunto de características acústicas, geralmente constituída das energias resultantes de um banco de filtros espectrais (Seção 2.3.4), e as saídas são as probabilidades *a porteriori* (*posteriors*) dos chamados senones (Figura 16). Um senone corresponde

[14] Os autores utilizam limitantes inferiores com solução variacional para as distribuições maginais.

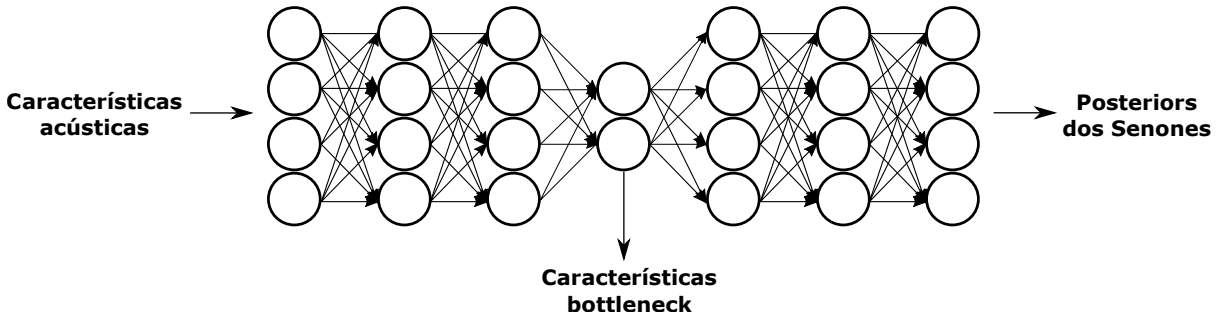

Figura 16 – Modelo de uma DNN utilizada por um sistema de ASR para a classificação do senone presente em uma determinada janela da locução.

a uma unidade de classificação fonética, tipicamente definida por um conjunto de trifones (arranjo de três fonemas). Tanto o conjunto de fonemas quanto o conjunto de senones são fixos e definidos em uma etapa anterior do sistema.

Os primeiros avanços em reconhecimento de locutores seguindo essa abordagem surgiram através da incorporação à modelagem com *i-vectors* de informações acústicas provenientes de DNNs treinadas para ASR. Em 2014, Kenny *et al.* propuseram a utilização da DNN para a modelagem de fundo, como uma alternativa ao UBM, utilizando os *posteriors* dos senones como as estatísticas de ordem zero (Equação 2.35), que são utilizadas para o cálculo das estatísticas de primeira ordem (Equação 2.36). Elas são então utilizadas para a estimativa da matriz de variabilidade total e, consequentemente, para geração dos *i-vectors* (KENNY et al., 2014). Também em 2014, a mesma abordagem foi proposta por Lei *et al.* (LEI et al., 2014b). Em 2015, McLaren *et al.* propuseram o uso das chamadas características de gargalo (*Bottleneck Features* - BFs) em conjunto com os coeficientes MFCC para formar o conjunto de características a ser utilizado na modelagem convencional com *i-vectors* (MCLAREN; LEI; FERRER, 2015). Tal abordagem já tinha sido utilizada anteriormente para identificação de idioma (SONG et al., 2013). As BFs são extraídas de uma camada intermediária da DNN treinada para ASR. Tal camada tem o objetivo de comprimir as informações aprendidas pelas camadas anteriores em um espaço de dimensão reduzida (ver Figura 16). Todas essas abordagens apresentaram ganhos de desempenho para sistemas de reconhecimento de locutores, porém, a utilização de DNNs para ASR requer bases de dados de treinamento com informações fonéticas classificadas (áudios com as falas transcritas), o que aumenta bastante a complexidade do sistema.

2.5.2 DNNs supervisionadas para classificação de locutores

A abordagem que se popularizou em seguida se tornou a mais bem sucedida e consiste na utilização de DNNs supervisionadas para classificação de locutores, como uma alternativa à modelagem com *i-vectors*. Variani *et al.* propuseram o uso de uma DNN para aprender novas representações para verificação de locutores dependente de texto (VARIANI et al., 2014). Nesse trabalho, o modelo consiste em uma DNN que realiza a classificação dos

locutores utilizando vetores de características espectrais de tempo curto (ver Figura 17). Assim como as DNNs treinadas para ASR, esse vetor é definido pelas energias resultantes pela passagem do banco de filtros, que é um passo intermediário da extração dos MFCCs. A rede é treinada para discriminar os vetores de tempo curto com relação aos locutores. O intuito é realizar um mapeamento entre os vetores e novas representações mais discriminativas. Após a retirada da camada de classificação, a resposta da última camada é utilizada como representação final e, a representação de uma locução é definida pela média das representações produzidas pelos vetores. Em alguns casos, foram observados resultados comparáveis aos dos *i-vectors* utilizando a similaridade do cosseno para realização da verificação. Essa representação ficou conhecida como *d-vector* (*deep vector*).

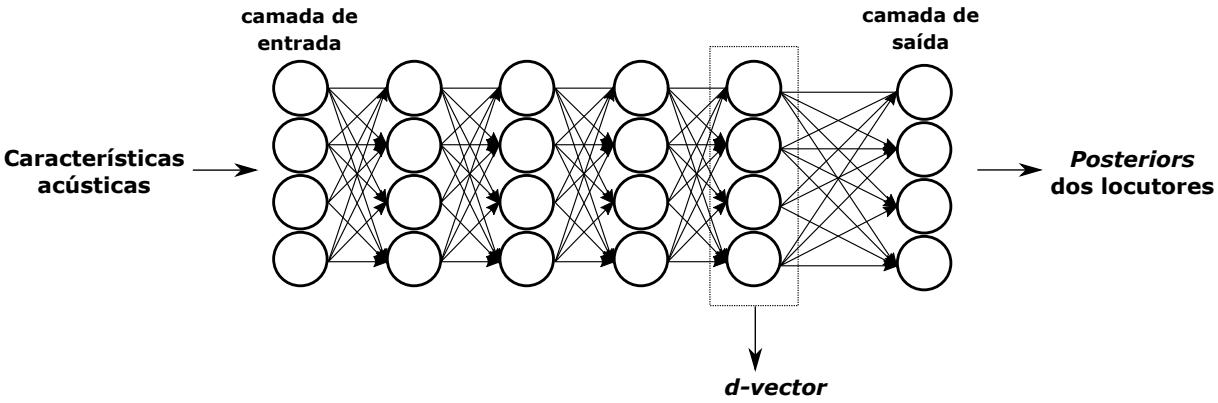

Figura 17 – Modelo DNN utilizado para aprender uma nova representação (*d-vector*) para o vetor de entrada utilizando as classes dos locutores. Após a retirada da camada de classificação, as saídas da última camada compõem o vetor extraído.

2.5.3 Sistemas fim-a-fim

Com a percepção de que DNNs são capazes de aprender representações discriminativas utilizando dados categorizados pelas informações dos locutores, algumas propostas surgiram com o intuito de produzir um único modelo profundo capaz de realizar a tarefa de verificação do início ao fim. O objetivo dessa abordagem é treinar um modelo que recebe como entrada um conjunto de locuções de cadastro de um determinado locutor e uma locução de teste e que produza como saída um *score* associado à hipótese de que a locução de teste foi gerada pelo locutor que produziu as locuções de cadastro. Uma visão geral desse modelo é mostrada na Figura 18. Em 2016, um sistema fim-a-fim foi proposto por Heigold *et al.* para verificação de locutores dependente de texto (HEIGOLD et al., 2016) e logo em seguida adaptado por Snyder *et al.* para verificação independente de texto (SNYDER et al., 2016).

Uma parte da rede é composta por uma DNN que é responsável pela geração de uma representação para um conjunto de locuções. As representações geradas através das locuções de cadastro e de teste são comparadas por outra parte da rede, que computa

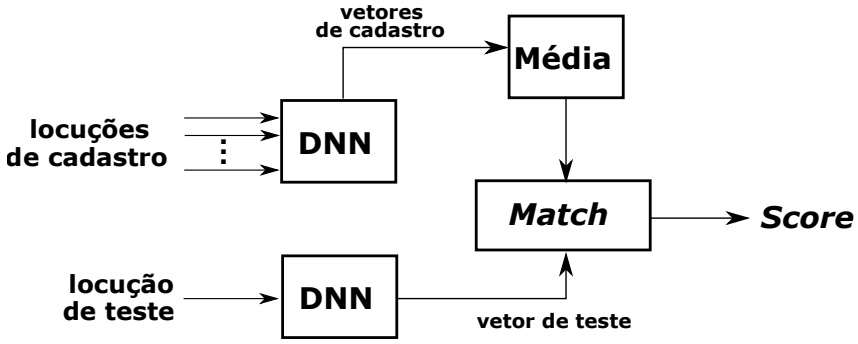

Figura 18 – Visão geral de um sistema fim-a-fim onde DNNs são utilizadas para extrair representações das locuções de cadastro e teste, que são comparadas pela função de *match*.

a similaridade entre o vetor médio de cadastro com o vetor de teste. Durante a fase de treinamento, a rede aprende a maximizar o *score* quando as locuções foram geradas pelo mesmo locutor e a minimizá-lo para o caso contrário.

A Figura 19 apresenta a DNN utilizada pelos autores (SNYDER et al., 2016). As locuções de entrada são definidas por vetores de características de tempo curto e, por essa razão, descritas em um espaço de dimensão variável. Para uma locução com T vetores de características de dimensão D, o objetivo da DNN é gerar uma representação de dimensão fixa, digamos $\boldsymbol{x} \in \mathbb{R}^E$, independentemente da quantidade de vetores T. Para esse propósito, foi introduzida uma camada escondida de *pooling* temporal, que computa estatísticas temporais (média e desvio-padrão) das respostas temporais da camada anterior e as concatena para produção do vetor de saída, que possui agora uma dimensão fixa. Essa representação intermediária é processada pelo restante da rede, e por último por uma camada linear, que produz o vetor de representação para cada locução. Para as camadas intermediárias não lineares, os autores utilizaram uma camada chamada de Rede-em-Rede (*Network-in-Network* - NIN)[15] (LIN; CHEN; YAN, 2013).

As representações geradas para as locuções de entrada são comparadas em outra parte da rede que computa a similaridade entre a representação média de cadastro e a representação gerada para a locução de teste. Dados o vetor médio de cadastro, $\overline{\boldsymbol{x}}_{\boldsymbol{c}}$, e o vetor de teste, $\boldsymbol{x_t}$, a similaridade entre eles é definida por

$$S(\overline{\boldsymbol{x}}_{\boldsymbol{c}}, \boldsymbol{x_t}) = \frac{1}{1 + \exp[-L(\overline{\boldsymbol{x}}_{\boldsymbol{c}}, \boldsymbol{x_t})]}, \tag{2.79}$$

onde $L(\boldsymbol{x}, \boldsymbol{y})$ é uma função que realiza um *match* entre os vetores $\boldsymbol{x}$ e $\boldsymbol{y}$ baseada no cálculo do *score* do modelo G-PLDA (Equação 2.76)[16]:

$$L(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^t\boldsymbol{y} - \boldsymbol{x}^t\boldsymbol{S}\boldsymbol{x} - \boldsymbol{y}^t\boldsymbol{S}\boldsymbol{y} + b, \tag{2.80}$$

[15] Uma NIN realiza o mapeamento de um vetor de entrada de dimensão D_i em um vetor de dimensão D_o, gerando internamente uma representação de dimensão D_h, através de diferentes redes neurais menores que compartilham pesos entre si.

[16] Em termos mais precisos, ela é baseada numa simplificação da função de *score* do G-PLDA, descrita em (BURGET et al., 2011).

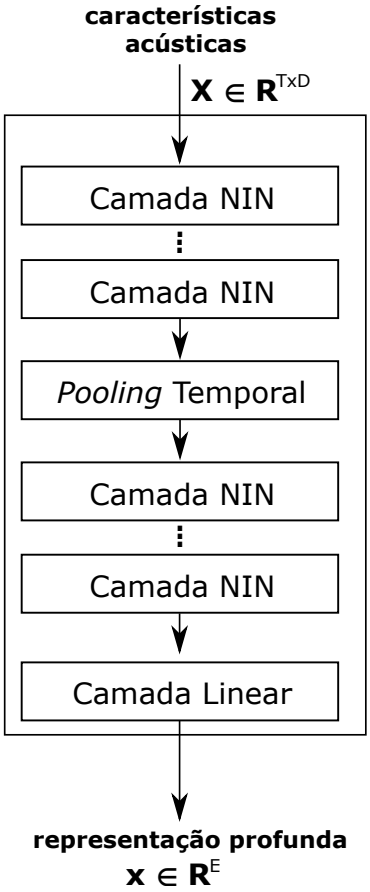

Figura 19 – Modelo DNN utilizado para o mapeamento entre as características acústicas de uma determinada locução, descrita através de um conjunto de vetores de características de tempo curto $X \in \mathbb{R}^{T \times D}$. A rede produz uma representação de dimensão fixa $\boldsymbol{x} \in \mathbb{R}^{E}$, independentemente da quantidade de vetores T presente na locução.

onde $\boldsymbol{S}$ e b são uma matriz quadrada de dimensões iguais às dos vetores e uma constante de *bias*, respectivamente, que são aprendidas durante o treinamento do modelo. A função de custo utilizada no treinamento da rede é baseada na entropia cruzada binária:

$$\mathcal{L} = -\sum_{\overline{\boldsymbol{x}}_{\boldsymbol{c}}, \boldsymbol{x}_{\boldsymbol{t}} \in P_{pos}} \ln S(\overline{\boldsymbol{x}}_{\boldsymbol{c}}, \boldsymbol{x}_{\boldsymbol{t}}) - K \sum_{\overline{\boldsymbol{x}}_{\boldsymbol{c}}, \boldsymbol{x}_{\boldsymbol{t}} \notin P_{neg}} \ln \left[1 - S(\overline{\boldsymbol{x}}_{\boldsymbol{c}}, \boldsymbol{x}_{\boldsymbol{t}})\right], \tag{2.81}$$

onde P_{pos} e P_{neg} são os conjuntos de treinamento onde as locuções foram produzidas pelo mesmo locutor e por locutores distintos, respectivamente. A constante K é utilizada para ponderar os dois termos de maneira que os conjuntos de exemplos positivos e negativos tenham o mesmo peso (fator de importância) na função de custo. Isso é necessário porque a quantidade de exemplos negativos é bem maior que a de exemplos positivos. Após o treinamento, o modelo é diretamente utilizado para comparar as locuções de cadastro e de teste, realizando assim a decisão do sistema.

Essa abordagem fim-a-fim, combinada com o *pooling* temporal, resultou em um avanço em termos de desempenho, apresentando em alguns casos resultados melhores que aqueles alcançados pelos *i-vectors*. Porém, para essa abordagem, existe a dificuldade proveniente

do fato de o modelo ser treinado utilizando pares de conjuntos de locuções de cadastro e de teste. Para enfrentar as incompatibilidades resultantes da diferença de contexto de treinamento e teste do sistema, uma grande quantidade de dados é necessária durante o treinamento. Geralmente, algum esquema para seleção dos melhores exemplos negativos é empregado entre diferentes épocas durante o treinamento (ZHANG; KOISHIDA, 2017).

2.5.4 Modelagem via *x-vectors*

Devido à complexidade envolvida no desenvolvimento de um único modelo fim-a-fim para verificação de locutores, Snyder *et al.* propuseram um retorno à abordagem anterior, separando o modelagem em duas tarefas distintas: (i) geração de uma representações robustas e discriminativas para as locuções e (ii) realizar a tarefa de verificação através da decisão se duas representações distintas foram produzidas ou não pelo mesmo locutor. Para realização da primeira tarefa, uma DNN supervisionada, treinada para discriminar locuções de diferentes locutores foi proposta, enquanto que para a segunda tarefa, os autores realizaram a decisão através da modelagem G-PLDA (SNYDER et al., 2017). Essa abordagem se aproveita do poder das DNNs para geração de representações discriminativas e também utiliza o poderoso modelo baseado em análise fatorial que foi um dos responsáveis pelo sucesso alcançado pela modelagem com *i-vectors*.

A DNN proposta para a geração das representações é semelhante àquela utilizada no sistema fim-a-fim proposto anteriormente em (SNYDER et al., 2016) (Figura 19). Porém, ao invés de a DNN produzir uma representação que é processada pelo restante da rede fim-a-fim, a rede proposta é treinada para discriminar as locuções com respeito aos rótulos dos locutores. Assim como em (VARIANI et al., 2014) (Figura 17), a DNN é um modelo supervisionado e a representação final é extraída de uma das camadas escondidas, ao retirar a camada de classificação. A Figura 20 apresenta a arquitetura da DNN proposta pelos autores. A entrada do modelo são locuções, descritas através de vetores MFCCs, e a saída corresponde às probabilidades *a posteriori* associadas às classes dos locutores.

A primeira parte da rede, composta por camadas em nível de janelas (*frame-level layers*), mapeia os vetores para novas representações, mantendo a dimensionalidade temporal. Digamos que uma determinada locução é descrita por T vetores MFCC com D componentes. A entrada, de dimensionalidade $\mathbb{R}^{T\times D}$, é mapeada pelas camadas a nível de janela para uma nova representação de dimensão, por exemplo, $\mathbb{R}^{T\times K}$. Essa parte do modelo é definida por uma rede com atraso temporal (*Time-delayed Neural Network* - TDNN) (PEDDINTI; POVEY; KHUDANPUR, 2015), que é composta por camadas que processam os dados levando em consideração um contexto temporal específico. Um determinado nó da camada possui um vetor de pesos associado a cada índice temporal presente no contexto da camada. Por exemplo, suponha que, para a i-ésima camada da TDNN opere no contexto temporal definido pelo conjunto $\{t-2,\ t,\ t+1\}$ e que a saída da camada anterior é composta pelo vetor $\boldsymbol{x}_{i-1} \in \mathbb{R}^{T\times D}$. O vetor $\boldsymbol{o}_i \in \mathbb{R}^T$ produzido por um nó da i-ésima

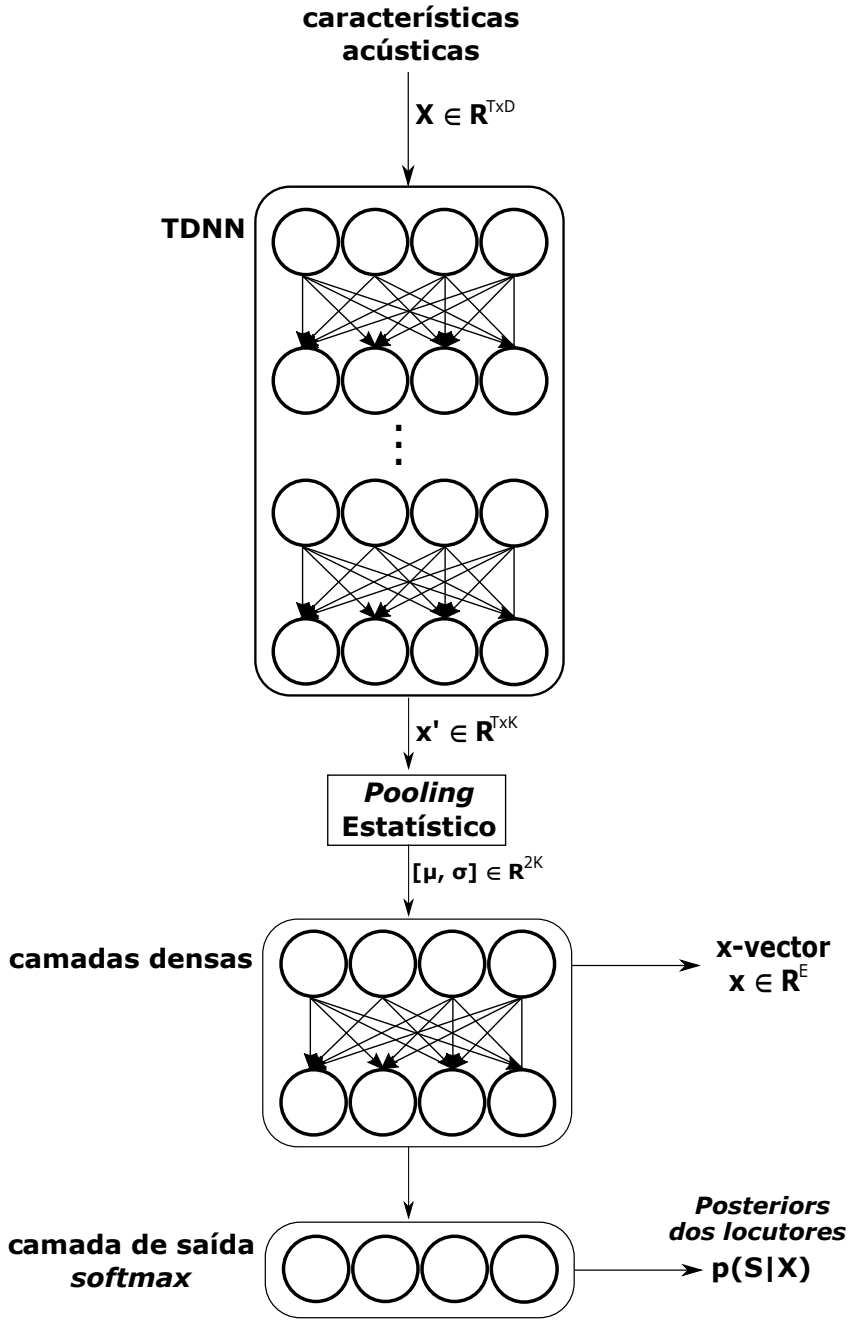

Figura 20 – Modelo DNN utilizado para o mapeamento entre as características acústicas de uma determinada locução, descrita através de um conjunto de vetores de características de tempo curto $X \in \mathbb{R}^{T \times D}$. A rede produz uma representação de dimensão fixa $\boldsymbol{x} \in \mathbb{R}^{E}$, independentemente da quantidade de vetores T presente na locução.

camada é definido por:

$$\begin{gathered} \boldsymbol{o_i}[t] = \boldsymbol{\omega}_{t-2}\boldsymbol{x}[t-2] \ + \ \boldsymbol{\omega}_{t}\boldsymbol{x}[t] \ + \ \boldsymbol{\omega}_{t+1}\boldsymbol{x}[t+1] \ + \ b, \\ t = 1, 2, ..., T, \end{gathered} \tag{2.82}$$

onde $\boldsymbol{\omega}_p$, $p \in \{t-2, \ t, \ t+2\}$, são os vetores de pesos de dimensionalidade $\mathbb{R}^D$, associados a cada um dos índices temporais presentes no contexto da camada e b é uma constante

bias. Dessa maneira, se a camada for composta por K nós, a saída da camada possui dimensionalidade $\mathbb{R}^{T\times K}$. A TDNN proposta pelos autores é formada por cinco camadas (SNYDER et al., 2016), cujas quantidades de nós e contextos temporais estão descritos na Tabela 1.

Tabela 1 – Quantidade de nós e contexto temporal de cada uma das cinco camadas presentes na TDNN da modelagem com *x-vectors*.

Camada	**Quantidade de Nós**	**Contexto Temporal**
1	512	[t-2, t+2]
2	512	{t-2, t, t+2}
3	512	{t-3, t, t+3}
4	512	{t}
5	1536	{t}

A saída da primeira parte do modelo é processada por uma camada de *pooling* estatístico[17], que computa os vetores de média e desvio-padrão a partir das representações temporais. Dessa maneira, se a saída da primeira parte do modelo consiste de uma representação de dimensionalidade $\mathbb{R}^{T\times K}$, essa camada computa os vetores das estatísticas e os concatena, gerando uma representação de dimensionalidade $\mathbb{R}^{2K}$. A partir desse momento, as representações para diferentes locuções, cujas dimensionalidades de entrada possuíam valores T dependentes da duração da locução, agora possuem dimensionalidade fixa. Tais representações são então processadas pela segunda parte da rede, composta por duas camadas densas com 512 nós cada. Por fim, as saídas da segunda parte da rede são processadas pela camada de classificação, contendo um nó para cada rótulo de locutor de treinamento e cuja função de ativação é uma função *softmax*, definindo, assim, as probabilidades *a porteriori* associadas a cada um dos locutores de treinamento.

Todas as camadas tanto da primeira quanto da segunda parte do modelo são compostas por unidades retificadoras lineares (*Rectified Linear Units* - ReLUs). A rede é treinada através do algoritmo de gradiente descendente estocástico (POVEY; ZHANG; KHUDANPUR, 2014), utilizando a entropia cruzada multi-classe como função de custo. Para um modelo treinado utilizando N locuções produzidas por L locutores, a função de custo é definida por

$$\mathcal{L} = -\sum_{n=1}^{N}\sum_{l=1}^{L} d_{nl}\ln[P(S_l|X_n)], \tag{2.83}$$

onde d_{nl} é um se a locução X_n foi produzida pelo locutor S_l e zero, caso contrário.

Uma vez treinado, o modelo é utilizado para gerar uma representação para uma determinada locução de entrada. Esse vetor é extraído da parte linear da primeira camada após o *pooling* estatístico (antes da aplicação da função de ativação). Para a tarefa de verificação, a modelagem G-PLDA é empregada da mesma maneira como é empregada para a

[17] Ela é idêntica à camada de *pooling* temporal utilizada nos modelos anteriores (HEIGOLD et al., 2016; SNYDER et al., 2016), mas a partir desse trabalho ela vêm sendo referenciada como *pooling* estatístico.

modelagem com *i-vectors*. Em (SNYDER et al., 2018), os autores nomearam a representação como *x-vector* e mostraram sua robustez em relação aos *i-vectors*, alcançando um desempenho consideravelmente superior. Foram apresentados, também, resultados envolvendo *data augmentation*, onde ruídos de diversas fontes (KO et al., 2017; SNYDER; CHEN; POVEY, 2015) foram adicionados aos segmentos de voz para geração de novas amostras. Os *x-vectors* apresentarem maiores ganhos de desempenho que os *i-vectors*, tomando maior proveito das novas amostras. Isso foi observado aplicando *data augmentation* apenas para a estimação do modelo G-PLDA, apenas para o treinamento da rede e para ambas as etapas conjuntamente.

Mesmo apresentando maior robustez e poder discriminativo que os *i-vectors*, métodos de pós-processamento, tais como LDA e normalização de comprimento, ainda são empregados aos *x-vectors* antes da modelagem G-PLDA. Porém, se no caso dos *i-vectors* esses métodos possuem o objetivo de aumentar a capacidade discriminativa das representações, para os *x-vectors* eles desempenham outro papel. Como demonstrado recentemente por Zhang *et al.*, o pós-processamento dos *x-vectors* pode ser visto como uma etapa de regularização do espaço das representações (ZHANG; LI; WANG, 2019). Como descrito anteriormente (Seção 2.4.5), o modelo G-PLDA decompõe o espaço dos *x-vectors* em duas componentes que seguem distribuições normais. Relembrando a Equação 2.70, na modelagem G-PLDA, um determinado *x-vector*, $\boldsymbol{x}$, é assumido seguir uma distribuição cuja decomposição em fatores é descrita por

$$\boldsymbol{x} = \boldsymbol{x_m} + \boldsymbol{\Phi\beta} + \boldsymbol{\epsilon_r}, \tag{2.84}$$

onde o fator correspondente ao locutor $\boldsymbol{\beta}$ segue uma distribuição normal padronizada e o termo residual $\boldsymbol{\epsilon_r}$ segue uma distribuição normal com média zero e matriz de covariância $\boldsymbol{\Sigma}$. Como os *x-vectors* são treinados apenas com o objetivo de distinguir locuções de diferentes locutores, nenhum controle é imposto sobre as distribuições dos vetores. Os métodos de pós-processamento então mapeiam os vetores para um espaço mais adequado ao G-PLDA, o que resulta em um ganho de desempenho. Mesmo com esse ganho de desempenho, tais métodos não são completamente adequados, uma vez que eles não impõem restrições às distribuições geradas.

Com o objetivo de fazer com que as distribuição dos *x-vectors* se aproximem de distribuições normais, (LI et al., 2019) propuseram uma mudança na camada de classificação e a adição de um termo de regularização à função de custo da DNN (Equação 2.83) definido pela norma L2 entre os *x-vectors* e o peso da camada de saída associado ao locutor correspondente, que por sua vez é fixado através do *x-vector* médio do locutor. Mais uma vez suponha um modelo treinado utilizando N locuções produzidas por L locutores. Além disso, suponha que X_l é o conjunto de locuções produzidas pelo locutor $S_l, l = 1, ..., L$ e que $\mathfrak{X}_l$ é o conjunto de *x-vectors* extraídos dessas locuções. Os autores propuseram

parametrizar a camada de classificação através dos *x-vector* médio de cada locutor:

$$\overline{\boldsymbol{x}}_{\boldsymbol{l}} = \frac{1}{|\mathfrak{X}_l|} \sum_{\boldsymbol{x}_l \in \mathfrak{X}_l} \boldsymbol{x}_{\boldsymbol{l}}, \quad l = 1, ..., L, \tag{2.85}$$

calculando as probabilidades *a posteriori* de cada locutor através do produto interno entre o *x-vector* gerado por uma determinada locução X, $\boldsymbol{x}$, e o *x-vector* médio associado ao locutor:

$$p(S_l|X) = \frac{\exp[\boldsymbol{x} \cdot \overline{\boldsymbol{x}}_{\boldsymbol{l}}]}{\sum_{j=1}^{L} \exp[\boldsymbol{x} \cdot \overline{\boldsymbol{x}}_{\boldsymbol{j}}]}. \tag{2.86}$$

Como a camada de classificação é parametrizada pelo *x-vector* médio, para treinar a DNN é necessário recorrer a um tipo de treinamento conhecido como treino-e-reposição (*train-and-replacement*) (LI et al., 2018), onde, a cada época, os *x-vectors* médios dos locutores são computados para toda a base e atualizados no modelo para a atualização dos pesos na época seguinte. Porém, tal abordagem aumenta o custo do treinamento, tornando o processo mais lento. Os autores então utilizaram uma segunda alternativa que consiste em manter os parâmetros da camada de classificação como pesos a serem aprendidos pela rede (como uma rede convencional) e adicionar um termo de regularização à função de custo, que é responsável pela minimização da norma L2 entre as os *x-vectors* de um determinado locutor e o parâmetro associado ao locutor correspondente. Basicamente, os autores substituíram os *x-vectors* médios, $\overline{\boldsymbol{x}}_{\boldsymbol{l}}$, da Equação 2.86 por parâmetros a serem aprendidos, $\boldsymbol{\theta}_{\boldsymbol{l}}$, e definiram o termo de regularização como

$$R = -\sum_{l=1}^{L} \sum_{\boldsymbol{x}_l \in \mathfrak{X}_l} ||\boldsymbol{x}_l - \boldsymbol{\theta}_{\boldsymbol{l}}||. \tag{2.87}$$

Adicionando tal termo de regularização à função de custo, temos:

$$\mathcal{L}' = \mathcal{L} + \alpha R, \tag{2.88}$$

onde $\mathcal{L}$ é a função de custo definida pela entropia cruzada multi-classe, normalmente utilizada (Equação 2.83), e α é peso que controla o grau de importância da regularização no treinamento da rede.

