Victor de Abreu Campos

Arcabouço para Reconhecimento de Locutor Baseado em Aprendizado Não Supervisionado

São José do Rio Preto
2017

Victor de Abreu Campos

Arcabouço para Reconhecimento de Locutor Baseado em Aprendizado Não Supervisionado

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Campus de São José do Rio Preto.

Financiadora: FAPESP – Proc. 2015/07934-4

Orientador: Prof. Dr. Daniel Carlos Guimarães Pedronette

São José do Rio Preto
2017

Campos, Victor de Abreu.
Arcabouço para reconhecimento de locutor baseado em aprendizado não supervisionado / Victor de Abreu Campos . -- São José do Rio Preto, 2017
85 f. : il., tabs.

Orientador: Daniel Carlos Guimarães Pedronette
Dissertação (mestrado) – Universidade Estadual Paulista “Júlio de Mesquita Filho”, Instituto de Biociências, Letras e Ciências Exatas

1. Computação. 2. Recuperação da informação. 3. Reconhecimento de padrões. 4. Reconhecimento automático da voz. 5. Fala – Reconhecimento automático. 6. Programas de aprendizado. I. Universidade Estadual Paulista "Júlio de Mesquita Filho". Instituto de Biociências, Letras e Ciências Exatas. II. Título.

CDU – 518.72

Ficha catalográfica elaborada pela Biblioteca do IBILCE
UNESP - Câmpus de São José do Rio Preto

Agradecimentos

Ao professor Daniel Carlos Guimarães Pedronette, pela dedicação, aconselhamento e incentivo às minhas iniciativas.

À minha família, que acreditou em mim e me apoiou em tempos de dificuldade.

À professora Kanae, que escutou minhas histórias e sempre ofereceu apoio moral.

Ao professor Carlos Norberto Fischer, que me ensinou sobre persistência e ousadia.

À todos que, direta ou indiretamente, contribuíram para minha formação.

À Fundação de Amparo à Pesquisa do Estado de São Paulo – FAPESP, pelo apoio financeiro (processo 2015/07934-4).

Resumo

A quantidade vertiginosa de conteúdo multimídia acumulada diariamente tem demandado o desenvolvimento de abordagens eficazes de recuperação. Nesse contexto, ferramentas de reconhecimento de locutor capazes de identificar automaticamente um indivíduo pela sua voz são de grande relevância. Este trabalho apresenta uma nova abordagem de reconhecimento de locutor modelado como um cenário de recuperação e usando algoritmos de aprendizado não supervisionado recentes.

A abordagem proposta considera Coeficientes Cepstrais de Frequência Mel (MFCCs) e Coeficientes de Predição Linear Perceptual (PLPs) como características de locutor, em combinação com múltiplas abordagens de modelagem probabilística, especificamente Quantização Vetorial, Modelos por Mistura de Gaussianas e *i-vectors*, para calcular distâncias entre gravações de áudio. Em seguida, métodos de aprendizado não supervisionado baseados em ranqueamento são utilizados para aperfeiçoar a eficácia dos resultados de recuperação e, com a aplicação de um classificador de K-Vizinhos Mais Próximos, toma-se uma decisão quanto a identidade do locutor.

Experimentos foram conduzidos considerando três conjuntos de dados públicos de diferentes cenários e carregando ruídos de diversas origens. Resultados da avaliação experimental demonstram que a abordagem proposta pode atingir resultados de eficácia altos. Adicionalmente, ganhos de eficácia relativos de até +318% foram obtidos pelo procedimento de aprendizado não supervisionado na tarefa de recuperação de locutor e ganhos de acurácia relativos de até +7,05% na tarefa de identificação entre gravações de domínios diferentes.

Palavras-chave: MFCC. PLP. VQ. GMM. i-vector. RL-Sim. ReckNN. reconhecimento de locutor. aprendizado não supervisionado.

Abstract

The huge amount of multimedia content accumulated daily has demanded the development of effective retrieval approaches. In this context, speaker recognition tools capable of automatically identifying a person through their voice are of great relevance. This work presents a novel speaker recognition approach modelled as a retrieval scenario and using recent unsupervised learning methods.

The proposed approach considers Mel-Frequency Cepstral Coefficients (MFCCs) and Perceptual Linear Prediction Coefficients (PLPs) as features along with multiple modelling approaches, namely Vector Quantization, Gaussian Mixture Models and i-vector to compute distances among audio objects. Next, rank-based unsupervised learning methods are used for improving the effectiveness of retrieval results and, based on a K-Nearest Neighbors classifier, an identity decision is taken. Several experiments were conducted considering three public datasets from different scenarios, carrying noise from various sources. Experimental results demonstrate that the proposed approach can achieve very high effectiveness results. In addition, effectiveness gains up to +318% were obtained by the unsupervised learning procedure in a speaker retrieval task. Also, accuracy gains up to +7,05% were obtained by the unsupervised learning procedure in a speaker identification task considering recordings from different domains.

Keywords: MFCC. PLP. VQ. GMM. i-vector. RL-Sim. ReckNN. speaker recognition. unsupervised learning.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

DFT	Discrete Fourier Transform
MFCC	Mel-Frequency Cepstral Coefficients
PLP	Perceptual Linear Prediction (coefficients)
VQ	Vector Quantization
GMM	Gaussian Mixture Model
UBM	Universal Background Model
SVM	Support Vector Machine
JFA	Joint Factor Analysis
DCT	Discrete Cosine Transform
EM	Expectation Maximization
WCCN	Within-Class Covariance Normalization
LDA	Linear Dicriminant Analysis
MAP	Mean Average Precision

Sumário

1 Introdução

Segundo (LEE et al., 2009), a análise e compreensão de conteúdo de áudio se tornou um dos maiores desafios dos tempos atuais. Dentre os vários níveis de informação presente em gravações de áudio, a extração de informações dependentes de locutor representa um foco de pesquisa há tempos, desde de trabalhos antigos como (WOLF, 1972; BURTON, 1987) a trabalhos atuais (SNYDER; GARCIA-ROMERO; POVEY, 2015; LEI et al., 2014; RICHARDSON; REYNOLDS; DEHAK, 2015; ALSULAIMAN; MAHMOOD; MUHAMMAD, 2017). Em parte atribui-se importância à esse tipo de conteúdo por sua frequente utilização dentro dos principais meios de comunicação. Essa condição ainda é realçada em meio aos constantes avanços tecnológicos na área de comunicação, fator que alavanca a produção diária de grandes quantidades de áudios sob diversas configurações: programas de televisão, rádio, diálogos por telefone ou vídeos *online*.

Portanto, a compreensão dessa vasta quantidade de dados de áudio se torna um pivô para o controle e organização de um dos principais meios de transmissão de informação. Em meio a esse contexto propõe-se o arcabouço descrito neste trabalho.

Um arcabouço para reconhecimento automático de locutor promove a integração entre funções de caracterização de agente locutor e métodos de modelagem probabilísticos com o objetivo de comparar e classificar gravações de áudio em termos de agente locutor. As funcionalidades propostas por tal sistema abrem novas oportunidades de automação de tarefas de análise e processamento de dados exclusivamente por conteúdo. São exemplos de aplicação a detecção e organização de interlocutores em faixas de áudio (SENOUSSAOUI et al., 2014), segmentando de forma automática falas de interlocutores em uma reunião ou recuperando de faixas de locutores alvos dentro de gravações longas (KINNUNEN; LI, 2010). O reconhecimento automático de locutor pode promover o aperfeiçoamento da interface entre sistemas inteligentes e usuários, ou ainda, capacitar o desenvolvimento de conjuntos de dados de outras pesquisas, como pela preparação de dados para treinamento de sistemas reconhecedores de fala especializados por locutor (HUANG; LEE, 1993; ABDEL-HAMID; JIANG, 2013; DODDIPATLA; HASAN; HAIN, 2014).

Uma propriedade relevante no processo de reconhecimento de locutor consiste no relacionamento entre fala e texto, isto é, se o locutor recita um texto predeterminado, ou se sua fala não depende de contexto e suas gravações possuem texto irrestrito. Enquanto o reconhecimento dependente de texto apresenta resultados mais confiáveis, uma vez que se espera uma palavra ou texto-chave sua modelagem e comparação possui mais restrições, essa abordagem depende da colaboração do locutor. O reconhecimento independente de texto é abrangente e pode ser aplicado inclusive no contexto forense, sem que o locutor

esteja ciente do reconhecimento.

A comparação entre gravações pode ser aplicada em três âmbitos, ou tarefas, principais:

- *identificar* um agente locutor desconhecido;
- *verificar* uma hipótese de identidade para um agente locutor desconhecido;
- *recuperar* áudios por semelhança de agente locutor.

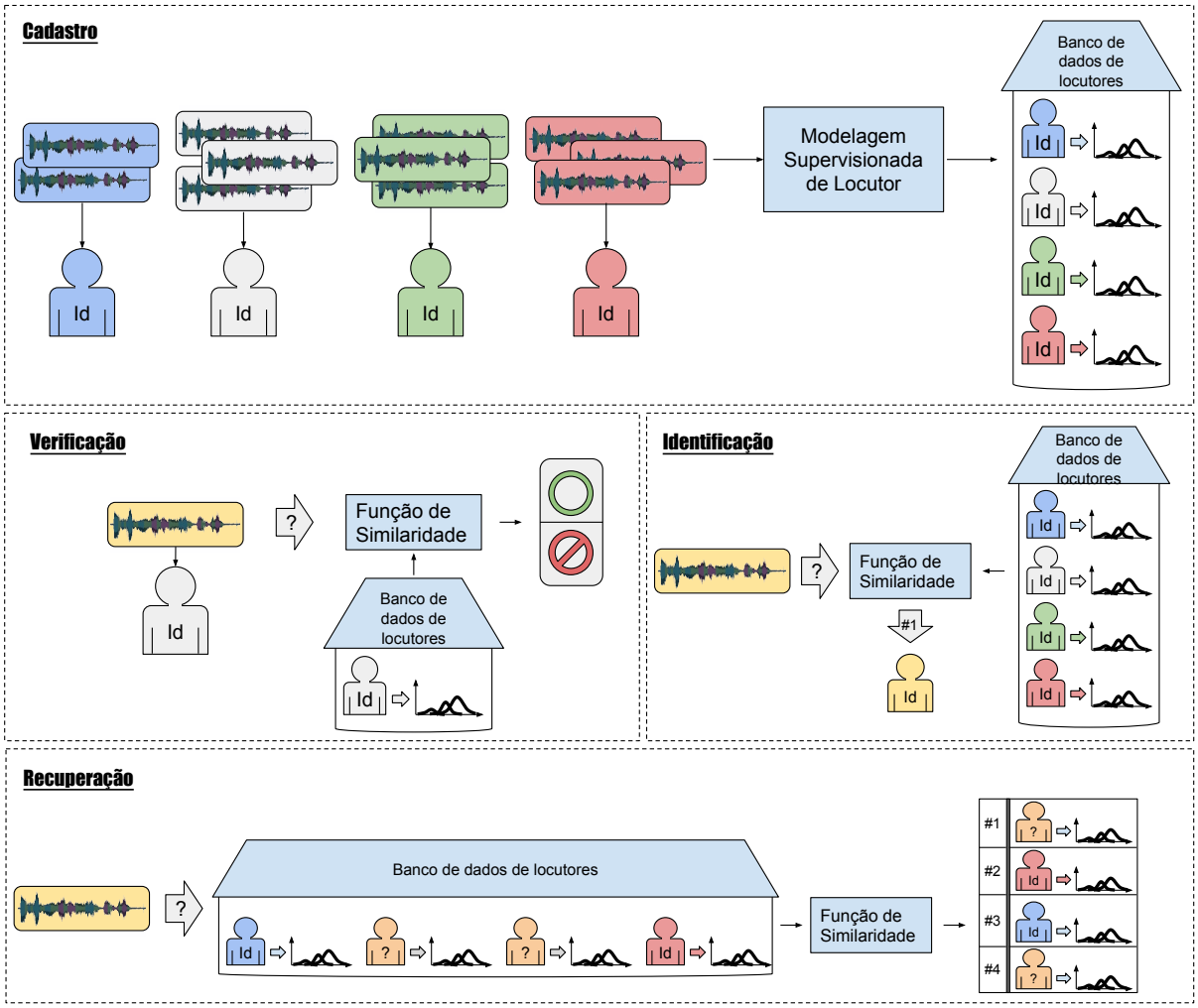
A *identificação* e a *verificação* de agente locutor requerem um banco de dados de agentes locutores cadastrados, preparado anteriormente à realização de tarefas de reconhecimento de locutor. Dessa forma, podem ser calculados por meio da submissão de uma nova gravação de locutor desconhecido, na *identificação* uma lista de classificação de agentes locutores registrados ordenada por semelhança, ou na *verificação* um valor de pontuação de confiança para a hipótese de identidade fornecida.

Entre *identificar* e *verificar* um agente locutor presente em uma faixa de áudio, observa-se que a identificação pode ser baseada em tarefas de *recuperação* de agente locutor seguida pela classificação por maior semelhança. Em contrapartida, a verificação de agente locutor toma uma decisão de aprovação ou rejeição baseada em dois tipos de pontuação de semelhança: a pontuação entre o áudio de origem não verificada e o modelo de agente locutor alvo e a pontuação entre aquele e um modelo de agente locutor genérico (HANSEN; HASAN, 2015). As três tarefas principais do reconhecimento de agente locutor estão representadas na Figura 1.

Toda aplicação que produz uma identidade como resposta requer o nível mínimo de supervisão que forneça a ligação entre conjuntos de amostras e suas respectivas identidades. Esse processo supervisionado de exemplificação, intrínseco à qualquer arcabouço identificador ou verificador de locutor, é esquematizado na etapa de cadastro representada na Figura 1. Nele são treinados modelos probabilísticos para cada locutor com base em amostras dos mesmos. No contexto de aprendizado de máquina, supervisão se refere ao conhecimento prévio da origem ou destino dos dados, conforme o tipo de tarefa ou aplicação sobre os mesmos. Em outras palavras, a existência de conjuntos de dados e respectivas "respostas certas" para previsões ou classificações dos conjuntos. Métodos que fazem uso desse tipo de conhecimento para replicar o que lhes foi exemplificado são denominados métodos de aprendizado supervisionado. A efetividade desses métodos tem dependência completa com a robustez dos exemplos fornecidos durante seu treinamento.

O fator de variabilidade entre gravações de mesmo locutor é alto. Neste contexto, podem ser destacados como principais fatores de variabilidade o canal de captação, o ambiente de captação e as condições de saúde e tipo de esforço vocal do locutor. Assim,

Figura 1 – Os três âmbitos do reconhecimento de locutor.



para compensar o grau de variabilidade das amostras durante o processo de treinamento de modelos de locutor, busca-se fornecer para os métodos de modelagem o número máximo de amostras para, consequentemente, produzir de modelos mais robustos.

Dessa forma, promover a efetividade de um sistema reconhecedor de locutor consiste em coletar uma quantidade de amostras dos locutores alvo que seja grande o suficiente para representar características dependentes de cada locutor. Como o volume suficiente de amostras relevantes de cada locutor é desconhecido, o processo de cadastro motiva o treinamento de modelos únicos de locutor, cada um tendo sido treinado com o número máximo de amostras de características dos locutores alvos disponíveis.

Além da quantidade numerosa de amostras esperada para a criação do modelo probabilístico, também há grande expectativa sobre a integridade das mesmas. Os métodos de extração de características não podem garantir a representação fiel do locutor sob qualquer configuração, uma vez que a efetividade da extração de características é limitada se as condições do conteúdo inicialmente são faltosas ou ausentes.

A falta observada nas abordagens supervisionadas está no fato de que, apesar de garantir melhores resultados de classificação de locutores, exige-se uma boa carga de amostras robustas em prol de um cadastro de locutor bem-sucedido. Porém a coleta de conjuntos de dados de locutores robustos corresponde a um custo alto, por depender de supervisão humana.

Atualmente uma quantidade vertiginosa de gravações é disponível publicamente, mantidas sem supervisão sobre seu conteúdo. Consequentemente, estas gravações não trazem garantias quanto à qualidade de características de locutor que podem ser extraídas das mesmas. Elas podem carregam ruídos de diversas formas como sons de ambiente, diálogos entre múltiplos locutores e outras variações adicionadas pelo aparelho de gravação. Assim, em detrimento da ausência de rótulos e falta de confiabilidade sobre seu conteúdo, impede-se o uso dessas gravações na criação de modelos de locutor.

Entretanto, não obstante a ausência de rótulos, um agente humano possui a capacidade de analisar gravações com níveis variados de ruído e inferir, com certo grau de confiança, similaridades entre gravações quanto ao seus locutores. De forma semelhante, a tarefa de recuperação de locutor navega entre gravações em termos de locutor por meio da comparação e ranqueamento das mesmas, guiada exclusivamente pela similaridade de seus conteúdos.

Em contraste com a classificação na identificação ou verificação, a recuperação de locutor retorna somente uma lista que relaciona uma gravação de consulta às demais gravações pertencentes a uma coleção. Dado que a tarefa é baseada exclusivamente em conteúdo, ela não recorre do cadastro inicial por locutor. Cada gravação é responsável por representar a si mesma em termos de seu locutor.

Portanto, a recuperação de locutor utiliza um modelo de locutor "fraco", criado sem rótulos de identidade nem número considerável de amostras, porém que permite que gravações sejam recuperadas de uma coleção sob o critério exclusivo de seu conteúdo. Apesar da resposta produzida pela tarefa de recuperação não carregar garantia substancial em termos de confiabilidade quanto a compatibilidade dos locutores retornados, essa resposta não depende de rótulos provenientes de um pré-cadastramento de agentes locutores. Adicionalmente, existem algoritmos de aprendizado não supervisionado que podem ser aplicados como pós-processamento à recuperação de locutor.

Algoritmos de aprendizado não supervisionado buscam aperfeiçoar as relações de similaridade entre objetos alvo por meio de heurísticas que não se apoiam em interferência de humana. De forma específica, para o contexto de recuperação de locutor, as relações de similaridade entre conjuntos de gravações se mostraram efetivas quando utilizadas em conjunto para calcular novas pontuações de similaridade que aproximassem gravações de vizinhanças similares e afastassem gravações com vizinhanças discrepantes.

A união da tarefa de recuperação de locutor ao potencial de aperfeiçoamento trazido por um pós-processamento não supervisionado sugere a aplicabilidade de arcabouço de reconhecimento de locutor completamente livre de interferência humana.

Tomando por base o cenário da recuperação de locutor, uma variante da abordagem de reconhecimento supervisionado pode ser considerada. Ao invés de modelos únicos de locutor, cada gravação é utilizada individualmente para a criação de um modelo de locutor anônimo. Assim, múltiplos modelos de locutor podem ser gerados para o mesmo locutor e gravações não supervisionadas, isto é, sem rótulos, também podem contribuir para a recuperação de outras consultas. Dada uma gravação de consulta, esta abordagem produz uma lista de ranqueamento que então é processada por algoritmos de aprendizado não supervisionado. Por fim, caso existam rótulos de locutor entre algumas das gravações da coleção considerada, a identificação de locutor pode ser realizada para a consulta com o auxílio de um classificador de K-Vizinhos Mais Próximos. O classificador analisa a lista de ranqueamento, contabiliza as gravações com rótulo de locutor e retorna o rótulo mais provável.

1.1 Objetivos e Contribuições

Atualmente a abordagem de reconhecimento de locutor se baseia fortemente na criação de modelos altamente supervisionados e centralizados em cada locutor. Porém a fragilidade dessa modelagem é uma dificuldade reconhecida em meio à crescente pluralidade multimídia. Isso é constatado em (SHUM et al., 2014), no qual leva-se em conta a queda de eficácia dos sistemas de reconhecimento de locutor devido a mudanças de domínio (aparelho de gravação, ruídos, etc.) e são buscadas maneiras de unir o treinamento supervisionado de modelos de locutor com técnicas de adaptação não supervisionada de domínio.