Os autores argumentam que, utilizando um fator de importância suficientemente grande à regularização, $\boldsymbol{\theta}_{\boldsymbol{l}}$ converge para o *x-vector* médio associado ao locutor e a distribuição dos *x-vectors* converge para $N(\boldsymbol{\theta}_{\boldsymbol{l}}, \mathbf{I})$. De fato, a abordagem se mostrou eficaz para ambas as representações *d-vectors* e *x-vectors*. Porém, diferentes fatores de importância foram atribuídos para a função de regularização e todos os resultados apresentados foram alcançados ao empregar LDA às novas representações, o que dificulta a análise do efeito da regularização nas distribuições das representações.

Já em (ZHANG; LI; WANG, 2019), um segundo modelo DNN foi proposto para projetar *x-vectors* já treinados em um novo espaço mais compacto e com distribuição controlada. O modelo consiste em um Auto-codificador Variacional (*Variational Autoencoder* - VAE)

(KINGMA; WELLING, 2013), que, assim como os auto-codificadores (*autoencoders*), é um modelo não-supervisionado (ou auto-supervisionado) treinado para reconstruir a própria entrada, gerando para isso uma representação intermediária, geralmente mais compacta que a original. Um VAE é um DNN que consiste de dois módulos, o codificador (*encoder*) e o decodificador (*decoder*) e possui o objetivo de aprender um mapeamento entre o espaço original complexo (no caso, o espaço dos *x-vectors*) de distribuição $p(\boldsymbol{x})$ para um espaço latente, geralmente mais compacto, que segue uma distribuição paramétrica. No caso mais comum, as variáveis latentes seguem a distribuição normal padronizada: $p(\boldsymbol{z}) = N(\boldsymbol{z}; \boldsymbol{0}, \boldsymbol{I})$. O VAE realiza essa tarefa seguindo uma abordagem variacional, aproximando a distribuição *a posteriori* $p(\boldsymbol{z}|\boldsymbol{x})$ por $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, definida pelos parâmetros do *encoder*, ϕ. O *encoder* é responsável pela geração dos parâmetros de $q(\boldsymbol{z}|\boldsymbol{x})$, que é assumido ser normalmente distribuído com matriz de covariância diagonal

$$q_\phi(\boldsymbol{z}|\boldsymbol{x}_i) = N(\boldsymbol{z}; \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2 \boldsymbol{I}), \tag{2.89}$$

onde $\boldsymbol{x}_i$ é uma amostra do espaço original dos *x-vectors*.

A abordagem VAE resolve a intratabilidade da distribuição *a priori* $p(\boldsymbol{x})$ através da maximização de um limitante inferior do logaritmo da verossimilhança das observações de $\boldsymbol{x}$ (*Evidence Lower Bound* - ELBO), utilizando a distribuição *a posteriori* $p_\theta(\boldsymbol{x}|\boldsymbol{z})$, definida pelos parâmetros do *decoder*, θ. Dada uma amostra $\boldsymbol{x}_i$, a função objetivo, $\mathcal{L}(\boldsymbol{x}_i, \phi, \theta) \leq \log p_\theta(\boldsymbol{x})$, é definida como

$$\mathcal{L}(\boldsymbol{x}_i, \phi, \theta) = -D_{KL}\Big[q_\phi(\boldsymbol{z}|\boldsymbol{x_i})||p(\boldsymbol{z})\Big] + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x_i})}\Big[\log[p_\theta(\boldsymbol{x_i}|\boldsymbol{z})]\Big], \tag{2.90}$$

onde a primeira parte é definida pela divergência KL entre $q_\phi(\boldsymbol{z}|\boldsymbol{x_i})$ e a distribuição *a priori* desejada $p(\boldsymbol{z})$, e a segunda parte corresponde à esperança associada ao mapeamento das variáveis latentes para o espaço original.

A Figura 21 apresenta a arquitetura genérica de um VAE. Para uma determinada amostra $\boldsymbol{x_i}$, o *encoder* produz os vetores de média e desvio-padrão, $\boldsymbol{\mu_i}, \boldsymbol{\sigma_i}$, e o vetor intermediário é definido por uma amostra aleatória da distribuição gerada, $\boldsymbol{z} \sim N(\boldsymbol{\mu_i}, \boldsymbol{\sigma}_i^2 I)$. Essa amostragem é realizada através do "truque de re-parametrização", onde uma amostra aleatória da distribuição normalizada é deslocada e escalada levando em consideração os parâmetros da distribuição gerada:

$$\boldsymbol{z_i} = \boldsymbol{\mu_i} + \boldsymbol{\epsilon} \odot \boldsymbol{\sigma_i}, \tag{2.91}$$

onde $\boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \boldsymbol{I})$ e $\odot$ é a operação de multiplicação elemento-a-elemento. Nesse cenário, e levando em consideração que a distribuição *a priori* desejada é a distribuição normal padronizada, a divergência KL presente na função de custo pode ser calculada da seguinte forma:

$$D_{KL}\Big[q_\phi(\boldsymbol{z}|\boldsymbol{x_i})||p(\boldsymbol{z})\Big] = \frac{1}{2}\sum_{j=1}^{K} \mu_{ij}^2 + \sigma_{ij}^2 - \log \sigma_{ij}^2 - 1, \tag{2.92}$$

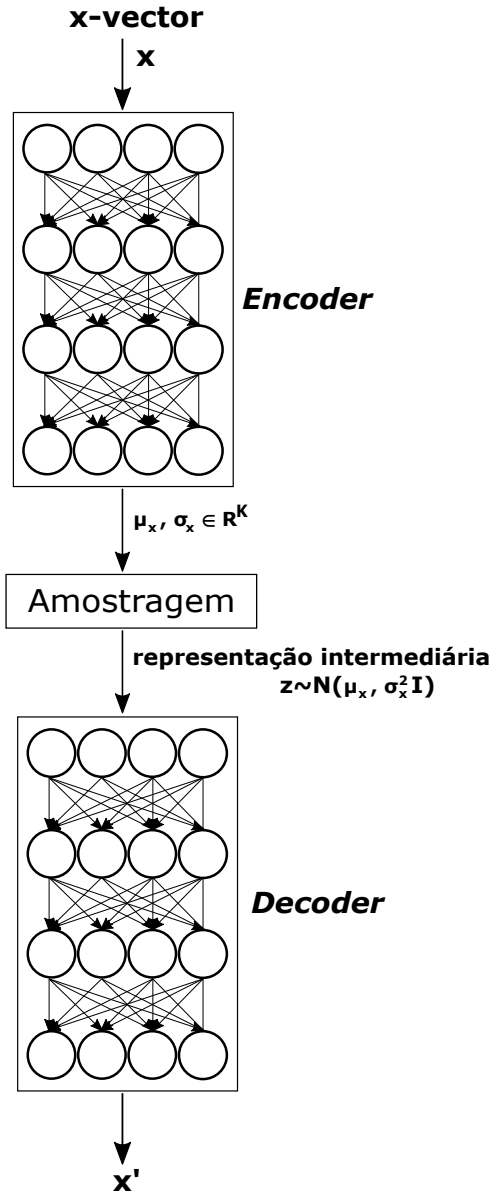

Figura 21 – Arquitetura geral de um Auto-codificador Variacional (VAE) aplicado para mapear os *x-vectors* para uma nova representação produzida através de um esquema de amostragem, sendo treinado para reconstruir a entrada a partir da representação gerada.

onde K é a dimensionalidade da variável latente ($\boldsymbol{\mu_i}, \boldsymbol{\sigma_i} \in \mathbb{R}^K$).

Zhang *et al.* propuseram essa abordagem como uma alternativa aos métodos de pós-processamentos das representações, uma vez que tanto regulariza o espaço dos *x-vectors* como também realiza uma redução de dimensionalidade. Os resultados apresentaram um ganho de desempenho em relação aos demais métodos, porém, não ao ponto de justificar o treinamento de um segundo modelo apenas para esse propósito. Os autores então apresentaram mais tarde uma utilidade mais apropriada para esse segundo modelo: o de adaptação de domínio (WANG; LI; WANG, 2019). Nesse cenário, a abordagem é utilizada para projetar *x-vectors* extraídos de um modelo treinado em um determinado domínio

(utilizando locuções telefônicas, por exemplo) em um espaço mais adequado para o domínio onde a tarefa ocorrerá (cadastramento e teste utilizando locuções gravadas com microfones, por exemplo).

2.6 NORMALIZAÇÃO DOS *SCORES* DO SISTEMA

Em verificação de locutores, o sistema computa um *score* correspondente à hipótese de que uma determinada locução foi produzida por um determinado locutor. Na modelagem GMM-UBM, por exemplo, esse *score* é definido pelo logaritmo da razão das verossimilhanças do modelo do locutor e do UBM (Equação 2.43). Por outro lado, na modelagem GMM-SVM, o *score* corresponde ao resultado do cálculo da função de classificação do SVM (Equação B.9). Com a modelagem PLDA, com *i-vectors* ou *x-vectors*, por exemplo, o *score* é definido pela Equação 2.76. Para a realização da tarefa, o *score* é comparado a um limiar de aceitação, que, para a autenticação da locução, deve ser inferior ao *score* calculado.

Como discutido anteriormente na Seção 1.3.1, a escolha do limiar de aceitação determina o ponto de operação do sistema. Para o desenvolvimento prático de sistemas de verificação de locutores, a escolha do ponto de operação é uma tarefa essencial, uma vez que ele determina as taxas de falsa aceitação (FAR) e falsa rejeição (FRR) esperadas do sistema. Além disso, é desejável que tanto o ponto de operação quanto o limiar de aceitação sejam independentes de locutor. Isto é, deseja-se utilizar um único limiar de aceitação para qualquer locutor cadastrado no sistema. Ao contrário, seria necessária a estimação de diferentes limiares de aceitação (um para cada locutor) de modo que o ponto de operação desejado fosse alcançado para qualquer processo de verificação.

Para a escolha do limiar de aceitação de um determinado locutor, S, por exemplo, utilizam-se as distribuições dos *scores* produzidos por locuções produzidas por S e por locuções produzidas por outros locutores, chamados de impostores. Obviamente, tais locuções não podem ter sido utilizadas para a estimação dos modelos do sistema, senão tais distribuições seriam enviesadas. Figura 22 ilustra um exemplo dessas distribuições. Para cada ponto de operação, as taxas de FAR e FRR são calculadas levando em consideração um determinado limiar de aceitação. O ponto de operação pode então ser definido a partir das taxas de erro satisfatórias.

A Figura 22 mostra possíveis distribuições de *scores* que podem ser encontradas no desenvolvimento de um sistema de verificação para um determinado locutor. Porém, espera-se que diferentes distribuições de *scores* sejam encontradas para diferentes locutores. Tal fato dificulta a escolha de um único ponto de operação independente de locutor. Outra maneira de analisar essa situação consiste em observar que um determinado sistema gera diferentes curvas ROC (Figura 2) para diferentes locutores.

Nesse contexto, o principal objetivo das técnicas de compensação de *scores* é mapear

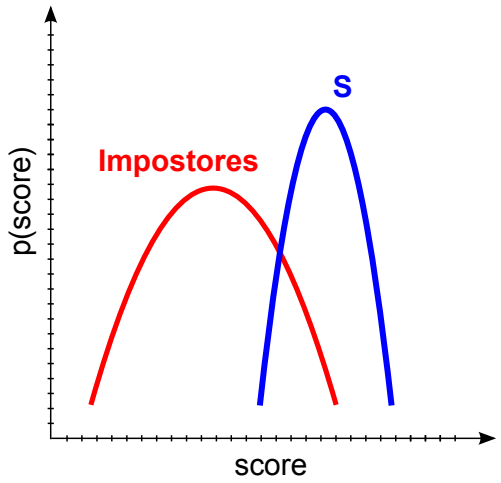

Figura 22 – Ilustrações de distribuições de *scores* utilizando locuções de um determinado locutor, S e locuções dos impostores.

os *scores* produzidos por diferentes locutores para intervalos semelhantes, de modo que seja possível encontrar um único limiar de aceitação satisfatório para todos os locutores cadastrados (AUCKENTHALER; CAREY; LLOYD-THOMAS, 2000). A mais bem sucedida abordagem para realizar tal mapeamento consiste em normalizar os *scores*. Por essa razão, técnicas de compensação de *scores* são comumente referenciadas como técnicas de normalização de *scores*. Tal normalização deve ser eficaz na eliminação de possíveis desvios encontrados nas distribuições produzidas por um determinado locutor.

Dado um determinado *score*, s, produzido por um sistema, a ideia básica desse tipo de compensação é realizar uma normalização da forma:

$$\hat{s} = \frac{s - \mu_{norm}}{\sigma_{norm}}, \tag{2.93}$$

onde $\hat{s}$ é o *score* normalizado. Os parâmetros μ_{norm} e σ_{norm} são a média e o desvio-padrão de uma determinada distribuição de *scores*. Para geração do conjunto de *scores* dos quais as estatísticas serão calculadas, geralmente consideram-se *scores* gerados por impostores. Nesse contexto, dependendo do método, esses *scores* podem ser produzidos utilizando tanto locuções de impostores quanto modelos de impostores. Esses conjuntos de modelos de locutores são comumente referenciados como *cohorts* (Seção 2.4.2).

As próximas seções descrevem os principais métodos de normalização utilizados em sistemas de verificação de locutores.

2.6.1 Normalização zero

A normalização zero (Z-norm) (REYNOLDS, 1997) foi uma das primeiras a serem amplamente utilizadas e possui a vantagem de computar os parâmetros μ_{norm} e σ_{norm} na fase de cadastramento, de modo que há pouco impacto no tempo de autenticação de uma

determinada locução. O conjunto de *scores* levado em consideração para calcular tais estatísticas é formado pelos *scores* produzidos pela tentativa de autenticação de impostores. Basicamente, produzem-se os *scores* utilizando locuções de vários locutores diferentes do locutor cadastrado. Geralmente, o mesmo conjunto de locuções é utilizado para todos os locutores cadastrados. Isso é feito produzindo um conjunto de locuções que provém de locutores que não estão cadastrados no sistema. Na prática, às vezes isso é inviável e, para um determinado locutor cadastrado S, utilizam-se locuções de outros locutores cadastrados, $S' \neq S$, para compor tal conjunto.

2.6.2 Normalização de teste

A normalização de teste (T-norm) (AUCKENTHALER; CAREY; LLOYD-THOMAS, 2000) é bem similar à Z-norm e também utiliza um conjunto de *scores* de impostores para o cálculo das estatísticas de média e desvio-padrão. Porém, esse processo é realizado no momento da verificação e considera os *scores* produzidos pela locução de teste utilizando modelos de outros locutores, chamados de modelos dos impostores. Suponha a verificação de uma determinada locução, X, com respeito a um determinado locutor S. O método então calcula o *score* produzido pelo modelo do locutor S e o normaliza levando em consideração os *scores* produzidos pelos modelos de outros locutores $S' \neq S$.

Apesar de apresentar maior custo computacional na verificação, a normalização realizada é mais eficaz quando há incompatibilidades nas locuções utilizadas para gerar os modelos dos locutores. Nesse caso, tais distorções impactam mais o método Z-norm, uma vez que o *score* produzido no teste será abaixo do esperado na distribuição estimada na fase de treinamento. Já no método T-norm, o *score* produzido no teste será abaixo do esperado em todos os modelos utilizados na normalização, mas espera-se que, ainda assim, o *score* produzido no modelo do locutor se sobressaia diante dos *scores* produzidos nos modelos dos impostores.

2.6.3 Normalização simétrica

Em (VOGT; BAKER; SRIDHARAN, 2005) foi proposta a combinação entre os métodos Z-norm e T-norm, onde eles eram executados em série. Na modelagem JFA, esse tipo de normalização foi bastante importante, mostrando ganhos de desempenho consideráveis (KENNY et al., 2008), passando a ser considerada uma parte essencial no desenvolvimento dos sistemas. Esse método é conhecido como ZT-norm. Em (KENNY, 2010), Kenny observou que a não-simetria do método o tornava inadequado para modelagens como o G-PLDA e HT-PLDA, onde o *score* compara duas representações e as hipóteses envolvidas no teste estatístico claramente impõem uma relação simétrica na comparação das representações. Ele então propôs uma mudança simples para manter a simetria dos conjuntos de *scores*,

onde o *score* final é definido por:

$$\hat{s} = \frac{1}{2}\left(\frac{s - \mu_{znorm}}{\sigma_{znorm}} + \frac{s - \mu_{tnorm}}{\sigma_{tnorm}}\right). \tag{2.94}$$

Esse método é referenciado como normalização simétrica (S-norm).

2.6.4 Normalização simétrica adaptativa

Em (STURIM; REYNOLDS, 2005), os autores propuseram uma seleção adaptativa dos modelos de impostores utilizados para compor o conjunto de *scores* dos impostores. Nesse método, o intuito é selecionar os modelos de impostores mais semelhantes ao cadastrado. Os autores propuseram essa abordagem e aplicaram para a normalização T-norm, e por essa razão esse método ficou conhecido como T-norm adaptativo. Apesar disso, a mesma abordagem pode ser realizada para qualquer um dos métodos descritos anteriormente, apenas considerando um conjunto de *cohorts* diferente para cada um dos locutores cadastrados. Nessa abordagem, utiliza-se tanto um conjunto de modelos de impostores quanto um conjunto de locuções de impostores. Cada modelo de impostor é caracterizado pelo vetor produzido pela concatenação dos *scores* produzidos pelas locuções, que foram geradas por locutores distintos daqueles dos modelos. A mesma caracterização é realizada para os modelos cadastrados e então os vetores de características dos modelos cadastrados e dos impostores são comparados (por distância *city block*, por exemplo). Para cada modelo cadastrado, selecionam-se os K modelos de impostores mais semelhantes. No momento da verificação, as estatísticas utilizadas nas normalizações são computadas desse conjunto de K modelos de impostores mais semelhantes do modelo cadastrado correspondente à alegação. Nos últimos anos, a abordagem mais utilizada consiste na abordagem adaptativa da normalização simétrica (S-norm adaptativo). Ela é considera atualmente a técnica padrão para normalização de *scores* (NAUTSCH et al., 2014; SNYDER et al., 2017; SNYDER et al., 2018; VILLALBA et al., 2019).

2.7 CONCLUSÕES

Neste capítulo, foram apresentados os principais métodos desenvolvidos para sistemas de verificação de locutores independente de texto. Na Seção 2.1 foi apresentada a definição do problema, em conjunto com a arquitetura genérica para um sistema desse tipo. As principais técnicas de pré-processamento foram apresentadas da Seção 2.2, enquanto que na Seção 2.3 descrevemos as características acústicas comumente extraídas dos segmentos de voz. Em especial, mostramos em detalhes o conjunto de características mais abrangentemente utilizado pelos sistemas, que é composto pelos coeficientes MFCC. Na Seção 2.4 apresentamos os métodos desenvolvidos para a modelagem dos locutores, desde o método GMM-UBM, passando pela abordagem GMM-SVM, até a modelagem através dos chamados *i-vectors*. Além disso, também foi descrita a principal modelagem utilizada

para a realização da tarefa de verificação, o G-PLDA. Já na Seção 2.5 apresentamos os principais modelos desenvolvidos seguindo a abordagem de aprendizagem profunda (DL), até a abordagem considerada atualmente o estado da arte, que consiste das representações conhecidas como *x-vectors*. Demos importante atenção à incompatibilidade existente entre as distribuições geradas pelos *x-vectors* e as premissas impostas pela modelagem G-PLDA, o que resulta na necessidade do pós-processamento das representações para alcançar melhores desempenhos. Além disso, descrevemos algumas técnicas propostas para controlar o espaço composto pelos *x-vectors*. Por fim, descrevemos na Seção 2.6 os métodos utilizados para normalização dos *scores* do sistema, etapa importante que está presente em praticamente todos os sistemas atualmente.

3 MÉTODOS PROPOSTOS

Assim como descrito no Capítulo 2 e mais especificamente na Seção 2.5, o estado-da-arte em verificação de locutores independente de texto consiste na utilização de uma DNN supervisionada para aprender o mapeamento entre locuções e representações robustas e discriminativas chamadas de *x-vectors*. Vimos também que, para a realização da tarefa de verificação, a modelagem G-PLDA é empregada e que o modelo resultante, assim como na modelagem utilizando *i-vectors* (Seção 2.4.5), calcula um *score* associado à hipótese de que dois *x-vectors* foram gerados pelo mesmo locutor.

Apesar de os *x-vectors* terem demonstrado possuir alto poder de discriminação entre diferentes locutores e alta robustez quando comparados aos *i-vectors* (SNYDER et al., 2018), métodos de pós-processamento como o LDA e normalização de comprimento (LN) ainda são empregados antes de serem modelados através do G-PLDA. O ganho de desempenho alcançado pela aplicação de tais métodos acontece por uma razão diferente daquela observada pela aplicação dos mesmos métodos aos *i-vectors*. Enquanto que os métodos possuem o objetivo de aumentar o poder discriminativo dos *i-vectors*, o mesmo não ocorre para os *x-vectors*, uma vez que eles naturalmente já possuem alto poder de discriminação. Zhang *et al.* mostraram recentemente que o papel dos métodos de pós-processamento no espaço dos *x-vectors* é o de regularização (ZHANG; LI; WANG, 2019).

Como descrito na Seção 2.4.5, a modelagem G-PLDA decompõe o espaço dos *x-vectors* em duas componentes que seguem distribuições normais. Relembrando a Equação 2.70, na modelagem G-PLDA, um determinado *x-vector*, $\boldsymbol{x}$, é assumido seguir uma distribuição cuja decomposição em fatores é descrita por:

$$\boldsymbol{x} = \boldsymbol{x_m} + \boldsymbol{\Phi\beta} + \boldsymbol{\epsilon_r}, \tag{3.1}$$

onde $\boldsymbol{x_m}$ é o *x-vector* médio, geralmente extraído da base de treinamento, $\boldsymbol{\beta}$ é a componente correspondente ao locutor e segue uma distribuição normal padronizada, enquanto o termo residual $\boldsymbol{\epsilon_r}$ segue uma distribuição normal com média zero e matriz de covariância $\boldsymbol{\Sigma}$. Ao observar as suposições impostas na decomposição dos supervetores GMM em *i-vectors* (Equação 2.67), pode-se perceber a semelhança entre tais suposições com aquelas realizadas pela modelagem G-PLDA. Tanto os *i-vectors* quanto as componentes de locutor ($\boldsymbol{\beta}$) são assumidos possuírem distribuição normal padronizada, o que resulta em relativa adequação entre o espaço dos *i-vectors* e a modelagem G-PLDA. Porém, o mesmo não ocorre para os *x-vectors*, que são extraídos de uma das camadas de uma DNN treinada exclusivamente para distinguir locuções de diferentes locutores, sem nenhuma imposição direta sobre as distribuições condicionadas aos locutores dos vetores gerados pelas camadas. Nesse cenário, pode-se observar que as técnicas de pós-processamento, LDA e LN, mapeiam os vetores para um espaço mais adequado ao G-PLDA, o que resulta em um

ganho de desempenho. Porém, tais métodos não são completamente adequados, uma vez que eles não foram desenvolvidos para esse propósito, e os ganhos de desempenho podem ser vistos mais como um efeito colateral das normalizações do que de fato como o resultado de uma técnica de compensação adequada para os *x-vectors*.

O objetivo principal deste trabalho consiste no desenvolvimento de técnicas capazes de gerar representações de melhor qualidade, levando em consideração as premissas impostas pela decomposição realizada durante a modelagem G-PLDA. Mais especificamente, os métodos aqui propostos são aplicados à rede DNN supervisionada treinada para aprender o mapeamento entre as locuções e os *x-vectors*, com o objetivo de aumentar a adequação dos mesmos à modelagem G-PLDA através do controle sobre as distribuições dos *x-vectors*, melhorando assim a acurácia do sistema.

3.1 HIPÓTESES

Assim como os outros métodos utilizados para controlar o espaço gerado pelos *x-vectors*, descritos na Seção 2.5.4, as técnicas deste trabalho foram desenvolvidas para controlar as distribuições subjacentes dos *x-vectors*. Em termos mais precisos, elas possuem o objetivo de controlar as distribuições *a priori* das representações e *a posteriori* dos locutores de maneira que, para um determinado *x-vector* $\boldsymbol{x}$ e um locutor S, tanto $p(\boldsymbol{x})$ quanto $p(S|\boldsymbol{x})$ sigam distribuições normais. Além disso, é desejável que as novas representações preservem a boa capacidade de discriminar diferentes locutores, já encontrada no espaço dos *x-vectors*.

Como apresentado na Seção 2.5.4, Li *et al.* propuseram a adição de um termo de regularização para restringir as distribuições condicionais $p(S|\boldsymbol{x})$ (LI et al., 2019), enquanto que Zhang *et al.* treinaram um segundo modelo baseado na abordagem variacional para mapear os *x-vectors* em um novo espaço com distribuição *a priori* normal padronizada (ZHANG; LI; WANG, 2019). Neste trabalho nós propomos abordagens específicas para controlar as probabilidades condicionais e *a priori* dos *x-vectors*, de maneira que elas possam ser integradas em um único modelo DNN.

Para controle das distribuições *a posteriori* dos locutores, $p(S|\boldsymbol{x})$, faz-se necessário que a rede aprenda a codificar as informações seguindo a distribuição normal. A DNN utilizada para geração dos *x-vectors* é composta por camadas cuja operação básica de mapeamento entre uma camada e outra é o produto interno. Dessa maneira, a rede é enviesada a codificar as informações aprendidas em diferentes direções do espaço. Por exemplo, a camada de classificação da rede é composta por um conjunto de nós, onde cada nó está associado a um locutor distinto, e a operação de classificação é definida pelo produto interno entre o vetor de saída da camada anterior e o vetor de pesos associado ao locutor. Uma consequência disso consiste na caracterização dos locutores através das direções dos vetores de peso e na distinção entre diferentes locutores através do ângulo

entre eles. Diante desse cenário, seguimos a abordagem de mudar algumas partes da rede para que ela codifique informações através de distribuições normais. Mais especificamente, alteramos as operações realizadas na camada de classificação e na camada de *pooling* temporal, de maneira que as informações geradas por elas fiquem dispostas sobre um conjunto de distribuições normais. Esses métodos estão descritos na Seção 3.2.

Através da Equação 3.1, pode-se observar que a variabilidade intra-locutor, descrita através do componente $\boldsymbol{\epsilon_r}$, é modelada no G-PLDA através de um espaço que também segue uma distribuição normal. Como estamos lidando com uma rede supervisionada, é natural imaginar que as camadas realizam operações com o objetivo apenas de discriminar diferentes locutores. Porém, na última camada, para a classificação dos locutores, as representações das locuções são comparadas com os parâmetros associados a eles, onde cada locutor é representado por um vetor de pesos apenas. Dessa maneira, podemos concluir que, além de discriminar os locutores entre si, a rede também aprende a minimizar a variabilidade entre as locuções de um mesmo locutor. Por outro lado, não é trivial analisar o espaço gerado por essas variabilidades, uma vez que ela não está diretamente expressa através de nenhuma camada, e o controle sobre a distribuição desse espaço não é uma tarefa fácil.

Assumindo que as distribuições condicionadas aos locutores seguem distribuições normais, uma maneira indireta de controle sobre a distribuição das componentes de variabilidade intra-locutor consiste no controle da distribuição *a priori* dos *x-vectors*. A hipótese aqui pode ser estabelecida de maneira simples utilizando a Equação 3.1: assumindo que $\boldsymbol{\beta}$ segue uma distribuição normal, controlar a distribuição de $\boldsymbol{x}$ implica em um controle indireto sobre a distribuição de $\boldsymbol{\epsilon_r}$. Para esse propósito, este trabalho propõe um termo de regularização baseado na abordagem variacional para controlar a distribuição *a priori* dos *x-vectors* gerados pela rede. Diferentemente de Zhang *et al.* (Seção 2.5.4), ao invés de utilizar um segundo modelo para esse propósito, treinado através de *x-vectors* já extraídos da DNN original, a abordagem adiciona um termo de regularização a ser utilizado durante o treinamento da própria rede, em conjunto com as outras abordagens para o controle simultânero tanto das distribuições *a priori* e como das distribuições condicionadas aos locutores. Esse método está descrito na Seção 3.3.

3.2 CLASSIFICAÇÃO E *POOLING* GAUSSIANOS

Como descrito anteriormente, um dos objetivos das abordagens propostas consiste no controle das distribuições condicionais associadas aos locutores e, para isso, propomos controlar a maneira como a rede codifica as informações aprendidas para diferenciar as locuções produzidas por diferentes locutores. O intuito aqui é fazer com que a rede aprenda a diferenciação entre locutores a partir de informações que seguem distribuições normais (gaussianas). Nesse cenário, a abordagem proposta consiste na modificação da DNN em

duas etapas importantes: na camada de classificação e na camada de *pooling* temporal.

Para restringir a codificação das informações aprendidas por uma determinada camada, utilizamos nós cuja operação é definida por uma função de base radial (*Radial Basis Function* - RBF). Uma RBF, $\phi(\boldsymbol{x}) : \mathcal{R}^D \rightarrow \mathcal{R}$, é uma função real cujos valores dependem apenas da norma entre o vetor de entrada e um ponto fixo, $\boldsymbol{c} \in \mathcal{R}^D$, chamado de centro:

$$\phi_{\boldsymbol{c}}(\boldsymbol{x}) = \phi(||\boldsymbol{x} - \boldsymbol{c}||). \tag{3.2}$$

Uma determinada RBF é parametrizada, portanto, pelo ponto fixo. Como nosso propósito é codificar os vetores correspondentes a um determinado locutor seguindo uma distribuição normal associada a ele, utilizamos a RBF gaussiana:

$$\phi_{\boldsymbol{c}}(\boldsymbol{x}) = \exp[-(\epsilon||\boldsymbol{x} - \boldsymbol{c}||)^2], \tag{3.3}$$

onde ϵ é um número real utilizado para escalar o *kernel* radial, que também pode ser observado como um parâmetro do nó a ser aprendido ou não pela rede. A Figura 23 apresenta os valores da RBF gaussiana centrada na origem para alguns valores de ϵ. Aprender os parâmetros de uma RBF gaussiana equivale a estimar os parâmetros de uma distribuição normal multivariada simples, com matriz de covariância diagonal e variâncias idênticas, parametrizadas através de ϵ.

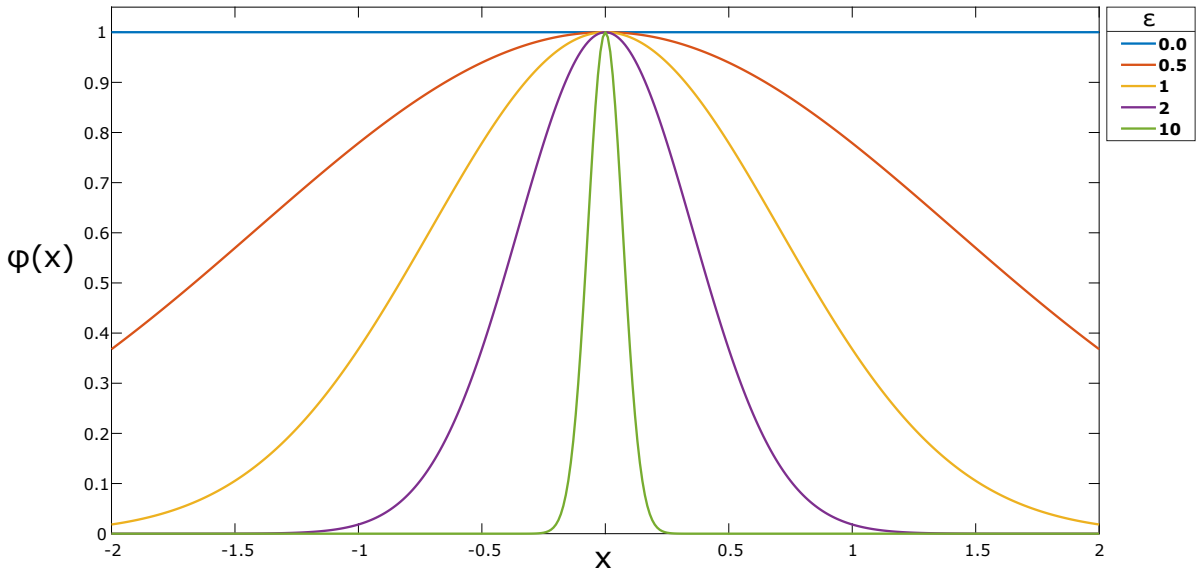

Figura 23 – Função de base radial (RBF) gaussiana centrada na origem para diferentes valores do parâmetro ϵ, que controla a escala radial da função.

Para a realização da classificação gaussiana, a abordagem consiste simplesmente na utilização de uma camada RBF para a classificação. Suponha que, para uma determinada locução de entrada X, a saída da camada anterior à camada de saída é composta pelo

vetor $\boldsymbol{o}$ e que a rede é treinada para classificar locuções provenientes de L locutores, $S_l, l = 1, 2, ..., L$. Nesse caso a probabilidade associada a cada um dos locutores é definida por:

$$p(S_l|X) = \frac{\phi_{\boldsymbol{c_l}}(\boldsymbol{o})}{\sum_{j=1}^{L} \phi_{\boldsymbol{c_j}}(\boldsymbol{o})}, \tag{3.4}$$

onde $\boldsymbol{c_l}$ é o centro associado ao locutor S_l. A Equação 3.4, combinada com a parte exponencial da Equação 3.3, é equivalente à função *softmax*, que é a função de ativação presente na camada de saída da DNN utilizada pela modelagem *x-vectors*. Para manter essa descrição, podemos manter uma camada com nós que realizam a operação presente no argumento das exponenciais da Equação 3.3 e então utilizar uma função de ativação *softmax*. Dessa maneira, a camada RBF é definida pela função

$$\phi^{\star}_{\boldsymbol{c}}(\boldsymbol{x}) = -\epsilon||\boldsymbol{x} - \boldsymbol{c}||^2, \tag{3.5}$$

e com função de ativação *softmax*, cuja saída define as probabilidades *a posteriori* dos locutores:

$$p(S_l|X) = \frac{\exp[\phi^{\star}_{\boldsymbol{c_l}}(\boldsymbol{o})]}{\sum_{j=1}^{L} \exp[\phi^{\star}_{\boldsymbol{c_j}}(\boldsymbol{o})]}. \tag{3.6}$$

Ao utilizar essa classificação, a rede mapeia a locução de entrada, camada a camada, até uma representação onde o modelo de um determinado locutor pode ser descrito através de uma distribuição normal e as representações referentes ao locutor se distribuem radialmente próximas ao centro associado ele.

A camada de classificação gaussiana possui a mesma quantidade de parâmetros que a camada de saída convencional (camada densa), uma vez que cada nó RBF é parametrizado pelo vetor de centro, cuja dimensionalidade é a mesma que o vetor de pesos de um neurônio, e pelo fator de escala, ϵ, que é um valor real como o *bias* do neurônio.

Mesmo no cenário em que o espaço onde a classificação é realizada (definido pela camada de saída) modela as representações produzidas pelos locutores utilizando distribuições normais, o controle sobre o espaço dos *x-vectors* ainda é limitado, visto que eles são extraídos a partir da camada anterior. De fato, quanto mais próxima uma determinada camada está da camada de saída, mais próximo o espaço gerado pela camada está do espaço onde a classificação dos locutores de treinamento é realizada. Esse espaço é majoritariamente definido pelas informações necessárias para diferenciar os locutores de treinamento e maior é a chance de a representação estar enviesada pelos dados de treinamento (*overfitting*). Essa é a principal razão pela qual as representações são extraídas da primeira camada após o *pooling* temporal (SNYDER et al., 2017; SNYDER et al., 2018).

Como os *x-vectors* são extraídos da parte linear da camada imediatamente após o *pooling* temporal, temos que eles são resultado de uma transformação linear da agregação realizada no *pooling*, o que torna essa camada uma excelente candidata para controle

da distribuição dos *x-vectors*. A abordagem proposta neste trabalho consiste, então, em modelar o espaço produzido pelo *pooling* temporal utilizando gaussianas.

A camada de *pooling* temporal agrega as representações temporais aprendidas pela rede e as agrega através do cálculo de estatísticas, como média e desvio-padrão para o caso dos *x-vectors*. A resposta da rede é composta pela concatenação dessas estatísticas para formar a representação, de dimensão fixa, para a locução inteira. A ideia da abordagem proposta é a mesma, porém, um controle é aplicado sobre a distribuição das representações temporais. Aqui, modela-se o espaço dessas representações através de um conjunto de RBFs, com o intuito de gerar representações provenientes de distribuições normais. Uma analogia direta pode ser realizada entre o conjunto de RBFs gaussianas com um modelo de fundo GMM. Cada RBF é responsável por modelar uma parte do espaço e um determinado vetor pode ser visto como pertencente a essas diferentes partes do espaço modelado pelo conjunto de gaussianas. Sob esse ponto de vista, podemos dizer que estamos modelando esse espaço através de um modelo generativo, caracterizado por um modelo de misturas de RBFs, cujos parâmetros serão aprendidos pela rede.

A própria RBF gaussiana (Equação 3.3) computa um valor para o grau de pertinência de um determinado vetor à partição[1] do espaço modelado por ela. Esse valor possui uma analogia direta com a verossimilhança de uma observação dada uma componente de mistura de um GMM, e, através da Equação 3.6 também podemos calcular a distribuição *a posteriori* das componentes. Nesse cenário, estatísticas como as de Baum–Welch (Seção 2.4.3) podem ser computadas nesse espaço caracterizado pelas RBFs. Tais estatísticas são utilizadas na caracterização dos supervetores GMM e possuem as informações necessárias para descrição do modelo gerado pela adaptação do UBM utilizando os vetores de características de uma determinada locução. De fato, as estatísticas de ordem zero e primeira ordem são suficientes, por exemplo, para a decomposição da matriz de variabilidade total dos *i-vectors*. Ao modelar o espaço através de um modelo de fundo, tais estatísticas agregam as informações temporais extraídas das locuções utilizando o grau de pertinência dos vetores a cada uma das componentes do modelo. Suponha uma camada formada por M RBFs gaussianas, parametrizadas pelos pontos centrais $\boldsymbol{c_j}$, $j = 1, ..., M$, e que a saída da camada anterior é composta, para uma determinada locução, de um conjunto de T vetores com representações temporais, $\boldsymbol{O} = \{\boldsymbol{o_1}, \boldsymbol{o_2}, ..., \boldsymbol{o_T}\}$. Para cada componente j do modelo de misturas, as estatísticas de ordem zero e primeira ordem são definidas por:

$$n_i = \sum_{t=1}^{T} p(i|\boldsymbol{o}_t), \tag{3.7}$$

$$E_i = \frac{1}{n_i} \sum_{t=1}^{T} p(i|\boldsymbol{o}_t)(\boldsymbol{o}_t - \boldsymbol{c_i}), \tag{3.8}$$

[1] Em termos mais precisos, estamos nos referindo a partições difusas (*fuzzy*), às quais podem ser atribuídas graus de pertinência de um determinado vetor do espaço.

respectivamente, e a probabilidade *a posteriori* da componente é calculada da mesma maneira como na Equação 3.4:

$$p(l|\boldsymbol{o}) = \frac{\phi_{\boldsymbol{c_l}}(\boldsymbol{o})}{\sum_{j=1}^{L} \phi_{\boldsymbol{c_j}}(\boldsymbol{o})}. \tag{3.9}$$

Enquanto que as estatísticas de ordem zero definem os graus de pertinência das representações temporais para as componentes de misturas, as estatísticas de primeira ordem extraem representações médias de deslocamento para cada uma delas. Assim como nas abordagens utilizando supervetores GMM e *i-vectors*, a combinação delas é assumida suficiente para caracterização da variabilidade contida nas representações temporais. A Figura 24 apresenta a arquitetura da DNN com as camadas de classificação e *pooling* gaussianos.

A saída da camada de *pooling* gaussiano proposta é composta pela concatenação das estatísticas de ordem zero e primeira ordem (Equações 3.7 e 3.8) de todas as componentes presentes na camada. Para uma camada formada por M componentes RBFs, cuja entrada e vetores de centro possuem dimensção K, a saída produzida possui dimensionalidade $M \times (K + 1)$. Uma importante diferença entre a camada de *pooling* proposta e a camada de *pooling* estatístico convencional é que a primeira desempenha um papel bem mais importante na modelagem do espaço que agrega as representações temporais. A camada convencional é composta pelos vetores de média e desvio-padrão das representações produzidas pela camada anterior. Dessa maneira, essa camada apenas produz uma agregação simples das representações anteriores, produzidas pelas camadas da primeira parte da rede, que são responsáveis pela robustez presente nas representações.

Já a camada proposta é responsável pela descrição do espaço das representações temporais em um espaço mais complexo, composto por misturas de RBFs. Dessa maneira, é natural assumir que ao diminuir a dimensionalidade da representação produzida pelas camadas temporais, a modelagem da variabilidade existente nesse espaço mais reduzido pode ser alcançada utilizando um número adequado de misturas. Por exemplo, na camada DNN convencional, as camadas temporais produzem representações de dimensionalidade $3 \times 512 = 1536$ para cada janela temporal, e a camada de *pooling* agrega as informações para geração do vetor de estatísticas de dimensão 3072. Com a camada proposta, a primeira parte da rede poderia gerar representações mais compactas, com, digamos, dimensionalidade 64, e 48 componentes de mistura poderiam ser utilizados para geração da representação final com dimensionalidade similar à da representação estatística na rede convencional.

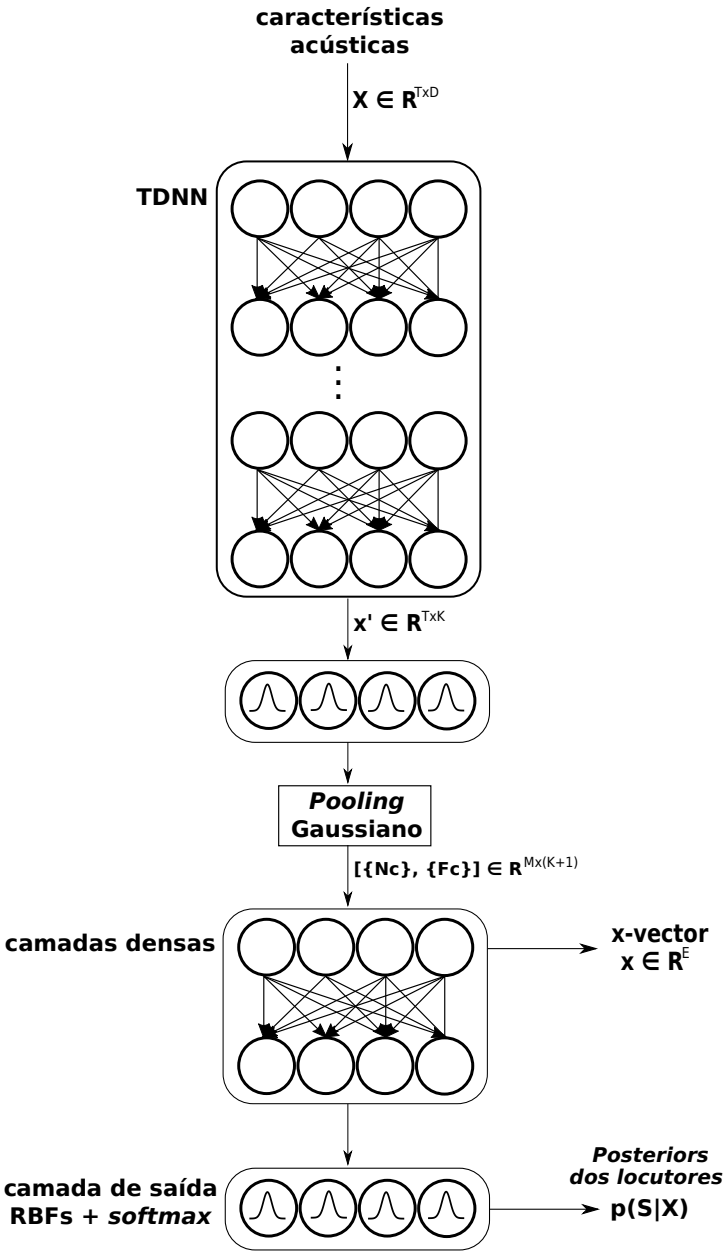

Figura 24 – DNN proposta com classificação e *pooling* gaussianos. Na classificação, uma camada RBF é utilizada para geração das probabilidades *a posteriori* dos locutores. Já para o *pooling* gaussiano, uma camada com M componentes RBFs é utilizada para a modelagem das representações temporais de entrada e a representação de saída é definida pela concatenação das estatísticas de ordem zero e primeira ordem extraídas de cada uma das componentes.

3.3 FUNÇÃO DE REGULARIZAÇÃO PARA CONTROLE DA DISTRIBUIÇÃO *A PRIORI* DOS *X-VECTORS*

Como visto anteriormente, propomos neste trabalho uma abordagem para controle da distribuição *a priori* dos *x-vectors* como uma maneira indireta de controle da distribuição do subespaço responsável pelas variabilidades intra-locutores. Relembrando a Equação

3.1, temos que o espaço dos *x-vectors* segue uma distribuição cuja decomposição assume-se seguir a forma

$$\boldsymbol{x} = \boldsymbol{x_m} + \boldsymbol{\Phi\beta} + \boldsymbol{\epsilon_r}, \tag{3.10}$$

composta por dois subespaços: o da componente do locutor $\boldsymbol{\beta}$ e o da componente residual $\boldsymbol{\epsilon_r}$. Partindo da suposição que as distribuições condicionais dos locutores seguem distribuições normais, a distribuição do subespaço residual fica condicionada a ser também uma distribuição normal se a mesma suposição puder ser realizada a $\boldsymbol{x}$.

O objetivo da abordagem é, portanto, restringir a distribuição dos *x-vectors* de uma maneira geral, sem levar em consideração as distribuições particulares dos vetores produzidos por um determinado locutor. Além disso, é desejável que tal abordagem possa ser integrada às abordagens propostas anteriormente, de maneira que os controles sobre as distribuições subjacentes dos *x-vectors* possam ser realizados conjuntamente. Para alcançar tais objetivos, propomos a adição de um termo de regularização à função objetivo utilizada para o treinamento da DNN. O método é baseado na abordagem variacional e a função de regularização computa uma medida de divergência entre a distribuição dos *x-vectors* gerados pela rede e uma distribuição desejável qualquer. Temos, portanto, que nenhuma mudança é realizada nas camadas já existentes na rede, o que torna o método compatível com as outras técnicas propostas. A rede permanece sendo treinada para discriminar locuções de diferentes locutores, mas sob a influência da função de regularização que torna o espaço das representações similar ao de uma distribuição desejável, que no nosso caso, é a distribuição normal.

Como visto na Seção 2.5.4, uma abordagem parecida para regularização da distribuição dos *x-vectors* foi proposta por Zhang *et al.*, onde um segundo modelo DNN foi treinado para projetar *x-vectors* em um subespaço que segue a distribuição normal padronizada (ZHANG; LI; WANG, 2019). O modelo consiste de um auto-codificador variacional (VAE) (KINGMA; WELLING, 2013), composto por dois módulos principais, o codificador (*encoder*) e o decodificador (*decoder*), que é treinado para realizar o mapeamento entre a representação original $\boldsymbol{x}$ e uma representação latente $\boldsymbol{z}$, tal que $p(\boldsymbol{z}) \sim N(\boldsymbol{z}; \boldsymbol{0}, \boldsymbol{I})$. Assim como os auto-codificadores (*autoencoders*) convencionais, o VAE é um modelo não-supervisionado (ou auto-supervisionado) treinado para reconstruir a própria entrada, gerando para isso a representação intermediária $\boldsymbol{z}$. A Figura 25 apresenta um exemplo desse processo. O VAE realiza essa tarefa utilizando um limitante inferior do logaritmo da verossimilhança das observações de $\boldsymbol{x}$ (*Evidence Lower Bound* - ELBO) para otimizar aproximações para a verossimilhança marginal dos dados, $p_\theta(\boldsymbol{x}|\boldsymbol{z})$, e para a distribuição de inferência, $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, onde ϕ e θ são os parâmetros do *encoder* e do *decoder*, respectivamente. Dada uma amostra $\boldsymbol{x}_i$, a função objetivo é definida como:

$$\mathcal{L}(\boldsymbol{x}_i, \phi, \theta) = -D_{KL}\Big[q_\phi(\boldsymbol{z}|\boldsymbol{x_i})||p(\boldsymbol{z})\Big] + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x_i})}\Big[\log[p_\theta(\boldsymbol{x_i}|\boldsymbol{z})]\Big] \leq \log[p_\theta(\boldsymbol{x})], \tag{3.11}$$

onde a primeira parte é definida pela divergência KL entre $q_\phi(\boldsymbol{z}|\boldsymbol{x_i})$ e a distribuição *a priori*

desejada $p(\boldsymbol{z})$, e a segunda parte corresponde à esperança associada ao mapeamento das variáveis latentes para o espaço original. Considerando o conjunto de dados observáveis D, a função objetivo maximizada durante o treinamento do VAE é definida por:

$$\mathcal{L}_{ELBO} = \mathbb{E}_{p_D(\boldsymbol{x})}\Big[\mathcal{L}(\boldsymbol{x}, \phi, \theta)\Big] \leq \mathbb{E}_{p_D(\boldsymbol{x})}\Big[\log[p_\theta(\boldsymbol{x})]\Big]. \tag{3.12}$$

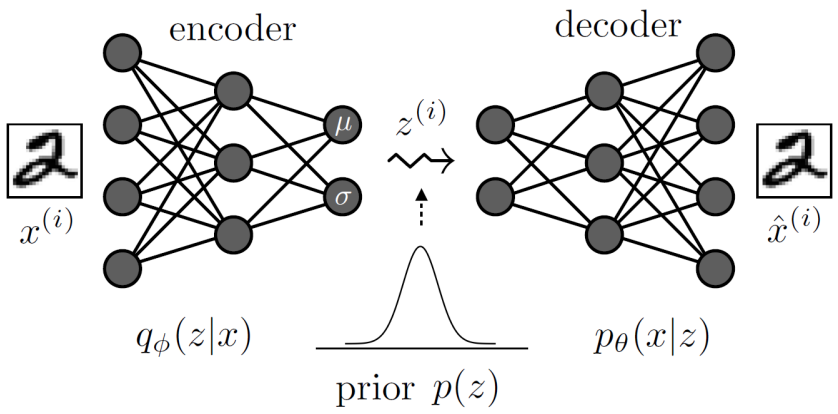

Figura 25 – Exemplo da abordagem VAE aplicada para aprender o mapeamento entre o espaço original, definido por uma imagem de caractere numérico, e um espaço latente com distribuição *a priori* normal. O *encoder* é responsável pela geração dos parâmetros da distribuição normal do espaço latente, enquanto que o *decoder* é utilizado para realizar a reconstrução da imagem de entrada através de uma amostra da distribuição gerada. Imagem adaptada de (TSCHANNEN; BACHEM; LUCIC, 2018).

A abordagem variacional seguida pelos VAEs desempenha um importante papel na área de aprendizado de representações, desencadeando o desenvolvimento de uma série de modelos generativos (TSCHANNEN; BACHEM; LUCIC, 2018), com aplicações, por exemplo, em processamento de linguagem natural (BOWMAN et al., 2015; HU et al., 2017; SERBAN et al., 2017), visão computacional (DENTON et al., 2017; VILLEGAS et al., 2017), aprendizado adversário (CHEN et al., 2016a; MATHIEU et al., 2016) e adaptação de domínio de imagem (ISOLA et al., 2017; ZHU et al., 2017).

Zhang *et al.* mostraram que o mapeamento dos *x-vectors* para representações gaussianas resulta em ganhos de desempenho quando a tarefa de verificação é realizada através da modelagem PLDA (ZHANG; LI; WANG, 2019). Mais recentemente, a mesma abordagem foi utilizada para adaptação do domínio dos *x-vectors* (WANG; LI; WANG, 2019; TU; MAK; CHIEN, 2019). Nesse cenário, a abordagem é utilizada para projetar os vetores extraídos em um determinado domínio (utilizando locuções telefônicas, por exemplo) em um espaço mais adequado para o domínio onde a tarefa ocorrerá (cadastramento e teste utilizando locuções gravadas com microfones, por exemplo).

Porém, pesquisadores descobriram que, dependendo da capacidade do modelo, da complexidade das distribuições marginais e da dimensionalidade atribuída a $\boldsymbol{z}$, a divergência KL presente na função objetivo ELBO (Equação 3.12) pode ser muito restrita (CHEN et

al., 2016b; ZHAO; SONG; ERMON, 2017), levando a duas situações indesejáveis. O maior problema consiste na definição de uma distribuição $q_\phi(z|x)$ que não aproxima bem a distribuição de inferência $q(\boldsymbol{z}|\boldsymbol{x})$. Isso pode ocorrer devido à baixa dimensionalidade de $\boldsymbol{z}$ (menor que a dimensão intrínseca, necessária para a devida reconstrução) ou devido à falta de capacidade do *decoder*. Quando o modelo não é capaz de maximizar ambos os termos da função objetivo, ele tende a maximizar a similaridade entre a distribuição de $\boldsymbol{z}$ e a distribuição *a priori* desejada em detrimento à verossimilhança marginal dos dados (para o VAE apresentado, a reconstrução dos dados). Nesse caso, as representações geradas tendem a seguir a distribuição desejada independentemente das amostras de entrada. Por outro lado, se o *decoder* for mais complexo do que o necessário, o modelo tende a ignorar as variáveis latentes para a maximização da verossimilhança dos dados. Nesse caso, no decorrer do treinamento, $p_\theta(x|z)$ se aproxima da distribuição observada, $p_D(\boldsymbol{x})$, independentemente de $\boldsymbol{z}$ e o espaço latente se torna inexpressivo.

Alternativas para superar tais problemas incluem o uso de funções menos restritas para maximizar a distribuição marginal dos dados $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ (CHEN et al., 2016b), o controle sobre os graus de importância entre os termos da função objetivo (HIGGINS et al., 2017), e o uso de termos de regularização para maximização da quantidade de informação mutualmente existente em $\boldsymbol{x}$ e $\boldsymbol{z}$ (CHEN et al., 2018; KIM; MNIH, 2018). Em 2017, Zhao *et al.* propuseram uma nova família de funções objetivos chamada de *InfoVAEs*, que generaliza a função ELBO e, explicitamente, pondera os termos correspondentes às maximizações da verossimilhança marginal dos dados observados, da similaridade entre a distribuição do espaço latente e a distribuição desejada e da informação mútua entre $\boldsymbol{x}$ e $\boldsymbol{z}$ (ZHAO; SONG; ERMON, 2017). Eles partiram de uma formulação alternativa, mas equivalente, da função objetivo ELBO:

$$\mathcal{L}(\phi,\theta) = -D_{KL}\Big[q_\phi(\boldsymbol{z})||p(\boldsymbol{z})\Big] - \mathbb{E}_{p(\boldsymbol{z})}\Big[D_{KL}[q_\phi(\boldsymbol{x}|\boldsymbol{z})||p_\theta(\boldsymbol{x}|\boldsymbol{z})]\Big], \tag{3.13}$$

onde $q_\phi(\boldsymbol{x}|\boldsymbol{z})$ é a distribuição a *posteriori* de $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ considerando o conjunto de dados observados D, isto é, $q_\phi(\boldsymbol{z}|\boldsymbol{x})p_D(\boldsymbol{x})/p(\boldsymbol{z})$. Além de adicionar um terceiro termo que mede a informação mútua existente entre um $\boldsymbol{x}$ e $\boldsymbol{z}$, fatores de escala são utilizados para o controle de suas importâncias:

$$\begin{aligned}\mathcal{L}_{INFO}(\phi,\theta) = \quad & -\mathbb{E}_{p(\boldsymbol{z})}\Big[D_{KL}[q_\phi(\boldsymbol{x}|\boldsymbol{z})||p_\theta(\boldsymbol{x}|\boldsymbol{z})]\Big] \\ & -\lambda\ D_{KL}\Big[q_\phi(\boldsymbol{z})||p(\boldsymbol{z})\Big] \\ & +\alpha\ I_q(\boldsymbol{x};\boldsymbol{z}),\end{aligned} \tag{3.14}$$

onde $I_q(\boldsymbol{x};\boldsymbol{z})$ define a informação mútua sob a distribuição $q(\boldsymbol{x},\boldsymbol{z}) = q_\phi(\boldsymbol{z}|\boldsymbol{x})p_D(\boldsymbol{x})$. Após

a manipulação dos termos, a função objetivo resultante é definida por:

$$\begin{aligned}\mathcal{L}_{INFO}(\phi,\theta) = \quad & \mathbb{E}_{p_D(\boldsymbol{x})}\ \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\Big[log[p_\theta(\boldsymbol{x}|\boldsymbol{z})]\Big] \\ & -(1-\alpha)\ \mathbb{E}_{p_D(\boldsymbol{x})}\Big[D_{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})]\Big] \\ & -(\alpha+\lambda-1)\ D_{KL}\Big[q_\phi(\boldsymbol{z})||p(\boldsymbol{z})\Big].\end{aligned} \tag{3.15}$$

Zhao *et al.* provaram que, por conta da intratabilidade em computar $\log q_\phi(\boldsymbol{z})$, o terceiro termo da função pode ser substituída por outras funções de divergência restritas, que podem ser otimizadas eficientemente por técnicas sem a computação das verossimilhanças[2]. Uma amostra $\boldsymbol{x} \sim p_D(\boldsymbol{x})$ é usada para produzir uma amostra $\boldsymbol{z} \sim q_\phi(\boldsymbol{z})$, e essas funções são utilizadas para estimar a divergência entre $\boldsymbol{z}$ e uma amostra da distribuição desejada ($p(\boldsymbol{z})$). Exemplos dessas funções são a Divergência de Jensen-Shannon (via treinamento adversário) (GOODFELLOW et al., 2014), o Gradiente Variacional de Stein (LIU; WANG, 2016), e a Máxima Discrepância-Média (*Maximum Mean Discrepancy* - MMD) (GRETTON et al., 2007). Particularmente, Zhao *et al.* definiram o chamado MMD-VAE como um caso particular da família dos InfoVAEs atribuindo uma importância máxima para o termo referente à informação mútua ($\alpha = 1$ na Equação 3.15), e utilizando a MMD como função de divergência. Nesse caso, a função objetivo é definida apenas em termos da divergência, MMD$[q_\phi(\boldsymbol{z})||p(\boldsymbol{z})]$, e da verossimilhança marginal dos dados, $log[p_\theta(\boldsymbol{x}|\boldsymbol{z})]$. O MMD-VAE mostrou-se mais estável do que o VAE convencional, mesmo utilizando outras funções de divergência, geralmente alcançando uma reconstrução acurada dos dados e sempre preservando um espaço latente informativo (ZHAO; SONG; ERMON, 2017).

Neste trabalho, propomos o controle da distribuição *a priori* dos *x-vectors* através da adição de um termo de regularização à função objetivo utilizada para treinar a DNN. Mais precisamente, utilizamos a função MMD para minimizar a divergência entre a distribuição dos *x-vectors* gerados e a distribuição desejável, que é a normal padronizada. A Figura 26 apresenta a arquitetura do método proposto.

Como um modelo de classificação, a DNN convencional utilizada para geração dos *x-vectors* é treinada através da função de entropia cruzada:

$$\mathcal{L}_C = \mathbb{E}_{p_D(\boldsymbol{x})}\Big[\log[p(s|\boldsymbol{x})]\Big], \tag{3.16}$$

onde $\boldsymbol{x}$ e s são o *x-vector* gerado por uma determinada locução e o rótulo do locutor correspondente, respectivamente. O termo de regularização a ser adicionado é definido por:

$$\mathcal{L}_V = -\lambda\ D_{KL}\Big[p(\boldsymbol{x})||p(\boldsymbol{z})\Big], \tag{3.17}$$

onde $p(\boldsymbol{z})$ corresponde à distribuição *a priori* desejada (no nosso caso, $p(\boldsymbol{z}) = N(\boldsymbol{z}; \mathbf{0}, \mathbf{I})$) e λ é um fator de escala que controla sua importância na função objetivo final.

[2] Tais técnicas são usualmente referenciadas como técnicas livres de verossimilhança (*likelihood-free techniques*).

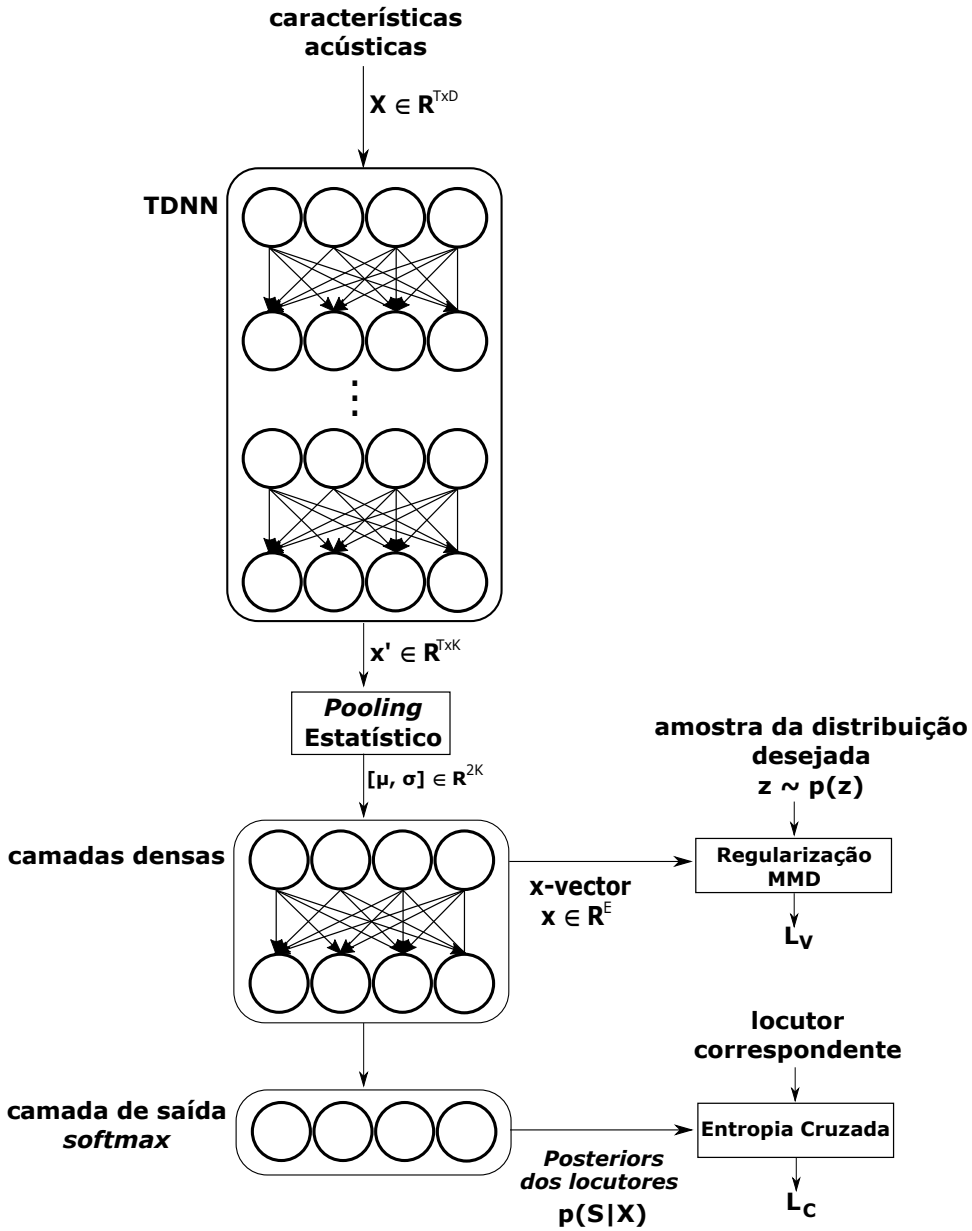

Figura 26 – DNN proposta com a adição do termo de regularização para controle da distribuição *a priori* dos *x-vectors*. Os vetores são comparados com um amostra da distribuição desejada, que é apresentada ao modelo como uma nova entrada. A função objetivo utilizada durante o treinamento é composta pelo termo de classificação $\mathcal{L}_C$, definido pela entropia cruzada e pelo termo de regularização $\mathcal{L}_V$, definido pela função de divergência não paramétrica MMD.