Esse trabalho apresenta um arcabouço que une técnicas de caracterização e modelagem de locutor e algoritmos de aprendizado não supervisionado como uma alternativa viável de reconhecimento de locutor. Neste arcabouço o reconhecimento é abordado como uma tarefa de recuperação de gravações. Cada gravação é transformada em um modelo de locutor que atua como base de comparação por conteúdo para com outras gravações. O reconhecimento de locutor é então realizado pela busca e classificação dos modelos da base de maior semelhança, dada uma gravação de consulta. Conforme explorado adiante nos Capítulos 2 e 3, a partir do potencial de reclassificação de objetos multimídia por relacionamentos globais apresentado por algoritmos de aprendizado não supervisionado, um classificador de vizinhos mais próximos pode atribuir identidade à gravação alvo considerando apenas a frequência dos rótulos de modelos melhores classificados.

A contribuição principal deste trabalho se encontra na adição do aprendizado não supervisionado como técnica de pós-processamento à modelagens clássicas de reconheci-

mento de locutor. Os resultados iniciais do arcabouço demonstraram ganhos significativos de precisão em três conjuntos de dados públicos e foram relatados na forma de artigo publicado no *International Workshop on Multimedia Analysis and Retrieval for Multimodal Interaction (MARMI)*, realizado em conjunto com a *ACM International Conference on Multimedia Retrieval (ICMR)* 2016.

1.2 Organização do texto

No Capítulo 2 é dado um breve panorama dos temas relacionados ao reconhecimento de locutor e e reclassificação de objetos multimídia por aprendizado não supervisionado. Em seguida o Capítulo 3 aprofunda a descrição do arcabouço proposto, junto dos métodos e algoritmos considerados na concepção do mesmo. O Capítulo 4 descreve a avaliação experimental, que foi elaborada para medir e comparar a eficácia do arcabouço, principalmente em termos de desempenho da fusão entre métodos de reconhecimento de locutor e algoritmos de pós-processamento não supervisionados. Ainda são apresentados os resultados dos experimentos e discussões sobre os mesmos. Finalmente, no Capítulo 5 são levantadas as conclusões sobre o desenvolvimento do arcabouço e possibilidades para futuras direções de pesquisa.

2 Trabalhos Relacionados

A fala como percebida pelos seres humanos é produto da interpretação de sinais captados por nervos sensoriais localizados no ouvido interno. Os sinais captados pelos nervos sensoriais são fruto de sua excitação por diferentes frequências transportadas primeiramente pela vibração do ar, e tendo sua origem no agente produtor da fala (SMITH, 1997). Esse capítulo apresenta uma breve explanação sobre o processo de captação e armazenamento da fala por computadores. Como breve introdução, a Seção 2.1 aborda o tema de digitalização do sinal sonoro. Em seguida, na Seção 2.2, é discutida a importância do processamento do sinal sonoro em busca de dados que quantifiquem informações da fala em meio à pluralidade de ruídos presente em qualquer gravação. Continuando o processamento de áudio, a Seção 2.3 introduz métodos de criação de modelos probabilísticos de locutor, sobre os quais, destaca-se seu fator redutor de redundâncias como principal motivador. A Seção 2.4 aborda o tema do aprendizado não supervisionado aplicado à tarefa de recuperação de objetos multimídia. Finalmente, conclui-se o capítulo com a Seção 2.5, na qual são apresentadas coleções de dados utilizadas com frequência em tarefas de reconhecimentos de locutor.

2.1 Captação e armazenamento da fala

As ondas sonoras transportadas pelo ar são traduzidas em corrente elétrica por um microfone. A corrente então passa por um conversor analógico-digital, que quantifica-a por números inteiros. Essa quantificação é realizada em função do tempo, tomando um número fixo de amostras por segundo. O número de bits usados para quantificar o sinal sonoro e a taxa de amostragem são os dois aspectos que influenciam o nível de confiança e qualidade de captação de um som. Para que sejam evitadas perdas do sinal, um número suficiente de *bits* precisa ser dedicado à quantificação. Além disso, segundo o teorema de Nyquist, é condição suficiente para a conservação do sinal que a taxa de amostragem do sinal seja pelo menos o dobro da maior frequência que o compõe (SHANNON, 1949).

Esse tipo de representação compreende o conteúdo do som de forma geral, não somente a fala. Informações referentes ao canal formado pela somatória de todos os componentes presentes entre o agente locutor, seu ambiente e o instrumento usado para captação, compõem a gravação de áudio tanto quanto a fala. Em decorrência dos múltiplos componentes que interferem no processamento da fala, o sinal de áudio costumeiramente passa por um processamento que extrai vetores de características ligados à fala.

2.2 Caracterização de locutor

A extração de características da fala, dada uma gravação genérica, está sujeita a diversos fatores que influenciam em sua qualidade final. A possibilidade de um individuo propositalmente alterar sua voz em relação ao que lhe é comum, como em sussurros, gritos ou imitações, ou mesmo de forma involuntária como observado pelo efeito Lombard (compensação do esforço vocal e decorrência de poluição sonora) (LANE; TRANEL, 1971) e pela alteração natural da voz com o avanço da idade (DELIYSKI STEVE AN XUE, 2001). Por estar sujeita a esses diversos fatores, as características extraídas da fala diferem de outras biometrias como reconhecimento por impressão digital, íris ou características faciais (HANSEN; HASAN, 2015). Além de fatores referentes ao agente locutor, o ambiente de captação e o tipo de microfone acrescentam ruídos a suas gravações. O próprio sinal sonoro está sujeito a degradação, principalmente em suas frequências mais altas, pela sua natureza vibratória, que vai contra a inércia de seu fluído transportador.

Conforme discutido na literatura (WOLF, 1972), características ideais da fala devem se encaixar nos seguintes termos:

- alto fator de variação entre agentes locutores diferentes;
- baixo fator de variação entre amostras de mesmo agente locutor;
- robustez diante de ruídos e distorções;
- ocorrência frequente e natural na fala;
- fácil extração;
- difícil imitação;
- independente de fatores de saúde ou variações de longo termo.

As limitações observadas por essa lista demonstram a dificuldade que uma tarefa de reconhecimento pode apresentar. Tendo isso em vista, o reconhecimento automático de locutor não depende somente de características, mas também usa de modelos probabilísticos para tomar decisões com maior segurança. Na questão de características da fala, características espectrais de curto termo são usadas com predominância, não somente no reconhecimento de locutor, mas em diversas tarefas que processam o sinal da fala (KINNUNEN; LI, 2010).

2.2.1 Características espectrais de curta duração

Levando em conta a onipresença de ruídos em gravações e motivados pela percepção humana da fala, os métodos de extração de características focam na análise das frequências

audíveis a humanos, buscando maior definição para frequências baixas e amplificação de frequências mais altas do sinal. Para a análise das frequências formantes de um sinal usa-se a transfomada discreta de Fourier (*discrete Fourier transform (DFT)*) ou a predição linear (*linear prediction (LP)*) sobre janelas de tempo de curta duração do sinal. Estima-se que a divisão em janelas de tempo de curta duração (20 a 30 milissegundos) produza amostras estacionárias do sinal, providenciando maior definição para sua decomposição em frequências (KINNUNEN; LI, 2010). Também é tomada curta sobreposição (10 milissegundos) entre as janelas de tempo para se estabelecer maior visibilidade de sua variação temporal (TEKTRONIX, 2009). Uma vez que o sinal é recortado em janelas de tempo sobrepostas, segue-se para a análise das frequências encontradas de acordo com o método apropriado. Entre os métodos mais populares de extração para reconhecimento de locutor encontram-se os coeficientes cepstrais de frequência mel (MFCCs) e coeficientes cepstrais de predição linear perceptual (PLP). Eles são populares principalmente pela simplicidade de extração e por descrever de forma compacta o sinal da fala. Além dos parâmetros cepstrais, é comum a concatenação de parâmetros de velocidade e aceleração entre as características extraídas das formas. Estes parâmetros são conhecidos com *deltas* e *double-deltas*, e representam propriedades dinâmicas das características de curta duração (HANSEN; HASAN, 2015).

Mais detalhes sobre o processo de extração de características acústicas são abordados na seção 3.1.

2.2.2 Compensação de canal

Em situações práticas, durante a aplicação do reconhecimento de locutor, o sinal processado está sujeito aos ruídos introduzidos pelo canal de transmissão, tanto como o tipo de ambiente, a qualidade do microfone utilizado ou do canal de transmissão, são todos fatores significativos na alteração do sinal. Dessa forma, compromete-se a robustez das características extraídas, afastando-as dos termos citados anteriormente sobre características ideais. Como reportado em (KINNUNEN; LI, 2010; REYNOLDS; ROSE, 1995), técnicas de compensação de canal são mais importantes que a escolha do tipo de características extraídas. Portanto, técnicas de compensação de canal, ou de normalização do sinal, tem por objetivo mitigar essas variações. Entre elas, de método simples e utilizada com frequência, a subtração por média cepstral (*Cepstral Mean Subtraction*) busca calcular o cepstro do sinal e subtraí-lo por sua média. Todo canal de captura possui seu próprio ruído, que é constante, e portanto o cepstro de um sinal gravado também carrega essa informação. O cepstro pode ser representado, de forma simplista, como uma soma entre o sinal do locutor $x[t]$ e os efeitos do canal de captura $h[t]$:

$$f[t] = x[t] + h[t] \tag{2.1}$$

Como somente $h[t]$ é constante, a subtração de $f[t]$ pela sua média reduz o efeito do canal sobre o sinal, como observado em:

$$f[t] - \bar{f[t]} = (x[t] + h[t]) - (\bar{x[t]} + h[t]) = x[t] - \bar{x[t]} \tag{2.2}$$

Apesar de sua grande importância, se os dados processados não possuem muito ruído, esse tipo de compensação pode degradar a acurácia do sistema (HAUTAMÄKI; KINNUNEN; FRÄNTI, 2008).

Além da compensação de canal, parte do processo de extração de características consiste em filtrar somente os instantes da gravação em que de fato ocorre a fala. Técnicas de detecção de atividade de voz costumam ser utilizadas para filtrar gravações de maior duração das quais espera-se segmentos de silêncio.

A extração de características de áudio é um passo essencial para o processo de reconhecimento de locutor. Contudo, como discutido anteriormente, as características isoladas não configuram dados biométricos ideais para a identificação, uma vez que há muita variação nas mesmas pela perspectiva intra-locutor. Com isso em mente, técnicas para criação de modelos probabilísticos foram adaptadas para o propósito do reconhecimento de locutor. O processo de modelagem tem por objetivo compensar a imperfeição das características extraídas, encaixando-as em modelos matemáticos de eficácia comprovada.

Outro fator que deve ser considerado é o que diz respeito à existência de múltiplos locutores por gravação. As técnicas de modelagem expostas nas seções seguintes consideram somente um indivíduo presente por gravação alvo. A seguir são descritos os métodos mais comuns de modelagem de locutor.

2.3 Modelagem de Locutor

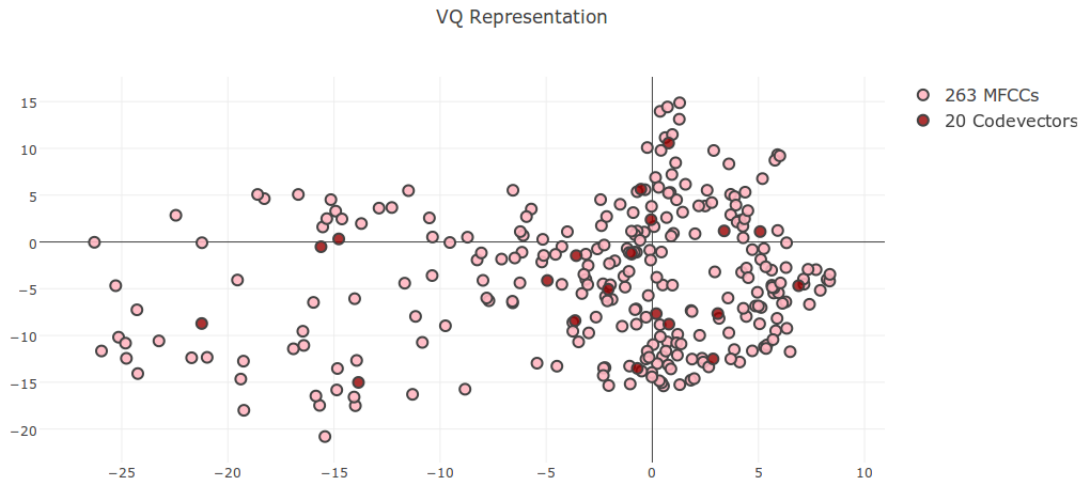
A extração de características de áudio é um passo essencial para o processo de reconhecimento de locutor. Contudo, como discutido anteriormente, as características extraídas não representam dados biométricos ideais para a identificação, uma vez que muita variação pode ser observada entre amostras de mesmo locutor. Assim, foram desenvolvidas técnicas de modelagem probabilística com o objetivo de analisar os vetores de características e reduzir redundâncias ou ruídos latentes.

2.3.1 Quantização vetorial

A modelagem por quantização vetorial (*vector quantization (VQ)*) consiste num dos métodos mais simples entre as modelagens clássicas (KINNUNEN; LI, 2010). Também utilizado na compressão de dados, o modelo gerado por esse método é composto por um conjunto de vetores de mesma dimensão dos vetores de amostra. Cada vetor desse conjunto

representa o centro de um agrupamento, ou centroide, calculado entre o espaço dos vetores de características. Como na compressão de dados, as amostras, ou vetores de características, são substituídos pelo seu centroide mais próximo. A distorção média, calculada como a média das distâncias entre cada amostra e seu respectivo centroide representante, é a utilizada para medir o erro de quantização de um modelo VQ. A similaridade ou distância entre um conjunto de vetores de características extraído para alguma gravação e um modelo de locutor gerado por VQ é calculado como a distorção média entre as amostras e seus respectivos centroides mais próximos. A Figura 2 representa o agrupamento realizado por esse método. Os centroides são também conhecidos como *codevectors* ou *codewords*, e o modelo formado por esse conjunto é nomeado *codebook*.

Figura 2 – Representação de duas dimensões de 263 vetores de MFCCs e seus codevectors gerados pela modelagem VQ.



Diversas abordagens podem ser tomadas para definir o número final de centroides e a forma como são calculados. Essa questão é discutida em (KINNUNEN; KILPELAINEN; FRANTI, 2011), porém de forma geral o algoritmo k-means pode ser usado e o número de centroides ajustado conforme o espaço amostral considerado. Apesar de não consistir num método utilizado atualmente com frequência em tarefas de reconhecimento de locutor, essa é de modelagem simples e eficaz para gravações caso não haja muito ruído ou diferenças de canal, como é constatado no Capítulo 4. A aplicação dessa modelagem é descrita em maiores detalhes na seção 3.2.1.

2.3.2 Modelos de mistura Gaussiana

Similar à modelagem por VQ, o GMM proposto em (REYNOLDS; ROSE, 1995) é um método de agrupamento não supervisionado do espaço de vetores de características. Em contraste com VQ, GMMs atribuem probabilidades não nulas para que um vetor

de amostra tenha se originado de cada um dos agrupamentos do modelo (KINNUNEN; LI, 2010). Um modelo GMM assume independência entre os vetores de características ao estimar curvas Gaussianas para representar os agrupamentos de dados. As curvas formam o modelo de locutor pela sua soma ponderada, conforme seus respectivos pesos de relevância e, dessa forma, representam a distribuição dos vetores de características de treinamento. Interpreta-se que pela fusão de Gaussianas é possível representar classes acústicas escondidas (pois não são usados rótulos) entre os vetores de características.

A densidade de uma mistura Gaussiana de M componentes dado um vetor aleatório $\vec{x}$ é uma superposição linear de Gaussianas defina pela equação:

$$p(\vec{x}|\theta) = \sum_{i=1}^{M} \pi_i b_i(\vec{x}), \tag{2.3}$$

na qual θ é o conjunto de parâmetros para as componentes $b_i, i = 1, ..., M$ e pesos de mistura $\pi_i, i = 1, ..., M$. Um modelo GMM é referenciado pelo seu conjunto de parâmetros:

$$\theta = \pi_i, \mu_i, \Sigma_i \;\; i = 1, ..., M \tag{2.4}$$

no qual μ_i e Σ_i são, respectivamente, as médias e covariâncias da componente Gaussiana b_i.

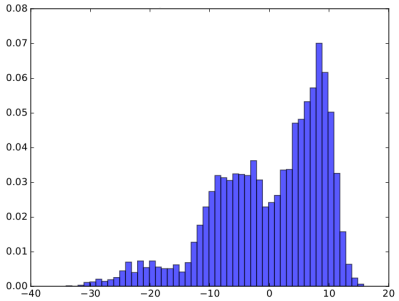
Essa modelagem, entre suas adaptações e fusões com outros métodos, se manteve parte do estado-da-arte (REYNOLDS; QUATIERI; DUNN, 2000; CAMPBELL et al., 2006; RICHARDSON; REYNOLDS; DEHAK, 2015) em reconhecimento de locutor independente de texto principalmente por possuir um alicerce probabilístico, métodos de treinamento escaláveis para grandes conjuntos de dados e boa acurácia no reconhecimento (CAMPBELL et al., 2006). A Figura 3 é uma representação comparativa entre as modelagens VQ e GMM sobre o mesmo conjunto de dados de treinamento.

Pode-se interpretar a abordagem GMM como um híbrido que une os centroides de VQ com a noção de densidade Gaussiana unimodal, que apresenta uma variância em torno da média, e dessa forma representa as características de um locutor com uma soma ponderada de Gaussianas, apresentando uma margem de erro suave em torno das médias. Uma comparação simplificada dessa modelagem pode ser feita com o algoritmo DFT, que representa um sinal complexo e imprevisível pela a soma previsível de senos. A seção 3.2.2 descreve em maiores detalhes a aplicação da modelagem GMM.

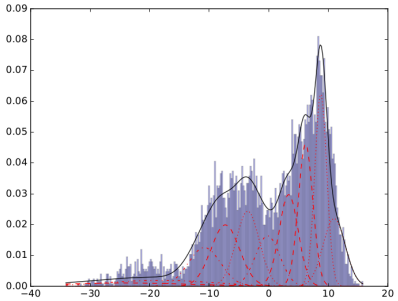
Uma evolução significativa para essa modelagem foi proposta em (REYNOLDS; QUATIERI; DUNN, 2000), nomeada GMM-UBM. Nesta nova abordagem, foi introduzido o *Universal Background Model* (UBM), um modelo de mistura Gaussiana independente de locutor e canal, treinado com centenas de horas de gravações de múltiplos locutores. Para um conjunto de características acústicas alvo de modelagem, o modelo específico

Figura 3 – Visualização de modelos de locutor sobre conjunto de características.

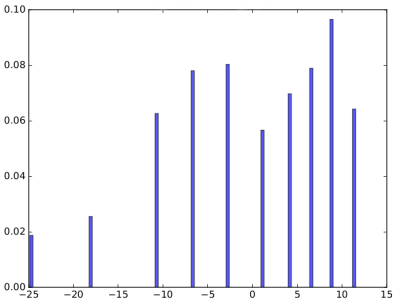
(a) Histograma de uma única dimensão de um conjunto de vetores de características



(b) Modelo GMM de 10 componentes gerado sobre o mesmo conjunto de vetores de características



(c) Histograma de 10 centroides gerados sobre o mesmo conjunto de vetores de características



de locutor é formado pela adaptação dos parâmetros do UBM. A adaptação é realizada pelo algoritmo *maximum a posteriori*. Essa abordagem se mostrou especialmente eficaz na verificação de locutor, por proporcionar um modelo universal de locutor que age como contraste para a semelhança calculada de um modelo específico de locutor alvo.

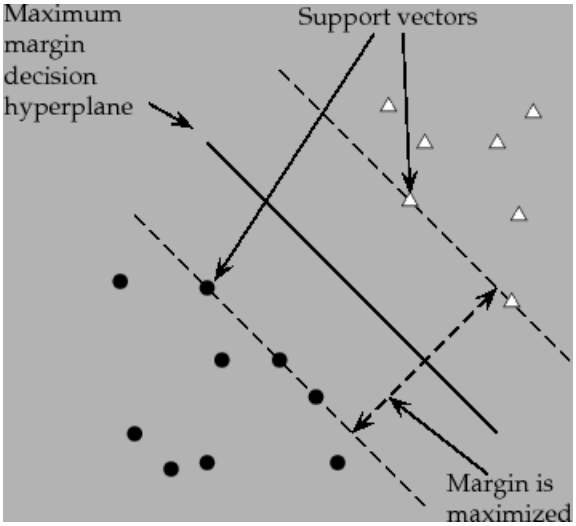
2.3.3 Modelos por máquinas de vetor suporte

Outro método relevante para verificação de locutor consiste na utilização do UBM para treinar uma máquina de vetor de suporte (SVM) para cada locutor. Segundo a proposta por (CAMPBELL et al., 2006), um UBM é utilizado da mesma forma para treinar modelos específicos de locutor, adaptando os valores de média global. Em seguida, os valores de média adaptados para um locutor são concatenados em um supervetor, de alta dimensionalidade, que por sua vez serve de entrada para um SVM (LEI et al., 2014).

SVMs são classificadores discriminativos, eles são normalmente utilizados para encontrar um separador entre duas classes na forma de um hiperplano (KINNUNEN; LI, 2010; DEHAK et al., 2009). O objetivo do treinamento de um SVM é o cálculo de um separador que maximize a distância de si próprio para qualquer ponto de conjunto de

treinamento (MANNING; RAGHAVAN; SCHÜTZE, 2008). A Figura 4 esquematiza um SVM de duas classes.

Figura 4 – Visualização do objetivo de um SVM. Fonte:(MANNING; RAGHAVAN; SCHÜTZE, 2008).



No reconhecimento de locutor, os supervetores de média do UBM agem como classe impostora enquanto que o supervetor de média do locutor alvo de treinamento, que foi adaptado do UBM, age como a classe do locutor (KINNUNEN; LI, 2010).

2.3.4 Modelos por análise fatorial

Junto com o crescimento da acessibilidade a grandes conjuntos de dados e capacidade de processamento computacional, o estado-da-arte em reconhecimento de locutor avançou no sentido de utilizar conjuntos de treinamento maiores para criar modelos resistentes a ruídos. Como já descrito, o modelo universal de locutor independente de locutor UBM é exemplo desse novo âmbito. Um UBM é um modelo GMM criado com centenas de horas de fala em contexto similar ao que espera-se reconhecer locutores. O modelo serve como base genérica para a criação de modelos de locutores que ressaltem as especificidades de acordo com discrepância entre os atributos de um conjunto de vetores de características e o UBM.

Seguindo os resultados positivos alcançados pela abordagem GMM-UBM, buscou-se por meio da análise fatorial reduzir a dimensionalidade dos modelos individuais, feitos a partir de supervetores de médias (concatenação dos vetores de média das componentes de um GMM). A nova modelagem propunha que um modelo dependente de locutor da modelagem GMM-UBM, representado por um supervetor μ_{loc}, seria formado pela combinação linear de fatores dependentes de locutor μ_s e fatores dependentes de canal μ_c, conforme a Equação 2.5. Neste contexto, um supervetor é um vetor formado pela

concatenação dos vetores de média de um GMM.

$$\mu_{loc} = \mu_s + \mu_c \tag{2.5}$$

Essa modelagem foi nomeada *Joint Factor Analysis* (JFA). Entretanto, em (DEHAK et al., 2011) foi identificado que o modelo de canal μ_c ainda continha informações dependentes de locutor, mostrando que locutores ainda poderiam ser identificados de certa forma com esse modelo de canal. Assim, foi proposta a união do espaço associado ao locutor e ao canal de JFA em um único, nomeado *Total Variability*. A nova combinação linear foi apresentada da seguinte forma:

$$\mu_s = \mu + T\vec{w} \tag{2.6}$$

Nessa nova modelagem, o vetor $\vec{w}$, nomeado *i-vector*, carrega os fatores dependentes de locutor e é uma representação de dimensionalidade reduzida em comparação com o supervetor de média dos GMMs. Na Equação 2.6, μ é o supervetor das médias concatenadas de um UBM.

Recentemente foi verificado que um UBM pode ser substituído por uma rede neural profunda que extraia estatísticas suficientes para a criação de *i-vectors* (RICHARDSON; REYNOLDS; DEHAK, 2015; LEI et al., 2014; SNYDER; GARCIA-ROMERO; POVEY, 2015).

2.3.5 Redes Neurais Profundas

A função do UBM para um sistema *i-vector* é providenciar o grau de associação de cada janela de tempo da gravação, ou vetor de características, as componentes do GMM, sendo cada componente representante de uma classe acústica latente. Essas estatísticas são durante o treinamento do sistema e utilizadas para formar o modelo de análise fatorial responsável pelo processo de redução de dimensionalidade (RICHARDSON; REYNOLDS; DEHAK, 2015). Tendo esta modelagem em vista, em (LEI et al., 2014) foi proposta a substituição do UBM por uma rede neural profunda treinada para o propósito de classificar as entradas em unidades sub-fonéticas, nomeadas *senones*, usadas no reconhecimento de voz. Neste novo arcabouço, as estatísticas de pertencimento do UBM são substituídas pelas probabilidades computadas pela rede neural profunda. Esse tipo de modelagem mostrou melhoria nos resultados de reconhecimento de locutor em comparação com a abordagem clássica com UBM.

Além de utilizadas para o treinamento de extratores de *i-vectors*, redes neurais profundas também são utilizadas para extrair características acústicas conhecidas como *bottleneck features* (RICHARDSON; REYNOLDS; DEHAK, 2015; ZHANG; CHUANG-

SUWANICH; GLASS, 2014). Nessa abordagem utiliza-se uma das camadas escondidas de uma rede neural profunda como o vetor de características acústicas. Uma vez que as camadas escondidas geralmente são de alta dimensionalidade, utiliza-se uma camada escondida com menos nós, que age como redutor de dimensionalidade. Esta é a camada *bottleneck*, da qual as ativações formam o novo vetor de características acústicas (RICHARDSON; REYNOLDS; DEHAK, 2015).

2.4 Recuperação de Informação Multimídia

A recuperação de informação pode ser definida, de forma genérica, como a área da computação responsável por recuperar automaticamente informações úteis ao usuário em uma coleção de dados. Embora haja singularidades de cada sub-área específica, o processo de recuperação pode considerar diferentes tipos de documento, seja ele em formato de imagem, texto, som ou outros dados multimídia.

Atualmente, temos vivenciado um cenário de significativas inovações tecnológicas nos processos de aquisição, armazenamento e compartilhamento de imagens, sons e vídeos. Como consequências de tais inovações, pode-se destacar o crescimento vertiginoso de coleções multimídia, atualmente acessíveis por meio de diversas tecnologias, assim como a expansão de aplicações. Nesse cenário, a demanda por métodos eficazes de recuperação, busca e reconhecimento é cada vez mais evidente.

Se originalmente a recuperação de dados multimídia era realizada por meio de análise textuais baseadas em metadados, estratégias atuais baseiam-se na análise direta do conteúdo multimídia. Dessa forma, os sistemas de recuperação baseados no conteúdo permitem a realização de buscas capazes de considerar o conteúdo dos objetos multimídia, analisando propriedades visuais ou sonoras. Tais sistemas possibilitaram o surgimento de inúmeras aplicações, inclusive aquelas relacionadas à identificação do locutor que constituem o foco desse trabalho.

De forma bastante ampla, um sistema de recuperação baseado no conteúdo pode ser modelado com o objetivo de recuperar objetos multimídia que estejam de acordo com as necessidades dos usuários, definidas por meio de especificações de consultas. Em geral, a definição da consulta é realizada utilizando-se um objeto (som, imagem, vídeo). Em seguida, o sistema busca os objetos mais similares na coleção de acordo com determinadas propriedades do seu conteúdo. O cenário de identificação de locutor, por exemplo, adéqua-se a esse modelo, considerando que o objeto de consulta é uma amostra da voz que se deseja identificar.

Em geral, a caracterização do conteúdo de um objeto multimídia é realizada em duas etapas: primeiro um algoritmo extrai características do objeto (codificando-as em vetores de características); e em seguida aplica uma função de distância ou modelo probabilístico

capaz de comparar as características. A similaridade entre dois objetos é geralmente calculada em função da distância de seus correspondentes vetores de características. Os objetos de uma dada coleção são ordenados em ordem crescente de distância, produzindo uma lista de resultados (*ranked list*).

Embora muito promissores, tais sistemas apresentam importantes desafios, dado que caracterizar o conteúdo multimídia é uma tarefa difícil. Nesse sentido, diversos esforços têm sido aplicados com o objetivo de aumentar a eficácia de tais sistemas. Grande parte desses esforços está relacionada ao uso de características mais eficazes. Outra vertente tem como foco a definição de funções de distância que sejam capazes de mensurar a distância entre vetores de características de maneira mais eficaz. Os sistemas de recuperação baseados no conteúdo têm evoluído apoiados pelo desenvolvimento de novos canais de transmissão e armazenamento, explorando novas características e medidas de distância.

No entanto, este modelo de distâncias se mostrou limitado ao relacionar características de baixo nível a conceitos de alto nível. Tais estratégias exploram apenas análises *par a par*, isto é, consideram o cálculo de medidas de distância considerando apenas pares de objetos, ignorando uma relevante fonte de informação codificada nos seus relacionamentos. Em contrapartida, o uso de informações contextuais pode representar importantes oportunidades para a recuperação de resultados mais eficazes. Os relacionamentos codificados nas distâncias e nas listas de resultados podem ser usadas para extrair informação contextual visando obter ganhos em eficácia.

Mais recentemente, visando resolver essa questão, novas estratégias de recuperação não diretamente relacionadas a aspectos de baixo nível têm sido aplicadas (LIU et al., 2007). Nesse cenário, métodos de pós-processamento estão atraindo atenção da comunidade científica, principalmente devido aos ganhos de eficácia significativos obtidos. Diversos métodos de aprendizado não supervisionado foram propostos, capazes de calcular uma medida de distância mais eficaz, sem a necessidade de intervenção do usuário (YANG; KOKNAR-TEZEL; LATECKI, 2009). O principal objetivo desses métodos é substituir o cálculo de distâncias par a par por medidas mais globais, capazes de analisar as coleções de forma geral (YANG; PRASAD; LATECKI, 2013). Vários métodos foram inicialmente desenvolvidos para recuperação de imagens, mas alguns já têm sido utilizados com sucesso em outros cenários de recuperação multimídia (ALMEIDA; PEDRONETTE; PENATTI, 2014).

Uma abordagem comum em tarefas de pós-processamento têm sido os processos de difusão (JIANG; WANG; TU, 2011; DONOSER; BISCHOF, 2013; YANG; KOKNAR-TEZEL; LATECKI, 2009). Tais algoritmos utilizam como entrada uma matriz de similaridade W, que pode ser interpretada como um grafo. O grafo $G = (V, E)$ relaciona n objetos entre si, consistindo de n nós $v_i \in V$ e arestas $e_{ij} \in E$, no qual as arestas são definidas pelos valores de similaridade w_{ij} . A partir das similaridade das arestas definidas

pela matriz W, os processos de difusão espalham as afinidades ao longo do grafo.

Embora os processos de difusão tenham alcançado importantes ganhos de eficácia, tais métodos são computacionalmente custosos, alcançando complexidade de $O(n^3)$. Como alternativa, métodos de reclassificação baseados em análise de ranqueamento têm sido propostos (SHEN et al., 2012; CHEN et al., ; PEDRONETTE; TORRES, 2013a; PEDRONETTE; TORRES, 2014b; BAI; BAI; WANG, 2015). Tais métodos são capazes de aliar significativos ganhos em eficácia à características relevantes como eficiência e escalabilidade (PEDRONETTE; ALMEIDA; TORRES, 2014; BAI; BAI; WANG, 2015). Dentre esses métodos, pode-se destacar o Algoritmo RL-Sim (PEDRONETTE; TORRES, 2013a) e o Algoritmo ReckNN (PEDRONETTE; PENATTI; TORRES, 2014), focos deste trabalho que serão discutidos nas próximas seções.

2.4.1 Algoritmo RL-Sim

O algoritmo RL-Sim (PEDRONETTE; TORRES, 2013a) é um método não supervisionado que visa aumentar a eficácia de tarefas de recuperação de objetos multimídia. Inicialmente, o algoritmo foi proposto para o cenário de reclassificação de imagens, porém a conjectura sobre a qual o algoritmo se baseia não é inerente ao tipo de objeto manipulado, o que torna ele aplicável sob diversos cenários (ALMEIDA; PEDRONETTE; PENATTI, 2014; ALMEIDA et al., 2016).

O RL-Sim explora a informação contextual codificada na similaridade entre listas de ranqueamento. Tais listas representam fonte relevante de informação, pois estabelecem relações entre os conjuntos de objetos contidos nelas de maneira global, e não apenas entre pares de objetos. Portanto, o algoritmo age em termos de informação de ranqueamento, o que torna ele aplicável em cenários de recuperação que se encaixem na conjectura de sua heurística.

O algoritmo apoia-se na conjectura de que, se dois objetos são similares, suas *top-k* posições nas listas de ranqueamento contém objetos semelhantes. O objetivo do algoritmo é aumentar a eficácia do sistema pelo cálculo de métricas de distância que considerem similaridades entre os conjuntos de gravações presentes nas *top-k* posições de dados dois objetos alvo.

O arcabouço proposto aplica o RL-Sim como técnica de pós-processamento, reclassificando áudios por similaridades em suas listas de ranqueamento. No Capítulo 3, define-se formalmente o algoritmo RL-Sim.

2.4.2 Algoritmo de grafos kNN recíprocos

Semelhante ao RL-Sim, o algoritmo de grafos kNN recíprocos (ReckNN) reclassifica objetos com base na informação contextual presente nas listas de ranqueamento. Este

algoritmo é uma técnica de *manifold learning* que calcula novas métricas de semelhança entre objetos através da geometria do conjunto de dados (PEDRONETTE; PENATTI; TORRES, 2014). Entretanto, o ReckNN busca alcançar mais objetos ao longo das listas de ranqueamento de um conjunto de dados ao unir conceitos de semelhança recíproca e grafos de vizinhos mais próximos.

Uma vez que o ReckNN age através de grafos de vizinhos mais próximos, a reclassificação ocorre de forma eficiente, evitando a analise de similaridade entre objetos mutualmente distantes. O algoritmo ReckNN é definido formalmente no Capítulo 3.

2.5 Coleções de Dados

Com o objetivo de possibilitar a realização de avaliações experimentais em tarefas de reconhecimento de locutor, diversas coleções de mídia na forma de áudio foram criadas. As coleções mais utilizadas em experimentos relatados na literatura da área são discutidos a seguir.

Essas são coleções de áudio amplamente utilizadas nos vários cenários do reconhecimento de locutor, cada uma com sua peculiaridade. O corpus TIMIT (JR; REYNOLDS et al., 1999) foi desenvolvido com o objetivo de servir de ferramenta experimental do reconhecimento automático de fala (JR; REYNOLDS et al., 1999). Assim, cada locutor do corpus só possui uma sessão de fala. Não é recomendado para avaliação de um sistema realista de reconhecimento de locutor pela boa condição do áudio colhido (gravado em estúdio sem ruídos de um ambiente prático), cenário que se afasta de muitas aplicações reais de um sistema desses (JR; REYNOLDS et al., 1999). Contudo, ainda vem sendo utilizado em trabalhos acadêmicos recentes (WANG, 2013).

O corpus YOHO (JR, 1995) foi gravado em um ambiente de escritório, com ruído de baixo nível. Ele é composto por locutores lendo dígitos, uma vez que foi produzido com a avaliação de sistemas de verificação de locutor dependente de texto. Foi utilizado recentemente em (WANG; JOHNSON, 2014). O corpus SWITCHBOARD (SWB) (GODFREY; HOLLIMAN; MCDANIEL, 1992) foi gravado por canal telefônico e possui conversações entre locutores. Avaliações de reconhecimento de locutor (SREs) organizadas pelo Instituto Nacional de Padrões e Tecnologia (em inglês, *National Institute of Standards and Technology* (NIST)) mais antigas (1998 e 1999) utilizaram um corpus experimental derivado do SWB (REYNOLDS; QUATIERI; DUNN, 2000) para formar o conjunto de dados de avaliação. Avaliações de reconhecimento de locutor mais recentes utilizaram coleções de dados próprias, coletadas pelo *Linguistic Data Consortium* (LDC), sendo trabalhos recentes de reconhecimento de locutor com frequência utilizam gravações do SWB ou de alguma SRE (HAUTAMÄKI; KINNUNEN; FRÄNTI, 2008; SENOUSSAOUI et al., 2010; SNYDER; GARCIA-ROMERO; POVEY, 2015; RICHARDSON; REYNOLDS;

DEHAK, 2015) .

O corpus KING (HIGGINS; VERMILYEA, 1995) foi gravado tanto por canal telefônico como por microfone e possui sessões entre 30s e 60s sobre descrições espontâneas de fotografias. O corpus MIXER (CIERI et al., 2004) foi usado durante as últimas avaliações SRE do NIST, ele é o único multi-lingual dos citados aqui e consiste de conversas por telefone fazendo uso de uma variedade de telefones diferentes, portanto acrescentando o fator de praticidade desse corpus (CIERI et al., 2004). O corpus VOICES (KAIN, 2001) produzido recentemente também consiste em diálogo telefônico e foi utilizado em (TZAGKARAKIS; MOUCHTARIS, 2010). Todos esses corpus são distribuídos pelo *Linguistic Data Consortium* (LDC). O acesso a essas coleções de dados é limitado a colaboradores do consórcio americano de dados linguísticos (LDC, 1992), ou mediante o pagamento de uma taxa que varia entre as coleções. Como alternativa viável, as seguintes coleções de acesso público também foram pesquisados e utilizados na avaliação experimental:

- CHAINS (GRIMALDI; CUMMINS, 2008);
- LapsBM1.4 (ALVES, s.d.);
- YouTube Speaker Dataset (SCHMIDT; SHARIFI; MORENO, 2014);

CHAINS é uma coleção que consiste leituras de fábulas e sentenças pré-determinadas. Além disso, as pronúncias foram gravadas em variadas modalidades:

- leitura de texto (SOLO/NORM);
- resumo dos textos de forma espontânea (RETELL);
- pares lendo textos tentando criar um diálogo sincronizado (SYNC);
- imitação de gravação (RSI);
- leitura acelerada (FAST);
- leitura sussurrada (WHSP).

Essa coleção possui gravações tomadas em dois cenários distintos: em estúdio de gravação ou em um ambiente de escritório silencioso. O primeiro cenário de gravação foi utilizado na leitura individual de forma natural (NORM), leitura sincronizada (SYNC) e na forma resumida (RETELL), e as demais no segundo cenário, realizado dois meses após o primeiro. O idioma do corpus é inglês americano. As diferentes modalidades de fala entre gravações são a principal vantagem desta coleção.