Assim como em (ZHAO; SONG; ERMON, 2017), nós substituímos a divergência KL pela função MMD. A MMD mede a divergência entre duas distribuições através de um teste estatístico não paramétrico de suas amostras (BORGWARDT et al., 2006; GRETTON et al., 2007; GRETTON et al., 2012). Dados conjuntos de amostras extraídas de duas distribuições, p_u and p_v, a distância entre as distribuições é definida na abordagem MMD pela distância

entre a representação média definida por uma função de mapeamento φ:

$$\text{MMD}[p_u, p_v] = \Big\| \mathbb{E}_{\boldsymbol{u}\sim p_u}[\varphi(\boldsymbol{u})] - \mathbb{E}_{\boldsymbol{v}\sim p_v}[\varphi(\boldsymbol{v})] \Big\|. \tag{3.18}$$

Para o caso mais simples onde $\varphi(\cdot)$ é a função indentidade, a MMD consiste da distância entre as médias das amostras. Porém, funções de mapeamento mais complexas podem ser empregadas para comparar momentos de mais alta ordem das amostras. Para esse propósito, uma abordagem útil consiste na utilização do truque de *kernel* (*kernel trick*):

$$\begin{aligned} \text{MMD}^2[p_u, p_v] \quad &= \mathbb{E}_{\boldsymbol{u},\boldsymbol{u'}\sim p_u}\Big[k(\boldsymbol{u}, \boldsymbol{u'})\Big] + \mathbb{E}_{\boldsymbol{v},\boldsymbol{v'}\sim p_v}\Big[k(\boldsymbol{v}, \boldsymbol{v'})\Big] \\ &\quad -2\mathbb{E}_{\boldsymbol{u}\sim p_u,\boldsymbol{v}\sim p_v}\Big[k(\boldsymbol{u}, \boldsymbol{v})\Big], \end{aligned} \tag{3.19}$$

onde $k(\boldsymbol{u}, \boldsymbol{v}) = \langle \varphi(\boldsymbol{u}), \varphi(\boldsymbol{v}) \rangle$ é um *kernel* semi-definido positivo. Quando um *kernel* universal é utilizado, o espaço produzido pelo mapeamento é de dimensão infinita, e corresponde ao espaço funcional de Hilbert com reprodução de *kernel* (*Reproducing Kernel Hilbert Space* - RKHS). Nesse caso, a MMD se torna assintoticamente restrita:

$$\text{MMD}[p_u, p_v] = 0 \iff p_u = p_v. \tag{3.20}$$

Em (ZHAO; SONG; ERMON, 2017), o *kernel* universal utilizado foi o *kernel* gaussiano, definido como

$$k(\boldsymbol{u}, \boldsymbol{v}) = \exp\Big[-\frac{1}{2\sigma^2}||\boldsymbol{u} - \boldsymbol{v}||^2\Big], \tag{3.21}$$

onde σ^2 é o parâmetro de largura. Pela expansão de Taylor da função de mapeamento correspondente $\varphi(\cdot)$, pode-se observar que o *kernel* gaussiano abrange as estatísticas de todas as ordens. Dessa maneira, a MMD computa a divergência entre as distribuições comparando todos os seus momentos.

Combinando ambas as funções objetivo de classificação (Equação 3.16) e regularização (Equação 3.17), a função objetivo se torna

$$\mathcal{L} = \mathbb{E}_{p_D(\boldsymbol{x})}\Big[\log[p(s|\boldsymbol{x})] - \lambda\ \text{MMD}^2[p(\boldsymbol{x}), p(\boldsymbol{z})]\Big], \tag{3.22}$$

onde λ controla a importância dada ao termo de regularização. Os *x-vectors* gerados durante treinamento em *mini-batch* são diretamente comparados com uma amostra da distribuição desejada $p(\boldsymbol{z})$. Uma importante característica da divergência MMD (Equação 3.19) é que não é necessário que os tamanhos das amostras sejam iguais. Visto que o conjunto de *mini-batch* é geralmente pequeno (dezenas de vetores), uma amostra maior da distribuição desejada pode ser utilizada para aumentar a evidência da comparação entre as distribuições.

Em comparação com as outras abordagens para o controle das distribuições dos *x-vectors*, algumas vantagens do método proposto podem ser observadas:

- Como em (ZHANG; LI; WANG, 2019), a distribuição dos *x-vectors* é controlada por uma função objetivo baseada na abordagem variacional. Porém, ao invés de treinar um segundo modelo para a regularização de vetores extraídos utilizando uma DNN treinada anteriormente, o termo de regularização variacional proposto é adicionado ao modelo convencional, de maneira que a regularização é realizada já durante o treinamento dos *x-vectors*.

- Uma vez que a MMD é uma função de divergência não paramétrica, nenhum parâmetro de distribuição precisa ser estimado pela rede e nenhum mecanismo de amostragem é necessário, como acontece no VAE, por exemplo. A comparação entre as distribuições é empregada utilizando a própria amostra dos *x-vectors* produzidos pela rede durante o treinamento em *mini-batch*.

- A amostra da distribuição desejada, $\boldsymbol{z} \sim p(\boldsymbol{z})$, é modelada como uma entrada do modelo, o que torna o método flexível com respeito à distribuição desejada. Nenhuma operação interna da rede precisa ser alterada no caso da utilização de outra distribuição desejada.

4 EXPERIMENTOS

Neste capítulo, são descritos os experimentos realizados com o objetivo de comparar os principais métodos presentes na literatura com as abordagens propostas neste trabalho. Após a descrição da base de dados utilizada, é apresentada a metodologia de avaliação empregada e as métricas de desempenho utilizadas. Os experimentos foram divididos em 3 partes, com objetivos específicos:

(i) **Avaliação e comparação entre as modelagens com *i-vectors* e *x-vectors***: nesta parte dos experimentos são avaliados os desempenhos das modelagens, levando em consideração aspectos como pós-processamento das representações e utilização de técnicas de normalização de *scores*.

(ii) **Avaliação de técnicas para controle das distribuições dos *x-vectors***: nesta etapa foram avaliadas as técnicas presentes na literatura que possuem o objetivo de gerar *x-vectors* mais apropriados para a modelagem G-PLDA. Especificamente, levamos em consideração as duas técnicas descritas na Seção 2.5.4: o método proposto por Li *et al.*, onde um termo de regularização é adicionado à função de custo utilizada para treinamento da DNN (LI et al., 2019), e a abordagem proposta por Zhang *et al.*, onde um VAE (*Variational Autoencoder*) modelo é treinado para regularização e mapeamento dos *x-vectors*.

(iii) **Avaliação e comparação dos métodos propostos**: a terceira parte dos experimentos foi conduzida com o intuito de avaliar o desempenho alcançado pelos métodos propostos no Capítulo 3. Primeiro avaliamos as abordagens propostas para o controle das distribuições dos *x-vectors* condicionadas aos locutores, que correspondem às camadas de classificação e de *pooling* gaussianos. Em seguida, nos concentramos no método proposto para o controle sobre a distribuição *a priori* das representações, onde um termo de regularização baseado na abordagem variacional é utilizado. E por fim, avaliamos a combinação das abordagens propostas.

4.1 BASE DE DADOS

Os experimentos foram conduzidos utilizando a segunda metade da base de dados chamada *Fisher English Training* (CIERI et al., 2005). Essa base foi desenvolvida pela *Linguistic Data Consortium* (LDC) para o propósito de reconhecimento de fala (CIERI; MILLER; WALKER, 2004) e ela é dividida em duas partes com publicações em 2004 e 2005. Ela não é pública e nosso acesso está restrito apenas à segunda parte.

Ela é constituída de 5849 chamadas telefônicas entre duas pessoas realizadas por 6485 pessoas, das quais 2594 são homens e 3891 são mulheres. Além disso, foram utilizados

locutores que moram em diferentes partes dos Estados Unidos. Dessa maneira, há bastante variabilidade quanto ao regionalismo da língua inglesa, com a presença de diferentes sotaques. Os áudios foram gravados com taxa de amostragem de 8kHz, com os dois canais independentes e as conversas possuem duração máxima de dez minutos. Além dos sinais de áudio, a base de dados também possui as transcrições das conversas. As locuções foram definidas através dos segmentos de áudio extraídos utilizando as marcações do início e do fim das frases e aquelas com duração menor do que um segundo foram descartadas. Como as chamadas possuem dois sinais de áudio independentes, não há locuções com pessoas falando simultaneamente.

4.2 METODOLOGIA DE AVALIAÇÃO

Dados os conjuntos de locutores e seus respectivos conjuntos de locuções, a metodologia de avaliação separa a base de dados em duas partes disjuntas:

(i) Base de treinamento (ou de fundo): composta pelas locuções produzidas pelos locutores de treinamento. Esse conjunto de dados é utilizado para a estimação dos modelos. Na modelagem com *i-vectors* ela é utilizada para o treinamento do modelo UBM, para a estimação da matriz de variabilidade total e para o treinamento do modelo G-PLDA. Já na modelagem com *x-vectors*, esse conjunto é utilizado para o treinamento da DNN e do modelo G-PLDA.

(ii) Base de avaliação (ou de teste): composta pelas locuções produzidas pelos locutores de teste. Essa parte da base é utilizada para as fases de cadastramento dos locutores e teste dos modelos. Para cada locutor de teste, suas locuções são divididas em dois conjuntos: o conjunto de cadastro e o de avaliação. As locuções de cadastro são utilizadas para a geração do *template* do locutor. Nas modelagens com *i-vectors* e *x-vectors*, cada locução de cadastro produz um vetor de representação e os *templates* dos locutores são definidos pelo vetor médio. A tarefa de verificação é realizada através dos conjuntos de locuções de teste e os *templates* cadastrados. O vetor de representação extraído por uma determinada locução de teste é comparado com todos os vetores de cadastro e o sistema produz um *score*. Quando uma locução é comparada com o *template* do mesmo locutor que a produziu, temos um *score* da classe positiva, enquanto que um *score* da classe negativa é produzido no caso contrário. Os conjuntos de *scores* positivos e negativos são então utilizados para cálculo das métricas de desempenho.

Para compor os conjuntos de dados, foram selecionados aleatoriamente, para cada gênero, 2000 locutores de treinamento e 500 de teste. No conjunto de treinamento, todas as locuções provenientes dos locutores foram utilizadas, o que resulta em um total de 224169 locuções para homens e 228011 para mulheres. A Tabela 2 sumariza a distribuição

dos segmentos de voz com respeito à duração e às quantidades percentual e média em cada intervalo. Para ambos os gêneros, cada locutor possui aproximadamente 100 locuções, das quais em torno de 60% possuem duração entre 1 e 3 s. Locuções longas (acima de 10 s) abrangem em torno de 3% da base de dados.

Tabela 2 – Sumário da base de dados de treinamento para ambos os gêneros e considerando diferentes intervalos de duração. Cada locutor possui um pouco mais de 100 locuções, das quais a maioria é curta (duração entre 1 e 3 s).

Gênero	**Duração**	**Quantidade (%)**	**Quantidade média de locuções por locutor**
Masculino	1-3 s	56,7	51,7
	3-5 s	20,0	22,4
	5-10 s	19,6	21,9
	>10 s	3,8	4,2
	Todas	-	112,1
Feminino	1-3 s	59,2	52,5
	3-5 s	19,6	22,3
	5-10 s	18,0	20,6
	>10 s	3,2	3,6
	Todas	-	114,0

Enquanto que nenhuma restrição foi imposta aos segmentos de voz presentes na base de treinamento, na base de teste alguns cuidados tiveram que ser tomados durante a separação entre os conjuntos de cadastro e teste. Para assegurar a variabilidade de sessão entre as locuções de cadastro e de avaliação, foram selecionados para o conjunto de teste locutores que participaram de pelo menos duas chamadas telefônicas. Os conjuntos de locuções de cadastro e avaliação foram compostos por locuções extraídas de chamadas distintas. Para cada locutor primeiramente uma das chamadas foi escolhida aleatoriamente para extração dos segmentos de voz de cadastro. Dessa chamada, foram selecionadas aleatoriamente 10 locuções com durações entre 1 e 5 s para compor o conjunto de locuções de cadastro. Ao somar as durações das locuções escolhidas temos que, na média, aproximadamente 30 s de áudio foram utilizados na fase de cadastro de cada locutor. A Tabela 3 apresenta a distribuição das durações de cadastro considerando todos os locutores de teste.

Para um determinado locutor cadastrado S, o protocolo utiliza as suas locuções de teste para calcular os *scores* positivos do sistema. Já para os *scores* negativos, todas as locuções de testes dos outros locutores, diferentes de S, são utilizadas. Consideramos nos experimentos a autenticação dependente de gênero. Isto é, o conjunto de locutores utilizados no teste de impostores de S é composto apenas de locutores do mesmo gênero

Tabela 3 – Distribuição das durações (em segundos) de cadastro dos locutores. Para cada locutor, 10 locuções com durações entre 1 e 5s foram escolhidas aleatoriamente para compor o conjunto de cadastro. Em média, os locutores foram cadastrados utilizando aproximadamente 30s de áudio.

Durações (s)	**Gênero**	
	Masculino	**Feminino**
Mínima	26,92	26,81
Média	31,19	31,24
Mediana	31,24	31,25
Máxima	35,42	35,52

de S[1]. Portanto, para cada locutor S, o conjunto de locuções de testes de impostores é formado pelas locuções de teste de um total de 499 locutores. Como a quantidade de locuções de teste varia para cada locutor, as quantidades de *scores* positivos e negativos também variam. Ao todo, os sistemas foram avaliados em oito condições de teste de maneira independente, levando em consideração o gênero e a duração da locução de teste. Os mesmos intervalos utilizados na descrição da base de dados foram utilizados. A Tabela 4 sumariza a quantidade de testes positivos e negativos em cada uma das oito condições de teste.

Tabela 4 – Quantidade total de testes positivos e negativos para cada uma das oito condições de teste, considerando o gênero do locutor de cadastrado e a duração do segmento de voz utilizado para autenticação.

Gênero	**Duração**	**Positivos ($\times 10^3$)**	**Negativos ($\times 10^6$)**
Masculino	1-3 s	22,9	11,4
	3-5 s	10,1	5,0
	5-10 s	10,9	5,5
	>10 s	4,3	2,2
Feminino	1-3 s	25,6	12,8
	3-5 s	10,3	5,1
	5-10 s	11,1	5,5
	>10 s	4,1	2,1

Em cada cenário de teste, os *scores* positivos e negativos produzidos pelo sistema foram utilizados para calcular as taxas de falsa aceitação (FAR) e falsa rejeição (FRR) independentes de locutor. Para um determinado limiar de aceitação θ, as taxas de FAR e FRR são definida como:

$$\mathrm{FRR} = \frac{1}{|S_p|} \sum_{s \in S_p} (1 - d_s), \tag{4.1}$$

[1] A escolha por essa metodologia se baseia no fato de autenticações entre gêneros não serem desafiadoras. Inclusive, nas competições de reconhecimento de locutores, esse tipo de autenticação nem é mais levada em consideração desde, pelo menos, 2004.

$$\text{FAR} = \frac{1}{|S_n|} \sum_{s \in S_n} d_s, \tag{4.2}$$

onde S_p e S_n são os conjuntos de *scores* positivos e negativos, e d_s é definido pela decisão do sistema levando em consideração um determinado *score* s:

$$d_s = \begin{cases} 1, & \text{se } s \geq \theta, \\ 0, & \text{caso contrário.} \end{cases} \tag{4.3}$$

Como visto na Seção 1.3.1, o limiar de aceitação θ define o ponto de operação do sistema, ao qual são associadas as taxas de erro, e ao variar os pontos de operação, uma curva ROC (*Receiver Operating Characteristic*) é gerada.

Neste trabalho, consideramos duas medidas de acurácia na comparação dos sistemas. A primeira delas é a taxa de erros iguais (*Equal Error Rate* - EER), que é definida pelo ponto de operação onde as taxas de FAR e FRR tomam o mesmo valor. A segunda medida é definida através do chamado custo de detecção:

$$\text{C}_{det}(\theta) = \text{C}_{FR} \times \text{FRR}(\theta) \times \text{P}_{pos} + \text{C}_{FA} \times \text{FAR}(\theta) \times (1 - \text{P}_{pos}), \tag{4.4}$$

onde os valores de FRR e FAR são ponderados levando em consideração seus custos C_{FR} e C_{FA}, respectivamente, e a proporção esperada de verdadeiros positivos P_{pos}. O valor de C_{det} associa, em um determinado ponto de operação, custos aos erros realizados pelo sistema. Ao variar os pontos de operação do sistema, os diferentes valores de custo de detecção geram uma curva, que é chamada de função de custo de detecção (*Detection Cost Function* - DCF). A segunda medida de desempenho utilizada nesse trabalho é a mesma utilizada na última competição da NIST, em 2019 (OMID; CRAIG, 2019), onde duas DCFs são geradas, considerando diferentes parâmetros de custos de detecção. Em ambas as curvas, os parâmetros C_{FR} e C_{FA} foram fixados em 1, enquanto que para a primeira e segunda curvas os parâmetros P_{pos} foram fixados em 0,01 e 0,005, respectivamente. Para fins de visualização, os valores de custo são normalizados através de um valor constante:

$$\text{C}^{\star}_{det}(\theta) = \frac{\text{C}_{det}(\theta)}{\text{C}_{const}}, \tag{4.5}$$

onde

$$\text{C}_{const} = \min\Big[\text{C}_{FR} \times \text{P}_{pos},\ \text{C}_{FA} \times (1 - \text{P}_{pos})\Big]. \tag{4.6}$$

Em ambas as DCFs utilizadas, $\text{C}_{const} = \text{P}_{pos}$. Finalmente, dadas as curvas normalizadas DCF_1 e DCF_2, a medida de desempenho final é definida pela média entre os valores mínimos das curvas:

$$\text{minDCF} = 0,5 \times \Big(\min[\text{DCF}_1] + \min[\text{DCF}_2]\Big). \tag{4.7}$$

Além dos valores de EER e minDCF, em alguns experimentos nós realizamos uma comparação visual das representações geradas pelos sistemas utilizando um conjunto de locuções. Para isso, os conjuntos de representações, definidas por vetores $\boldsymbol{x} \in \mathbb{R}^D$, foram projetadas para o plano $\mathbb{R}^2$ utilizando a técnica conhecida como t-SNE (*t-Distributed Stochastic Neighbor Embedding*), que é uma técnica de redução de dimensionalidade não linear amplamente utilizada para visualização de dados de alta dimensionalidade projetando-os em um espaço de baixa dimensionalidade (MAATEN; HINTON, 2008).

4.3 CARACTERÍSTICAS ACÚSTICAS

Neste trabalho, todos os sistemas, sejam eles baseados em *i-vectors* ou *x-vectors*, foram avaliados utilizando o mesmo conjunto de características acústicas. Dessa maneira, fixamos todo o processo de pré-processamento e extração de características dos sistemas. A Tabela 5 sumariza os parâmetros utilizados.

Durante o pré-processamento das locuções, apenas a operação de pré-ênfase (Seção 2.2.1) foi considerada. Como são disponibilizadas as transcrições das conversas telefônicas e as posições temporais em que ocorreram as locuções, não vimos necessidade de se executar nenhum método de detecção de voz (Seção 2.2.2). Em seguida, os sinais de voz foram segmentados em janelas de 20 ms, a cada 10 ms, utilizando a janela de Hamming (Equação 2.7). Por fim, coeficientes MFCC (Seção 2.3.4) foram extraídos utilizando um banco de filtros constituído de 26 filtros triangulares[2] igualmente espaçados na escala Mel. 19 coeficientes MFCC foram extraídos juntamente com a energia. Além disso, os coeficientes dinâmicos de primeira e segunda (Seção 2.3.5.1) ordem foram anexados ao conjunto de características, resultando em vetores de dimensionalidade 60. Por fim, os coeficientes foram pós-processados pela subtração de média cepstral (CMS, Seção 2.3.6.1) utilizando janelas de até 3 segundos.

Decidimos fixar a técnica de compensação para aquela que é a mais utilizada atualmente (por exemplo, na modelagem com *x-vectors* (SNYDER et al., 2018) e na técnica proposta por Li *et al.*, que também iremos avaliar neste trabalho). No trabalho que apresenta a modelagem com *i-vectors*, Dehak *et al.* utilizaram a técnica de deformação de características (FW, Seção 2.3.6.3) ao invés do CMS (DEHAK et al., 2011). Em experimentos preliminares, avaliamos as diferentes combinações de técnicas de compensação de características (Seção 2.3.6.4) utilizando a modelagem com *i-vectors*. Como não observamos diferenças significativas entre as técnicas FW e CMS (elas se mostraram um pouco melhores que as demais), escolhemos por sempre utilizar CMS durante as comparações.

[2] Seguimos a seguinte fórmula, comumente utilizada para fixar a quantidade de filtros presentes no banco: $n = \lfloor 3 \times \log Fs \rfloor$, onde n é a quantidade de filtros e Fs é a frequência de amostragem (RABINER; JUANG, 1993). Para as locuções da base de dados Fisher com $Fs = 8Khz$, temos $n = 26$.

Tabela 5 – Visão geral dos parâmetros utilizados durante a extração das características acústicas das locuções.

Parâmetro	**Valor**
Pré-ênfase	sim
VAD	não
Janelamento	janelas de 20 ms a cada 10ms
Função da Janela	Hamming
Banco de Filtros	26 filtros triangulares (escala Mel)
Coeficientes MFCC	19
Log engergia	sim
Coeficientes dinâmicos	de primeira e segunda ordem
Compensação	CMS com janelas de 3 s

4.4 AVALIAÇÃO DAS MODELAGENS UTILIZANDO *I-VECTORS* E *X-VECTORS*

A primeira parte dos experimentos foi conduzida com o objetivo de avaliar e comparar as modelagens utilizando as duas representações mais bem sucedidas até o momento, os *i-vectors* e os *x-vectors*. Para ambas as abordagens, um único modelo foi treinado para ambos os gêneros, isto é, seguimos uma abordagem independente de gênero. Além disso, todas as locuções do conjunto de treinamento foram consideradas para treinar todos os modelos.

Na abordagem com *i-vectors* (Seção 2.4.5), um UBM composto por 1024 misturas com matrizes de covariância diagonais foi estimado através da combinação de dois UBMs de 512 misturas, estimados utilizando as locuções de treinamento dos homens e das mulheres, respectivamente. A combinação entre os UBMs dependentes de gênero foi realizada da maneira mais simples: os conjuntos de misturas foram unidos e os pesos associados a cada uma das misturas foram divididos pela metade. Em seguida, uma matriz de variabilidade total foi estimada para projeção das locuções em *i-vectors* de dimensionalidade 600.

Já para a abordagem com *x-vectors* (Seção 2.5.4), as mesmas especificações da DNN apresentada pelos autores foram utilizadas. A primeira parte da rede é composta por uma TDNN (*Time-delayed Neural Network*), formada por cinco camadas de atraso temporal, que é seguida pela camada de *pooling* estatístico, que gera uma representação de dimensão fixa e igual a 3072 para cada locução de entrada da rede. Tal representação é então processada por duas camadas densas de 512 nós cada e finalmente pela camada de saída. Todas os nós da rede são formadas por ReLUs, com excessão da camada de saída, que possui função de ativação *softmax*. A Tabela 6 sumariza a quantidade de nós e contexto temporais das camadas presentes na DNN. Como a rede foi treinada utilizando todas as locuções do conjunto de treinamento, a camada de saída é composta por 4000 nós, que é

Tabela 6 – Quantidade de nós e contexto temporal das camadas de atraso temporal e densas presentes na DNN utilizada para geração dos *x-vectors*.

Camada	**Quantidade de nós**	**Contexto temporal**
TD-1	512	[t-2, t+2]
TD-2	512	{t-2, t, t+2}
TD-3	512	{t-3, t, t+3}
TD-4	512	{t}
TD-5	1536	{t}
Densa-1	512	{t}
Densa-2	512	{t}
Saída	4000	{t}

a quantidade total de locutores. Uma vez que os *x-vectors* são definidos pela resposta da função afim da primeira camada densa após o *pooling* estatístico, temos que os *x-vectors* possuem dimensionalidade 512. A rede completa possui 6,59 milhões de parâmetros, e para a extração dos *x-vectors*, 4,28 milhões deles são utilizados. Durante o treinamento da rede, a acurácia do conjunto de treinamento foi observada. As redes foram treinadas até que uma taxa de acerto de 95% na classificação das locuções ocorresse por três épocas consecutivas. Neste trabalho, todos os métodos baseados na modelagem com *x-vectors* foram treinados respeitando esse mesmo critério de parada.

Enquanto para os sistemas baseados em *i-vectors*, todas as caracterísitcas acústicas são utilizadas durante a modelagem (isto é, coeficientes MFCC acrescido dos coeficientes dinâmicos de primeira e segunda ordem), na modelagem com *x-vectors*, os coeficientes dinâmicos não são utilizados. Dessa maneira, enquanto que vetores de características acústicas de dimensão 60 são extraídas das locuções para a modelagem com *i-vectors*, a DNN dos *x-vectors* é treinada utilizando segmentos de voz caracterizados por vetores de dimensão 20. Para ambas as modelagens, um modelo G-PLDA é enfim estimado utilizando os vetores de representações extraídas das locuções do conjunto de treinamento.

A primeira avaliação das representações foi realizada levando em consideração a modelagem mais simples, isto é, sem a utilização de nenhum método de pós-processamento para as representações e sem utilizar técnicas de normalização de *scores*. Nesse caso, as representações geradas para as locuções de treinamento foram diretamente utilizadas na modelagem G-PLDA, e o modelo resultante foi então utilizado para a geração dos *scores* dos sistemas. A Tabela 7 apresenta os desempenhos alcançados pelos métodos.

Primeiramente, podemos observar que independentemente do gênero e da representação, o desempenho do sistema melhora com o aumento da duração da locução de teste. De fato, os piores resultados são observados no cenário onde a verificação é realizada utilizando locuções curtas (durações entre 1 e 3 s). No caso dos *i-vectors*, as taxas de erro nesse cenário de teste mais que dobram quando comparadas às taxas alcançadas utili-

Tabela 7 – Desempenhos alcançados pelas modelagens utilizando *i-vectors* e *x-vectors*. Nesse experimento, as representações são diretamente modeladas através do G-PLDA. Nenhuma técnica de pós-processamento ou normalização de *scores* foi utilizada.

EER (%) / minDCF x 10					
Gênero	**Modelagem**	**Duração**			
		1-3 s	**3-5 s**	**5-10 s**	**>10 s**
Masculino	*i-vectors*	21,72/9,91	12,55/9,62	10,86/9,39	9,88/9,12
	x-vectors	**12,63/9,26**	**8,40/8,45**	**7,79/7,48**	**7,49/6,69**
Feminino	*i-vectors*	24,78/9,74	13,77/8,89	11,78/8,58	10,19/8,15
	x-vectors	**13,11/9,45**	**8,71/8,42**	**7,67/7,33**	**7,35/6,55**

zando locuções mais longas (durações superiores a 10 s). Comparando ambas as representações, observamos que os desempenhos alcançados com *x-vectors* são consideravelmente superiores àqueles alcançados pelos *i-vectors*, principalmente quando locuções curtas são utilizadas. As taxas de erro alcançadas pelos *x-vectors* foram menores em todos os casos de teste.

Em seguida, avaliamos a importância do módulo de normalização de *scores* nas modelagens. Como visto na Seção 2.6, atualmente essa é uma etapa que está sempre presente nos sistemas, independente da abordagem seguida para a modelagem dos locutores. Em testes preliminares, avaliamos, para a modelagem com *i-vectors*, as quatro técnicas descritas neste trabalho: normalização zero, normalização de teste, normalização simétrica e normalização simétrica adaptativa. Observamos que os sistemas se beneficiaram da utilização das técnicas, melhorando os resultados em quase todos os casos de teste. Além disso, observamos que as normalizações simétrica e simétrica adaptativa apresentaram resultados similares, mas melhores que as outras. Como atualmente a técnica de normalização simétrica adaptativa (Seção 2.6.4) é a mais abrangentemente utilizada (NAUTSCH et al., 2014; SNYDER et al., 2018; VILLALBA et al., 2019), decidimos fixar as comparações utilizando essa técnica. A Tabela 8 apresenta os resultados alcançados pelos sistemas.

Comparando os resultados das Tabelas 7 e 8, pode-se observar que, de fato, a normalização de *scores* melhorou o desempenho dos sistemas. Os maiores ganhos de desempenho são observados para a modelagem com *i-vectors*, porém, o desempenho dos *x-vectors* foi novamente superior. A partir desse momento, todos os experimentos foram realizados normalizando *scores* dos sistemas através do método simétrico adaptativo.

Por último, avaliamos a importância dos métodos de pós-processamento das representações antes da modelagem G-PLDA. Duas foram as técnicas de pós-processamento consideradas: a redução de dimensionalidade através da análise dos discriminantes lineares (*Linear Discriminant Analysis* - LDA) e a normalização de comprimento (*Length Normalization* - LN). Consideramos os casos onde apenas um dos métodos é empregado

Tabela 8 – Desempenhos alcançados pelas modelagens utilizando *i-vectors* e *x-vectors*. Nesse experimento, as representações são diretamente modeladas através do G-PLDA e os *scores* gerados pelos sistemas são normalizados através do método simétrico adaptativo (Seção 2.6.4).

EER (%) / minDCF x 10					
Gênero	**Modelagem**	**Duração**			
		1-3 s	**3-5 s**	**5-10 s**	**>10 s**
Masculino	*i-vectors*	20,18/9,52	11,59/8,11	10,05/7,48	9,40/6,69
	x-vectors	**12,53/8,66**	**8,20/7,22**	**7,57/7,02**	**7,32/6,19**
Feminino	*i-vectors*	22,79/9,59	12,74/8,16	10,74/7,41	9,10/6,83
	x-vectors	**13,09/8,84**	**7,54/7,26**	**7,49/6,86**	**7,31/6,05**

e também o caso onde os vetores são processados pelo LDA e depois pela normalização, como é feito na modelagem com *x-vectors* (SNYDER et al., 2018). O objetivo dessa avaliação é fixar *baselines* de sistemas nos dois contextos, pós-processando ou não as representações. Para ambas as representações, o LDA foi utilizado para mapear o espaço original em um espaço mais compacto, de dimensão 200. A Tabela 9 apresenta os resultados alcançados pelos sistemas nos quatro casos utilizados para comparação.