Também composto por gravações em inglês, a coleção de dados colhido do YouTube é uma sequência de vídeos de palestras do canal Google Tech Talk do YouTube (YOUTUBE,

2007). Cada locutor possui um vídeo de em média 30 minutos, no geral com bastante ruído. 74 dos locutores possuem mais de um vídeo, gravado sob diferentes condições. Sua qualidade numerosa e ampla variedade entre gravações formam fatores importantes de avaliação.

Representante da língua portuguesa, a coleção de dados LapsBM1.4, encurtado para Laps ao longo desse projeto, é usado pelo grupo FalaBrasil para avaliar sistema de reconhecimento de fala. A coleção é composta por 700 frases gravadas por homens e mulheres em ambiente de escritório e de duração curta. Essas condições se encaixam em um cenário de ocorrência frequente no mundo real, por isso a relevância desse conjunto.

Mais detalhes sobre as coleções de dados públicas são apresentados no Capítulo 4.

3 Arcabouço para Reconhecimento de Locutor Baseado em Aprendizado Não Supervisionado

O arcabouço desenvolvido neste projeto é descrito a seguir pela apresentação da organização e fluxo de dados, assim como os métodos que o compõem. O arcabouço proposto é baseado em um modelo genérico de ranqueamento, de forma que pode ser utilizado em tarefas de recuperação ou identificação de locutor.

O arcabouço é composto por cinco componentes esquematizadas na Figura 5: (i) caracterização de áudio; (ii) modelagem de locutor; (iii) consulta por locutor; (iv) aprendizado não supervisionado e; (v) identificação de locutor. Diferentes técnicas podem ser combinadas em cada componente. Neste projeto a caracterização de áudio foi feita com MFCCs e PLPs, a modelagem de locutor com VQs, GMMs e *i-vectors* e o aprendizado não supervisionado realizado com os algoritmos RL-Sim e ReckNN.

Como entrada, o arcabouço recebe uma coleção de gravações, cujos locutores não são necessariamente identificados. Os métodos utilizados pelo arcabouço assumem um único locutor por gravação.

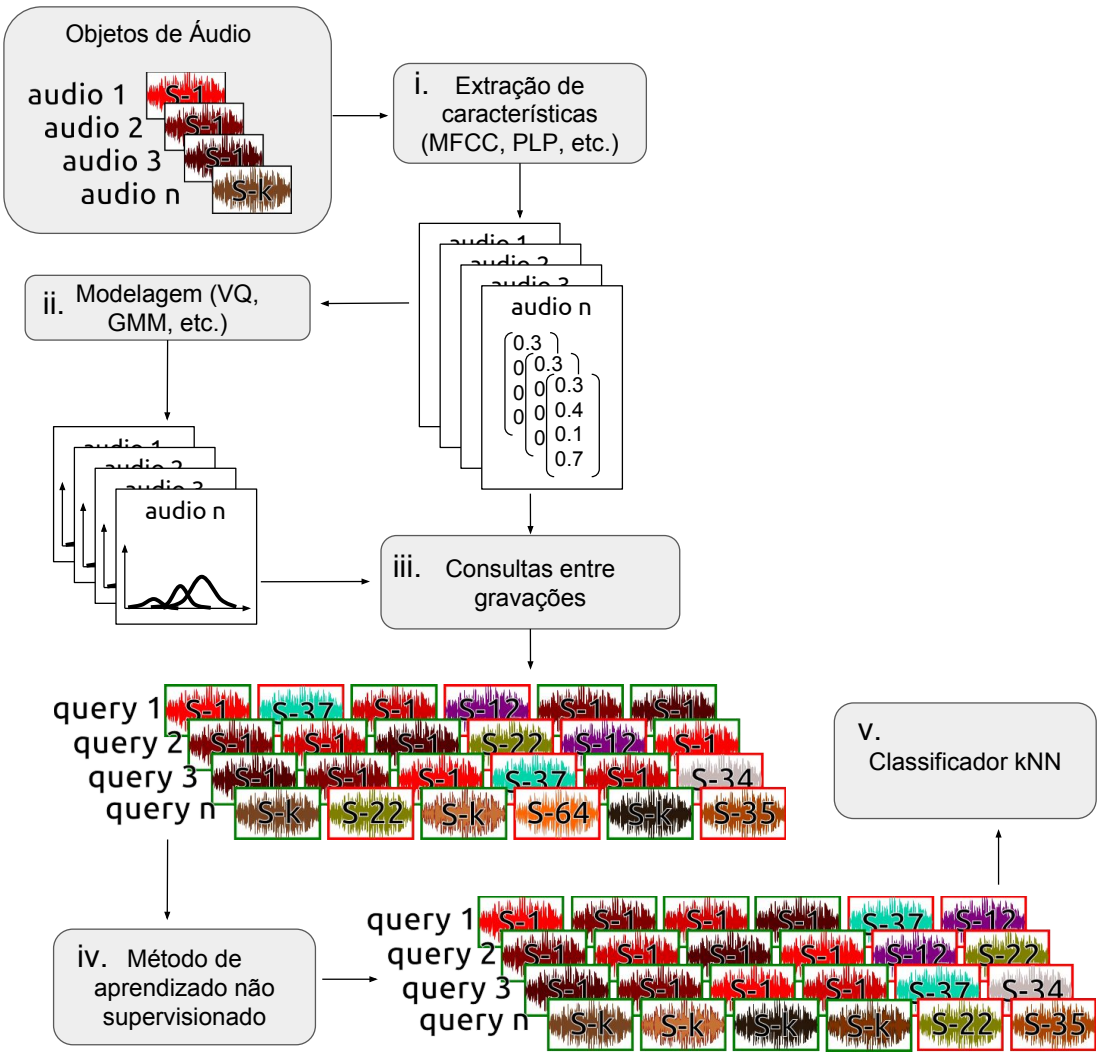
Tomando uma coleção inicial de dados com n gravações, estas passam pelo processo que modela cada uma conforme características dependentes de locutor, dessa forma gerando a coleção de modelos de locutores $\mathcal{M} = \{m_1^i, m_2^j, ..., m_n^k\}$. Os subscritos na representação $\mathcal{M}$ correspondem aos índices do conjunto de gravações, enquanto que os sobrescritos correspondem à identidade do locutor, caso esta seja conhecida. Destaca-se que os índices dos locutores $\{i, j, ..., k\}$ não precisam ser fornecidos, dado o caráter não supervisionado de todos os métodos empregados pelo arcabouço. Entretanto, a identificação de locutor, executada como último passo, naturalmente requer rótulos de locutor para que possa retornar alguma identificação.

As componentes do arcabouço são apresentadas a seguir.

i. **Função de caracterização de locutor:** a finalidade deste componente é, dada uma gravação, extrair de forma compacta e robusta características dependentes de locutor. As técnicas de caracterização, representadas pelo item (i.) da Figura 5, como MFCC, PLP, etc., reduzem o espaço amostral inicial, frequentemente ruidoso demais para a realização de tarefas de reconhecimento.

ii. **Modelagem de locutor:** como discutido no Capítulo 2, a extração de característi-

Figura 5 – Esquema do arcabouço para reconhecimento de locutor com aprendizado não supervisionado.



cas de locutor não são ideais devido à diversidade de fatores que podem prejudicar a análise da fala. Ainda que a extração de características de locutor fosse eficaz em quantizar perfeitamente seus atributos, os vetores de características acústicas produzidos são numerosos e carregam muitas redundâncias. Para possibilitar comparações mais eficazes entre gravações em termos de locutor, são utilizadas técnicas de modelagem de locutor, que contabilizam as características extraídas e calculam representações mais compactas e robustas do locutor alvo. Esse passo é representado no item (ii.) da Figura 5.

iii. Consultas entre gravações: nessa etapa calcula-se a similaridade entre as características de locutor e os modelos criados na etapa anterior, retornando para cada gravação uma lista de gravações ordenada por semelhança de locutor. Trata-se, portanto, de uma recuperação de locutor tomada cada gravação como consulta.

iv. Aprendizado Não Supervisionado: a funcionalidade principal deste projeto,

representada pelo item (iv.) da Figura 5, consiste na utilização de algoritmos de aprendizado não supervisionado como pós-processamento à recuperação de locutor. O objetivo desta etapa é aperfeiçoar a medição de similaridade entre gravações com base na informação contextual da coleção de gravações. O aprendizado não supervisionado é realizado com base nas listas de ranqueamento retornadas pela etapa anterior. Por meio da análise e comparação destas, infere-se novas conexões de proximidade entre gravações pelo contexto de cada consulta na coleção de gravações. Dessa forma, os valores de similaridade são recalculados com base na informação global da coleção de dados, frequentemente substituindo de forma positiva a similaridade inicial calculada na etapa anterior.

v. **Identificação:** Dependendo da tarefa desejada (recuperação/identificação) e a presença/ausência de gravações rotuladas, um classificador de k-vizinhos mais próximos pode ser aplicado à lista de ranqueamento e dessa forma retornar a identidade mais provável entre os locutores disponíveis no coleção de gravações.

A seguir são descritos os métodos usados para compôr as componentes do arcabouço. Como mencionado, o arcabouço não é limitado exclusivamente à esses métodos. Quaisquer métodos de caracterização e modelagem de locutor, assim como de aprendizado não supervisionado sobre listas de ranqueamento, podem ser combinados, com a condição que se mantenha o fluxo dados na sequência: dados de gravação; dados de locutor e; dados de ranqueamento por similaridade entre as gravações.

3.1 Extração de características

Inicialmente, as gravações da coleção alvo $\mathcal{C} = \{rec_1, rec_2, ..., rec_n\}$ são processadas em vetores de características, que reduzem redundâncias e ruídos presentes em qualquer tipo de gravação. Como convenção, a cada 10 milissegundos de gravação é computado um vetor de características, sendo por MFCCs ou PLPs. Como descrito na seção 2.2.1, essa segmentação permite que a análise do sinal seja feita por unidades fonéticas estáticas.

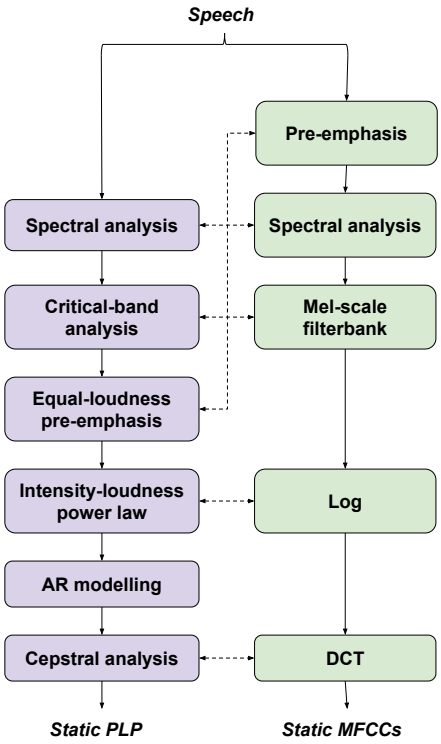
3.1.1 Coeficientes cepstrais de frequência Mel

A extração de MFCCs inicia-se com a pré-ênfase do sinal, na qual a densidade de potência das frequências é melhor distribuída entre suas amplitudes. Essa etapa age como compensação pela queda natural de potência das frequências mais altas, análogo ao realizado pelo cérebro humano (BEIGI, 2011). A seguir o sinal é transformado do domínio do tempo para o da frequência, processo conhecido como análise espectral, realizado pela DFT. Para que esta etapa seja realizada corretamente o sinal passa por uma função de janela, como a janela de Hamming, cujo objetivo é reduzir a descontinuidade nas

extremidades do sinal de cada janela de tempo. A descontinuidade, introduzida pelo processo de segmentação do sinal, deve ser mitigada em favor da implicação de uso da DFT, que assume a periodicidade do sinal. Uma vez no domínio da frequência, a informação de fase das frequências é desprezada e sua magnitude é deformada por um banco de filtros motivado pela acústica da audição humana. O banco de filtros é espaçado conforme a escala mel (STEVENS; VOLKMANN, 1940). Calcula-se o logaritmo das frequências acumuladas pelos bancos para modelar a relação não-linear entre a intensidade do som e o volume percebido. Os coeficientes cepstrais são por fim calculados com a transformada discreta de cosseno (*discrete cosine transform (DCT)*). Um diagrama desse processo é representado na Figura 6.

Figura 6 – Comparação entre a extração de coeficientes PLP e MFCC.

Adaptado de (MILNER, 2002)



3.1.2 Coeficientes cepstrais de predição linear perceptual

Como representado na Figura 6 os coeficientes por PLP são computados de forma análoga a MFCCs, visto a equivalência de suas etapas de processamento. Em comparação com MFCCs, PLP aplica um banco de filtros espaçado pela escala bark, e em seguida, como na pré-ênfase, as amplitudes dos bandas críticas denotadas pela escala bark são escaladas para compensar pela sensibilidade acústica desigual da biologia humana. Em seguida, segundo a lei de potencial publicada por Stevens (STEVENS, 1957), a intensidade do sinal é convertida em uma escala de percepção de volume, aproximada pela raiz cúbica

da frequência. Esta última etapa relaciona-se com a compressão logarítmica realizada nos MFCCs. Sobre o resultante aplica-se a transformada inversa DFT. O sinal transformado é usado como função de autocorrelação usada para gerar os coeficientes de predição linear pelo método Levinson-Durbin. Um algoritmo recursivo computa os coeficientes cepstrais de PLP, também conhecidos como PLP Cepstra (BEIGI, 2011).

Ao final do processo de extração de características, as gravações da coleção $\mathcal{C}$ são representadas pelo novo conjunto $\mathcal{C}_x = \{X_1, X_2, ..., X_n\}$, cada X_i sendo um conjunto de vetores de características de dimensão $D \times C$. D é definido em função da duração da gravação base e o tamanho do passo tomado entre as janelas de tempo (podendo também ser afetado pela aplicação de algum método de detecção de atividade de voz). A dimensão C varia conforme aplicações, de costume sendo formada pela concatenação dos primeiros 12-19 coeficientes cepstrais com seus parâmetros *delta* e *double-delta*.

3.1.3 Ferramentas de extração de características de locutor

As duas principais alternativas para extração de características de áudio são disponibilizadas no ambiente Matlab/Octave ou na ferramenta HTK (YOUNG et al., 2006), desenvolvida originalmente pela universidade de Cambridge. A implementação em Matlab/Octave foi desenvolvida por Daniel Ellis (ELLIS, 2005) permite melhor visualização do processo de extração, uma vez que utiliza funções de alto nível do ambiente. Portanto é uma implementação mais didática e de fácil adaptação. A ferramenta HTK é um *software* mais robusto, que dentre outras várias funcionalidades para processamento da fala, apresenta uma interface para extração de características acústicas. Essa ferramenta exige um arquivo de configuração como o da Figura 7.

Figura 7 – Arquivo de configuração para extração de características da fala.

(a) MFCC

```
SOURCEFORMAT = WAV
TARGETKIND = MFCC_E_D_A
TARGETRATE = 100000.0
SAVECOMPRESSED = F
SAVEWITHCRC = T
WINDOWSIZE = 250000.0
USEHAMMING = T
PREEMCOEF = 0.97
NUMCHANS = 24
CEPLIFTER = 22
NUMCEPS = 19
ENORMALISE = T
```

(b) PLP

```
SOURCEFORMAT = WAV
TARGETKIND = PLP_E_D_A
TARGETRATE = 100000.0
SAVECOMPRESSED = F
SAVEWITHCRC = T
WINDOWSIZE = 250000.0
USEHAMMING = T
PREEMCOEF = 0.97
NUMCHANS = 24
CEPLIFTER = 22
NUMCEPS = 19
ENORMALISE = T
USEPOWER = T
LPCORDER = 19
```

3.2 Modelagem de locutor

O processo seguinte consiste em transformar a nova representação do conjunto de vetores de características $\mathcal{C}_x$ em um conjunto de dados mais robusto e compacto, que irá configurar a base de modelos de locutor $\mathcal{M} = \{m_1^i, m_2^j, ..., m_n^k\}$. Apesar do termo modelagem de locutor estar fortemente relacionado à noção de treinamento supervisionado, ou cadastro de locutor, os métodos explorados nessa seção podem ser aplicados sem informações externas às gravações alvo, dessa forma podendo ser tratados como um passo extra de extração de características de locutor. A extração de i-vectors requer um modelo GMM de larga escala em seu processamento, entretanto a criação deste modelo também ocorre sem supervisão.

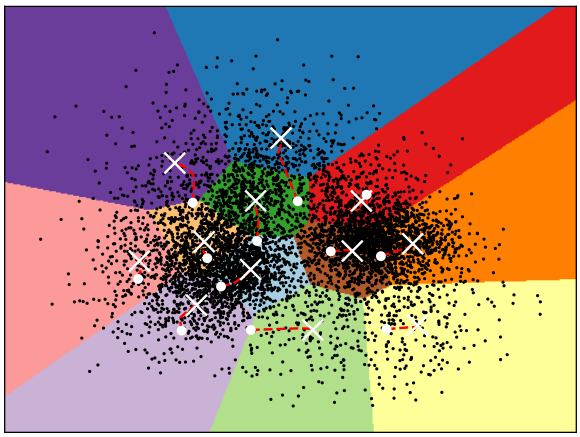
3.2.1 Quantização vetorial

O modelo de locutor VQ é o produto de uma compressão com perda de dados dos vetores de características da gravação alvo. Os vetores de uma gravação alvo são substituídos por protótipos de mesma dimensão, cujo objetivo central é representar de forma fiel e reduzida as recorrências de dados acumuladas em agrupamentos. Esse método é inerentemente não supervisionado, sendo que o algoritmo de agrupamento *k-means* é utilizado com frequência para geração de centroides (KINNUNEN; LI, 2010).

O algoritmo *k-means* tem por objetivo separar os dados de entrada em k agrupamentos, representados por valores de centro nomeados centroides. Em sua implementação mais simples, inicia-se o algoritmo pela seleção de k amostras entre os dados como os centroides iniciais. Em seguida e iterativamente, as amostras restantes são distribuídas entre os centroides conforme proximidade, por distância Euclidiana, e os centroides são atualizados como a média das amostras que representam. O algoritmo é finalizado quando não há movimentação significativa dos centroides (ARTHUR; VASSILVITSKII, 2007). A Figura 8 representa um exemplo de avanço pelas iterações do *k-means* aplicado sobre um conjunto de vetores de características. Um vez que o algoritmo é inicializado ele sempre converge para o erro mínimo local, e em virtude disso é recomendado que o algoritmo seja realizado múltiplas vezes sobre o mesmo conjunto de dados porém com inicializações aleatórias. Dessa forma existem mais chances de se alcançar melhores agrupamentos.

A similaridade entre um conjunto de vetores de características e um modelo VQ é calculada como a distorção média entre as amostras e o modelo VQ. Da mesma forma como foi criado o modelo, as amostras do locutor desconhecido são distribuídas entre os centroides conforme proximidade. A distância média entre centroides e suas respectivas amostras é usada como o valor de similaridade, quanto menor for a distância maior a similaridade entre as amostras de um locutor e o modelo VQ.

Figura 8 – Trajetória de atualização dos centroides para o algoritmo *k-means*.



3.2.2 Modelo de mistura Gaussiana

Conforme mencionado na seção 2.3.2 um GMM é definido como uma superposição de Gaussianas cujo propósito é providenciar uma classe superior de densidade de probabilidade para um conjunto de amostras de treinamento se comparado a um modelo VQ ou a uma Gaussiana de densidade unimodal. Ele é representado na forma:

$$p(\vec{x}|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\vec{x}|\mu_k, \Sigma_k) \tag{3.1}$$

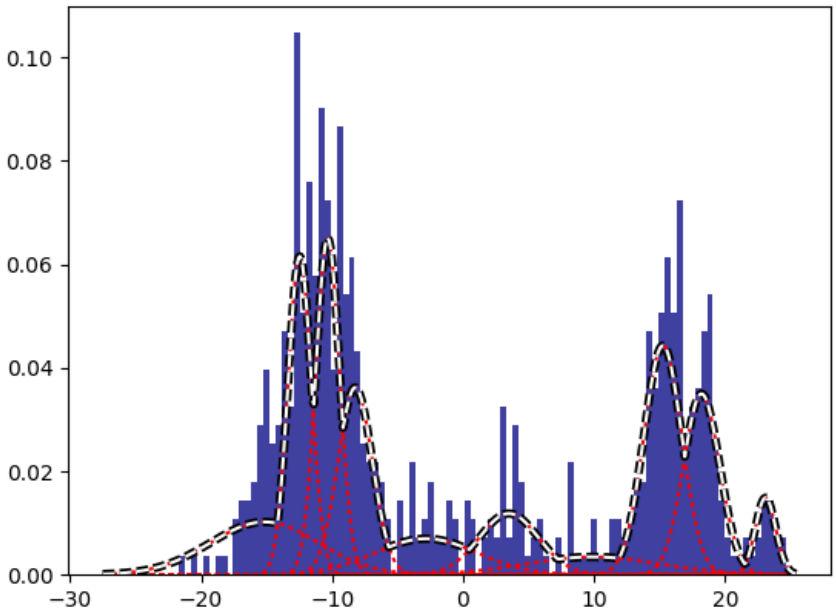
Esse tipo de modelagem assume que as amostras de um conjunto de vetores de características acústicas $D-$dimensionais $X = (\vec{x}_1, ..., \vec{x}_n)$ são geradas sem dependência entre si por uma distribuição de probabilidade complexa, que porém pode ser aproximada com precisão pela soma de distribuições Gaussianas $D-$dimensionais $\mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$. Um exemplo dessa aproximação com $D = 1$ é representado na Figura 9. Nessa modelagem, um modelo θ é formado pela coleção dos pesos de mistura, médias e matrizes de covariância. Respectivamente, $\theta = \{\pi_k, \mu_k, \Sigma_k\}$ $k = 1, ..., M$.

Dado um conjunto de vetores aleatórios e independentes X, pode-se estimar a probabilidade que estes tenham sido gerados por um modelo $\theta = \{\pi, \mu, \Sigma\}$. Essa probabilidade é calculada pela multiplicação das probabilidades de cada membro de X:

$$p(X|\theta) = \prod_{n=1}^{N} p(\vec{x}_n|\theta) \tag{3.2}$$

Entretanto, para um conjunto de vetores de características acústicas extraídos de uma gravação, a verdadeira distribuição de probabilidade geradora θ desse conjunto é desconhecida. Portanto o treinamento de um modelo por GMM parte da maximização de 3.2, com base nos vetores de características acústicas observados. Essa maximização é realizada com o algoritmo *Expectation Maximization* (EM), que age sobre os dados de forma similar os algoritmo *k-Means*, atualizando os parâmetros de θ conforme adquire-se

Figura 9 – Soma linear de curvas Gaussianas modelando a distribuição dos vetores representados pelo histograma em azul.



maior valor de verossimilhança calculada por 3.2 (BISHOP, 2006; REYNOLDS; ROSE, 1995).

O algoritmo EM é uma forma geral de estimativa de parâmetros para modelos estatísticos que dependem de variáveis latentes. Segundo a interpretação intuitiva proposta por (REYNOLDS; ROSE, 1995), as Gaussianas de um GMM representam classes acústicas ocultas, que por sua vez carregam informações sobre a identidade do locutor. As classes ocultas podem ser denotadas por uma variável binária latente de dimensionalidade correspondente ao número de componentes Gaussianas. O algoritmo EM estima iterativamente valores de probabilidade de variável latente para cada componente Gaussiana, buscando convergência em valor de probabilidade máxima local.

A maximização do logaritmo da Equação 3.2 traz as seguintes fórmulas para atualização dos parâmetros π_k, μ_k e Σ_k, que garantem o aumento da verossimilhança do modelo GMM (REYNOLDS; ROSE, 1995):

$$\pi_k = \frac{1}{N} \sum_{n=1}^{N} p(k|\vec{x}_n, \theta) \tag{3.3}$$

$$\mu_k = \frac{\sum_{n=1}^{N} p(k|\vec{x}_n, \theta)\vec{x}_n}{\sum_{n=1}^{N} p(k|\vec{x}_n, \theta)} \tag{3.4}$$

$$\Sigma_k = \frac{\sum_{n=1}^{N} p(k|\vec{x}_n, \theta) x_n^2}{\sum_{n=1}^{N} p(k|\vec{x}_n, \theta)} - \mu_k^2 \tag{3.5}$$

sendo que a probabilidade da variável latente M-dimensional para a componente k de um GMM é obtida pela fórmula:

$$p(k|\vec{x}_n, \theta) = \frac{\pi_k \mathcal{N}(\vec{x_n}|\mu_k, \Sigma_k)}{\sum_{i=1}^{M} \pi_i \mathcal{N}(\vec{x_n}|\mu_i, \Sigma_i)} \tag{3.6}$$

Na identificação de locutor, o objetivo é achar o modelo GMM que possui a maior probabilidade *a posteriori* dada uma sequência de vetores de características acústicas $X = \{\vec{x}_1, ..., \vec{x}_T\}$. Dessa forma, a pontuação de similaridade entre X e um modelo θ é computada como a log-verossimilhança média, calculada pela fórmula:

$$\psi(X, \theta) = \frac{1}{T} \sum_{t=1}^{T} log\ p(\vec{x}_t|\theta) \tag{3.7}$$

3.2.3 i-vector

O modelo i-vector foi proposto em (DEHAK et al., 2009; DEHAK et al., 2011) e representa uma das abordagens mais recentes de modelagem de locutor. O modelo se trata de uma adaptação da modelagem por análise fatorial, conhecida como *Joint Factor Analysis* (KENNY et al., 2007). A adaptação utiliza análise fatorial para treinar um subespaço de baixa dimensão a partir de um GMM independente de locutor, também conhecido como UBM. Dada uma gravação, seu conjunto de vetores de características acústicas, ou janelas de tempo, são projetados nesse subespaço de baixa dimensionalidade, tendo como vetor de coordenadas o *i-vector* da respectiva gravação (GARCIA-ROMERO; ESPY-WILSON, 2011). O calculo de i-vector para uma gravação rec_i de vetores de características acústicas $X = \{\vec{x}_1, \vec{x}_2, ..., \vec{x}_n\}$, usa o UBM θ composto por C componentes para estimar as seguintes estatísticas Baum-Welche:

$$N_c = \sum_t \gamma_t(c), \tag{3.8}$$

$$F_c = \sum_t \gamma_t(c) \vec{x}_t, \tag{3.9}$$

$$\tilde{F}_c = \sum_t \gamma_t(c)(\vec{x}_t - m_c), \tag{3.10}$$

nas quais, para cada observação t, $\gamma_t(c)$ é a probabilidade posterior, ou grau de associação, do vetor de características acústicas $\vec{x}_t$ em relação à componente c e m_c é o vetor de média da componente c do UBM. O subespaço T e uma matriz de covariância Σ são treinados

pela análise fatorial e, junto as estatísticas Baum-Welch, são usados para estimar o i-vector de uma gravação rec_i pela equação:

$$w = (I + T^t\Sigma^{-1}N(i)T)^{-1}T^t\Sigma^{-1}\tilde{F}(i), \tag{3.11}$$

na qual $N(i)$ é a matriz diagonal formada pelos N_c $(c = 1, ..., C)$ e $\tilde{F}(i)$ é um supervetor de concatenação das estatísticas $\tilde{F}_c$ da gravação (DEHAK et al., 2009). Portanto, a extração de um i-vector para alguma gravação depende do UBM, T e Σ, que são calculados de forma não supervisionada, fator que implica em sua utilização sem a necessidade de múltiplas gravações de cadastro por locutor.

Por se tratar de uma projeção de baixa dimensão dos vetores de características acústicas, diz-se que essa abordagem é na verdade uma nova forma de extração de características de locutor, uma vez que a criação de *i-vectors* não requer múltiplas gravações para cadastro de locutor (DEHAK et al., 2009; DEHAK et al., 2011).

A comparação de similaridade entre *i-vectors* costuma ser realizada pelo cálculo da distância cosseno normalizada pela covariância encontrada entre *i-vectors* de locutores impostores. Esse método é conhecido como *Within-Class Covariance Normalization* (WCCN) e usa a matriz inversa de covariância entre *i-vectors* extraídos de gravações de mesmo locutor usadas na estimativa do UBM (DEHAK et al., 2011).

3.2.4 Ferramentas de extração e modelagem de locutor

Os métodos de modelagem de locutor explorados neste projeto são implementações clássicas utilizadas em variados contextos como técnicas de aprendizado de máquina ou compressão de dados. A quantização vetorial, que consistiu na aplicação do algoritmo k-means sobre um conjunto de vetores de características acústicas, não se trata de um algoritmo demasiadamente sofisticado, portanto possui inúmeras implementações. Como referência, nos experimentos foi utilizada a implementação da biblioteca Scikit-learn (PEDREGOSA et al., 2011) do ambiente Python.

A modelagem por GMM consiste em uma implementação mais sofisticada, portanto menos acessível que o algoritmo k-means. Como base inicial utilizou-se a implementação (BROOKES, s.d.) no ambiente Matlab/Octave. Conforme o desenvolvimento do projeto mudou-se para o uso da implementação disponibilizada pelo Scikit-learn, por ser uma versão mais atual e consistente com o ambiente utilizado para a modelagem VQ.

Para aplicação de modelagens de locutor mais robustas existe a plataforma ALIZE (BONASTRE; WILS; MEIGNIER, 2005) que implementa os métodos mais recentes de modelagem de locutor como GMM-UBM, extração de *i-vectors* e métodos de pontuação como a distância cosseno com WCCN.

3.3 Aprendizado não supervisionado

Os métodos discutidos nas seções anteriores correspondem todos à abordagens clássicas utilizadas em tarefas de reconhecimento de locutor. A seguir discute-se sobre algoritmos de pós-processamento à recuperação de objetos multimídia por conteúdo. Esses algoritmos se baseiam na conjectura que quaisquer métodos utilizados para caracterizar e comparar conteúdo multimídia, no caso deste arcabouço quanto à identidade de locutor, são suficientes para orientar heurísticas baseadas na informação contextual de similaridades, codificadas em listas de ranqueamento de conjuntos de objetos multimídia.

O modelo de recuperação por conteúdo é definido formalmente a seguir. Em seguida são apresentados os algoritmos de aprendizado não supervisionado RL-Sim e ReckNN, ambos utilizados na avaliação experimental do arcabouço.

3.3.1 Definição do Modelo de Recuperação e Ranqueamento

Seja $\mathcal{M}$=$\{m_1, m_2, \ldots, m_n\}$ uma coleção de modelos de locutores treinados, respectivamente, a partir dos conjuntos de características acústicas de $\mathcal{C}_x = \{X_1, X_2, \ldots, X_n\}$. Dado ainda que os conjuntos de $\mathcal{C}_x$ foram extraídos da coleção de gravações $\mathcal{C} = \{rec_1, rec_2, \ldots, rec_n\}$. Seja $n = |\mathcal{C}|$ o tamanho da coleção $\mathcal{C}$. Seja $\rho(rec_i, rec_j)$ uma função que calcula um valor real que representa a distância entre as características acústicas $X_i \in \mathcal{C}_x$ e um modelo de locutor $m_j \in \mathcal{M}$. A notação $\rho(i, j)$ é utilizada ao longo do texto visando maior facilidade de leitura.

A distância $\rho(i, j)$ entre todos as gravações rec_i, $rec_j \in \mathcal{C}$ pode ser calculada para obter-se uma matriz quadrada $n \times n$, de forma que $A_{ij} = \rho(i, j)$. Também, em resposta a uma consulta de uma gravação rec_c, uma lista de ranqueamento τ_c pode ser computada.

A lista de ranqueamento τ_c=$(rec_1, \ldots, rec_n)$ pode ser definida como uma permutação da coleção $\mathcal{C}$. Uma permutação τ_c é uma bijeção de um conjunto $\mathcal{C}$ para o conjunto $[N] = \{1, \ldots, n\}$. Para uma permutação τ_c, podemos interpretar $\tau_c(i)$ como a posição do objeto rec_i na lista de ranqueamento τ_c. Podemos dizer que, se rec_i está posicionado antes de rec_j na lista de ranqueamento de rec_c, isto é, $\tau_c(i) < \tau_c(j)$, então $\rho(c, i) \leq \rho(c, j)$. Também podemos tomar todo objeto $rec_i \in \mathcal{C}$ como um objeto de consulta rec_c, a fim de obter um conjunto $\mathcal{R} = \{\tau_1, \ldots, \tau_n\}$ de listas de ranqueamento para cada objeto da coleção $\mathcal{C}$.

O objetivo de um algoritmo de aprendizado não supervisionado consiste em redefinir a distância inicial ρ computando uma função de distância mais eficaz. O objetivo geral é aumentar a eficácia das distâncias, usando a informação contextual codificada nas listas de ranqueamento definidas pelo conjunto $\mathcal{R}$. Mais formalmente, podemos definir o algoritmo como uma função f_r:

$$\hat{A} = f_r(\mathcal{R}, A). \tag{3.12}$$

Uma nova matriz de distância $\hat{A}$ pode ser calculada pela função f_r, a qual toma como entrada um conjunto de listas de classificação $\mathcal{R}$.

3.3.2 RL-Sim

Dado um conjunto inicial de listas de classificação, o algoritmo RL-Sim (PEDRONETTE; TORRES, 2013a) utiliza uma abordagem iterativa que considera alguma métrica de distância baseada na similaridade entre as listas $\mathcal{R}$ desse conjunto. Formalmente, seja $^{(t)}$ a notação utilizada para a iteração atual, um novo e mais efetivo conjunto de listas de classificação $\mathcal{R}^{(t+1)}$ é computado considerando as distancias entre as componentes desse conjunto. Em seguida, $\mathcal{R}^{(t+1)}$ é usado para a próxima execução do algoritmo de reclassificação e assim sucessivamente. Estes passos são repetidos ao longo de iterações com o objetivo de melhorar a eficácia de forma incremental. Depois de um número T de iterações, é realizada uma reclassificação definitiva.

A métrica contextual de distância calculada a cada iteração é baseada na suposição que os objetos mais bem classificadas (top-k) são semelhantes entre si e suas listas de ranqueamento contêm alguns objetos em comum (PEDRONETTE; TORRES, 2013a). Neste cenário, uma estratégia simples para o cálculo da similaridade entre os objetos depende da utilização de métricas de correlação de classificação.

Considerando o conjunto de vizinhança $\mathcal{N}(i,k)$ de um objeto rec_i, o qual contém os k objetos mais similares de rec_i, de acordo com uma determinada distância ρ. O conjunto $\mathcal{N}(i,k)$ pode ser obtido pelos k vizinhos mais próximos, em que a cardinalidade do conjunto é denotado por $\mid \mathcal{N}(i,k) \mid = k$.

Seja $d(\tau_i, \tau_j, k)$ uma métrica de correlação entre listas de classificação considerando suas primeiras posições, definidas pelos conjuntos $\mathcal{N}(i,k)$ e $\mathcal{N}(j,k)$, definida no intervalo $[0,1]$. Uma métrica contextual de distância $\rho_c(i,j)$, baseada na comparação das listas de ranqueamento τ_i, τ_j, pode ser definida do seguinte modo:

$$\rho_c(i,j) = d(\tau_i, \tau_j, k) \tag{3.13}$$

Com base na conjectura de que a métrica de distância ρ_c representa uma distância mais eficaz entre objetos (PEDRONETTE; TORRES, 2013a), a distância entre todos os objetos em uma coleção pode ser recalculada baseada nesta métrica. Entretanto, um novo conjunto de listas de classificação pode ser obtido, de tal modo que a distância contextual também pode ser recalculada e o processo pode ser repetido de modo iterativo. Seja $^{(t)}$ a atual iteração e seja $\tau_i^{(t)}$ a lista de classificação da iteração t. Seja $\rho_c^{(0)}$ a distância contextual

da primeira iteração, a qual é igual a distância original, tal que $\rho_c^{(0)}(i,j) = \rho(i,j)$ para todos os objetos $rec_i, rec_j \in \mathcal{C}$. A métrica iterativa é definida como:

$$\rho_c^{(t+1)}(i,j) = d(\tau_i^{(t)}, \tau_j^{(t)}, k) \tag{3.14}$$

Espera-se que a eficácia da medida de distância melhore ao longo de iterações, então os objetos não relevantes são removidos das primeiras posições das listas de classificação. Desta maneira, o tamanho da vizinhança k pode ser aumentado por considerar mais objetos ao longo das iterações. Portanto, a métrica pode ser redefinida como:

$$\rho_c^{(t+1)}(i,j) = d(\tau_i^{(t)}, \tau_j^{(t)}, k+t) \tag{3.15}$$

Depois de um determinado número T de iterações, uma nova distância é calculada baseada na métrica de distância contextual:

$$\hat{\rho}(i,j) = \rho_c^{(T)}(i,j) \tag{3.16}$$

Finalmente, usando a distância $\hat{\rho}$, uma nova matriz de distância pode ser calculada como $\hat{A}_{ij} = \hat{\rho}(i,j)$. Baseada em $\hat{A}$, um novo conjunto de listas de classificação $\hat{\mathcal{R}}$ pode ser também calculado.

A métrica de distância contextual utilizada neste projeto foi a métrica da interseção, também utilizada em (PEDRONETTE; TORRES, 2013a). Ela consiste na acumulação iterativa do conjunto de interseção $\mathcal{N}(i,k) \cap \mathcal{N}(j,k)$, dado as duas listas de classificação dos objetos rec_i e rec_j. Essa medida consiste na seguinte similaridade:

$$\psi(rec_i, rec_j, k) = \frac{\sum_{k_c=1}^{k} \mathcal{N}(i,k) \cap \mathcal{N}(j,k)}{k} \tag{3.17}$$

Para adequar a métrica ao contexto de matriz de distâncias, usa-se a inversão proporcional da similaridade ψ como:

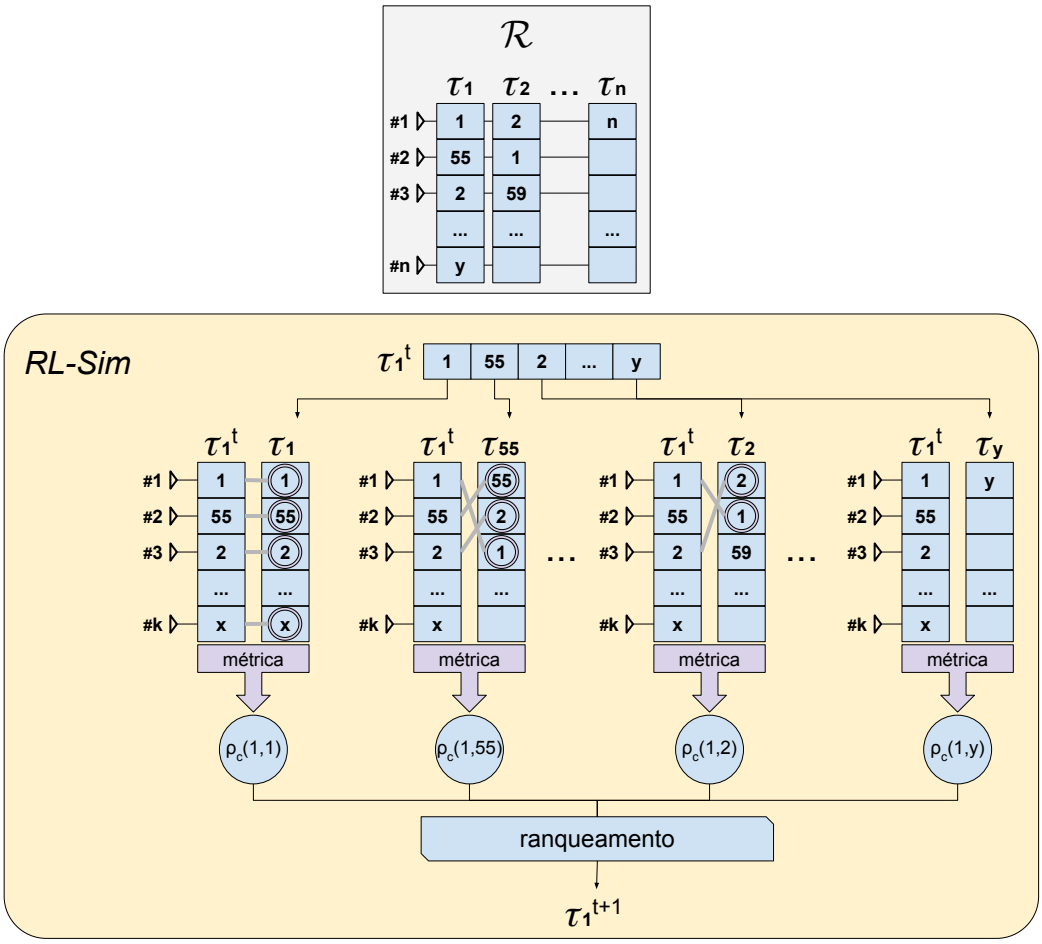
$$dist_\psi(rec_i, rec_j) = \frac{1}{1 + \psi(rec_i, rec_j, k)} \tag{3.18}$$

A Figura 10 é uma representação do funcionamento do algoritmo sobre um conjunto de listas de classificação (*ranked lists*).