Para ambas as representações, a combinação entre as técnicas LDA e LN foi a que mais melhorou os resultados dos sistemas. Para a modelagem com *i-vectors*, os valores de minDCF foram menores em todos os casos de teste, enquanto que as taxas de EER diminuiram em quase todos. O mesmo pode ser afirmado para os *x-vectors*. A Tabela 10 apresenta os ganhos de desempenho dos sistemas quando comparamos o caso LDA-LN com os resultados obtidos sem a realização do pós-processamento.

Os ganhos de desempenho são mais expressivos para os valores de minDCF. Para ambas as representações, os ganhos são mais limitados para locuções curtas, enquanto que ganhos expressivos são observados para locuções longas. Para os *i-vectors*, os ganhos de desempenho nas taxas de EER foram de mais de 10% para durações superiores a 10 s, enquanto que os ganhos para os valores de minDCF foram de aproximadamente 20%. Já para os *x-vectors*, os ganhos nos valores de minDCF chegaram a expressivos 30% em alguns casos de teste. Em média, os *i-vectors* se beneficiaram mais que os *x-vectors* nas taxas de EER, enquanto que o contrário pode ser afirmado para os valores de minDCF. De toda forma, os melhores desempenhos continuam sendo alcançados pelos *x-vectors*, com taxas de erro expressivamente menores que os *i-vectors*. A partir desse momento, como os experimentos foram voltados para a modelagem com *x-vectors*, definimos dois *baselines* para essa representação: utilizando pós-processamento (via LDA-LN) e sem utilizá-lo.

Através dos resultados observados durante essa fase dos experimentos, podemos salientar:

- No geral, o desempenho dos sistemas melhora com o aumento da duração das locu-

Tabela 9 – Resultados apresentados pelas modelagens com *i-vectors* e *x-vectors* utilizando técnicas de pós-processamento das representações. Duas técnicas foram consideradas: a redução de dimensionalidade via LDA e a normalização de comprimento (LN). São avaliados os casos em que as técnicas são utilizadas isoladamente e também o caso onde o LDA é aplicado seguido pelo LN.

EER (%) / minDCF x 10					
Gênero	**Modelagem**	**Duração**			
		1-3 s	**3-5 s**	**5-10 s**	**>10 s**
Masculino	*i-vectors*	20,18/9,52	11,59/8,11	10,05/7,48	9,40/6,69
	i-vectors/LN	20,93/9,71	12,91/8,16	11,64/6,86	9,65/5,74
	i-vectors/LDA	19,37/9,58	11,53/8,34	10,06/7,59	9,46/6,91
	i-vectors/LDA-LN	21,04/9,22	11,30/6,96	9,44/5,78	8,40/5,01
	x-vectors	12,53/8,66	8,20/7,22	7,57/7,02	7,32/6,19
	x-vectors/LN	12,47/8,29	8,73/6,75	7,58/6,68	7,18/6,03
	x-vectors/LDA	12,59/8,17	**8,16**/5,81	**7,16**/5,17	7,06/5,17
	x-vectors/LDA-LN	**12,35/7,91**	8,25/**5,48**	7,17/**4,91**	**6,98/4,84**
Feminino	*i-vectors*	22,79/9,59	12,74/8,16	10,74/7,41	9,10/6,83
	i-vectors/LN	23,33/9,80	13,52/8,51	11,83/7,36	9,37/6,22
	i-vectors/LDA	21,88/9,68	12,76/8,51	11,16/7,84	9,45/7,30
	i-vectors/LDA-LN	22,04/9,47	12,34/7,48	9,82/6,33	7,96/5,49
	x-vectors	13,09/8,84	7,54/7,26	7,49/6,86	7,31/6,05
	x-vectors/LN	12,95/8,41	**7,28**/6,92	7,24/6,38	7,54/5,71
	x-vectors/LDA	12,85/8,76	7,48/6,84	7,43/5,56	6,99/4,91
	x-vectors/LDA-LN	**12,76/8,36**	7,46/**6,36**	**7,23/4,72**	**6,88/4,73**

Tabela 10 – Ganhos de desempenho alcançados pelo pós-processamento das representações através do método LDA-LN.

Gênero	**LDA-LN - Ganhos de desempenho (%) - EER / minDCF**					
	Modelagem	**Duração**				**Média**
		1-3 s	**3-5 s**	**5-10 s**	**>10 s**	
Masculino	*i-vectors*	-4,26/3,15	2,50/14,18	6,07/22,73	10,64/25,11	3,74/16,29
	x-vectors	1,44/8,66	-0,61/24,10	5,28/30,06	4,64/21,81	2,69/21,16
Feminino	*i-vectors*	3,29/1,25	3,14/8,33	8,57/14,57	12,53/19,62	6,88/10,94
	x-vectors	2,52/5,43	1,06/12,40	3,47/31,20	5,88/21,82	3,23/17,71

ções de teste.

- A fase de normalização de *scores* se mostrou de fato importante e, por essa razão, ela foi fixada nos experimentos. Em especial, fixamos o método de normalização simétrica adaptativa.

- Os *x-vectors* se mostraram melhores que os *i-vectors* em ambos os casos, utilizando ou não pós-processamento. As taxas de EER e valores de minDCF foram expressivamente menores para os *x-vectors*.

- Os métodos de pós-processamento se mostraram importantes para ambas as representações. A partir desse momento, fixaremos a etapa de pós-processamento dos *x-vectors* através do método LDA-LN, que foi o que apresentou os melhores resultados.

4.5 AVALIAÇÃO DOS MÉTODOS DA LITERATURA PARA CONTROLE DOS *X-VECTORS*

Esta etapa dos experimentos foi conduzida para avaliar os métodos, presentes na literatura, propostos para o controle da distribuição dos *x-vectors*. Tais técnicas foram descritas na Seção 2.5.4 e possuem o objetivo de gerar representações mais adequadas para a modelagem G-PLDA. Em termos mais precisos, elas possuem o objetivo de fazer com que os *x-vectors* sigam uma distribuição normal. As técnicas avaliadas são:

- O método proposto por Li *et al.*, que consiste em uma mudança na operação da camada de classificação da DNN dos *x-vectors* e na adição de um termo de regularização à função de custo (LI et al., 2019). Dessa maneira, a quantidade de parâmetros da rede proposta é a mesma da DNN convencional dos *x-vectors*. Através da avaliação da acurácia da rede durante o treinamento, fixamos o peso associado ao termo de regularização para 0,01. Assim como os autores, referenciamos tal método como *x-vectors*/Gauss.

- A abordagem proposta por Zhang *et al.*, que utiliza um auto-codificador variacional (VAE) para mapear os *x-vectors* já treinados em um novo espaço que segue uma distribuição normal padronizada (ZHANG; LI; WANG, 2019). Utilizamos os mesmos parâmetros utilizados pelos autores, onde o VAE é composto por uma DNN com sete camadas densas, e gera representações de dimensão 200. O *encoder* e o *decoder* são compostos por 3 camadas com 1800 ReLUs cada. Ao todo, o VAE possui 15,94 milhões de parâmetros, mas durante a extração das novas representações, 8,15 milhões de parâmetros são utilizados. Esse método é referenciado como *x-vectors*/VAE.

A primeira comparação realizada considerou o caso onde nenhum método de pós-processamento é empregado. A Tabela 11 apresenta uma comparação dos valores de EER e minDCF alcançados pelas abordagens e pela modelagem convencional. Nesse caso, o método *x-vectors*/Gauss apresentou melhores resultados que os *x-vectors* convencionais em todos os casos de teste. Já a abordagem *x-vectors*/VAE melhorou os valores de minDCF em todos os casos, mas para três casos femininos, as taxas de EER aumentaram.

Tabela 11 – Comparação entre os desempenhos alcançados pelas abordagens *x-vectors*/Gauss, *x-vectors*/VAE e a modelagem convencional, no contexto onde nenhuma técnica de pós-processamento é empregada antes da modelagem com G-PLDA.

EER (%) / minDCF x 10					
Gênero	**Modelagem**	**Duração**			
		1-3 s	**3-5 s**	**5-10 s**	**>10 s**
Masculino	*x-vectors*	12,53/8,66	8,20/7,22	7,57/7,02	7,32/6,19
	x-vectors/Gauss	12,18/8,27	8,14/**6,68**	**7,13/6,46**	**7,09/5,66**
	x-vectors/VAE	**12,01/8,23**	**8,01**/6,98	7,16/6,78	7,04/5,95
Feminino	*x-vectors*	13,09/8,84	7,54/7,26	7,49/6,86	7,31/6,05
	x-vectors/Gauss	**12,99/8,45**	**7,49/6,64**	**7,45/6,10**	7,31/**5,28**
	x-vectors/VAE	13,51/8,47	7,83/7,05	7,68/6,58	**7,23**/5,66

A Tabela 12 apresenta os ganhos percentuais das duas técnicas em relação à modelagem convencional. No geral, os melhores valores de minDCF foram alcançados pela abordagem *x-vectors*/Gauss, mas melhoras nos valores de EER ocorreram também. O sistema *x-vectors*/VAE também apresentou melhoras nos valores de minDCF, enquanto que uma melhora média no EER ocorreu apenas para os testes com homens. No geral, a única técnica que melhorou o desempenho da modelagem *x-vectors* em todos os casos de teste foi a abordagem *x-vectors*/Gauss. Apesar do ganho de desempenho apresentado pelo modelo VAE em alguns casos, pode-se observar que no geral sua contribuição é mais limitada que o sistema *x-vectors*/Gauss. Como visto anteriormente, a abordagem com VAE considera o treinamento de um segundo modelo apenas para realização da regularização dos *x-vectors*, o que aumenta bastante o custo de desenvolvimento do sistema. Nesse sentido, a abordagem *x-vectors*/Gauss possui a vantagem de não acrescentar custo de processamento ao sistema convencional.

Em seguida, avaliamos a influência do pós-processamento das representações nos desempenhos do sistema. A Tabela 13 compara os desempenhos alcançados pelas abordagens

Tabela 12 – Ganhos de desempenho percentuais alcançados pelas abordagens *x-vectors*/Gauss, *x-vectors*/VAE em relação à modelagem convencional. Nessa comparação, nenhum pós-processamento é aplicado às representações.

Gênero	**Ganhos de desempenho (%) - EER / minDCF**					
	Modelagem	**Duração**				**Média**
		1-3 s	**3-5 s**	**5-10 s**	**>10 s**	
Masculino	*x-vectors*/Gauss	2,79/4,50	0,73/7,48	5,81/7,98	3,14/8,56	3,12/7,13
	x-vectors/VAE	4,15/4,97	2,32/3,32	5,42/3,42	3,83/3,88	3,93/3,90
Feminino	*x-vectors*/Gauss	0,76/4,41	0,66/8,54	0,53/11,08	0,00/12,73	0,49/9,19
	x-vectors/VAE	-3,21/4,19	-3,85/2,89	-2,54/4,08	1,09/6,45	-2,13/4,40

Tabela 13 – Comparação entre os desempenhos alcançados pelas abordagens *x-vectors*/Gauss, *x-vectors*/VAE e a modelagem convencional para o caso onde as representações são pós-processadas através do método LDA-LN.

LDA-LN - EER (%) / minDCF x 10					
Gênero	**Modelagem**	**Duração**			
		1-3 s	**3-5 s**	**5-10 s**	**>10 s**
Masculino	*x-vectors*	12,35/7,91	8,25/5,48	7,17/4,91	6,98/4,84
	x-vectors/Gauss	12,12/7,68	**8,25/5,17**	7,10/**4,54**	**6,65/4,36**
	x-vectors/VAE	**11,75/7,59**	8,26/5,49	**6,99**/5,03	6,92/4,96
Feminino	*x-vectors*	12,76/8,36	7,46/6,36	7,23/4,72	6,88/4,73
	x-vectors/Gauss	**11,79**/8,06	**6,67/5,91**	**6,60/4,39**	6,26/**4,38**
	x-vectors/VAE	12,23/**7,88**	6,92/6,26	6,70/4,81	**6,20**/4,87

com a modelagem convencional utilizando o método LDA-LN. Mais uma vez, a abordagem *x-vectors*/Gauss foi a única que melhorou os valores de EER e minDCF em todos os casos de teste. O método *x-vectors*/VAE foi o melhor em um dos casos (locuções curtas masculinas), e também apresentou melhores resultados de EER (ou minDCF) em alguns casos. Em outros casos o resultado foi pior que a abordagem convencional. Comparando as Tabelas 11 e 13, pode-se observar também que a aplicação do pós-processamento melhorou ambas as abordagens consideravelmente. Isso indica que tais métodos realizaram algum tipo de regularização que as abordagens não foram capazes de fazer.

Analisando os ganhos de desempenho alcançados pelas abordagens (Tabela 14), podemos observar que, na média, não houve melhoras nos valores de minDCF para o sistema *x-vectors*/VAE, enquanto que para a abordagem *x-vectors*/Gauss os ganhos foram superiores a 6%. Os maiores ganhos ocorreram para os casos de teste femininos, onde as taxas de EER diminuíram pelos menos 7% para ambos os métodos. Para os testes com homens, os ganhos nas taxas de EER foram aproximadamente 2%.

Levando em consideração os experimentos realizados e os resultados obtidos nessa

Tabela 14 – Comparação entre os ganhos de desempenho proporcionados pelas abordagens *x-vectors*/Gauss, *x-vectors*/VAE, em relação à modelagem convencional, ao pós-processar as representações através do LDA-LN.

Gênero	**LDA-LN - Ganhos de desempenho (%) - EER / minDCF**					
	Modelagem	**Duração**				**Média**
		1-3 s	**3-5 s**	**5-10 s**	**>10 s**	
Masculino	*x-vectors*/Gauss	1,86/2,91	0,00/5,66	0,98/7,54	4,73/9,92	1,89/6,51
	x-vectors/VAE	4,86/4,05	-0,12/-0,18	2,51/-2,44	0,86/-2,48	2,03/-0,26
Feminino	*x-vectors*/Gauss	7,60/3,59	10,59/7,08	8,71/6,99	9,01/7,40	8,98/6,27
	x-vectors/VAE	4,15/5,74	7,24/1,57	7,33/-1,91	9,88/-2,96	7,15/0,61

etapa dos experimentos, as principais observações são:

- Em termos de complexidade, o método proposto por Zhang *et al.* (*x-vectors*/VAE) apresenta aumento de complexidade consideravelmente superior, uma vez que uma segunda DNN com aproximadamente 15 milhões de parâmetros é treinada para a regularização. Nesse sentido, a técnica proposta por Li *et al.* (*x-vectors*/Gauss) não aumenta a complexidade da rede e nenhum modelo adicional é empregado.
- A abordagem *x-vectors*/Gauss foi a que apresentou os melhores resultados, consistentemente melhorando os resultados da modelagem convencional em todos os casos de teste. Tal fato foi observado em ambos os contextos: com e sem pós-processamento das representações.
- Considerável ganho de desempenho foi observado ao aplicar LDA-LN em ambas as abordagens, o que indica um tipo de limitação das abordagens em regularizar as representações convencionais.

4.6 AVALIAÇÃO DOS MÉTODOS PROPOSTOS

A última parte dos experimentos teve por objetivo a avaliação das abordagens propostas neste trabalho e na comparação das mesmas com as abordagens avaliadas anteriormente. Essa parte dos experimentos foi dividida em três etapas:

(i) Avaliação da abordagem proposta para controle das distribuições dos *x-vectors* condicionadas aos locutores, descrita na Seção 3.2.

(ii) Avaliação do método proposto para controle da distribuição *a priori* dos *x-vectors*, descrita na Seção 3.3.

(iii) Avaliação da combinação entre as abordagens para controle conjunto das distribuições subjacentes dos *x-vectors*.

4.6.1 Classificação e *pooling* gaussianos

Na primeira etapa das avaliações dos métodos propostos, focamos na abordagem desenvolvida para controle das distribuições dos *x-vectors* condicionadas aos locutores. Essa abordagem está descrita na Seção 3.2. A abordagem proposta consiste na mudança de duas partes da DNN: na camada de classificação e na camada de *pooling* estatístico. Apesar de a abordagem proposta consistir na mudança conjunta dessas duas partes da rede, nessa parte dos experimentos também comparamos os desempenhos alcançados pela utilização das camadas isoladamente. Referenciamos o sistema utilizando as camada de classificação e *pooling* gaussianos como G-Class e G-Pool, respectivamente, enquanto que o sistema referente à abordagem completa é referenciado como G-Class-Pool.

Como visto na Seção 3.2, a camada de classificação gaussiana possui a mesma quantidade de parâmetros que a camada de saída da DNN convencional. Dessa maneira, não houve mudança na quantidade de parâmetros para o G-Class. Por outro lado, a camada de *pooling* estatístico convencional não possui parâmetros (ela apenas computa os vetores de média e desvio-padrão dos vetores temporais de entrada da camada), enquanto que a camada de *pooling* gaussiano proposta possui um total de $M \times (K + 1)$ parâmetros, onde M é a quantidade de misturas da camada e K é a dimensionalidade dos vetores temporais de entrada. Coincidentemente, a dimensionalidade do vetor de saída da camada é igual à quantidade de parâmetros. A entrada da camada de *pooling* estatístico na abordagem convencional possui dimensionalidade 1536, e a saída da camada, 3072 (ver Tabela 6). Com o intuito de produzir uma dimensionalidade similar à camada convencional e controlar a quantidade de parâmetros total da rede, utilizamos uma camada de *pooling* gaussiano com 32 misturas e dimensionalidade de entrada igual a 64. Para controlar a dimensionalidade de entrada da camada de *pooling*, precisamos modificar a primeira parte da rede, composta pela TDNN. Para produzir uma saída de dimensão 64, substituimos a última camada da TDNN, que possui 1536 nós, por duas camadas, com 512 e 64 nós, respectivamente. Essa mudança produziu um modelo com um total 4,56 milhões de parâmetros, dos quais 3,27 milhões são utilizados, após o treinamento, para a extração das representações.

A avaliação da abordagem foi realizada sem pós-processar as representações geradas e, para comparação, levamos em consideração a modelagem convencional em ambos os casos: sem pós-processamento e com a utilização do LDA-LN. Os resultados obtidos são mostrados na Tabela 15. Observando os desempenhos alcançados pelos sistemas G-Class e G-Pool, percebemos que, em alguns casos, a utilização das camadas de classificação e *pooling* gaussianos melhoram o desempenho da modelagem convencional com *x-vectors* sem pós-processamento. Porém, comparando o sistema G-Class com *x-vectors* processados através do LDA-LN, vemos que em quase todos os casos o desempenho é pior. Para o sistema G-Pool, apenas em alguns casos ele se mostrou superior. Dessa maneira, podemos concluir que pouco ganho é alcançado ao controlar apenas uma parte da rede. Por outro lado, o controle conjunto da camada de classificação e de *pooling* alcançou uma melhora de desempenho significativa. A Tabela 16 mostra os ganhos de desempenho alcançados pelas abordagens quando comparados com os *x-vectors* com aplicação da técnica LDA-LN.

Como podemos observar, na média, nenhum ganho de desempenho é alcançado pelo sistema G-Class. Já com a utilização do *pooling* gaussiano, observou-se uma melhora nos valores de minDCF para as locuções longas, em ambos os gêneros. Na média, o ganho de desempenho ocorreu apenas para os casos de teste femininos. Já o sistema G-Class-Pool, que realiza o controle de ambas as partes de classificação e *pooling* da rede, ganhos de desempenho foram observados em quase todos os testes. Na média, aproximadamente 4% e 6% de melhora foram observados nos valores de minDCF, e 3% e 1% nas taxas de EER. Os melhores ganhos foram observados nas locuções longas, com aproximadamente 17%

Tabela 15 – Desempenhos alcançados pela abordagem proposta para controle das distribuições dos *x-vectors* condicionadas aos locutores. A abordagem é composta pelas camadas de classificação e *pooling* gaussianos, G-Class e G-Pool, respectivamente. A abordagem foi comparada sem a utilização de técnicas de pós-processamento, e levou-se em consideração a utilização isolada das camadas e a utilização conjunta (G-Class-Pool).

EER (%) / minDCF x 10					
Gênero	**Modelagem**	**Duração**			
		1-3 s	**3-5 s**	**5-10 s**	**>10 s**
Masculino	*x-vectors*	12,53/8,66	8,20/7,22	7,57/7,02	7,32/6,19
	x-vectors/LDA-LN	12,35/7,91	8,25/5,48	7,17/4,91	6,98/4,84
	G-Class	12,34/8,33	8,12/5,75	7,40/5,65	7,25/5,94
	G-Pool	**12,11**/8,02	7,97/5,81	7,15/5,21	6,99/4,53
	G-Class-Pool	12,21/**7,78**	**7,75/5,58**	**6,90/4,83**	**6,77/4,04**
Feminino	*x-vectors*	13,09/8,84	7,54/7,26	7,49/6,86	7,31/6,05
	x-vectors/LDA-LN	12,76/8,36	7,46/6,36	7,23/4,72	6,88/4,73
	G-Class	12,97/8,44	8,01/6,62	7,31/5,46	6,93/4,80
	G-Pool	12,82/8,30	**7,52**/6,29	**7,17/4,58**	6,73/4,22
	G-Class-Pool	**12,74/8,26**	7,56/**6,06**	7,21/4,66	**6,44/3,87**

Tabela 16 – Ganhos de desempenho das abordagens G-Class, G-Pool e G-Class-Pool em relação à modelagem convencional com *x-vectors* pós-processados utilizando LDA-LN.

Gênero	**Ganhos de desempenho (%) - EER / minDCF**					
	Modelagem	**Duração**				**Média**
		1-3 s	**3-5 s**	**5-10 s**	**>10 s**	
Masculino	G-Class	0,08/-5,31	1,58/-4,93	-3,21/-15,07	-3,87/-22,73	-1,36/-12,01
	G-Pool	1,94/-1,39	3,39/-6,02	0,28/-6,11	-0,14/6,40	1,37/-1,78
	G-Class-Pool	1,13/1,64	6,06/-1,82	3,77/1,63	3,01/16,53	3,49/4,50
Feminino	G-Class	-1,65/-0,96	-7,37/-4,09	-1,11/-15,68	-0,73/-1,48	-2,72/-5,55
	G-Pool	-0,47/0,72	-0,80/1,10	0,83/2,97	2,18/10,78	0,44/3,89
	G-Class-Pool	0,16/1,20	-1,34/4,72	0,28/1,27	6,40/18,18	1,38/6,34

de ganho nos valores de minDCF.

4.6.2 Controle sobre a distribuição dos *x-vectors*

Na segunda etapa dos experimentos envolvendo os métodos propostos, avaliamos a abordagem desenvolvida para o controle sobre a distribuição *a priori* dos *x-vectors*. Essa abordagem é descrita na Seção 3.3 e consiste na adição de um termo de regularização para função de custo utilizada para o treinamento da DNN. Tal método é baseado na abordagem variacional e a função de regularização computa uma medida de divergência

não paramétrica, chamada de máxima discrepância-média (*Maximum Mean Discrepancy* - MMD), entre os *x-vectors* e uma amostra de uma distribuição desejada.

Neste trabalho, consideramos a distribuição normal padronizada como a desejada para regularização dos *x-vectors* e referenciamos essa abordagem como G-MMD. Como mostrado anteriormente, o fato de a função utilizada no termo de regularização realizar um teste não paramétrico entre as amostras das distribuições gerada e desejada, nenhuma mudança precisou ser realizada no restante da rede e nenhum novo parâmetro foi adicionado. Dessa maneira, a quantidade de parâmetros do modelo proposto é a mesma da modelagem convencional.

O termo de regularização proposto é parametrizado pelo fator de importância e pelo tamanho da amostra da distribuição desejada, que é passada para a rede como uma nova entrada. Como mencionado anteriormente, as redes foram treinadas levando em consideração uma condição de parada baseada na acurácia de classificação. Em testes preliminares, observamos que o valor atingido pela função MMD (Equação 3.19) não sofre influência do fator de importância[3]. Por essa razão, fixamos esse valor em 500, que é o mesmo utilizado pelo MMD-VAE (ZHAO; SONG; ERMON, 2017). Além disso, como é comumente feito nos treinamentos de VAEs (WANG; LI; WANG, 2019; ZHANG; LI; WANG, 2019), em cada *batch*, uma amostra aleatória composta por 100 observações da distribuição desejada é utilizada para a regularização.

Os desempenhos alcançados pela abordagem proposta estão apresentados na Tabela 17. Realizamos a avaliação sem empregar nenhuma técnica de pós-processamento às representações geradas, mas comparamos a abordagem levando em consideração ambos os *baselines*, com e sem pós-processamento pelo método LDA-LN. A abordagem proposta alcançou desempenhos superiores que ambos os *baselines*, com taxas de EER e valores de minDCF menores em todos os casos de teste. A Tabela 18 apresenta os ganhos de desempenho.

Os maiores ganhos de desempenho ocorreram nos valores de minDCF, com média de aproximadamente 8% para os testes com homens e de aproximadamente 13% com mulheres. Além disso, os maiores ganhos com relação às taxas de EER ocorreram também com as mulheres, com ganhos de, em média, 5,30%. Para locuções longas, a melhora foi ainda maior, com ganhos de 11% e 21% para as taxas de EER e minDCF, respectivamente.

Entre as abordagens propostas, os melhores resultados foram alcançados através do controle sobre a distribuição *a priori* dos *x-vectors*. Como mencionado anteriormente, essa abordagem tem como objetivo um controle indireto sobre o espaço das variabilidades intra-locutor, algo que realmente falta tanto na abordagem proposta anteriormente (G-Class-Pool) como nos outros métodos da literatura, avaliados na Seção 4.5.

[3] De fato, ao avaliar o MMD-VAE, da família dos InfoVAEs, os autores afirmam que basta o fator de importância ser "suficientemente"alto (ZHAO; SONG; ERMON, 2017).

Tabela 17 – Desempenhos alcançados pela abordagem proposta para controle da distribuição *a priori* dos *x-vectors* (G-MMD). A abordagem foi avaliada sem empregar técnicas de pós-processamento e comparada com ambos os *baselines*, com e sem pós-processamento via LDA-LN.

EER (%) / minDCF x 10					
Gênero	**Modelagem**	**Duração**			
		1-3 s	**3-5 s**	**5-10 s**	**>10 s**
Masculino	*x-vectors*	12,53/8,66	8,20/7,22	7,57/7,02	7,32/6,19
	x-vectors/LDA-LN	12,35/7,91	8,25/5,48	7,17/4,91	6,98/4,84
	G-MMD	**11,84/7,83**	**8,18/5,32**	**7,05/4,50**	**6,58/3,86**
Feminino	*x-vectors*	13,09/8,84	7,54/7,26	7,49/6,86	7,31/6,05
	x-vectors/LDA-LN	12,76/8,36	7,46/6,36	7,23/4,72	6,88/4,73
	G-MMD	**12,06/7,87**	**7,28/5,35**	**7,11/4,27**	**6,08/3,72**

Tabela 18 – Ganhos de desempenho da abordagem proposta G-MMD em relação à modelagem convencional com *x-vectors* pós-processados utilizando LDA-LN.

G-MMD - Ganhos de desempenho (%) - EER / minDCF					
Gênero	**Duração**				**Média**
	1-3 s	**3-5 s**	**5-10 s**	**>10 s**	
Masculino	4,13/1,01	0,85/2,92	1,67/8,35	5,73/20,25	3,10/8,13
Feminino	5,49/5,86	2,41/15,88	1,66/9,53	11,63/21,35	5,30/13,16

4.6.3 Combinação entre as abordagens propostas

Por fim, realizamos a avaliação da combinação das abordagens propostas neste trabalho. Nessa abordagem, tanto o método G-Class-Pool quanto o método G-MMD são empregados durante o treinamento da rede. Isto é, são empregados controles tanto sobre a distribuição *a priori* dos *x-vectors* quanto sobre as distribuições condicionadas aos locutores, de maneira conjunta. Os mesmos parâmetros fixados nos experimentos anteriores foram utilizados nessa etapa. Referenciamos essa abordagem como G-MMD-Class-Pool. Mais uma vez, para a avaliação da abordagem, nenhum método de pós-processamento foi aplicado às representações. A Tabela 19 apresenta os desempenhos alcançados pela abordagem conjunta, comparando-os com os desempenhos alcançados pela modelagem convencional com *x-vectors* processados pela técnica LDA-LN, e também com os métodos avaliados anteriormente.

Apesar de ambas as abordagens G-MMD e G-Class-Pool auxiliarem na geração de *x-vectors* mais apropriados, elas foram desenvolvidas com objetivos distintos e, por essa razão, impactam o espaço dos *x-vectors* de maneira diferente. Porém, como visto na Seção 2.4.5, a modelagem G-PLDA assume que tanto a componente do locutor quanto a componente das variabilidades intra-locutor seguem distribuições normais. Dessa maneira,

Tabela 19 – Desempenhos alcançados pela combinação entre as abordagens propostas para controle sobre as distribuições subjacentes dos *x-vectors*, composta pelas abordagens G-MMD e G-Class-Pool, sendo empregas conjuntamente durante o treinamento da DNN.

EER (%) / minDCF x 10					
Gênero	**Modelagem**	**Duração**			
		1-3 s	**3-5 s**	**5-10 s**	**>10 s**
Masculino	*x-vectors*/LDA-LN	12,35/7,91	8,25/5,48	7,17/4,91	6,98/4,84
	G-Class-Pool	12,21/7,78	7,75/5,58	6,90/4,83	6,77/4,04
	G-MMD	11.84/7.83	8.18/5.32	7.05/4.50	6.58/3.86
	G-MMD-Class-Pool	**10,43/7,35**	**7,66/5,13**	**6,60/4,27**	**6,25/3,68**
Feminino	*x-vectors*/LDA-LN	12,76/8,36	7,46/6,36	7,23/4,72	6,88/4,73
	G-Class-Pool	12,74/8,26	7,56/6,06	7,21/4,66	6,44/3,87
	G-MMD	12.06/7.87	7.28/5.35	7.11/4.27	6.08/3.72
	G-MMD-Class-Pool	**11,15/7,33**	**7,19/4,91**	**6,31/4,04**	**5,30/3,59**

apesar de as abordagens propostas atuarem sobre as distribuições das componentes de maneira separada (a abordagem G-Class-Pool atua mais na componente do locutor, enquanto que a abordagem G-MMD atua mais na componente de variabilidade), elas se completam sob a ótica imposta pela modelagem G-PLDA. A combinação entre as abordagens, de maneira que os controles sobre os diferentes aspectos das representações sejam realizados simultaneamente, resulta em um ganho de desempenho ao sistema. De fato, a combinação dos métodos apresentou desempenhos superiores ao G-MMD em todos os casos de teste, mostrando que ela foi capaz de gerar representações mais apropriadas.

A Tabela 20 apresenta, de maneira geral, os ganhos de desempenho de todas as abordagens avaliadas neste trabalho. Os ganhos foram calculados levando em consideração a modelagem convencional com *x-vectors* sendo pós-processados pela técnica LDA-LN. Além das técnicas propostas neste trabalho, também estão as técnicas da literatura, avaliadas na Seção 4.5.