3.3.3 Grafos kNN recíprocos

O algoritmo ReckNN age com base na mesma conjectura utilizada pelo RL-Sim, que é similar a hipótese de agrupamento: documentos de um agrupamento tendem a ser relevantes aos mesmo pedidos de informação (PEDRONETTE; PENATTI; TORRES,

Figura 10 – Exemplo do algoritmo RL-Sim.



2014; MANNING; RAGHAVAN; SCHÜTZE, 2008). Entretanto, o ReckNN age em maior profundidade que o RL-Sim, ramificando a análise por similaridades através das vizinhanças $\mathcal{N}(i,k)$ de objetos mais próximos. Considerando a coleção de objetos alvo como um grafo cujas arestas são definidas por uma matriz de distâncias, este algoritmo atualiza essas distâncias com base na geometria descrita entre os vizinhos mais próximos de cada consulta. Para uma gravação rec_i, cuja vizinhança é definida por $\mathcal{N}(i,k)$ de cardinalidade $\mid \mathcal{N}(i,k) \mid = k$, o algoritmo analisa o conjunto formado pela geometria de vizinhança descrita por $\mathcal{N}(i,k)$ e que se estende pelas vizinhanças $\mathcal{N}(j,k)$ com $j \in \mathcal{N}(i,k)$. Especificamente, o ReckNN une três conceitos para reclassificar objetos:

- vizinhança recíproca: objetos mutualmente relevantes devem se referenciar no topo de suas listas de classificação;
- colaboração de ranqueamento: objetos presentes no topo de listas de classificação provavelmente são similares entre si;
- densidade do grafo de vizinhança: um bom topo de lista de classificação possui referências recorrentes em seu grafo de vizinhança.

Esses conceitos são implementados na forma de três pontuações. A primeira é chamada de pontuação de autoridade da lista de ranqueamento e é uma medida não supervisionada da eficácia deste ranqueamento(PEDRONETTE; PENATTI; TORRES, 2014). Essa pontuação é definida pela equação:

$$A_s(c,k) = \frac{\sum_{i \in \mathcal{N}(c,k)} \sum_{j \in \mathcal{N}(i,k)} f_{in}(j,c)}{k^2}, \tag{3.19}$$

na qual $f_{in}(j,c) = 1$ se rec_j existe em $\mathcal{N}(c,k)$ e 0 caso contrário. No caso ideal, todos as gravações de $\mathcal{N}(c,k)$ também formam as subsequentes $\mathcal{N}(j,k)$ e a pontuação de autoridade totaliza em 1.

A segunda pontuação, consiste na propagação do A_s das listas de ranqueamento entre os objetos que formam suas vizinhanças mais próximas. Esta pontuação se relaciona ao segundo ponto conceitual citado anteriormente, que busca recompensar conjuntamente os objetos que produziram a pontuação de autoridade. A propagação é realizada pela acumulação do quadrado da pontuação de autoridade sobre os objetos componentes de $\mathcal{N}(c,k)$. Utiliza-se o quadrado da pontuação para penalizar pontuações baixas (PEDRONETTE; PENATTI; TORRES, 2014):

$$C_s(c,i,k) = \sum_{k_c=1}^{k} \sum_{j \in \mathcal{C}} A_s(j,k_c)^2 f_{in}(c,i,j), \tag{3.20}$$

em que $f_{in} = 1$ se $obj_c, obj_i \in \mathcal{N}(j,k)$ e 0 caso contrário.

A terceira pontuação age como contra-peso para falsos positivos, ou seja, objetos não relevantes à c porém presentes em $\mathcal{N}(c,k)$. Esta pontuação é nomeada Pontuação kNN Recíproca e usa a maior posição pela qual dois objetos tenham se tornado vizinhos recíprocos:

$$R_s(c,i) = \frac{max(\tau_c(i), \tau_i(c))}{n_s}, \tag{3.21}$$

na qual n_s é definido como o tamanho da coleção de gravações $\mathcal{C}$.

A nova distância entre duas gravações definida pelo ReckNN utiliza R_s e C_s da seguinte forma para caso $C_s(c,i,k) > 0$:

$$\rho_(c,i) = \frac{R_s(c,i)}{1 + C_s(c,i,k)}, \tag{3.22}$$

Caso $C_s(c,i,k) = 0$ a distância é atualizada como $\rho(c,i) = \tau_c(i)$.

3.3.4 Caso especial de consultas e coleção de objetos multimídia disjuntas

Conforme definido em 3.3.1, dada uma coleção de gravações $\mathcal{C}$ e tomadas cada uma de suas gravações como consulta rec_c, obtém-se a matriz de distâncias A e as listas de ranqueamento $\mathcal{R}$, sendo que ambos codificam informações contextuais entre as gravações de $\mathcal{C}$. O RL-Sim toma as duas entradas $\mathcal{R}$ e A, enquanto que o ReckNN toma somente $\mathcal{R}$, e as usa para calcular iterativamente novas matrizes A^t e listas de ranqueamento $\mathcal{R}^t$ utilizando as informações contextuais nelas contidas. Por fim, converge-se na matriz de distâncias $\hat{A}$ e listas de ranqueamento $\hat{\mathcal{R}}$ que refletem as informações contextuais entre as gravações de $\mathcal{C}$.

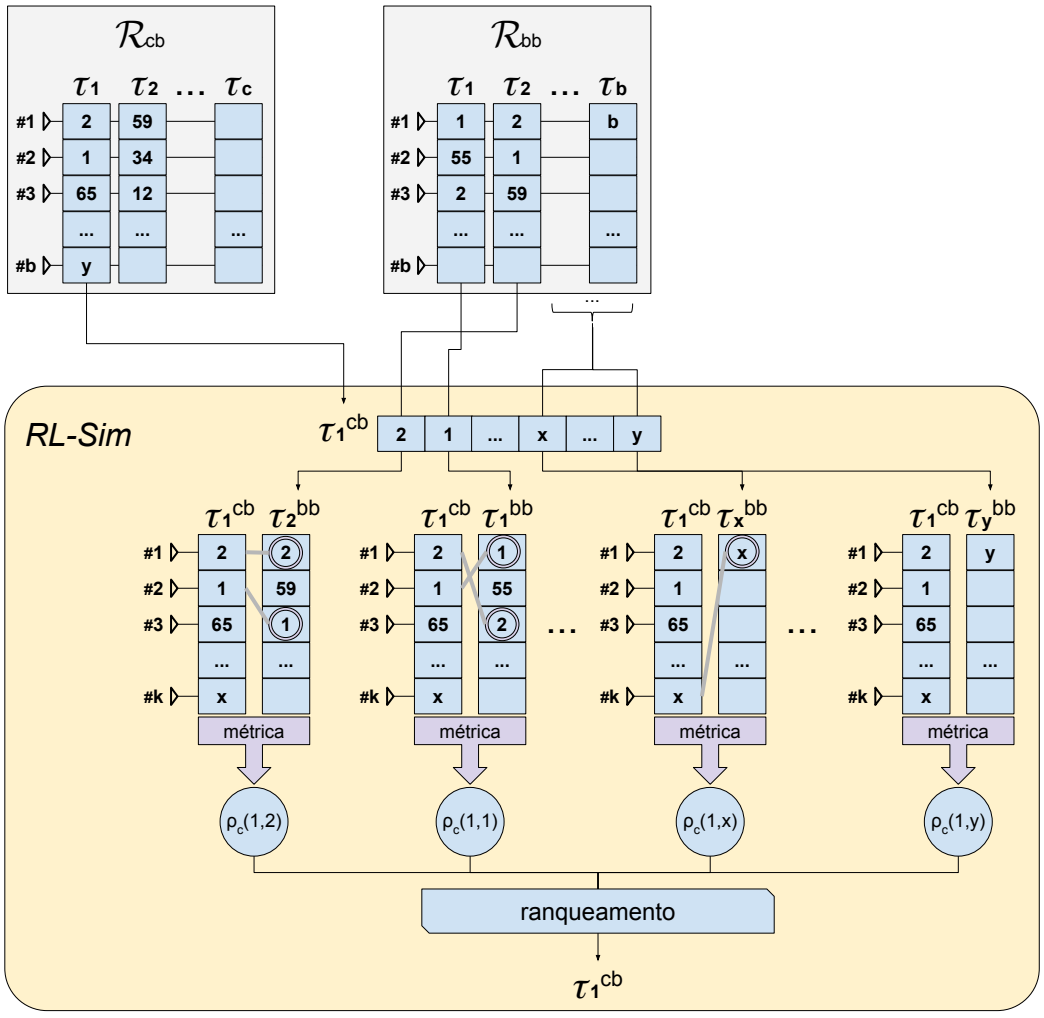
Em cenários alternativos, pode ser impossível ou inviável recalcular distâncias tomando todos os objetos de uma coleção $\mathcal{C}$. Tomando uma coleção de larga escala $\mathcal{C}_b$ composta por nb gravações de interesse e cujas informações de consulta entre seus elementos A_{bb} e $\mathcal{R}_{bb}$ já são conhecidas. Adicionalmente, devido a limitações trazidas pelo seu tamanho, não é desejável recalcular A_{bb} e $\mathcal{R}_{bb}$. Considerando outra coleção $\mathcal{C}_c$, composta por nc gravações e disjunta de $\mathcal{C}_b$, como consulta à $\mathcal{C}_b$. O resultado das consultas entre $\mathcal{C}_c$ e $\mathcal{C}_b$ é a matriz $nc \times nb$ A_{cb} e as listas de ranqueamento $\mathcal{R}_{cb}$. Ocorre que A_{cb} e $\mathcal{R}_{cb}$ contém informações contextuais em relação à coleção $\mathcal{C}_b$, da qual inicialmente tem-se informações contextuais sobre si própria na matriz $nb \times nb$ A_{bb} e $\mathcal{R}_{bb}$. Sobre este cenário, foram propostas as seguintes alterações nos algoritmos RL-Sim e ReckNN com o objetivo de transferir as informações contextuais de $\mathcal{C}_b$ no âmbito de aperfeiçoar as relações de $\mathcal{R}_{cb}$ sem o recalculo de A_{bb} ou $\mathcal{R}_{bb}$.

- **RL-Sim:** para cada consulta de $\mathcal{C}_c$, a métrica contextual de distância 3.14 é calculada entre os pares $rec_i \in \mathcal{C}_c$ e $rec_j \in \mathcal{C}_b$, pela comparação de $\tau_i \in \mathcal{R}_{cb}$ com $\tau_j \in \mathcal{R}_{bb}$.
- **ReckNN:** foram feitas duas alterações sobre a formulação original. A primeira foi sobre o cálculo da pontuação de colaboração $C_s(rec_i, rec_j, k)$, definida pela Equação 3.20. Esta foi alterada para acumular a pontuação de autoridade $A_s(rec_i, k)$ não somente pela vizinhança $\mathcal{N}(rec_i, k)$ da consulta, mas também pelas vizinhanças dos objetos de $\mathcal{N}(rec_i, k)$. A segunda alteração foi sobre a distância contextual $\rho(rec_i, rec_j)$, definida pela Equação 3.22. Esta foi alterada para utilizar somente a posição da gravação rec_j em $\tau_i \in \mathcal{R}_{cb}$ como freio à aproximação de objetos não relevantes. Assim como na alteração ao RL-Sim, $C_s(rec_i, rec_j, k)$ e $\rho(rec_i, rec_j)$ é definido para todo $rec_i \in \mathcal{C}_c$ e $rec_j \in \mathcal{C}_b$.

Nas Figuras 11 e 12 são esquematizadas as reclassificações alteradas para o cenário alternativo. A única alteração feita sobre o RL-Sim diz respeito ao domínio das consultas e das listas de ranqueamento τ_j utilizadas em 3.14. Em contra-partida, como o ReckNN tem base forte no cálculo de reciprocidade dos objetos considerados pelo algoritmo e, uma

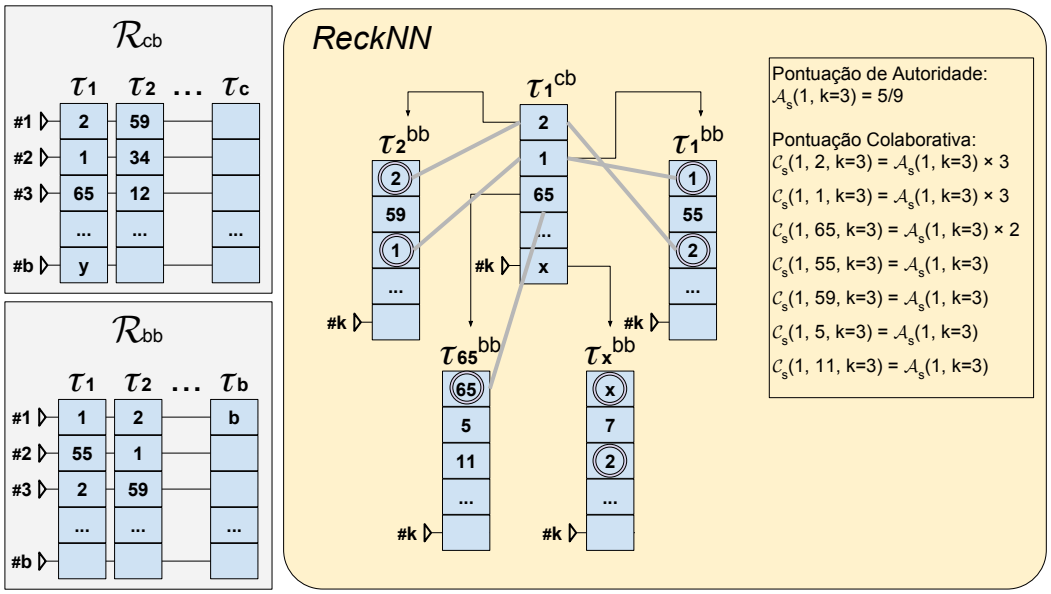
vez que esse fator é mitigado neste cenário alternativo, as alterações no algoritmo tiveram que ser mais profundas. Uma proposta com menos alterações sobre o ReckNN, na qual, de forma semelhante às alterações do RL-Sim, mudou-se apenas o domínio das gravações rec_i e rec_j assim como as vizinhanças $\mathcal{N}(rec_i, k)$ e $\mathcal{N}(rec_j, k)$, provou-se ineficaz para recalcular distâncias de A_{cb}.

Figura 11 – Esquema do RL-Sim modificado.



As principais características do ReckNN são sua habilidade de alcançar objetos relevantes que foram recuperados em posições mais distantes da consulta e seu equilíbrio de recuperação. Os dois âmbitos são controlados pela análise das posições recíprocas, que evita aproximar objetos não relevantes. Ambas as características são formuladas com forte dependência ao fator recíproco. Justifica-se a alteração feita sobre o cálculo da pontuação C_s como uma tentativa de, de forma semelhante ao algoritmo original, alcançar objetos relevantes mais distantes. O fator limitador desse alcance é definido pelo uso das posições dos objetos alvo nas listas de ranqueamento $\tau_i \in \mathcal{R}_{cb}$ como freio à aproximação de objetos

Figura 12 – Esquema do ReckNN modificado.



não relevantes. Para efeito de clareza as equações modificadas do ReckNN são as seguintes:

$$C_s(rec_i, rec_j, k) = \sum_{k_c=1}^{k} \sum_{c \in \mathcal{C}_c} A_s(rec_c, k_c)^2 f_{in}(rec_i, rec_j), \tag{3.23}$$

$$\rho(rec_i, rec_j) = \frac{\frac{\tau_i(j)}{n_b}}{1 + C_s(rec_i, rec_j, k)}, \tag{3.24}$$

na qual n_b é o tamanho da base $\mathcal{C}_b$, $f_{in}(rec_i, rec_j)$ retorna 1 se $rec_j \in \mathcal{N}(rec_i, k) \vee rec_j \in \mathcal{N}(rec_l, k)$, com $rec_l \in \mathcal{C}_b$, caso contrário retorna 0. Também, semelhante ao ReckNN, caso $C_s(rec_i, rec_j, k) = 0$ a nova distância é definida como a posição $\tau_i(rec_j)$.

3.4 Classificação para identificação de locutor

Os resultados produzidos pela recuperação de locutor se encontram na forma de listas de ranqueamento. Dessa forma, para uma gravação de consulta rec_i, com lista de ranqueamento μ_i, o classificador kNN contabiliza os grupos de mesmo locutor $L_i = \{l_1, l_2, ..., l_n\}$ presentes na vizinhança $\mathcal{N}(i, k_{ck})$. Os grupos de L_i tem tamanho variável, conforme a presença de gravações de mesmo locutor em $\mathcal{N}(i, k_{cl})$. O tamanho de um grupo de locutor l_x pertencente a L_i pode ser contabilizado pela verificação dos rótulos, se presentes, das gravações de $\mathcal{N}(i, k_{cl})$.

Um classificador kNN clássico pode simplesmente retornar como resposta de classificação o locutor x cujo grupo l_x tem maior tamanho η_x entre os locutores de L_i. Alternativamente, outras métricas que levem em conta as distâncias $\rho(rec_i, rec_j^x)$ das gravações rec_j^x, $j = (1, ..., \eta_x)$ de cada $l_x \in L_i$ podem ser utilizadas.

Neste arcabouço, para o calculo final da distância entre uma consulta rec_i e um rótulo de locutor, foi utilizada uma métrica que leva em conta a distância média do locutor conforme sua taxa de aparição em $\mathcal{N}(i, k_{cl})$.

Para uma consulta rec_i, a distância final adotada $dist(rec_i, x)$, entre a consulta e o locutor x de grupo $l_x = \{rec_1^x, ..., rec_{\eta_x}^x\}, l_x \in L_i$, é definida como:

$$dist(rec_i, x) = \frac{\sum_{j=0}^{\eta_x} \rho(rec_i, rec_j^x)}{\eta_x^{r_x}} \tag{3.25}$$

sendo $rec_j^x \in l_x$ e r_x a taxa de aparição do locutor x em $\mathcal{N}(i, k_{cl})$, que é definida como:

$$r = \frac{\eta_x}{k_{cl}} \tag{3.26}$$

Uma gravação de teste foi considerada como classificada corretamente se o locutor de menor distância $dist(rec_i, x)$ combina com o fornecido pelo rótulo da gravação.

4 Avaliação experimental

O processo de avaliação experimental conduzido para aferir a eficácia do arcabouço proposto é descrito neste capítulo. Este processo consistiu na seleção de conjuntos de dados acústicos com diferentes propósitos, e de acesso livre ao público geral, que apresentassem objetos de áudio com diferentes condições de gravação. Dessa forma, buscou-se aumentar a abrangência do estudo realizado sobre o arcabouço proposto sobre múltiplos cenários de aplicação.

A Seção 4.1 descreve os conjuntos de dados utilizados pelos experimentos. A Seção 4.2 traz uma descrição detalhada do protocolo experimental conduzido. A partir da Seção 4.3 são apresentados os resultados dos experimentos e discussões sobre os mesmos, com enfoque no efeito dos algoritmos de aprendizado não supervisionado e seus respectivos parâmetros sobre a eficácia do arcabouço. Na Seção 4.3.1, inicia-se com a apresentação dos resultados da tarefa de recuperação de gravações por locutor, assim como o efeito da variação do parâmetro de vizinhança k do RL-Sim e ReckNN. Os resultados de cada conjunto de dados são apresentados e discutidos individualmente. Conclui-se a apresentação dos resultados de recuperação com uma discussão sobre a eficácia dos algoritmos RL-Sim e ReckNN frente às configurações de distâncias fornecidas à eles. Em seguida, na Seção 4.3.1, os resultados da tarefa de identificação (classificação por kNN) são apresentados de moda semelhante à da recuperação. Finalmente, o capítulo é concluído com resultados e discussões sobre o experimento realizado no cenário desafiador de reconhecimento de locutor sob discrepância completa entre gravações de consulta e de base de locutor.

4.1 Conjuntos de dados

Os três conjuntos de dados selecionados para avaliação experimental são resumidos na Tabela 1. Mais detalhes de suas respectivas configurações são discutidas nesta seção. Pode-se verificar que foram considerados diferentes idiomas, níveis de ruído e quantidade de locutores.

Tabela 1 – Conjuntos de dados utilizados na avaliação experimental.

Conjunto	**Tamanho**	**Locutores**	**Segmentos por locutor**	**Língua**	**Duração por segmento**	**Ruído**
CHAINS	3996	36	111	Inglês	Frases (2s-3s) & Leituras (30s)	Baixo & Médio - Estúdio & Escritório
LapsBM1.4	700	35	20	Português	Frases (2s-3s)	Médio - Escritório
YouTube	9950(55400)	995	10	Inglês	10s	Alto - Apresentação

4.1.1 CHAINS

O conjunto de dados CHAINS, de língua inglesa, foi coletado com o objetivo de providenciar gravações de mesmo locutor apresentando tanto variações de forma de fala, por exemplo sussurros e leituras rápidas, como variações de ambiente e material de gravação. As gravações utilizadas podem ser divididas em três categorias divergentes: fala normal gravada em estúdio, fala em sussurro gravada em ambiente de escritório e fala rápida também gravada em ambiente de escritório. Cada locutor do conjunto possui 37 gravações para cada categoria de fala, totalizando 111 gravações por locutor.

Pode-se discutir que, por causa do baixo nível de ruído e população de locutores limitada, as gravações da categoria de fala normal, gravadas em estúdio, são de fácil distinção entre si para um sistema reconhecedor, evento que de fato foi averiguado em (CAMPOS; PEDRONETTE, 2016). Entretanto, essa afirmação também pode ser estendida para gravações ruidosas de mesmo locutor que originam-se de mesma sessão e canal de gravação, porém divergem quanto à configurações de sessão e canal de gravações de outros locutores, fato observado em (CAMPOS; PEDRONETTE, 2016) nos experimentos com o conjunto de dados YouTube.

Dessa forma, o cenário proposto pela mistura das três categorias representa uma configuração flexível e de acordo com a variabilidade inerente à aplicações de reconhecimento de locutor em mundo real. Espera-se que as divergências introduzidas no canal de gravação, sessão e modalidade de fala reduzam a acurácia do reconhecimento de locutor.

Este conjunto de dados foi apresentado em (GRIMALDI; CUMMINS, 2008) como conjunto de dados de avaliação do método de extração de características proposto pelos autores, também no contexto de identificação de locutor.

4.1.2 Laps

O conjunto de dados Laps (ALVES, s.d.) é o representante da língua portuguesa entre os conjunto de dados considerados. Ele é composto por 700 gravações, sendo estas de 35 locutores com 20 frases cada. Entre os locutores, 25 são homens e 10 são mulheres. O ambiente de gravação não é controlado, portanto este conjunto de dados é um representante do espectro ruidoso do campo amostral. Apesar de pequeno em comparação com os outros conjuntos, Laps é composto por gravações realizadas em ambientes de escritório com nível médio de ruído, formato que representa um tipo de condição de gravação comum e esperado pelo arcabouço.

4.1.3 YouTube

O conjunto de dados extraído do canal GoogleTalks do YouTube (YOUTUBE, 2007), coletado em (SCHMIDT; SHARIFI; MORENO, 2014), representa um cenário de

reconhecimento de locutor de larga escala. As gravações do conjunto foram extraídas de vídeos, cada um contendo apresentações de em média trinta minutos sobre variados assuntos relacionados com tecnologia. Os vídeos de apresentação não passaram por quaisquer alterações de edição de conteúdo, portanto em seus áudios encontram-se numerosas fontes de ruídos, incluindo até mesmo porções com falas de outros locutores, diferentes do locutor alvo, ou sem fala alguma.

Dentre os 998 locutores distintos, 73 possuem mais de um vídeo na coleção, gravado sob diferentes configurações de ambiente e contexto. Dessa forma, este tipo de conjunto apresenta o maior nível de dificuldade de reconhecimento, especialmente entre diferentes vídeos de mesmo locutor. A presença numerosa de ruídos no canal de gravação e ambiente entre as amostras dificilmente é ignorada pelos métodos de modelagem do arcabouço, isso é, sem a utilização de metodologias supervisionadas. Esses dados, porém, são originados do YouTube, site que disponibiliza o serviço de publicação de vídeos por qualquer um de seus mais de um bilhão usuários (quase um terço de todas as pessoas na internet) (YOUTUBE, 2017). Pela origem de difícil supervisão dos dados, uma abordagem não supervisionada, assim como em (SCHMIDT; SHARIFI; MORENO, 2014), se torna o meio exclusivo de reconhecimento de locutor em larga escala.

O reconhecimento de locutor foi relizado sobre segmentos de 10 segundos. Para facilitar o processamento e acurácia do sistema, a segmentação foi realizada sequencialmente, a partir da marca dos 30 segundos iniciais, período sem áudio nas apresentações, até a marca anterior aos 10 minutos finais, período frequentemente utilizado para perguntas e respostas, portanto com a fala de outros locutores. Os segmentos de 10 segundos foram tomados como gravações independentes e modelados dessa forma.

4.2 Protocolo experimental

Três modalidades de experimentos foram conduzidas para avaliar a eficácia do arcabouço representado na Figura 5. O primeiro experimento diz respeito à eficácia de recuperação de locutor, o segundo quanto à acurácia de identificação de locutor e o terceiro experimento quanto à eficácia do arcabouço frente ao cenário de completa divergência de canal e sessão entre gravações de consulta e de base de locutor. Os experimentos foram realizados separadamente para cada coleção de dados e consideram dois conjuntos de gravações: o conjunto de modelos de locutores $\mathcal{M} = \{m_1, m_2, ..., m_{n_b}\}$ e o conjunto de consultas $\mathcal{C} = \{X_1, X_2, ..., X_{n_c}\}$.

Nas duas primeiras modalidades de experimento, as gravações que deram origem aos conjuntos $\mathcal{C}$ e $\mathcal{M}$ foram as mesmas. Portanto essas duas modalidades se encaixam nas mesmas formulações originais dos algoritmos de aprendizado não supervisionado descritas na Seção 3.3.2 e 3.3.3. Cada modelo m_i foi calculado exclusivamente com base

nas características acústicas X_i, referentes à gravação rec_i e, portanto, $n_c = n_b$.

A segunda modalidade de experimentação transcorreu após a recuperação de locutor e consistiu na utilização do classificador kNN descrito na Seção 3.4 sobre o resultado de recuperação das consulta. Esse tipo de experimento só pode ser conduzido sob a condição de existência de gravações com rótulos de locutor. A média da acurácia de identificação entre todas as consultas foi utilizada como o parâmetro de avaliação desta modalidade.

Na terceira modalidade avaliou-se o caso especial em que as gravações de mesmo locutor entre os conjuntos $\mathcal{C}$ e $\mathcal{M}$ diferiam quanto à características do canal de captação e sessão. Quanto à modelagem, utilizou-se i-vectors, para a qual o calculo de distância entre duas gravações é simétrica. Assim, sobre a definição dos conjuntos $\mathcal{C}$ e $\mathcal{M}$, esta pode ser simplificada para $\mathcal{C} = \{iv_1, iv_2, ..., iv_{n_c}\}$ e $\mathcal{M} = \{iv_1, iv_2, ..., iv_{n_b}\}$, sob a condição $\mathcal{C} \cap \mathcal{M} = \varnothing$.

A condição imposta sobre a última modalidade define que as condições de captação (canal e sessão) das gravações que deram origem à $\mathcal{C}$ e $\mathcal{M}$ sejam diferentes. Este cenário representa a condição mais esperada em aplicações no mundo real e, portanto, consiste em um importante desafio de pesquisa na área do reconhecimento de locutor (SHUM et al., 2014). Adiante, essa modalidade é referenciada como *holdout*, em alusão ao experimento original, *holdout10*, conduzido em (SCHMIDT; SHARIFI; MORENO, 2014) e no qual este foi baseado. Para os experimentos realizados sobre essa modalidade com a coleção YouTube, $n_c \neq n_b$.

A modelagem por i-vector requer uma quantidade considerável de amostras de contextos similares ao de utilização dos modelos. Por conta disso esta modelagem foi reservada exclusivamente para o terceiro tipo de experimento, realizado somente sobre o conjunto de dados de larga escala do YouTube. Como discutido no Capítulo 3, quanto à modelagem i-vector, apesar dela apresentar efetividade superior às outras modelagens estudadas neste trabalho, seu desempenho como ferramenta de reconhecimento de locutor depende em grande parte de dados supervisionados. Uma vez que o cenário de recuperação explorado pelo arcabouço não espera dados rotulados, mas sim gravações que são modeladas exclusivamente pelo próprio conteúdo, o potencial de reconhecimento do i-vector experimentado neste trabalho não alcança os padrões apresentados pelo estado-da-arte.

Os procedimentos para recuperação e identificação são detalhados a seguir.

4.2.1 Recuperação

O processo de recuperação é iniciado pela extração de características e modelagem individual de todas a gravações relativas à base $\mathcal{M}$. No processo de extração de características de locutor, gravações foram segmentadas em janelas de tempo de 25 milissegundos. De cada janela de tempo foram computados 19 MFCCs ou PLPs, a energia média da

janela de tempo (primeiro coeficiente cepstral) e os valores *delta* e *double-delta*. O vetor acústico de uma janela de tempo é composto pela concatenação desses valores, totalizando 60 dimensões por vetor. Foi utilizado um passo de 10 milissegundos para segmentação em janela de tempo, o que resultou em uma matriz de dimensões $60 \times d * 100$ de vetores de características acústicas por gravação, na qual d denota a duração da gravação em segundos.

Em termos de modelagem, VQs foram criados com $k = 20$ centroides e GMMs de covariância diagonal com 32 componentes Gaussianas. O número de centroides do VQ foi decidido experimentalmente como o valor mínimo permitido pela quantização vetorial da ferramenta HTK, considerando a curta duração média das gravações. O número de componentes Gaussianas utilizado foi baseado nos experimentos de (REYNOLDS; ROSE, 1995).

As gravações de consulta, relativas à $\mathcal{C}$, passaram pelo mesmo processo de extração de características. Em seguida, o procedimento de consulta consistiu no cálculo de distâncias entre consultas $\mathcal{C}$ e modelos de locutor $\mathcal{M}$. A distância entre os vetores de características acústicas de uma consulta X_i e um modelo m_j foi definida, na modelagem VQ, como a distorção média os vetores de características acústicas e os centroides de m_j ou, na modelagem GMM, como a log-verossimilhança média dos vetores de características acústicas com a mistura m_j invertida proporcionalmente para distância.

Na terceira modalidade de experimento, *i-vectors* foram extraídos para todas as gravações, tomando como base para extração um UBM treinado com as gravações relativas a $\mathcal{M}$. Neste experimento, os locutores presentes em $\mathcal{C}$ e $\mathcal{M}$ foram os do grupo de 73 locutores que possuem dois ou mais vídeos no conjunto de dados YouTube.

Tomando os 73 locutores, um de seus vídeos foi escolhido como base de treinamento do UBM e subespaço T. As gravações para treinamento do UBM somam 27213, cada uma com 10 segundos de duração, o que totaliza em torno de 75 horas de áudio, sendo que a presença da fala entre as amostras é inferior a esse valor. Feito o treinamento de UBM e T, o processo de extração gerou um *i-vector* de 30 dimensões para cada gravação, sendo que o número de dimensões foi definido experimentalmente como o mais eficaz para separar locutores pela distância cosseno. Apesar dos *i-vectors* de (SCHMIDT; SHARIFI; MORENO, 2014) serem de 200 dimensões, experimentos mostraram que este valor de dimensões não foi eficaz quando aplicado sem a redução de dimensionalidade por LDA e normalização por WCCN, como realizado em (SCHMIDT; SHARIFI; MORENO, 2014).

A seleção dos conjuntos $\mathcal{C}$ e $\mathcal{M}$ ocorreu de forma aleatória. Para cada locutor, um dos vídeos não utilizados no treinamento do UBM foi selecionado como fonte de 10 *i-vectors* aleatórios para $\mathcal{C}$, enquanto os *i-vectors* de um de seus outros vídeos foram adicionados à $\mathcal{M}$. O método utilizado para medir a distância entre i-vectors foi a distância cosseno, semelhante a (SCHMIDT; SHARIFI; MORENO, 2014).

O treinamento do UBM e do subespaço T, assim como a extração de *i-vectors* foi realizada com o auxílio da ferramenta ALIZE (BONASTRE; WILS; MEIGNIER, 2005).

O resultado final das consultas nas modalidades de experimento é uma matriz de distâncias A_{cb}, que relaciona as gravações de consulta às da base de locutores em termos de distância. Nas primeiras duas modalidades de experimento, A_{cb} é uma matriz quadrada de dimensão ditada pelo tamanho do conjunto de dados considerado. Na terceira modalidade, a matriz A_{cb} é $730 \times D$ dimensional, sendo 730 o total de consultas e D denotado pelo tamanho de $\mathcal{M}$, valor variável em torno de 26000 e 27000 pela natureza aleatória da seleção.

O algoritmo RL-Sim recebeu como parâmetro o limite de vizinhança das listas de classificação k e o número de iterações T, recalculando similaridades pelo cômputo cumulativo das interseções encontradas entre topos de listas $\mathcal{N}(i,k)$ em T iterações. De forma semelhante aplicou-se o algoritmo ReckNN até o limite k de vizinhos mais próximos, com o número de iterações suficiente para que a pontuação de convergência do algoritmo não sofresse alteração superior ao valor do parâmetro ϵ. Com base nos estudos de variação de k, T e ϵ em (PEDRONETTE; TORRES, 2013a; PEDRONETTE; PENATTI; TORRES, 2014), as constantes de limite de vizinhança k para os dois algoritmos foram definidas com o valor $k = 15$, o número de iterações do RL-Sim como $T = 3$ e constante de convergência do ReckNN $\epsilon = 0.0125$. Finalizada a atualização das matrizes de distância, calculou-se as métricas de avaliação de recuperação *mean average precision*, precisões em pontos de referência e precisões interpoladas por pontos de referência de revocação, discutidas no final desta seção.

Na terceira modalidade de experimento, foram utilizadas as definições dos algoritmos de aprendizado não supervisionado descritos na Seção 3.3.4, uma vez que sua matriz A_{cb} não carrega informações recíprocas necessárias para a aplicação do aprendizado não supervisionado por RL-Sim ou ReckNN.

4.2.2 Identificação

A identificação, realizada para toda consulta de $\mathcal{C}$, foi realizada pela submissão da matriz de distância para o classificador kNN descrito na Seção 3.4. O classificador, definido com valor arbitrário k_{cl} de vizinhos mais próximos e gravação rec_i, contabilizou a recorrência dos locutores pertencentes ao topo de lista $\mathcal{N}(i, k_{cl})$. A média da precisão de identificação para cada consulta foi contabilizada como medida de eficácia e o valor de k_{cl} foi variado de forma a demonstrar o efeito dos algoritmos de aprendizado não supervisionado sobre as matrizes de distância.

Com o objetivo de avaliar os resultados de experimentos considerando tarefas de recuperação de objetos de multimidia, a precisão e a precisão média são frequentemente

utilizadas de forma a facilitar a comparação e promover visualização de diferenças entre abordagens. Definida pela fração entre os objetos relevantes e o total de objetos recuperados em uma dada consulta, a precisão é normalmente avaliada considerando os resultados obtidos nas primeiras posições das listas de resultados até a n-ésima, denotada por $P@n$. Formalmente, podemos definir:

$$precisão = \frac{|\{objetos\ recuperados\} \cap \{objetos\ relevantes\}|}{|\{objetos\ recuperados\}|} \tag{4.1}$$

em que $|\{objetos\ recuperados\}| = n$.

A precisão média (*Mean Average Precision* - MAP), baseia-se no cálculo da precisão a cada vez que um objeto relevante apareça na lista de resultados. Em seguida, é calculada a média dos valores das precisões obtidas. Formalmente, seja q um objeto de consulta e N_r o número de objetos relevantes à consulta q. Seja $(r_i|i = 1, 2, ..., d)$ um vetor de relevância ordenado até a profundidade d, no qual r indica a relevância da i-ésima pontuação que o item alcançou, sendo 0 (não relevante) ou 1 (relevante), a precisão média (AP) é definida como:

$$AP = \frac{1}{N_r} \sum_{i=1}^{d} \left(\frac{r_i}{i} \sum_{j=1}^{i} r_j \right). \tag{4.2}$$

A medida MAP é definida calculando-se a precisão média para uma série de consultas.

4.3 Resultados e Discussão

Os resultados dos experimentos de recuperação e identificação de locutor são apresentados nessa seção, incluindo discussões sobre aspectos relacionados às características de locutor, modelagens e algoritmo de aprendizado não supervisionado.

4.3.1 Resultados para recuperação de locutor

Conforme descrito anteriormente, a tarefa de recuperação de locutor é o passo anterior à identificação que tem por objetivo recuperar todas as gravações relevantes do banco de locutores para uma dada consulta. Gravações relevantes são gravações de mesmo locutor. A seguir são apresentados os resultados obtidos em tarefas de recuperação para cada uma das coleções consideradas.

4.3.1.1 CHAINS

Os resultados de recuperação para o conjunto de dados CHAINS são apresentados na Tabela 2, considerando diferentes características (MFCC e PLP), modelagens (VQ e

GMM) e algoritmos de aprendizado não supervisionado (RL-Sim e ReckNN). Em seguida o gráfico de precisão em função da revocação é apresentado na Figura 13. Nas tabelas adiante, a notação "s.a." indica resultados sem algoritmo de pós-processamento.