Na média, todas as técnicas contribuem, em alguma escala, para a modelagem convencional. De maneira geral, as abordagens propostas G-MMD e G-Class-Pool, em conjunto com o método *x-vectors*/Gauss, consistentemente apresentam ganhos de desempenho em relação aos *x-vectors*. Porém, ao combinarmos as duas abordagens propostas neste trabalho para realização simultânea dos dois tipos de controle, os ganhos de desempenho se mostraram superiores aos demais. Com exceção de um caso de teste, onde a abordagem *x-vectors*/Gauss apresentou melhor ganho na taxa de EER, o método G-MMD-Class-Pool supera os demais, apresentando ganhos de desempenho, em média, superiores a 10% tanto para as taxas de EER quanto os valores de minDCF.

Tabela 20 – Ganhos de desempenho das abordagens avaliadas neste trabalho. Além dos métodos propostos, são também apresentados os ganhos alcançados pelos métodos da literatura. Os ganhos foram calculados levando em consideração a modelagem convencional com *x-vectors* pós-processados utilizando LDA-LN.

Gênero	**Ganhos de desempenho (%) - EER / minDCF**					
	Modelagem	**Duração**				**Média**
		1-3 s	**3-5 s**	**5-10 s**	**>10 s**	
Masculino	*x-vectors*/Gauss	1,86/2,91	0,00/5,66	0,98/7,54	4,73/9,92	1,89/6,51
	x-vectors/VAE	4,86/4,05	0,12/-0,18	2,51/-2,44	0,86/-2,48	2,09/-0,26
	G-Class-Pool	1,13/1,64	6,06/-1,82	3,77/1,63	3,01/16,53	3,49/4,50
	G-MMD	4,13/1,01	0,85/2,92	1,67/8,35	5,73/20,25	3,10/8,13
	G-MMD-Class-Pool	**15,55/7,08**	**7,15/6,39**	**7,95/13,03**	**10,46/23,97**	**10,28/12,62**
Feminino	*x-vectors*/Gauss	7,60/3,59	**10,59**/7,08	8,71/6,99	9,01/7,40	8,98/6,27
	x-vectors/VAE	4,15/5,74	7,24/1,57	7,33/-1,91	9,88/-2,96	7,15/0,61
	G-Class-Pool	0,16/1,20	-1,34/4,72	0,28/1,27	6,40/18,18	1,38/6,34
	G-MMD	5,49/5,86	2,41/15,88	1,66/9,53	11,63/21,35	5,30/13,16
	G-MMD-Class-Pool	**12,62/12,32**	3,62/**22,80**	**12,72/14,41**	**22,97/24,10**	**12,98/18,41**

4.6.4 Visualização das representações

Com o objetivo de analisar o impacto das abordagens propostas sobre o espaço gerado pelas DNNs, uma inspeção visual sobre uma amostra dos dados foi realizada. Através da técnica t-SNE (MAATEN; HINTON, 2008), as representações correspondentes a um conjunto de locuções foram projetadas no plano para possibilitar a visualização. Essa técnica é amplamente utilizada para fins de visualização de dados de alta dimensionalidade, inclusive aqueles gerados por DNNs (DONAHUE et al., 2014; AYTAR; VONDRICK; TORRALBA, 2016). A técnica, em si, otimiza as relações de vizinhança entre os dados observados de maneira não-supervisionada. Dessa maneira, mesmo estando no plano, consegue-se analisar aspectos como separação entre as classes dos locutores ou dispersão das amostras de um mesmo locutor.

Para realização da análise, escolhemos aleatoriamente 20 locutores (10 de cada gênero) do conjunto de teste. Isto é, escolhemos amostras de dados que não foram utilizadas para o treinamento das redes. Além disso, todas as amostras de teste foram utilizadas, resultando em aproximadamente 90 amostras para cada locutor e 1883 amostras no total. Além da representação convencional dos *x-vectors*, levamos em consideração as representações geradas pelas duas abordagens propostas (G-Class-Pool e G-MMD) e a combinação entre elas (G-MMD-Class-Pool). Cada conjunto de vetores, correspondente a cada uma das técnicas, foi mapeado para o espaço bidimensional. A Figura 27 apresenta as representações bidimensionais geradas por cada uma das técnicas. Uma cor diferente foi atribuída a cada um dos locutores, onde os homens estão descritos através de círculos e as mulheres através de cruzes.

Primeiramente, pode-se observar que os locutores de mesmo gênero tendem a estar

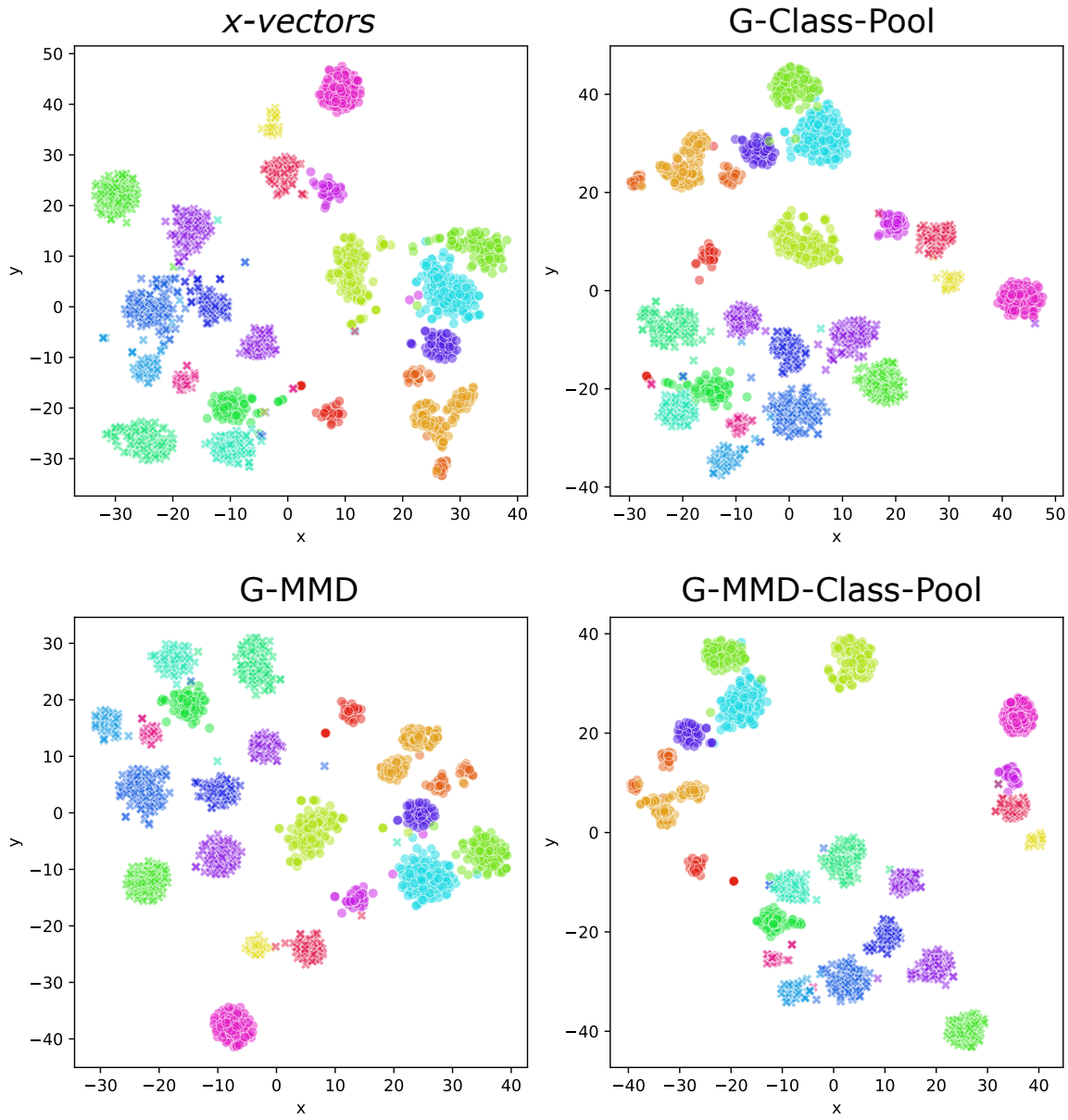

Figura 27 – Visualização das projeções bidimensionais das representações geradas pela abordagem convencional com *x-vectors* e pelas abordagens propostas neste trabalho. Foram consideradas representações extraídas das locuções geradas por 20 locutores do conjunto de teste, distribuídos igualmente entre os gêneros masculino (círculos) e feminino (cruzes), e categorizados por diferentes cores. A projeção entre o espaço original e a representação bidimensional foi realizada através da técnica t-SNE (MAATEN; HINTON, 2008), e x e y são as duas dimensões resultantes da projeção.

mais próximos entre si. Além disso, constata-se que, de fato, os *x-vectors* carregam informações discriminativas com respeito aos locutores, mesmo sem a aplicação de técnicas de pós-processamento. É possível atribuir a uma determinada partição do espaço uma categoria correspondente a um único locutor predominante, mesmo que ocorram erros em alguns pontos. Uma determinada partição do espaço na qual pode-se atribuir uma única

classe é geralmente referenciada como agrupamento ou *cluster*. No nosso caso, as representações podem ser analisadas através da forma como os *clusters* dos locutores ficam dispostos no espaço. Um alto poder discriminativo gera representações onde os *clusters* de diferentes locutores permanecem distantes uns dos outros. Porém, como apresentado no Capítulo 3, a qualidade do espaço gerado pelas abordagens vai além do poder discriminativo e considera também a maneira como os vetores se distribuem. Sob esse aspecto, além de analisar quão separadas estão as diferentes classes de locutores, podemos verificar quão dispersas estão as representações geradas para um mesmo locutor.

Nesse sentido, podemos verificar que a representação convencional dos *x-vectors* gera, para alguns locutores, amostras que se dispersam de maneira assimétrica, diferentemente do desejado, que seria descrito por uma distribuição normal ou, da maneira como os dados estão descritos na imagem, concentradas nas vizinhanças de um determinado centro. Podemos encontrar vetores que se aproximam mais de *clusters* associados a outros locutores, o que, muito provavelmente, resultam em erros de decisão na modelagem G-PLDA. Comparando as abordagens G-Class-Pool e G-MMD com o espaço original, conseguimos perceber melhoras significativas em alguns locutores. As representações tendem a se concentrar mais, melhorando a qualidade dos *clusters* e delimitando melhor a separação entre diferentes locutores. Para a modelagem G-MMD, podemos observar uma melhora considerável, por exemplo, nos *clusters* de três locutoras, descritas por cores azuis em cruzes. A separação dessas locutoras se torna bem mais evidente quando comparamos com o espaço original. Em um determinado locutor masculino (laranja mais escuro), vemos que as representações geradas por ele estão dispostas em dois *clusters*, separados pelo *cluster* de um outro locutor. Essa disposição continua presente na abordagem G-Class-Pool, mas na abordagem G-MMD esses *clusters* ficam mais próximos entre si e distantes do outro locutor. Já na abordagem conjunta, G-MMD-Class-Pool, temos a melhor disposição de *clusters* do experimento. A quantidade de erros (vetores mais próximos de *clusters* de outros locutores) é mais reduzida e a distribuição dos vetores nos *clusters* aparenta ser mais bem comportada, isto é, parecem estar dispostas na vizinhança do centro do *cluster*, com menos amostras dispersas. Além disso, ele foi capaz de aumentar a discrepância entre os diferentes locutores, o que sugere um aumento de poder de discriminação. Enquanto nas outras abordagens os espaços entre os grupos (em branco) é reduzido, nessa nova representação esses espaços são maiores. Tais espaços podem estar ocupados por outros tipos de informações, talvez associadas a outros locutores que não foram vistos nesse experimento, por exemplo.

Apesar de a análise visual ter sido realizada em uma escala bem reduzida (20 locutores de um total de 1000), os locutores foram escolhidos aleatoriamente e apenas locuções de teste foram consideradas, isto é, apenas amostras que não foram observadas durante os treinamentos dos modelos. Isso possibilitou uma análise da qualidade das representações que foram geradas pelas abordagens. O mais interessante é que, de fato, as melhorias nos

espaços, constatadas pela análise visual, são confirmadas pelos desempenhos observados nos experimentos anteriores. Isto é, percebe-se que os melhores desempenhos foram alcançados pela técnica que apresentou as melhores visualizações (G-MMD-Class-Pool), com as amostras dos locutores dispostas nas vizinhanças dos *clusters* correspondentes e com melhor separação entre os *clusters* correspondentes a diferentes locutores.

4.6.5 Considerações finais

A última parte dos experimentos teve por objetivo a análise dos desempenhos alcançados pelas abordagens propostas neste trabalho e na comparação das mesmas com as outras abordagens propostas na literatura. Diante dos resultados expostos, pode-se salientar:

- Ambas as abordagens propostas para controle das distribuições subjacentes aos *x-vectors*, aplicadas isoladamente, resultaram em ganhos de desempenho quando comparados com a modelagem convencional, mesmo utilizando técnicas de pós-processamento como o LDA-LN.

- A primeira técnica (G-Class-Pool), desenvolvida para controle das distribuições condicionadas aos locutores, é composta pelas camadas de classificação e *pooling* gaussianos, que, ao serem utilizadas isoladamente, trouxeram ganhos de desempenho limitados à modelagem convencional. Dentre elas, a utilização da camada de *pooling* proposta foi a que alcançou os melhores desempenhos. Ao utilizarmos a abordagem completa, vimos que modificar ambas as partes da rede resultou em um ganho de desempenho razoável, comparável às outras técnicas da literatura (*x-vectors*/Gauss e *x-vectors*/VAE). Os maiores ganhos de desempenho foram nos valores de minDCF para locuções longas, ultrapassando 15% de ganho em ambos os gêneros.

- Ao analisarmos as abordagens separadamente, a técnica G-MMD foi a que apresentou os melhores resultados entre todas as técnicas avaliadas. Esse método foi proposto para controle da distribuição *a priori* das representações e melhoras de desempenho foram observados nos valores de EER e minDCF em todos os casos de teste. Para os valores de minDCF em locuções longas, o ganho de desempenho foi superior a 20% em ambos os gêneros.

- Uma importante características das abordagens propostas neste trabalho é que elas foram desenvolvidas para o controle de diferentes aspectos do espaço das representações e, além disso, foram desenvolvidas para que pudessem ser aplicadas em conjunto. Enquanto que a abordagem G-Class-Pool consiste nas camadas de classificação e *pooling* gaussianos, a técnica G-MMD é composta por um termo de regularização, baseado na abordagem variacional, adicionado à função objetivo utilizada no treinamento da rede. As técnicas são compatíveis entre si e são combinadas para

o controle conjunto das distribuições *a priori* e condicionadas aos locutores dos *x-vectors*. A abordagem completa (G-MMD-Class-Pool) foi a que alcançou os melhores resultados nos experimentos realizados, superando, em média, os resultados obtidos pelas outras técnicas, levando em consideração os valores de EER e minDCF. Em média, os ganhos de desempenho foram superiores a 10% em ambos os gêneros. Bons ganhos de desempenho também foram observados nas locuções curtas, em ambos os gêneros, com melhoras de aproximadamente 15% e 12% nas taxas de EER, e de aproximadamente 7% e 12% para os valores de minDCF, para homens e mulheres, respectivamente.

- Por fim, realizamos uma análise visual dos espaços das representações gerados pelas abordagens propostas. Apesar de a análise ter sido realizada em uma escala menor, com 20 locutores, melhoras significativas foram observadas pelas técnicas no que diz respeito à qualidade dos *clusters* gerados para os locutores, levando em consideração a separação dos grupos e como as amostras estão dispersas em cada um deles. A abordagem principal G-MMD-Class-Pool foi capaz de separar bem os diferentes grupos de locutores e diminuir a variabilidade existente nos vetores produzidos por um mesmo locutor.

5 CONCLUSÕES

Este trabalho focou no desenvolvimento de sistemas de verificação de locutores independentes de texto. A definição do problema e os desafios envolvidos no desenvolvimento desses sistemas foram descritos no Capítulo 1. A dificuldade no desenvolvimento de sistemas dessa natureza provém dos mais diversos fatores que podem influenciar na geração dos sinais de voz. O desafio consiste em extrair das locuções representações robustas, capazes de distinguir os locutores diante dessas incompatibilidades. O foco deste trabalho consistiu no desenvolvimento dessas representações.

No Capítulo 2, foram descritos os métodos mais bem sucedidos propostos através dos anos para esse propósito. A representação mais predominantemente utilizada consistiu nos chamados vetores-identidade (*i-vectors*). Nessa modelagem, um modelo universal de fundo (*Universal Background Model* - UBM) probabilístico é primeiramente estimado utilizando características de tempo curto, como os Coeficientes Mel-cepstrais (*Mel-Frequency Cepstral Coefficientss* - MFCCs). Os *i-vectors* são então definidos pela projeção das estatísticas do UBM - levando em consideração os MFCCs extraídos de uma determinada locução - para um espaço de dimensionalidade reduzida. Essa projeção é realizada através de uma decomposição não supervisionada do espaço, seguindo a abordagem de análise fatorial. Geralmente, um pós-processamento dos *i-vectors* é realizado através de operações como normalização de comprimento (*Length Normalization* - LN) e análise de discriminantes lineares (*Linear Discriminant Analysis* - LDA), com o intuito de aumentar o poder discriminativo das representações. Nesse tipo de abordagem, uma autenticação é realizada ao decidir se dois *i-vectors* foram produzidos pelo mesmo locutor ou não. Tal decisão é realizada através de um modelo de análise probabilística de discriminante linear (*Probabilistic Linear Discriminant Analysis* - PLDA). Mais precisamente, um tipo particular de PLDA é utilizado, assumindo que a componente do locutor segue uma distribuição normal. Tal modelo é chamado de PLDA gaussiano (*Gaussian Probabilistic Linear Discriminant Analysis* - G-PLDA).

Nos últimos anos, diversas abordagens utilizando redes neurais profundas (*Deep Neural Networkss* - DNNs) vêm sendo propostas para a geração de novas representações mais robustas. Dentre tais representações, a que mais se destacou consiste nos chamados *x-vectors*, onde uma DNN supervisionada é treinada para diferenciar locuções de diversos locutores. Nessa abordagem, as locuções também são descritas através dos coeficientes MFCCs e uma representação vetorial para a locução é gerada através de uma camada de *pool* estatístico, que agrega os diversos vetores da locução, computando seus vetores de média e desvio-padrão. A partir dessa camada, a rede discrimina locuções inteiras utilizando as classes dos locutores que as produziram. Assim como os *i-vectors*, a autenticação é realizada através do modelo G-PLDA e a acurácia do sistema também depende

de métodos de pós-processamento como o LDA e o LN. Porém, enquanto que a etapa de pós-processamento dos *i-vectors* possui o objetivo de tornar as representações mais discriminantes, para os *x-vectors*, o objetivo dessa etapa é o de regularização. Isso ocorre porque a modelagem G-PLDA assume que as probabilidades condicionais e *a priori* dos vetores dos locutores seguem uma distribuição normal. Como os *x-vectors* são treinados apenas com o objetivo de distinguir locuções de diferentes locutores, nenhuma imposição é aplicada sobre as distribuições condicionais dos vetores. Os métodos de pós-processamento então mapeiam os vetores para um espaço mais adequado ao G-PLDA, o que resulta em um ganho de desempenho. Mesmo com esse ganho de desempenho, tais métodos não são completamente adequados, uma vez que eles não impõem restrições às distribuições geradas.

Nos últimos anos, alguns métodos foram desenvolvidos para a geração de *x-vectors* seguindo a distribuição normal. Li *et al.* propuseram a adição de um termo de regularização à função de custo da DNN que minimiza a norma da diferença entre os *x-vectors* e o peso da camada de saída, associado ao locutor correspondente (LI et al., 2019). Referenciamos esse trabalho como *x-vectors*/Gauss. Já em (ZHANG; LI; WANG, 2019), um segundo modelo DNN foi proposto para projetar *x-vectors* já treinados em um novo espaço mais compacto e com distribuição normal padronizada. O modelo consiste de um Auto-codificador Variacional (*Variational Autoencoder* - VAE), que, assim como os auto-codificadores (*autoencoders*), é um modelo não-supervisionado treinado para reconstruir a própria entrada, gerando para isso uma representação intermediária. VAEs possuem uma função de regularização variacional definida pela divergência entre as distribuições das variáveis intermediárias e uma distribuição paramétrica desejada (nesse caso, uma distribuição gaussiana padrão). Esse sistema foi referenciado como *x-vectors*/VAE.

Neste trabalho, propusemos um conjunto de abordagens capazes de melhorar a qualidade dos *x-vectors* tornando-os mais apropriados para a modelagem G-PLDA. Essas abordagens estão descritas no Capítulo 3. Propusemos duas abordagens, desenvolvidas para o controle das distribuições condicionadas aos locutores e da distribuição *a priori* dos *x-vectors*. Enquanto o controle sobre as distribuições condicionadas aos locutores age sobre como as componentes dos locutores se comportam no espaço gerado pela DNN, o controle sobre a distribuição *a priori* realiza um controle indireto sobre a componente de variabilidade intra-locutor.

Para controle das distribuições condicionadas aos locutores, propusemos mudanças nas camadas de classificação e *pooling* da rede (Seção 3.2). Tais camadas são compostas por nós definidos por funções de base radial (RBFs), que realizam operações mais apropriadas para a geração de espaços onde as informações estão dispostas sob distribuições gaussianas. Referenciamos as abordagens correspondentes à utilização dessas camadas como G-Class e G-Pool, e a abordagem completa, como G-Class-Pool. Enquanto a camada de classificação modela as componentes referentes aos locutores como gaussianas, a camada de *pooling*

gaussiano modela o espaço das representações temporais geradas pela primeira parte da rede como uma mistura de RBFs. Nesse caso, as representações temporais de uma determinada locução são agregadas através das estatísticas de ordem zero e primeira ordem das misturas, similarmente à descrição vetorial utilizada na decomposição da modelagem com *i-vectors*.

Já para o controle sobre a distribuição *a priori* dos *x-vectors*, propusemos a adição de um termo de regularização, baseado na abordagem variacional, que calcula a divergência entre a distribuição dos vetores gerados pela rede e uma distribuição desejada. A função de regularização é definida pela Máxima Discrepância-Média (*Maximum Mean Discrepancy* - MMD), que realiza um teste não paramétrico entre amostras de duas distribuições quaisquer. Uma amostra da distribuição desejada (no nosso caso, a distribuição normal padronizada) é então apresentada à rede como uma nova entrada e o termo de regularização então minimiza a divergência entre as distribuições. Esse método está descrito na Seção 3.3 e foi referenciado como G-MMD.

Como pode ser observado, as abordagens foram desenvolvidas para propósitos diferentes e elas atuam em aspectos diferentes do espaço de representações gerado pela DNN. Além disso, elas foram desenvolvidas com o intuito de poderem ser utilizadas juntas. Dessa maneira, unindo ambas as abordagens, conseguimos realizar o treinamento da rede, empregando um controle simultâneo das distribuições subjacentes dos *x-vectors*. Esse sistema final é referenciado como G-MMD-Class-Pool.

Os experimentos realizados estão descritos no Capítulo 4 e foram conduzidos utilizando a base de dados *Fisher English Training* (CIERI et al., 2005), que é composta por chamadas telefônicas realizadas por milhares de indivíduos de ambos os gêneros. Na metodologia empregada, 2000 locutores de cada gênero foram utilizados para treinamento dos modelos, enquanto que as locuções geradas por 500 locutores de cada gênero foram utilizadas para avaliação. Ao todo, foram levadas em consideração oito condições de teste, dependendo do gênero do locutor e a duração da locução de teste. Para comparação dos sistemas utilizamos a taxa de erros iguais (*Equal Error Rate* - EER) e a métrica utilizada na última competição da NIST (OMID; CRAIG, 2019), definida pelo valor mínimo da chamada de função de custo de detecção (*Detection Cost Function* - DCF), o minDCF. Os experimentos foram divididos em três partes, com objetivos específicos.

Na primeira parte dos experimentos (Seção 4.4), realizamos a comparação entre as modelagens convencionais *i-vectors* e *x-vectors*. Nessa fase dos experimentos ficou evidenciada a robustez dos *x-vectors*, que apresentou melhores resultados em todos os casos de teste, nos dois contextos: com e sem utilização de métodos de pós-processamento antes da modelagem através do G-PLDA. As taxas de EER e valores de minDCF foram expressivamente menores para os *x-vectors*. Além disso, também observamos que a composição entre as técnicas LDA e LN conduz aos melhores desempenhos dos sistemas.

Já na segunda parte dos experimentos (Seção 4.5), realizamos a comparação entre

os métodos propostos para controle dos *x-vectors*, presentes na literatura, os sistemas *x-vectors*/Gauss e *x-vectors*/VAE. Nessa parte dos experimentos, destacamos que a utilização de um segundo modelo (VAE) para regularização do espaço dos *x-vectors* aumenta consideravelmente a complexidade do sistema. Além disso, observamos que a abordagem *x-vectors*/Gauss foi a que apresentou os melhores resultados, consistentemente melhorando os resultados da modelagem convencional em todos os casos de teste. Tal fato foi observado em ambos os contextos: com e sem pós-processamento das representações. Porém, ainda vimos um considerável ganho de desempenho ao aplicar a técnica LDA-LN para pós-processamento das representações geradas, o que indica um tipo de limitação das abordagens em regularizar as representações.

Na terceira fase dos experimentos (Seção 4.6), avaliamos as abordagens propostas neste trabalho e as comparamos com a modelagem convencional dos *x-vectors* e com as abordagens presentes na literatura. Observamos que ambas as abordagens propostas (G-Class-Pool e G-MMD), aplicadas isoladamente, resultaram em ganhos de desempenho quando comparados com a modelagem convencional, mesmo utilizando técnicas de pós-processamento como o LDA-LN. Ao analisar a técnica G-Class-Pool pudemos observar que, ao utilizar as mudanças nas camadas de classificação e *pooling* de maneira isolada, ganhos limitados de desempenho foram alcançados. Dentre elas, a camada de *pooling* gaussiano foi a que alcançou os melhores desempenhos. Já para a abordagem completa, ganhos de desempenho razoáveis, comparáveis às outras técnicas da literatura (*x-vectors*/Gauss e *x-vectors*/VAE) foram alcançados, em especial para os valores de minDCF para locuções longas, ultrapassando 15% de ganho em ambos os gêneros.

Ao analisarmos as abordagens separadamente, a técnica G-MMD foi a que apresentou os melhores resultados entre todas as técnicas avaliadas neste trabalho. Em comparação com os *x-vectors* pós-processados através do LDA-LN, melhores resultados foram observados em todos os casos de teste para ambas as métricas de desempenho. Os ganhos para os valores de minDCF em locuções longas foram superiores a 20% em ambos os gêneros. Já a abordagem completa, G-MMD-Class-Pool, foi a que apresentou os melhores resultados, superando, em média, os desempenhos obtidos pelas outras técnicas, levando em consideração os valores de EER e minDCF. Em média, os ganhos de desempenho foram superiores a 10% em ambos os gêneros. Consideráveis ganhos de desempenho também foram observados nas locuções curtas, em ambos os gêneros, com melhoras de aproximadamente 15% e 12% nas taxas de EER, e de aproximadamente 7% e 12% para os valores de minDCF, para homens e mulheres, respectivamente.

Por fim, realizamos uma análise visual dos espaços das representações gerados pelas abordagens propostas (Seção 4.6.4) e comparamos com o espaço gerado pela modelagem convencional. A partir da análise dos *clusters* associados aos locutores, pudemos analisar como as abordagens separam os vetores gerados por diferentes locutores e como as representações dos locutores se distribuem entre si. Neste sentido, pudemos fixar algumas

características de qualidade que dizem respeito à separação entre os *clusters* e a maneira como os vetores se comportam dentro do *cluster*. Para esse propósito utilizamos os vetores extraídos das locuções de teste de 20 locutores, 10 de cada gênero. Pudemos observar o ganho de qualidade resultante das abordagens propostas neste trabalho, que provocam uma melhor distinção entre os *clusters* e uma melhor distribuição dos vetores, o que corrobora para os resultados observados nas fases anteriores dos experimentos.

5.1 DIRECIONAMENTOS FUTUROS

Como visto anteriormente, a abordagem proposta para controle da distribuição *a priori* das representações foi a que apresentou os melhores desempenhos, utilizando a modelagem G-PLDA. Tal abordagem foi desenvolvida como um controle indireto sobre o espaço correspondente às variabilidades intra-locutor. Vimos também que, ao utilizar uma DNN supervisionada, treinada apenas para classificar as locuções de entrada, a modelagem desse espaço de variabilidade é uma tarefa difícil de ser realizada. Isso acontece porque, nesse tipo de DNN, esse espaço não está acessível diretamente em nenhuma camada da rede. Basicamente, o que a rede faz é mapear representações, camada a camada, até que seja possível, na camada de saída, a classificação das locuções a partir de algum tipo de operação, como produto interno (camadas convencionais) ou distância radial (camadas RBFs).

Nesse contexto, o desenvolvimento de abordagens para a modelagem explícita do espaço de variabilidades intra-locutor facilitaria a utilização de algum controle direto sobre esse espaço. Ao analisar a decomposição realizada pela modelagem G-PLDA, temos que o espaço original pode ser descrito através de componentes distintas, mas que apenas uma delas é utilizada de fato para a classificação. Sob essa ótica, podemos imaginar uma rede que realize o mapeamento entre o espaço original em dois espaços, dos quais um é utilizado apenas para classificação e ambos são utilizados para a reconstrução da entrada (e geração do espaço original). Uma combinação entre rede de classificação e reconstrução (como o VAE, ou qualquer outra abordagem baseada em auto-codificadores, por exemplo) parece fazer sentido. Porém, fazer com que a rede combine ambas as componentes para recomposição do espaço original não é uma tarefa simples. A solução parece estar no desenvolvimento de novas funções custo, mais apropriadas e que levem em consideração esses espaços simultaneamente. Atualmente, as tarefas de classificação e reconstrução aparecem integradas apenas por conta da arquitetura da rede (através do compartilhamento dos pesos), uma vez que as funções de custo correspondentes a elas são empregas de maneira independente, a menos dos fatores de importância, para compor a função objetivo final.

REFERÊNCIAS

ADAMI, A. G. Prosodic modeling for speaker recognition based on sub-band energy temporal trajectories. In: IEEE. *2005 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. [S.l.], 2005. v. 1, p. I–189.

ADAMI, A. G.; MIHAESCU, R.; REYNOLDS, D. A.; GODFREY, J. J. Modeling prosodic dynamics for speaker recognition. In: IEEE. *International Conference on Acoustics, Speech, and Signal Processing*. [S.l.], 2003. v. 4, p. IV–788.

AJILI, M.; BONASTRE, J.-F.; ROSSETTO, S.; KAHN, J. Inter-speaker variability in forensic voice comparison: a preliminary evaluation. In: IEEE. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2016. p. 2114–2118.

AL-ALI, A. K. H.; SENADJI, B.; NAIK, G. R. Enhanced forensic speaker verification using multi-run ica in the presence of environmental noise and reverberation conditions. In: IEEE. *2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*. [S.l.], 2017. p. 174–179.

ATAL, B. S. Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *the Journal of the Acoustical Society of America*, Acoustical Society of America, v. 55, n. 6, p. 1304–1312, 1974.

AUCKENTHALER, R.; CAREY, M.; LLOYD-THOMAS, H. Score normalization for text-independent speaker verification systems. *Digital Signal Processing*, Elsevier, v. 10, n. 1, p. 42–54, 2000.

AYTAR, Y.; VONDRICK, C.; TORRALBA, A. Soundnet: Learning sound representations from unlabeled video. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2016. p. 892–900.

BEIGI, H. *Fundamentals of speaker recognition*. [S.l.]: Springer, 2011.

BILMES, J. A. et al. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. *International Computer Science Institute*, v. 4, n. 510, p. 126, 1998.

BOË, L.-J. Forensic voice identification in france. *Speech Communication*, Elsevier, v. 31, n. 2-3, p. 205–224, 2000.

BONASTRE, J.-F.; BIMBOT, F.; BOË, L.-J.; CAMPBELL, J. P.; REYNOLDS, D. A.; MAGRIN-CHAGNOLLEAU, I. Person authentication by voice: A need for caution. In: *Eighth European Conference on Speech Communication and Technology*. [S.l.: s.n.], 2003.

BONASTRE, J.-F.; KAHN, J.; ROSSATO, S.; AJILI, M. Forensic speaker recognition: Mirages and reality. *S. Fuchs/D*, p. 255, 2015.

BORGWARDT, K. M.; GRETTON, A.; RASCH, M. J.; KRIEGEL, H.-P.; SCHÖLKOPF, B.; SMOLA, A. J. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, Oxford University Press, v. 22, n. 14, p. e49–e57, 2006.

BOWMAN, S. R.; VILNIS, L.; VINYALS, O.; DAI, A. M.; JOZEFOWICZ, R.; BENGIO, S. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

BURGES, C. J. C. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, Springer, v. 2, n. 2, p. 121–167, 1998.

BURGET, L.; PLCHOT, O.; CUMANI, S.; GLEMBEK, O.; MATĚJKA, P.; BRÜMMER, N. Discriminatively trained probabilistic linear discriminant analysis for speaker verification. In: IEEE. *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2011. p. 4832–4835.

CAMPBELL, J. P. Speaker recognition: a tutorial. *Proceedings of the IEEE*, IEEE, v. 85, n. 9, p. 1437–1462, 1997.

CAMPBELL, J. P.; REYNOLDS, D. A.; DUNN, R. B. Fusing high-and low-level features for speaker recognition. In: *Interspeech*. [S.l.: s.n.], 2003.

CAMPBELL, J. P.; SHEN, W.; CAMPBELL, W. M.; SCHWARTZ, R.; BONASTRE, J.-F.; MATROUF, D. Forensic speaker recognition. In: INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. [S.l.], 2009.

CAMPBELL, W. M. Generalized linear discriminant sequence kernels for speaker recognition. In: IEEE. *International Conference on Acoustics, Speech, and Signal Processing*. [S.l.], 2002. v. 1, p. I–161.

CAMPBELL, W. M.; ASSALEH, K. T. Polynomial classifier techniques for speaker verification. In: IEEE. *International Conference on Acoustics, Speech, and Signal Processing*. [S.l.], 1999. v. 1, p. 321–324.

CAMPBELL, W. M.; CAMPBELL, J. P.; GLEASON, T. P.; REYNOLDS, D. A.; SHEN, W. Speaker verification using support vector machines and high-level features. *IEEE Transactions on Audio, Speech, and Language Processing*, IEEE, v. 15, n. 7, p. 2085–2094, 2007.

CAMPBELL, W. M.; STURIM, D. E.; REYNOLDS, D. A. Support vector machines using GMM supervectors for speaker verification. *IEEE Signal Processing Letters*, IEEE, v. 13, n. 5, p. 308–311, 2006.

CAMPBELL, W. M.; STURIM, D. E.; REYNOLDS, D. A.; SOLOMONOFF, A. SVM based speaker verification using a GMM supervector kernel and NAP variability compensation. In: IEEE. *International Conference on Acoustics, Speech and Signal Processing*. [S.l.], 2006. v. 1, p. I–I.

CHEN, T. Q.; LI, X.; GROSSE, R. B.; DUVENAUD, D. K. Isolating sources of disentanglement in variational autoencoders. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2018. p. 2610–2620.

CHEN, X.; DUAN, Y.; HOUTHOOFT, R.; SCHULMAN, J.; SUTSKEVER, I.; ABBEEL, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2016. p. 2172–2180.

CHEN, X.; KINGMA, D. P.; SALIMANS, T.; DUAN, Y.; DHARIWAL, P.; SCHULMAN, J.; SUTSKEVER, I.; ABBEEL, P. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016.

CIERI, C.; GRAFF, D.; KIMBALL, O.; MILLER, D.; WALKER, K. *Fisher English Training Part 2*. 2005. <https://catalog.ldc.upenn.edu/ldc2005t19>.

CIERI, C.; MILLER, D.; WALKER, K. The fisher corpus: a resource for the next generations of speech-to-text. In: *LREC*. [S.l.: s.n.], 2004. v. 4, p. 69–71.

CONWAY, J. B. *A course in functional analysis*. [S.l.]: Springer, 2010. v. 96.

DAVIS, S.; MERMELSTEIN, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, IEEE, v. 28, n. 4, p. 357–366, 1980.

DEHAK, N. *Discriminative and generative approaches for long-and short-term speaker characteristics modeling: application to speaker verification*. Tese (Doutorado) — École de technologie supérieure, 2009.

DEHAK, N.; KENNY, P.; DEHAK, R.; GLEMBEK, O.; DUMOUCHEL, P.; BURGET, L.; HUBEIKA, V.; CASTALDO, F. Support vector machines and joint factor analysis for speaker verification. In: IEEE. *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. [S.l.], 2009. p. 4237–4240.

DEHAK, N.; KENNY, P. J.; DEHAK, R.; DUMOUCHEL, P.; OUELLET, P. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, IEEE, v. 19, n. 4, p. 788–798, 2011.

DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum likelihood from incomplete data via the EM algorithm. In: . [S.l.: s.n.], 1977. v. 39, n. 1, p. 1–38.

DENG, L.; HINTON, G.; KINGSBURY, B. New types of deep neural network learning for speech recognition and related applications: An overview. In: IEEE. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. [S.l.], 2013. p. 8599–8603.

DENG, L.; O'SHAUGHNESSY, D. *Speech processing: a dynamic and optimization-oriented approach*. [S.l.]: CRC Press, 2003.

DENTON, E. L. et al. Unsupervised learning of disentangled representations from video. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2017. p. 4414–4423.

DO, M. N. Fast approximation of Kullback-Leibler distance for dependence trees and hidden Markov models. *IEEE Signal Processing Letters*, IEEE, v. 10, n. 4, p. 115–118, 2003.

DODDINGTON, G. R. Speaker recognition - identifying people by their voices. *Proceedings of the IEEE*, IEEE, v. 73, n. 11, p. 1651–1664, 1985.

DODDINGTON, G. R. et al. Speaker recognition based on idiolectal differences between speakers. In: *Interspeech*. [S.l.: s.n.], 2001. p. 2521–2524.

DODDINGTON, G. R.; PRZYBOCKI, M. A.; MARTIN, A. F.; REYNOLDS, D. A. The NIST speaker recognition evaluation–overview, methodology, systems, results, perspective. *Speech communication*, Elsevier, v. 31, n. 2-3, p. 225–254, 2000.

DONAHUE, J.; JIA, Y.; VINYALS, O.; HOFFMAN, J.; ZHANG, N.; TZENG, E.; DARRELL, T. Decaf: A deep convolutional activation feature for generic visual recognition. In: *International Conference on Machine Learning*. [S.l.: s.n.], 2014. p. 647–655.

DRYGAJLO, A.; HARAKSIM, R. Biometric evidence in forensic automatic speaker recognition. In: *Handbook of Biometrics for Forensic Science*. [S.l.]: Springer, 2017. p. 221–239.

DUDA, R. O.; HART, P. E. *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

FANT, G. *Acoustic theory of speech production*. [S.l.]: Walter de Gruyter, 1970.

FBI. *Business e-mail compromise the 12 billion dollar scam*. 2018. <https://www.ic3.gov/media/2018/180712.aspx>. Acessado em Julho de 2019.

FERRER, L.; LEI, Y.; MCLAREN, M.; SCHEFFER, N. Study of senone-based deep neural network approaches for spoken language recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, IEEE, v. 24, n. 1, p. 105–116, 2015.

FLANAGAN, J. L. *Speech analysis: Synthesis and perception.* [S.l.]: Springer-Verlag, 1972.

FURUI, S. Cepstral analysis technique for automatic speaker verification. *IEEE Transactions on Acoustics, Speech and Signal Processing*, IEEE, v. 29, n. 2, p. 254–272, 1981.

FURUI, S. Recent advances in speaker recognition. In: SPRINGER. *Audio-and Video-based Biometric Person Authentication*. [S.l.], 1997. p. 235–252.

GARCIA-ROMERO, D.; ESPY-WILSON, C. Y. Analysis of i-vector length normalization in speaker recognition systems. In: *Twelfth annual conference of the international speech communication association*. [S.l.: s.n.], 2011.

GARCIA-ROMERO, D.; ZHOU, X.; ESPY-WILSON, C. Y. Multicondition training of Gaussian PLDA models in i-vector space for noise and reverberation robust speaker recognition. In: IEEE. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2012. p. 4257–4260.

GAUVAIN, J. L.; LEE, C.-H. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. In: *IEEE Transactions on Speech and Audio Processing*. [S.l.: s.n.], 1994. v. 2, n. 2, p. 291–298.

GISH, H.; KARNOFSKY, K.; KRASNER, M.; ROUCOS, S.; SCHWARTZ, R.; WOLF, J. Investigation of text-independent speaker indentification over telephone channels. In: IEEE. *1985 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. [S.l.], 1985. v. 10, p. 379–382.

GLEMBEK, O.; BURGET, L.; DEHAK, N.; BRUMMER, N.; KENNY, P. Comparison of scoring methods used in speaker recognition with joint factor analysis. In: IEEE. *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. [S.l.], 2009. p. 4057–4060.

GONZÁLEZ-RODRÍGUEZ, J.; ORTEGA-GARCÍA, J.; MARTÍN, C.; HERNÁNDEZ, L. Increasing robustness in GMM speaker recognition systems for noisy and reverberant speech with low complexity microphone arrays. In: IEEE. *Proceeding of Fourth International Conference on Spoken Language Processing (ICSLP)*. [S.l.], 1996. v. 3, p. 1333–1336.

GOODFELLOW, I.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDE-FARLEY, D.; OZAIR, S.; COURVILLE, A.; BENGIO, Y. Generative adversarial nets. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2014. p. 2672–2680.

GRETTON, A.; BORGWARDT, K.; RASCH, M.; SCHÖLKOPF, B.; SMOLA, A. J. A kernel method for the two-sample-problem. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2007. p. 513–520.

GRETTON, A.; BORGWARDT, K. M.; RASCH, M. J.; SCHÖLKOPF, B.; SMOLA, A. A kernel two-sample test. *Journal of Machine Learning Research*, v. 13, n. Mar, p. 723–773, 2012.

GUDNASON, J.; BROOKES, M. Voice source cepstrum coefficients for speaker identification. In: IEEE. *International Conference on Acoustics, Speech and Signal Processing*. [S.l.], 2008. p. 4821–4824.

HARRINGTON, J.; CASSIDY, S. *Techniques in speech acoustics*. [S.l.]: Springer, 1999. v. 8.

HARSHA, B. V. A noise robust speech activity detection algorithm. In: IEEE. *Proceedings of International Symposium on Intelligent Multimedia, Video and Speech Processing*. [S.l.], 2004. p. 322–325.

HATCH, A. O.; KAJAREKAR, S.; STOLCKE, A. Within-class covariance normalization for svm-based speaker recognition. In: *Ninth international conference on spoken language processing*. [S.l.: s.n.], 2006.

HAUTAMÄKI, V.; TUONONEN, M.; NIEMI-LAITINEN, T.; FRÄNTI, P. Improving speaker verification by periodicity based voice activity detection. In: *International Conference on Speech and Computer (SPECOM)*. [S.l.: s.n.], 2007. p. 645–650.

HEIGOLD, G.; MORENO, I.; BENGIO, S.; SHAZEER, N. End-to-end text-dependent speaker verification. In: IEEE. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2016. p. 5115–5119.

HERMANSKY, H. Perceptual linear predictive (PLP) analysis of speech. *The Journal of the Acoustical Society of America*, Acoustical Society of America, v. 87, n. 4, p. 1738–1752, 1990.

HERMANSKY, H.; MORGAN, N. RASTA processing of speech. *IEEE Transactions on Speech and Audio Processing*, IEEE, v. 2, n. 4, p. 578–589, 1994.

HERMANSKY, H.; MORGAN, N.; BAYYA, A.; KOHN, P. RASTA-PLP speech analysis technique. In: IEEE. *International Conference on Acoustics, Speech, and Signal Processing*. [S.l.], 1992. v. 1, p. 121–124.

HIGGINS, A. L.; BAHLER, L.; PORTER, J. Speaker verification using randomized phrase prompting. In: *Digital Signal Processing*. [S.l.: s.n.], 1991. v. 1, n. 2, p. 89–106.

HIGGINS, I.; MATTHEY, L.; PAL, A.; BURGESS, C.; GLOROT, X.; BOTVINICK, M.; MOHAMED, S.; LERCHNER, A. beta-VAE: Learning basic visual concepts with a constrained variational framework. *ICLR*, v. 2, n. 5, p. 6, 2017.

HINTON, G.; DENG, L.; YU, D.; DAHL, G. E.; MOHAMED, A.-r.; JAITLY, N.; SENIOR, A.; VANHOUCKE, V.; NGUYEN, P.; SAINATH, T. N. et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, IEEE, v. 29, n. 6, p. 82–97, 2012.

HU, Z.; YANG, Z.; LIANG, X.; SALAKHUTDINOV, R.; XING, E. P. Toward controlled generation of text. In: JMLR. ORG. *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. [S.l.], 2017. p. 1587–1596.

HUANG, X.; ACERO, A.; HON, H.-W.; BY-REDDY, R. F. *Spoken language processing: A guide to theory, algorithm, and system development*. [S.l.]: PTR Prentice Hall, 2001.

IAFPA. *IAFPA Resolution - Voiceprints*. 2013. <http://www.iafpa.net/voiceprintsres.htm>. Acessado em Julho de 2019.

ISOLA, P.; ZHU, J.-Y.; ZHOU, T.; EFROS, A. A. Image-to-image translation with conditional adversarial networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 1125–1134.

JAAKKOLA, T.; HAUSSLER, D. Exploiting generative models in discriminative classifiers. *Advances in Neural Information Processing Systems*, MIT; 1998, p. 487–493, 1999.

JR, H. L. P.; SIEGEL, G. M.; FOX, P. W.; GARBER, S. R.; KEARNEY, J. K. Inhibiting the lombard effect. *The Journal of the Acoustical Society of America*, Acoustical Society of America, v. 85, n. 2, p. 894–900, 1989.

JUNQUA, J.-C. The lombard reflex and its role on human listeners and automatic speech recognizers. *The Journal of the Acoustical Society of America*, Acoustical Society of America, v. 93, n. 1, p. 510–524, 1993.

KANAGASUNDARAM, A.; VOGT, R.; DEAN, D. B.; SRIDHARAN, S.; MASON, M. W. I-vector based speaker recognition on short utterances. In: INTERNATIONAL SPEECH COMMUNICATION ASSOCIATION (ISCA). *Proceedings of the 12th Annual Conference of the International Speech Communication Association*. [S.l.], 2011. p. 2341–2344.

KANAGASUNDARAM, A.; VOGT, R. J.; DEAN, D. B.; SRIDHARAN, S. PLDA based speaker recognition on short utterances. In: ISCA. *The Speaker and Language Recognition Workshop (Odyssey 2012)*. [S.l.], 2012.

KENNY, P. Joint factor analysis of speaker and session variability: Theory and algorithms. *CRIM, Montreal,(Report) CRIM-06/08-13*, v. 215, 2005.

KENNY, P. Bayesian speaker verification with heavy-tailed priors. In: *Odyssey*. [S.l.: s.n.], 2010. p. 14.

KENNY, P.; BOULIANNE, G.; DUMOUCHEL, P. Eigenvoice modeling with sparse training data. *IEEE Transactions on Speech and Audio Processing*, IEEE, v. 13, n. 3, p. 345–354, 2005.

KENNY, P.; BOULIANNE, G.; OUELLET, P.; DUMOUCHEL, P. Factor analysis simplified [speaker verification applications]. In: IEEE. *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.* [S.l.], 2005. v. 1, p. I–637.

KENNY, P.; BOULIANNE, G.; OUELLET, P.; DUMOUCHEL, P. Joint factor analysis versus eigenchannels in speaker recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, IEEE, v. 15, n. 4, p. 1435–1447, 2007.

KENNY, P.; BOULIANNE, G.; OUELLET, P.; DUMOUCHEL, P. Speaker and session variability in gmm-based speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, IEEE, v. 15, n. 4, p. 1448–1460, 2007.

KENNY, P.; OUELLET, P.; DEHAK, N.; GUPTA, V.; DUMOUCHEL, P. A study of interspeaker variability in speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, IEEE, v. 16, n. 5, p. 980–988, 2008.

KENNY, P.; STAFYLAKIS, T.; OUELLET, P.; GUPTA, V.; ALAM, M. J. Deep neural networks for extracting Baum-Welch statistics for speaker recognition. In: *Odyssey*. [S.l.: s.n.], 2014. v. 2014, p. 293–298.

KERSTA, L. G. Voiceprint identification. *Nature*, Nature Publishing Group, v. 196, n. 4861, p. 1253, 1962.

KIM, H.; MNIH, A. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

KINGMA, D. P.; WELLING, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

KINNUNEN, T.; ALKU, P. On separating glottal source and vocal tract information in telephony speaker verification. In: IEEE. *International Conference on Acoustics, Speech and Signal Processing*. [S.l.], 2009. p. 4545–4548.

KINNUNEN, T.; LI, H. An overview of text-independent speaker recognition: from features to supervectors. *Speech Communication*, Elsevier, v. 52, n. 1, p. 12–40, 2010.

KO, T.; PEDDINTI, V.; POVEY, D.; SELTZER, M. L.; KHUDANPUR, S. A study on data augmentation of reverberant speech for robust speech recognition. In: IEEE. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2017. p. 5220–5224.

KOLLEWE, J. *HSBC rolls out voice and touch ID security for bank customers*. 2016. <https://www.theguardian.com/business/2016/feb/19/hsbc-rolls-out-voice-touch-id-security-bank-customers>. Acessado em Julho de 2019.

LEI, Y.; SCHEFFER, N.; FERRER, L.; MCLAREN, M. A novel scheme for speaker recognition using a phonetically-aware deep neural network. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.: s.n.], 2014. p. 1695–1699.

LEI, Y.; SCHEFFER, N.; FERRER, L.; MCLAREN, M. A novel scheme for speaker recognition using a phonetically-aware deep neural network. In: IEEE. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2014. p. 1695–1699.

LEXISNEXIS. *O Verdadeiro Custo da Fraude 2018 - Brasil*. 2018. <https://risk.lexisnexis.com.br/global/-/media/files/financial%20services/infographics/lnrs-tcof-brazil_infographic-nxr12661-02-0619-pt-la.pdf>. Acessado em Julho de 2019.

LI, J.; DENG, L.; HAEB-UMBACH, R.; GONG, Y. *Robust automatic speech recognition: a bridge to practical applications*. [S.l.]: Academic Press, 2015.

LI, L.; TANG, Z.; SHI, Y.; WANG, D. Gaussian-constrained training for speaker verification. In: IEEE. *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2019. p. 6036–6040.

LI, L.; TANG, Z.; WANG, D.; ZHENG, T. F. Full-info training for deep speaker feature learning. In: IEEE. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2018. p. 5369–5373.

LIN, M.; CHEN, Q.; YAN, S. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

LIU, Q.; WANG, D. Stein variational gradient descent: A general purpose bayesian inference algorithm. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2016. p. 2378–2386.

LYU, S.; SIMONCELLI, E. P. Nonlinear extraction of independent components of natural images using radial gaussianization. *Neural Computation*, MIT Press, v. 21, n. 6, p. 1485–1519, 2009.

MAATEN, L. v. d.; HINTON, G. Visualizing data using t-SNE. *Journal of Machine Learning Research*, v. 9, n. Nov, p. 2579–2605, 2008.

MAKHOUL, J. Linear prediction: A tutorial review. *Proceedings of the IEEE*, IEEE, v. 63, n. 4, p. 561–580, 1975.

MAMMONE, R. J.; ZHANG, X.; RAMACHANDRAN, R. P. Robust speaker recognition: A feature-based approach. *IEEE Signal Processing Magazine*, IEEE, v. 13, n. 5, p. 58, 1996.

MATEJKA, P.; ZHANG, L.; NG, T.; GLEMBEK, O.; MA, J. Z.; ZHANG, B.; MALLIDI, S. H. Neural network bottleneck features for language identification. In: *Odyssey*. [S.l.: s.n.], 2014.

MATHIEU, M. F.; ZHAO, J. J.; ZHAO, J.; RAMESH, A.; SPRECHMANN, P.; LECUN, Y. Disentangling factors of variation in deep representation using adversarial training. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2016. p. 5040–5048.

MCLAREN, M.; LEI, Y.; FERRER, L. Advances in deep neural network approaches to speaker recognition. In: IEEE. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2015. p. 4814–4818.

MICROSOFT. *Speaker Recognition*. 2019. <https://azure.microsoft.com/pt-br/services/cognitive-services/speaker-recognition/>. Acessado em Julho de 2019.

MING, J.; STEWART, D.; VASEGHI, S. Speaker identification in unknown noisy conditions-a universal compensation approach. In: IEEE. *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. [S.l.], 2005. v. 1, p. 617–620.

MOON, T. K. The expectation-maximization algorithm. In: . [S.l.: s.n.], 1996. v. 13, n. 6, p. 47–60.

NAUTSCH, A.; RATHGEB, C.; BUSCH, C.; REININGER, H.; KASPER, K. Towards duration invariance of i-vector-based adaptive score normalization. In: *Odyssey*. [S.l.: s.n.], 2014.

NILSON. *Card Fraud Losses Reach $27.85 Billion*. 2019. <https://nilsonreport.com/mention/407/1link/>. Acessado em Dezembro de 2019.

NIST. *NIST Speaker Recognition Evaluation*. 1996. <https://www.nist.gov/itl/iad/mig/speaker-recognition>.

NUANCE. *Multi-modal biometrics: Simpler, stronger customer authentication*. 2019. <https://www.nuance.com/omni-channel-customer-engagement/security/identification-and-verification.html>. Acessado em Julho de 2019.

OGLESBY, J.; MASON, J. S. Radial basis function networks for speaker recognition. In: IEEE. *International Conference on Acoustics, Speech, and Signal*. [S.l.], 1991. p. 393–396.

OMID, S.; CRAIG, G. *NIST 2019 Speaker Recognition Evaluation*. 2019. <https://www.nist.gov/itl/iad/mig/nist-2019-speaker-recognition-evaluation>.

ORTEGA-GARCÍA, J.; GONZÁLEZ-RODRÍGUEZ, J. Overview of speech enhancement techniques for automatic speaker recognition. In: IEEE. *Proceeding of Fourth International Conference on Spoken Language Processing (ICSLP)*. [S.l.], 1996. v. 2, p. 929–932.

O'SHAUGHNESSY, D. *Speech Communications: Human And Machine*. [S.l.]: Universities Press, 1987.

PEDDINTI, V.; POVEY, D.; KHUDANPUR, S. A time delay neural network architecture for efficient modeling of long temporal contexts. In: *Sixteenth Annual Conference of the International Speech Communication Association*. [S.l.: s.n.], 2015.

PEER, I.; RAFAELY, B.; ZIGEL, Y. Reverberation matching for speaker recognition. In: IEEE. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2008. p. 4829–4832.

PELECANOS, J.; SRIDHARAN, S. Feature warping for robust speaker verification. International Speech Communication Association (ISCA), 2001.

PLUMPE, M. D.; QUATIERI, T. F.; REYNOLDS, D. A. Modeling of the glottal flow derivative waveform with application to speaker identification. *IEEE Transactions on Speech and Audio Processing*, IEEE, v. 7, n. 5, p. 569–586, 1999.

POVEY, D.; ZHANG, X.; KHUDANPUR, S. Parallel training of deep neural networks with natural gradient and parameter averaging. *arXiv preprint arXiv:1410.7455*, Citeseer, 2014.

PRASANNA, S. R. M.; GUPTA, C. S.; YEGNANARAYANA, B. Extraction of speaker-specific excitation information from linear prediction residual of speech. *Speech Communication*, Elsevier, v. 48, n. 10, p. 1243–1261, 2006.

PRINCE, S. J. D.; ELDER, J. H. Probabilistic linear discriminant analysis for inferences about identity. In: IEEE. *International Conference on Computer Vision*. [S.l.], 2007. p. 1–8.

RABINER, L. R.; JUANG, B.-H. *Fundamentals of speech recognition*. [S.l.]: PTR Prentice Hall, 1993. v. 14.

RAMIREZ, J.; SEGURA, J. C.; BENITEZ, C.; TORRE, A. D. L.; RUBIO, A. Efficient voice activity detection algorithms using long-term speech information. *Speech Communication*, Elsevier, v. 42, n. 3, p. 271–287, 2004.

RAO, K. R.; YIP, P. *Discrete cosine transform: algorithms, advantages, applications*. [S.l.]: Academic press, 2014.

REYNOLDS, D. A. A Gaussian mixture modeling approach to text-independent speaker identification. *PhD Thesis*, Georgia Institute of Technology, 1992.

REYNOLDS, D. A. Experimental evaluation of features for robust speaker identification. *IEEE Transactions on Speech and Audio Processing*, IEEE, v. 2, n. 4, p. 639–643, 1994.

REYNOLDS, D. A. Speaker identification and verification using Gaussian mixture speaker models. *Speech Communication*, Elsevier, v. 17, n. 1, p. 91–108, 1995.

REYNOLDS, D. A. Comparison of background normalization methods for text-independent speaker verification. In: *Eurospeech*. [S.l.: s.n.], 1997.

REYNOLDS, D. A. Channel robust speaker verification via feature mapping. In: IEEE. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. [S.l.], 2003. v. 2, p. II–53.

REYNOLDS, D. A.; QUATIERI, T. F.; DUNN, R. B. Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, Elsevier, v. 10, n. 1, p. 19–41, 2000.

REYNOLDS, D. A.; ROSE, R. C. Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing*, IEEE, v. 3, n. 1, p. 72–83, 1995.

REYNOLDS, D. A.; ROSE, R. C.; SMITH, M. J. T. PC-based TMS320C30 implementation of the Gaussian mixture model text-independent speaker recognition system. In: *Proceedings of the International Conference on Signal Processing Applications and Technology*. [S.l.: s.n.], 1992. p. 967–973.

ROSE, P. *Forensic speaker identification*. [S.l.]: CRC Press, 2003.

ROSENBERG, A. E.; DELONG, J.; LEE, C.-H.; JUANG, B.-H.; SOONG, F. K. The use of cohort normalized scores for speaker verification. In: *International Conference Speech Language Processing*. [S.l.: s.n.], 1992. p. 599–602.

SAMBUR, M. Selection of acoustic features for speaker identification. *IEEE Transactions on Acoustics, Speech and Signal Processing*, IEEE, v. 23, n. 2, p. 176–182, 1975.

SCHMIDT, M.; GISH, H. Speaker identification via support vector classifiers. In: IEEE. *International Conference on Acoustics, Speech, and Signal Processing*. [S.l.], 1996. v. 1, p. 105–108.

SERBAN, I. V.; SORDONI, A.; LOWE, R.; CHARLIN, L.; PINEAU, J.; COURVILLE, A.; BENGIO, Y. A hierarchical latent variable encoder-decoder model for generating dialogues. In: *Thirty-First AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2017.

SHRIBERG, E.; FERRER, L.; KAJAREKAR, S.; VENKATARAMAN, A.; STOLCKE, A. Modeling prosodic feature sequences for speaker recognition. *Speech Communication*, Elsevier, v. 46, n. 3, p. 455–472, 2005.

SNYDER, D.; CHEN, G.; POVEY, D. MUSAN: A music, speech, and noise corpus. *arXiv preprint arXiv:1510.08484*, 2015.

SNYDER, D.; GARCIA-ROMERO, D.; POVEY, D.; KHUDANPUR, S. Deep neural network embeddings for text-independent speaker verification. In: *Interspeech*. [S.l.: s.n.], 2017. p. 999–1003.

SNYDER, D.; GARCIA-ROMERO, D.; SELL, G.; POVEY, D.; KHUDANPUR, S. X-vectors: Robust DNN embeddings for speaker recognition. In: IEEE. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2018. p. 5329–5333.

SNYDER, D.; GHAHREMANI, P.; POVEY, D.; GARCIA-ROMERO, D.; CARMIEL, Y.; KHUDANPUR, S. Deep neural network-based speaker embeddings for end-to-end speaker verification. In: IEEE. *2016 IEEE Spoken Language Technology Workshop (SLT)*. [S.l.], 2016. p. 165–170.

SOHN, J.; KIM, N. S.; SUNG, W. A statistical model-based voice activity detection. *IEEE Signal Processing Letters*, IEEE, v. 6, n. 1, p. 1–3, 1999.

SOLEWICZ, Y. A.; JESSEN, M.; VLOED, D. van der. Null-hypothesis llr: A proposal for forensic automatic speaker recognition. In: *INTERSPEECH*. [S.l.: s.n.], 2017. p. 2849–2853.

SOLOMONOFF, A.; CAMPBELL, W. M.; BOARDMAN, I. Advances in channel compensation for svm speaker recognition. In: IEEE. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. [S.l.], 2005. v. 1, p. I–629.

SONG, Y.; JIANG, B.; BAO, Y.; WEI, S.; DAI, L.-R. I-vector representation based on bottleneck features for language identification. *Electronics Letters*, IET, v. 49, n. 24, p. 1569–1570, 2013.

SOONG, F. K.; ROSENBERG, A. E.; JUANG, B.-H.; RABINER, L. R. A vector quantization approach to speaker recognition. *AT&T Technical Journal*, Alcatel-Lucent, v. 66, n. 2, p. 14–26, 1987.

STADTSCHNITZER, M.; PHAM, T. V.; CHIEN, T. T. Reliable voice activity detection algorithms under adverse environments. In: IEEE. *International Conference on Communications and Electronics (ICCE)*. [S.l.], 2008. p. 218–223.

STEVENS, S. S.; VOLKMANN, J. The relation of pitch to frequency: A revised scale. *The American Journal of Psychology*, JSTOR, p. 329–353, 1940.

STEVENS, S. S.; VOLKMANN, J.; NEWMAN, E. B. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, Acoustical Society of America, v. 8, n. 3, p. 185–190, 1937.

STURIM, D. E.; CAMPBELL, W. M.; KARAM, Z. N.; REYNOLDS, D. A.; RICHARDSON, F. S. The MIT Lincoln laboratory 2008 speaker recognition system. In: *Interspeech*. [S.l.: s.n.], 2009. p. 2359–2362.

STURIM, D. E.; REYNOLDS, D. A. Speaker adaptive cohort selection for tnorm in text-independent speaker verification. In: IEEE. *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.* [S.l.], 2005. v. 1, p. I–741.

TAYLOR, P. *Text-to-speech synthesis*. [S.l.]: Cambridge university press, 2009.

TOGNERI, R.; PULLELLA, D. An overview of speaker identification: Accuracy and robustness issues. *IEEE Circuits and Systems Magazine*, IEEE, v. 11, n. 2, p. 23–61, 2011.

TSCHANNEN, M.; BACHEM, O.; LUCIC, M. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*, 2018.

TU, Y.; MAK, M.-W.; CHIEN, J.-T. Variational domain adversarial learning for speaker verification. *Interspeech*, p. 4315–4319, 2019.

VAPNIK, V. *The nature of statistical learning theory*. [S.l.]: springer, 2000.

VAPNIK, V. N. *Estimation of dependences based on empirical data [Em russo]*. [S.l.]: Nauka, Moscow, 1979.

VAPNIK, V. N.; VAPNIK, V. *Statistical learning theory*. [S.l.]: Wiley New York, 1998. v. 2.

VARIANI, E.; LEI, X.; MCDERMOTT, E.; MORENO, I. L.; GONZALEZ-DOMINGUEZ, J. Deep neural networks for small footprint text-dependent speaker verification. In: IEEE. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2014. p. 4052–4056.

VERGIN, R.; O'SHAUGHNESSY, D. Pre-emphasis and speech recognition. In: IEEE. *Proceedings 1995 Canadian Conference on Electrical and Computer Engineering*. [S.l.], 1995. v. 2, p. 1062–1065.

VESELỲ, K.; GHOSHAL, A.; BURGET, L.; POVEY, D. Sequence-discriminative training of deep neural networks. In: *Interspeech*. [S.l.: s.n.], 2013. v. 2013, p. 2345–2349.

VIIKKI, O.; LAURILA, K. Cepstral domain segmental feature vector normalization for noise robust speech recognition. *Speech Communication*, Elsevier, v. 25, n. 1, p. 133–147, 1998.

VILLALBA, J.; CHEN, N.; SNYDER, D.; GARCIA-ROMERO, D.; MCCREE, A.; SELL, G.; BORGSTROM, J.; RICHARDSON, F.; SHON, S.; GRONDIN, F. et al. State-of-the-art speaker recognition for telephone and video speech: the JHU-MIT submission for NIST SRE18. *Proc. Interspeech 2019*, p. 1488–1492, 2019.

VILLEGAS, R.; YANG, J.; HONG, S.; LIN, X.; LEE, H. Decomposing motion and content for natural video sequence prediction. *arXiv preprint arXiv:1706.08033*, 2017.

VOGT, R. J.; BAKER, B. J.; SRIDHARAN, S. Modelling session variability in text independent speaker verification. International Speech Communication Association (ISCA), 2005.

VOGT, R. J.; LUSTRI, C. J.; SRIDHARAN, S. Factor analysis modelling for speaker verification with short utterances. IEEE, 2008.

VUUREN, S. V. Comparison of text-independent speaker recognition methods on telephone speech with acoustic mismatch. In: IEEE. *International Conference on Spoken Language*. [S.l.], 1996. v. 3, p. 1788–1791.

WAN, V.; CAMPBELL, W. M. Support vector machines for speaker verification and identification. In: CITESEER. *Neural Networks for Signal Processing. Proceedings of the IEEE Signal Processing Society Workshop.* [S.l.], 2000. v. 2, p. 775–784.

WAN, V.; RENALS, S. Evaluation of kernel methods for speaker verification and identification. In: IEEE. *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*. [S.l.], 2002. v. 1, p. I–669.

WAN, V.; RENALS, S. Speaker verification using sequence discriminant support vector machines. *IEEE Transactions on Speech and Audio Processing*, IEEE, v. 13, n. 2, p. 203–210, 2005.

WANG, X.; LI, L.; WANG, D. VAE-based domain adaptation for speaker verification. *arXiv preprint arXiv:1908.10092*, 2019.

WARMAN, M. *Say goodbye to the pin: voice recognition takes over at Barclays Wealth*. 2013. <https://www.telegraph.co.uk/technology/news/10044493/Say-goodbye-to-the-pin-voice-recognition-takes-over-at-Barclays-Wealth.html>. Acessado em Julho de 2019.

WAYMAN, J. L. Error rate equations for the general biometric system. *IEEE Robotics & Automation Magazine*, v. 6, n. 1, p. 35–48, 1999.

WOLF, J. J. Efficient acoustic parameters for speaker recognition. *The Journal of the Acoustical Society of America*, Acoustical Society of America, v. 51, n. 6B, p. 2044–2056, 1972.

XIANG, B.; CHAUDHARI, U. V.; NAVRATIL, J.; RAMASWAMY, G. N.; GOPINATH, R. A. Short-time gaussianization for robust speaker verification. In: IEEE. *International Conference on Acoustics, Speech, and Signal Processing*. [S.l.], 2002. v. 1, p. I–681.

ZHANG, C.; KOISHIDA, K. End-to-End text-independent speaker verification with triplet loss on short utterances. In: *Interspeech*. [S.l.: s.n.], 2017. p. 1487–1491.

ZHANG, Y.; LI, L.; WANG, D. VAE-based regularization for deep speaker embedding. *arXiv preprint arXiv:1904.03617*, Apr 2019.

ZHAO, S.; SONG, J.; ERMON, S. InfoVAE: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017.

ZHENG, N.; LEE, T.; CHING, P.-C. Integration of complementary acoustic features for speaker recognition. *IEEE Signal Processing Letters*, IEEE, v. 14, n. 3, p. 181–184, 2007.

ZHOU, X.; GARCIA-ROMERO, D.; DURAISWAMI, R.; ESPY-WILSON, C.; SHAMMA, S. Linear versus mel frequency cepstral coefficients for speaker recognition. In: IEEE. *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. [S.l.], 2011. p. 559–564.

ZHU, J.-Y.; PARK, T.; ISOLA, P.; EFROS, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: *Proceedings of the IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2017. p. 2223–2232.

APÊNDICE A – 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 (DEMPSTER; LAIRD; RUBIN, 1977). 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 amostras. Geralmente é utilizado quando as equações não podem ser solucionadas de forma direta (MOON, 1996; BILMES et al., 1998). 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.

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 alguma condição de parada seja alcançada, como a estabilização das verossimilhanças ou um número máximo de iterações ser atingido.

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, Z, 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 componente a variável conhecida provém.

Sabe-se 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.

Para Modelos de Misturas Gaussianas, por exemplo, encontrar a solução que maximiza a verossimilhança do modelo requer o cálculo das 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 28 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, λ, associados a uma função de verossimilhança $L(\lambda; X, Z) = p(X, Z|\lambda)$, a estimativa do máximo de verossimilhança (MLE[1]) dos parâme-

[1] Maximum likelihood estimate.

tros desconhecidos é dada pela verossimilhança marginal dos dados observados:

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

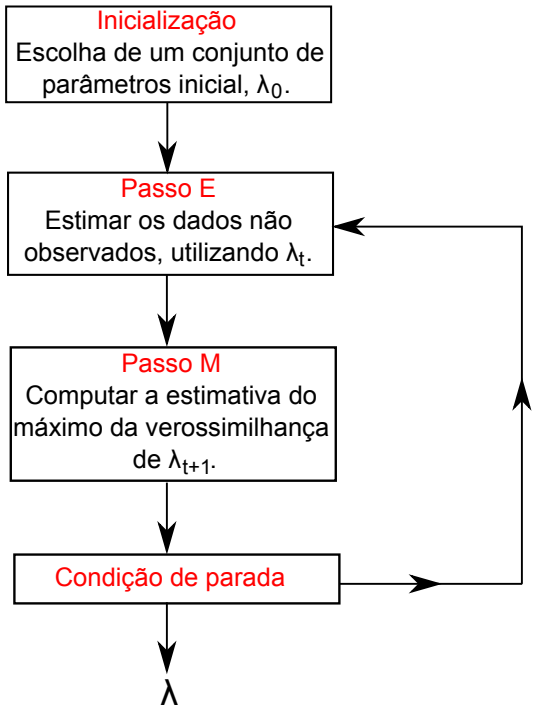

Figura 28 – Visão geral do algoritmo EM. Os passos E e M são alternados até que a estimativa dos parâmetros convirja.

O algoritmo procura encontrar o MLE iterativamente aplicando dois passos:

- **Passo de expectativa (Passo E)**: calcular o valor esperado do logaritmo da verossimilhança com respeito à distribuição condicional de Z e X com respeito ao valor corrente dos parâmetros $\lambda^{(t)}$:

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

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

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

A.1 *EXPECTATION-MAXIMIZATION* 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

enxergar as variáveis ω_i como as variáveis latentes desconhecidas. Estimar $P(\omega_i)$ significa, portanto, estimar o peso final da distribuição i.

Suponha $X = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ e que tenhamos M distribuições no modelo, cujos parâmetros são referenciados como λ, então:

$$p(\mathbf{x}_k|\lambda) = \sum_{j=1}^{M} p(\mathbf{x}_k|\omega_j, \lambda_j)P(\omega_j). \tag{A.4}$$

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

$$p(X|\lambda) = \prod_{k=1}^{N} p(\mathbf{x}_k|\lambda), \tag{A.5}$$

e a estimativa do máximo da verossimilhança, λ', é o valor de λ que maximiza $p(X|\lambda)$.

Se assumirmos que $p(X|\lambda)$ é uma função diferenciável em λ, então podemos derivar as condições necessárias para os valores de λ'. Definimos então L como sendo o logaritmo da verossimilhança e $\nabla_{\lambda_i} L$ o gradiente de L com respeito a λ_i, que é um subconjunto de λ, referente aos parâmetros da i-ésima distribuição. Então:

$$L = \sum_{k=1}^{N} \log p(\mathbf{x}_k|\lambda) \tag{A.6}$$

e

$$\nabla_{\lambda_i} L = \sum_{k=1}^{N} \frac{1}{p(\mathbf{x}_k|\lambda)} \nabla_{\lambda_i} [\sum_{j=1}^{M} p(\mathbf{x}_k|\omega_j, \lambda_j)P(\omega_j)]. \tag{A.7}$$

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

$$P(\omega_i|\mathbf{x}_k, \lambda) = \frac{p(\mathbf{x}_k|\omega_i, \lambda_i)P(\omega_i)}{p(\mathbf{x}_k|\lambda)}, \tag{A.8}$$

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

$$\nabla_{\lambda_i} L = \sum_{k=1}^{N} P(\omega_i|\mathbf{x}_k, \lambda) \nabla_{\lambda_i} [\log p(\mathbf{x}_k|\omega_i, \lambda_i)]. \tag{A.9}$$

Uma vez que o gradiente deve desaparecer em λ_i que maximiza L, a estimativa do máximo de verossimilhança, λ_i', deve satisfazer a condição:

$$\sum_{k=1}^{N} P(\omega_i|\mathbf{x}_k, \lambda') \nabla_{\lambda_i} [\log p(\mathbf{x}_k|\omega_i, \lambda_i')] = 0, \tag{A.10}$$

$$i = 1, ..., M. \tag{A.11}$$

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

$$P(\omega_i)' = \frac{1}{N} \sum_{k=1}^{N} P(\omega_i|\mathbf{x}_k, \lambda), \tag{A.12}$$

$$\mu_i' = \frac{\sum_{k=1}^{N} P(\omega_i|\mathbf{x}_k, \lambda)\mathbf{x}_k}{\sum_{k=1}^{N} P(\omega_i|\mathbf{x}_k, \lambda)}, \tag{A.13}$$

$$\Sigma_i' = \frac{\sum_{k=1}^{N} P(\omega_i|\mathbf{x}_k, \lambda)(\mathbf{x}_k - \mu_i)(\mathbf{x}_k - \mu_i)^T}{\sum_{k=1}^{N} P(\omega_i|\mathbf{x}_k, \lambda)}. \tag{A.14}$$

APÊNDICE B – MÁQUINAS DE VETORES SUPORTE

Máquinas de vetores suporte (*Support Vector Machines* - SVMs) constituem uma técnica de aprendizado que vem recebendo cada vez mais atenção na área de reconhecimento de padrões. Os livros (VAPNIK; VAPNIK, 1998) e (VAPNIK, 2000) possuem excelentes descrições das SVMs. Além disso, um bom tutorial pode ser encontrado em (BURGES, 1998). Apesar dos primeiros trabalhos datarem da década de 1970 (VAPNIK, 1979), essa técnica vem ganhando destaque desde a última década pela disponibilidade de novos algoritmos de aprendizagem. O sucesso de SVMs se dá em grande parte ao fato de ser uma técnica sofisticada sob o ponto de vista computacional e também fácil de ser analisada teoricamente. Isto é, aliam o poder necessário para utilização em problemas reais difíceis e a facilidade de análise dos resultados.

Basicamente, SVMs são classificadores binários, que se propõem a estimar uma função de classificação,

$$f : \mathbb{R}^D \to \{-1, +1\}, \tag{B.1}$$

a partir de um conjunto de amostras (vetores de dimensão D, $\mathbf{x}_i$, e classes correspondentes, y_i) de treinamento,

$$\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\} \in \mathbb{R}^D \times \{-1, +1\}, \tag{B.2}$$

de maneira que a classificação de novas amostras possa ser realizada a partir do cálculo do sinal da função f.

SVMs são baseadas na teoria de aprendizado estatístico (*statistical learning theory*), ou teoria VC [1] (VAPNIK, 2000), que mostra que é crucial a imposição de restrições às classes de funções para f. Se nenhuma imposição for feita, a minimização do erro de treinamento não necessariamente implica uma pequena taxa de erro de classificação de novas amostras. Tal teoria mostra a necessidade de se restringir f a uma classe de funções que possua capacidade de aprendizado apropriada para a quantidade de dados disponíveis. Tal restrição se mostra necessária para uma bom poder de generalização das máquinas de aprendizado. Por essa razão, SVMs são baseadas na classe de funções de hiperplanos:

$$f(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}) + b, \qquad \mathbf{w} \in \mathbb{R}^D, \quad b \in \mathbb{R}. \tag{B.3}$$

Pode-se mostrar que o hiperplano ótimo, sob o ponto de vista de generalização, é definido como aquele que apresenta a maior margem de separação entre as amostras das duas classes (Figura 29). Tal hiperplano pode ser construído unicamente ao solucionar um problema de otimização quadrática com restrições. A solução desse problema de otimização possui a seguinte expansão:

$$w = \sum_i \alpha_i y_i \mathbf{x}_i, \tag{B.4}$$

[1] Vapnik-Chervonenski.

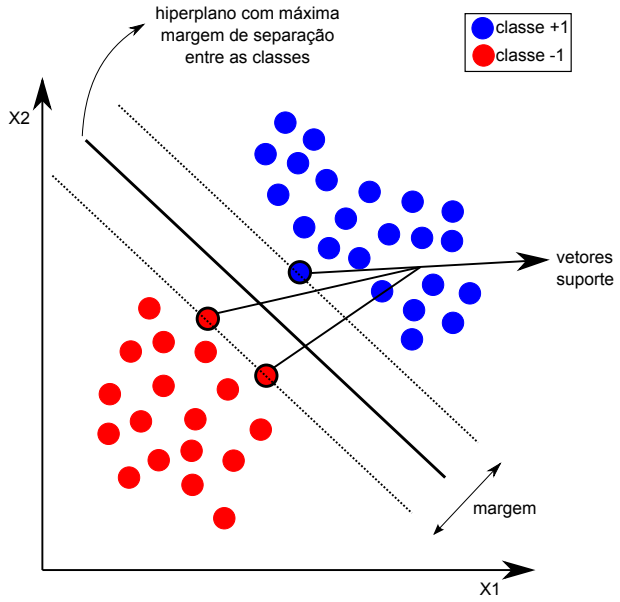

Figura 29 – Hiperplano com máxima margem de separação entre as classes.

onde $\alpha_i \in \mathbb{R}$, $\alpha_i \geq 0$ e $y_i \in \{-1, +1\}$. Isto é, a solução é escrita em termos dos vetores de treinamento. Mais precisamente, o hiperplano de separação é definido pelos vetores que se encontram na margem de classificação. Tais amostras, chamadas de vetores suporte, carregam toda a informação relevante a respeito do problema de classificação e apresentam $\alpha_i > 0$.

Um detalhe importante a respeito do problema de otimização quadrática utilizado para maximizar a margem do hiperplano é que ele depende apenas do produto interno das amostras. Dessa maneira, a função de decisão é definida como:

$$f(\mathbf{x}) = sign(\sum_i \alpha_i y_i (\mathbf{x} \cdot \mathbf{x}_i) + b), \tag{B.5}$$

onde $sign(\cdot)$ é a função sinal e $\mathbf{x}_i$ são os vetores suporte encontrados na aprendizagem. Como a aprendizagem e a decisão são realizadas apenas a partir dos produtos internos das amostras, é possível a generalização das SVMs para problemas não-lineares.

A ideia básica por trás das SVMs consiste em realizar um mapeamento dos dados para outro espaço de dimensão mais alta, F, chamado de espaço de características (*feature space*), utilizando um mapeamento não-linear:

$$\phi : \mathbb{R}^D \rightarrow F, \tag{B.6}$$

e realizar o aprendizado e a decisão no espaço F, ao invés do espaço original. Esse mapeamento é realizado de modo que as classes possam ser separadas por um hiperplano no novo espaço, fato que não seria possível no espaço original.

Como geralmente a dimensão de F é alta, o cálculo dos produtos internos nessa dimensão é bastante custoso computacionalmente. SVMs utilizam, então, as chamadas funções de *kernel* para realizar essas operações de maneira eficiente. Além disso, o mapemento não

linear proporcionado por tais funções permite a geração, no espaço original, de fronteiras de decisão não lineares. Mais precisamente, escolhem funções de *kernel* que possam ser escritas da seguinte maneira:

$$k(\mathbf{x}, \mathbf{y}) = (\phi(\mathbf{x}) \cdot \phi(\mathbf{y})), \tag{B.7}$$

isto é, funções que possam ser escritas em termos do produto interno na dimensão mais alta, F.

Utilizando funções desse tipo, SVMs realizam a otimização quadrática no espaço F sem de fato realizar nenhuma computação nesse espaço, e sim, no espaço original, utilizando as funções de *kernel* correspondentes.

Pode-se provar que, para toda função de *kernel* cuja matriz

$$(k(\mathbf{x}_i, \mathbf{x}_j))_{i,j} \tag{B.8}$$

é positivamente definida, existe um mapeamento ϕ correspondente e essa função pode ser descrita na forma apresentada na Equação B.7. Nesse tipo de situação, basta substituir os produtos internos $(\phi(\mathbf{x}) \cdot \phi(\mathbf{y}))$ do problema de otimização quadrática pelas funções de *kernel* para encontrar o hiperplano de separação no espaço F.

A decisão sobre uma amostra de teste é, portanto, realizada a partir do cálculo da função de *kernel* e os vetores suporte a partir da seguinte função de classificação:

$$f(\mathbf{x}) = \sum_i \alpha_i y_i (k(\mathbf{x}, \mathbf{x}_i)) + b. \tag{B.9}$$

Vetores que apresentam $f(\mathbf{x}) \geq 0$ são classificadas como a classe $+1$, enquanto que o restante deles é classificado como pertencente à classe -1. Além dessa classificação absoluta utilizando o sinal de $f(\mathbf{x})$, seu valor pode ser utilizado como *score* e a classificação seria realizada comparando-o com outro limiar diferente de zero.

A função de *kernel* deve ser apropriada para cada problema, porém, as funções mais comumente utilizadas são:

- o *kernel* polinomial (de ordem n):

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^n; \tag{B.10}$$

- o *kernel* de função de base radial (RBF):

$$k(\mathbf{x}, \mathbf{y}) = e^{\frac{-\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}; \tag{B.11}$$

- e o *kernel* sigmoidal (com ganho χ e deslocamento θ):

$$k(\mathbf{x}, \mathbf{y}) = \tanh(\chi(\mathbf{x} \cdot \mathbf{y}) + \theta). \tag{B.12}$$

APÊNDICE C – ESTIMAÇÃO DA MATRIZ DE *EIGENVOICES* NA MODELAGEM JFA

Como visto na Seção 2.4.5, na modelagem JFA, um determinado supervetor $\boldsymbol{M}$, definido através da adaptação de um UBM com C componentes de mistura e vetores de dimensão D, segue uma distribuição cuja decomposição em variáveis latentes é definida por:

$$\boldsymbol{M} = \boldsymbol{m} + \boldsymbol{V}\boldsymbol{y} + \boldsymbol{R}\boldsymbol{z} + \boldsymbol{U}\boldsymbol{x}, \tag{C.1}$$

onde $\boldsymbol{m}$ é um supervetor de dimensão CD, $\boldsymbol{V}$ e $\boldsymbol{U}$ são matrizes retangulares de *rank* reduzido, $\boldsymbol{R}$ é uma matriz diagonal e $\boldsymbol{y}$, $\boldsymbol{z}$ e $\boldsymbol{x}$ seguem a distribuição normal padronizada $N(0, \boldsymbol{I})$ correspondente a cada dimensionalidade. O supervetor $\boldsymbol{m}$ é o supervetor médio, independente de locutor e de sessão, e geralmente é extraído do UBM. Enquanto $\boldsymbol{V}$ é a matriz de *eigenvoice* que define a projeção para o subespaço referente ao locutor, $\boldsymbol{V}$ é a matriz de *eigenchannels* que define o subespaço correspondente à sessão e $\boldsymbol{R}$ define a projeção para o subespaço residual.

As matrizes retangulares de decomposição são estimadas utilizando os dados disponíveis, que são categorizados através das informações dos locutores. Primeiramente, a matriz de *eigenvoice* é estimada assumindo que $\boldsymbol{R}$ e $\boldsymbol{U}$ são nulos. No segundo passo, um método semelhante ao utililizado para estimação de $\boldsymbol{V}$ é empregado para $\boldsymbol{m}$ e $\boldsymbol{R}$. E por fim, as variabilidades intra-locutor observadas entre os supervetores $\boldsymbol{M} - \boldsymbol{C}\boldsymbol{x}$ são utilizadas para estimação da matriz de *eigenchannels*. Neste trabalho, vamos nos limitar à estimação apenas da matriz de *eigenvoice* porque ela é a única utilizada na modelagem com vetores-identidade (*i-vectors*). O método foi definido em (KENNY, 2005) e (KENNY et al., 2008), e a estimação é realizada através do algoritmo iterativo EM (*Expectation Maximization*).

Considere que as amostras disponíveis para treinamento consistem de vetores de características extraídos de locuções provenientes de diferentes locutores. Dessa maneira, dado o conjunto de locutores L, para cada um determinado locutor $l \in L$, temos um conjunto de vetores de características $\boldsymbol{X_l}$.

O primeiro passo consiste em extrair as estatísticas de primeira e segunda ordem de cada um dos C componentes de mistura do UBM[1], para cada um dos locutores:

$$N_{lc} = \sum_{\boldsymbol{x} \in \boldsymbol{X_l}} Pr(c|\boldsymbol{x}), \tag{C.2}$$

$$\boldsymbol{F_{lc}} = \sum_{\boldsymbol{x} \in \boldsymbol{X_l}} Pr(c|\boldsymbol{x})\boldsymbol{x}, \tag{C.3}$$

[1] Assim como na modelagem GMM-UBM, Equações 2.35-2.37.

$$\boldsymbol{S_{lc}} = \text{diag}\left(\sum_{\boldsymbol{x}\in\boldsymbol{X_l}} Pr(c|\boldsymbol{x})\boldsymbol{x}\boldsymbol{x}^t\right), \tag{C.4}$$

onde $Pr(c|\boldsymbol{x})$ é a probabilidade *a posteriori* da mistura c com respeito a $\boldsymbol{x}$ e $diag(\cdot)$ de uma matriz quadrada considera apenas a sua diagonal, zerando os outros valores. N refere-se à estatística de ordem nula, enquanto que $\boldsymbol{F}$ e $\boldsymbol{S}$ referem-se às estatísticas de primeira e segunda ordem, respectivamente. Além disso, observemos que $\boldsymbol{N_{lc}} \in \mathbb{R}$, $\boldsymbol{F_{lc}} \in \mathbb{R}^D$ e que $\boldsymbol{S_{lc}} \in \mathbb{R}^{D\times D}$. Em seguida, centralizam-se as estatísticas de primeira e segunda ordem com respeito às médias das componentes de mistura do UBM:

$$\widetilde{\boldsymbol{F_{lc}}} = \boldsymbol{F_{lc}} - N_{lc}\ \boldsymbol{\mu_c}, \tag{C.5}$$

$$\widetilde{\boldsymbol{S}}_{\boldsymbol{lc}} = \boldsymbol{S_{lc}} - \text{diag}\left(\boldsymbol{F_{lc}}\ \boldsymbol{\mu_c}^t + \boldsymbol{\mu_c}\boldsymbol{F_{lc}}^t - N_{lc}\ \boldsymbol{\mu_c}\boldsymbol{\mu_c}^t\right), \tag{C.6}$$

onde $\boldsymbol{\mu_c}$ é a média da componente c do UBM.

Todas estatísticas são então organizadas em matrizes, partindo para uma descrição que compacta as estatísticas de todas as misturas, facilitando a formalização do problema de minimização a ser resolvido. Considere o mesmo valor da estatística nula de uma determinada componente, c, para cada uma das dimensões dos vetores de características:

$$\hat{\boldsymbol{N}}_{\boldsymbol{lc}} = N_{lc}\boldsymbol{I}, \tag{C.7}$$

onde $\boldsymbol{I}$ é matriz identidade em $\mathbb{R}^{D\times D}$. As matrizes que compactam as estatísticas de todas as misturas são:

$$\boldsymbol{N_l} = \begin{bmatrix} \hat{\boldsymbol{N}}_{\boldsymbol{l1}} & & & \\ & \hat{\boldsymbol{N}}_{\boldsymbol{l2}} & & \\ & & \ddots & \\ & & & \hat{\boldsymbol{N}}_{\boldsymbol{lC}} \end{bmatrix}, \tag{C.8}$$

$$\boldsymbol{F_l} = \begin{bmatrix} \widetilde{\boldsymbol{F}}_{\boldsymbol{l1}} \\ \widetilde{\boldsymbol{F}}_{\boldsymbol{l2}} \\ \vdots \\ \widetilde{\boldsymbol{F}}_{\boldsymbol{lC}} \end{bmatrix}, \tag{C.9}$$

e

$$\boldsymbol{S_l} = \begin{bmatrix} \widetilde{\boldsymbol{S}}_{\boldsymbol{l1}} & & & \\ & \widetilde{\boldsymbol{S}}_{\boldsymbol{l2}} & & \\ & & \ddots & \\ & & & \widetilde{\boldsymbol{S}}_{\boldsymbol{lC}} \end{bmatrix}, \tag{C.10}$$

onde $\boldsymbol{N_l} \in \mathbb{R}^{CD \times CD}$, $\boldsymbol{F_l} \in \mathbb{R}^{CD}$ e $\boldsymbol{S_l} \in \mathbb{R}^{CD \times CD}$.

A seguir realizam-se repetidas iterações entre as etapas de expectativa (*expectation*) e maximização (*maximization*). Partindo de uma estimativa inicial para $\boldsymbol{V}$, que realiza o mapeamento entre o supervetor (aqui expresso através das estatísticas de primeira ordem, $\boldsymbol{F_l}$) e os fatores de eigenvoice, $\boldsymbol{y_l} \in \mathbb{R}^E$, onde E é a dimensão dos fatores. Dessa maneira, podemos expressar a minimização como:

$$\mathrm{V} = \underset{\boldsymbol{V} \in \mathbb{R}^{CD \times E}, \boldsymbol{y_l} \in \mathbb{R}^E}{\arg\min} ||\boldsymbol{F_l} - \boldsymbol{V}\boldsymbol{y_l}||^2. \tag{C.11}$$

Na fase de expectativa, encontra-se o valor esperado da distribuição *a posteriori* de $\boldsymbol{y_l}$ dado $\boldsymbol{V}$. A verossimilhança de $\boldsymbol{V}$ com respeito a $\boldsymbol{y_l}$ é:

$$\boldsymbol{L_{Vl}} = \boldsymbol{I} + \boldsymbol{V}^t \boldsymbol{\Sigma}^{-1} \boldsymbol{N_l} \boldsymbol{V}, \tag{C.12}$$

onde $\boldsymbol{I}$ é a matriz identidade e $\boldsymbol{\Sigma}$ é construída pela concatenação das matrizes diagonais de covariância do UBM (assim como na Equação C.10).

Assumindo a distribuição normal da probabilidade *a posteriori* de $\boldsymbol{y_l}$ com respeito a $\boldsymbol{V}$, temos:

$$\boldsymbol{y_l} \sim N(\boldsymbol{L_{Vl}^{-1}} \boldsymbol{V^t} \boldsymbol{\Sigma^{-1}} \boldsymbol{F_l},\ \boldsymbol{L_{Vl}^{-1}}) \tag{C.13}$$

e o seu valor esperado é expresso por:

$$\bar{\boldsymbol{y}}_{\boldsymbol{l}} = \mathbb{E}[\boldsymbol{y_l}] = \boldsymbol{L_{Vl}^{-1}} \boldsymbol{V^t} \boldsymbol{\Sigma^{-1}} \boldsymbol{F_l}. \tag{C.14}$$

Já na etapa de maximização, para cada mistura do UBM, acumulam-se as estatísticas dos diferentes locutores:

$$N_c = \sum_{l \in L} N_{lc}, \tag{C.15}$$

$$\boldsymbol{A_c} = \sum_{l \in L} N_{lc} \boldsymbol{L_{Vl}}^{-1}, \tag{C.16}$$

e

$$\mathbb{C} = \sum_{l \in L} \boldsymbol{F_l} \bar{\boldsymbol{y}}_{\boldsymbol{l}}, \tag{C.17}$$

onde $\mathbb{C}$ pode ser exposta em função das componentes do UBM:

$$\mathbb{C} = \begin{bmatrix} \mathbb{C}_1 \\ \mathbb{C}_2 \\ \vdots \\ \mathbb{C}_C \end{bmatrix}. \tag{C.18}$$

E por fim, computa-se uma nova estimativa para $\mathbf{V}$ utilizando as estatísticas acumuladas dos locutores:

$$\mathbf{V} = \begin{bmatrix} \boldsymbol{A_1}^{-1}\mathbb{C}_1 \\ \boldsymbol{A_2}^{-1}\mathbb{C}_2 \\ \vdots \\ \boldsymbol{A_C}^{-1}\mathbb{C}_C \end{bmatrix}. \tag{C.19}$$

As etapas de expectativa e maximização (Equações C.15-C.19) são executadas até um critério de parada ser alcançado (como a estabilização do valor esperado da probabilidade *a posteriori*, Equação C.14). Empiricamente, observou-se que algo em torno de vinte iterações são suficientes (KENNY et al., 2008).

Definida matriz de mapeamento $\mathbf{V}$, os fatores *eigenvoices* de uma determinada locução, $\mathbf{y}$, são definidos através da estimativa do seu valor médio (Equação C.14):

$$\boldsymbol{y} = \left(\boldsymbol{I} + \boldsymbol{V^t\Sigma^{-1}NV}\right)^{-1}\left(\boldsymbol{V^t\Sigma^{-1}F}\right), \tag{C.20}$$

onde $\boldsymbol{N}$ e $\boldsymbol{F}$ são as matrizes definidas nas Equações C.8 e C.9, respectivamente, utilizando as estatísticas de ordem nula e primeira ordem dos vetores de características que compõem a locução, em relação ao UBM.