Tabela 2 – Resultados de recuperação para o conjunto de dados CHAINS.

(a) MFCC+GMM

	s.a.	RL-Sim	ReckNN
MAP [%]	9.08	37.99	10.13
P@5 [%]	96.45	99.92	95.05
P@10 [%]	50.77	99.79	51.02
P@15 [%]	34.76	99.67	34.93
P@20 [%]	26.72	99.49	26.96
P@25 [%]	21.95	99.31	22.29
P@30 [%]	18.75	99.02	19.30
P@35 [%]	16.43	98.66	17.28
P@40 [%]	14.71	91.65	15.87
P@45 [%]	13.38	81.98	14.85
P@50 [%]	12.35	74.18	14.12
P@55 [%]	11.50	67.79	13.56

(b) PLP+GMM

	s.a.	RL-Sim	ReckNN
MAP [%]	12.08	36.88	14.33
P@5 [%]	96.56	99.68	96.44
P@10 [%]	54.62	99.48	62.20
P@15 [%]	39.46	99.22	46.53
P@20 [%]	31.59	98.96	38.82
P@25 [%]	26.83	98.69	33.44
P@30 [%]	23.75	98.31	29.62
P@35 [%]	21.48	97.69	26.82
P@40 [%]	19.86	90.84	24.77
P@45 [%]	18.62	81.31	23.22
P@50 [%]	17.67	73.55	22.02
P@55 [%]	16.99	67.20	21.18

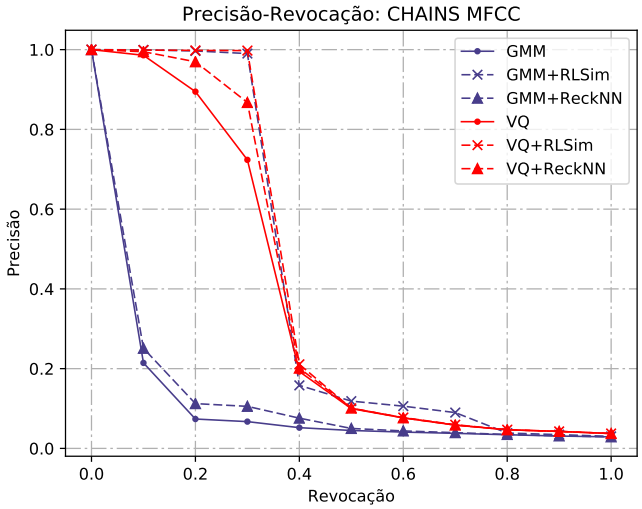
(c) MFCC+VQ

	s.a.	RL-Sim	ReckNN
MAP [%]	31.61	38.01	34.42
P@5 [%]	99.50	99.86	99.17
P@10 [%]	95.79	99.75	99.05
P@15 [%]	92.39	99.71	98.07
P@20 [%]	89.05	99.70	96.70
P@25 [%]	85.27	99.70	94.41
P@30 [%]	80.71	99.69	90.80
P@35 [%]	75.69	99.66	85.43
P@40 [%]	70.20	92.46	78.25
P@45 [%]	65.12	82.40	71.45
P@50 [%]	60.51	74.31	65.62
P@55 [%]	56.45	67.69	60.58

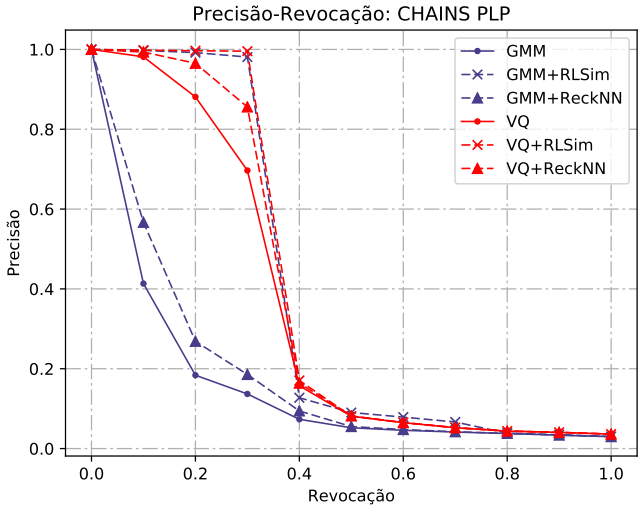
(d) PLP+VQ

	s.a.	RL-Sim	ReckNN
MAP [%]	30.42	37.28	33.54
P@5 [%]	99.41	99.82	99.04
P@10 [%]	95.15	99.67	98.86
P@15 [%]	91.49	99.59	97.84
P@20 [%]	87.95	99.50	96.36
P@25 [%]	83.87	99.46	94.07
P@30 [%]	79.25	99.42	90.35
P@35 [%]	74.05	99.38	84.83
P@40 [%]	68.64	92.20	77.56
P@45 [%]	63.63	82.17	70.76
P@50 [%]	59.13	74.07	64.91
P@55 [%]	55.14	67.42	59.89

As gravações relevantes por locutor neste conjunto totalizam 111, dentro das quais dois terços são compostos por gravações de qualidade inferior e com fala alterada. Os resultados iniciais, antes do pós-processamento, transparecem de forma clara essa condição do conjunto de dados. A baixa precisão de recuperação antes da metade da revocação é um sinal de que, até certo ponto, as modelagens não foram robustas o suficiente e quantidade/qualidade das amostras não se mostraram suficientes para modelagem. Neste caso, o desempenho baixo pode ser atribuído majoritariamente à falta de amostras para geração de modelos significantes de locutor, dado o alto nível de variabilidade presente entre as gravações de mesmo locutor. Costuma-se treinar modelos tomando por base múltiplas sentenças de dado locutor, ao invés de frases curtas, como realizado nos experimentos. O fato que as características de locutor e as técnicas de modelagem não serem completamente robustas por si próprias, mesmo em cenários com múltiplas amostras rotuladas, foi discutido no Capítulo 3.



(a) MFCC



(b) PLP

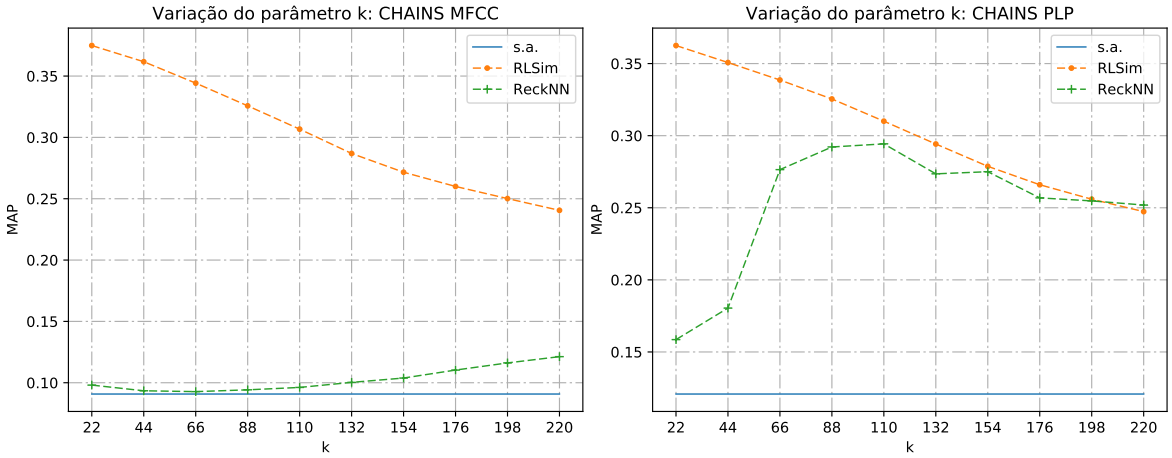
Figura 13 – Gráfico de precisão versus revocação para conjunto de dados CHAINS.

Os resultados mostraram que a melhor recuperação de locutor neste conjunto de dados é alcançada com a extração de MFCCs e modelagem por VQ. Apesar da base teórica superior apresentada pela modelagem GMM, é possível que a quantidade insuficiente de amostras no processo de modelagem seja o motivo de sua eficácia menor em comparação com a modelagem VQ.

Não obstante o baixo desempenho das técnicas clássicas de reconhecimento de locutor, ambos os algoritmos de pós-processamento recalcularam distâncias entre gravações, de forma a aperfeiçoar significativamente os resultados obtidos, atingindo, por exemplo, ganhos relativos de até +318% em termos de MAP (MFCC+GMM × MFCC+GMM+RL-Sim, Tabela 2a.

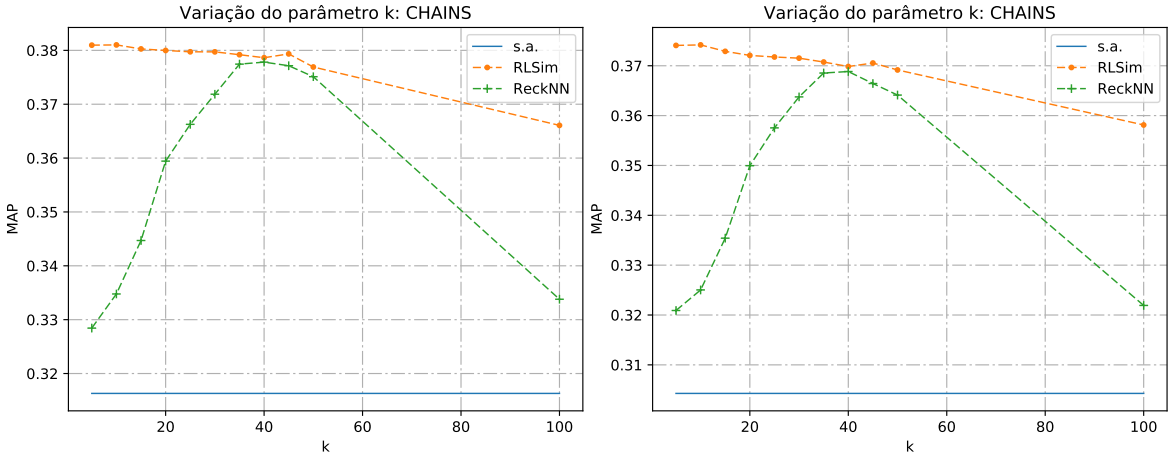
Comparando os resultados entre os algoritmos de aprendizado não supervisionado observa-se que o ReckNN teve desempenho inferior ao RL-Sim em todas as configurações, o que demonstra os efeitos entre suas diferentes abordagens para cálculo de novas distâncias.

A incapacidade do algoritmo ReckNN de recuperar gravações de forma tão efetiva quanto o RL-Sim neste conjunto de dados está relacionada ao valor do parâmetro k de profundidade de vizinhança. Os gráficos das Figuras 14 e 15 mostram a variação das precisões em função do valor de k conforme cada abordagem considerada no experimento.



(a) MFCC+GMM (b) PLP+GMM

Figura 14 – Gráficos da eficácia dos algoritmos de pós-processamento sobre o conjunto de dados CHAINS com modelagem GMM em função do parâmetro k.



(a) MFCC+VQ (b) PLP+VQ

Figura 15 – Gráficos da eficácia dos algoritmos de pós-processamento sobre o conjunto de dados CHAINS com modelagem VQ em função do parâmetro k.

Observa-se nos gráficos que, neste conjunto de dados, valores superiores a $k = 10$ degradam a precisão do RL-Sim, porém afetam positivamente o ReckNN. Destaca-se na Figura 14b o comportamento crescente do ReckNN proporcional ao aumento de vizinhança máxima k. De fato a variação do valor de k no intervalo $[35, 50]$, na modelagem VQ, traz resultados equivalentes entre os algoritmos RL-Sim e ReckNN, sendo os resultados do RL-Sim sempre levemente superiores.

4.3.1.2 Laps

A seguir na Tabela 3 são apresentados os resultados de recuperação de locutor no conjunto Laps, junto ao gráfico de precisão em função da revocação na Figura 16.

Tabela 3 – Resultados de recuperação para o conjunto de dados Laps.

(a) MFCC

	Laps – MFCC+GMM		
	s.a.	RL-Sim	ReckNN
MAP [%]	69.28	98.93	94.21
P@5 [%]	91.54	99.02	97.88
P@10 [%]	83.15	98.84	97.07
P@15 [%]	73.34	98.61	95.76
P@20 [%]	63.42	98.03	90.80
P@25 [%]	54.44	79.32	75.20
P@30 [%]	47.61	66.21	63.14
P@35 [%]	42.26	56.87	54.57
P@40 [%]	38.05	49.85	48.06
P@45 [%]	34.59	44.33	42.87
P@50 [%]	31.76	39.90	38.67
P@55 [%]	29.37	36.28	35.22

(b) PLP

	Laps – PLP+GMM		
	s.a.	RL-Sim	ReckNN
MAP [%]	89.20	99.45	99.53
P@5 [%]	98.71	99.65	99.82
P@10 [%]	96.2	99.35	99.75
P@15 [%]	91.81	99.23	99.69
P@20 [%]	82.94	98.94	99.04
P@25 [%]	70.17	79.59	79.63
P@30 [%]	60.12	66.41	66.44
P@35 [%]	52.41	57.04	57.00
P@40 [%]	46.47	49.99	49.90
P@45 [%]	41.70	44.44	44.36
P@50 [%]	37.83	40.00	39.93
P@55 [%]	34.61	36.36	36.30

(c) MFCC

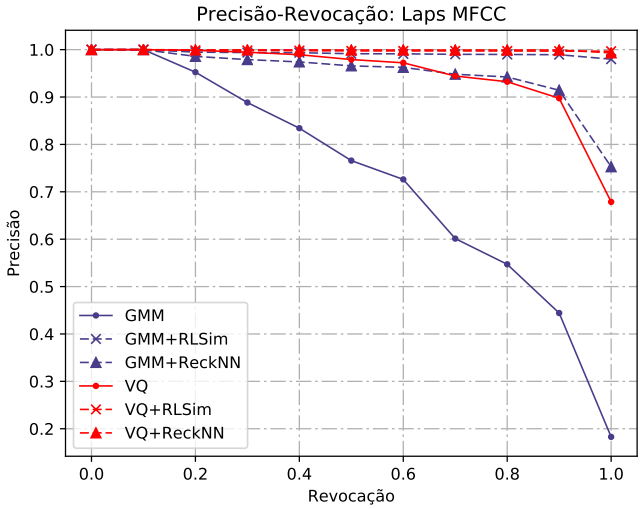
	Laps – MFCC+VQ		
	s.a.	RL-Sim	ReckNN
MAP [%]	94.30	99.67	99.90
P@5 [%]	99.6	99.71	99.94
P@10 [%]	98.41	99.48	99.94
P@15 [%]	95.91	99.53	99.94
P@20 [%]	89.66	99.54	99.75
P@25 [%]	74.56	79.82	79.90
P@30 [%]	63.22	66.57	66.60
P@35 [%]	54.78	57.10	57.09
P@40 [%]	48.3	49.99	49.97
P@45 [%]	43.14	44.44	44.44
P@50 [%]	38.99	40.00	40.00
P@55 [%]	35.51	36.36	36.36

(d) PLP

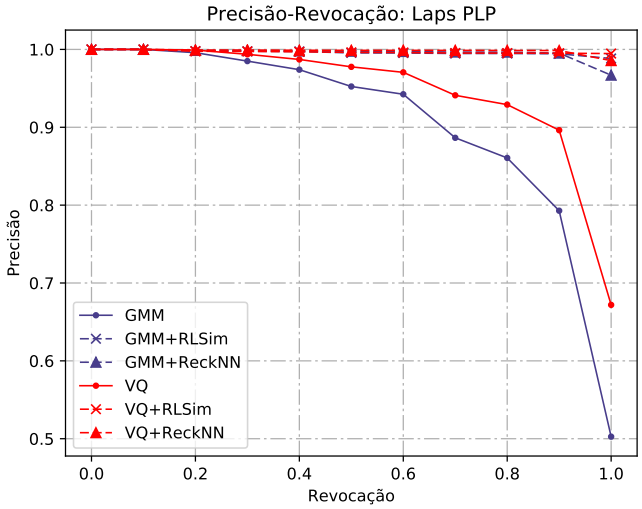
	Laps – PLP+VQ		
	s.a.	RL-Sim	ReckNN
MAP [%]	94.11	99.51	99.80
P@5 [%]	99.62	99.62	99.85
P@10 [%]	98.11	99.31	99.88
P@15 [%]	95.67	99.26	99.88
P@20 [%]	89.57	99.28	99.55
P@25 [%]	74.44	79.73	79.85
P@30 [%]	63.08	66.48	66.56
P@35 [%]	54.66	57.04	57.07
P@40 [%]	48.16	49.95	49.97
P@45 [%]	43.06	44.42	44.43
P@50 [%]	38.94	39.99	39.99
P@55 [%]	35.52	36.36	36.35

A precisão de recuperação elevada neste conjunto de dados é tida majoritariamente pela baixa variabilidade entre gravações de mesmo locutor, gravadas todas em uma única sessão, com as mesmas condições de canal e ambiente. A baixa quantidade total de gravações também pode contribuir no melhor desempenho do arcabouço.

Concordando com o aumento da precisão, os algoritmos de aprendizado não supervisionado aperfeiçoaram a recuperação de locutor em todas as configurações consideradas. Com exceção da extração por MFCCs unida à modelagem GMM, o algoritmo ReckNN teve o melhor desempenho de pós-processamento com margem pequena, apresentando precisões equivalentes ao pós-processamento com RLSim. A exceção, observada na Tabela 3a, corrobora com a conjectura sobre o desempenho do ReckNN sobre listas de classificação



(a) MFCC



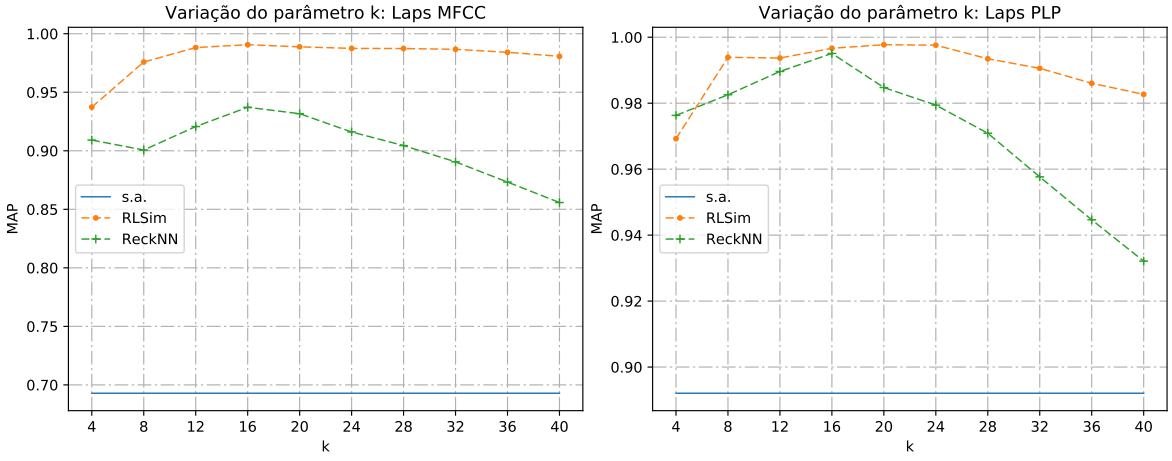
(b) PLP

Figura 16 – Gráfico de precisão versus revocação para conjunto de dados Laps.

de precisão inicial inferior. Nesta tabela observa-se o menor valor inicial de P@5 a P@15 entre as abordagens consideradas.

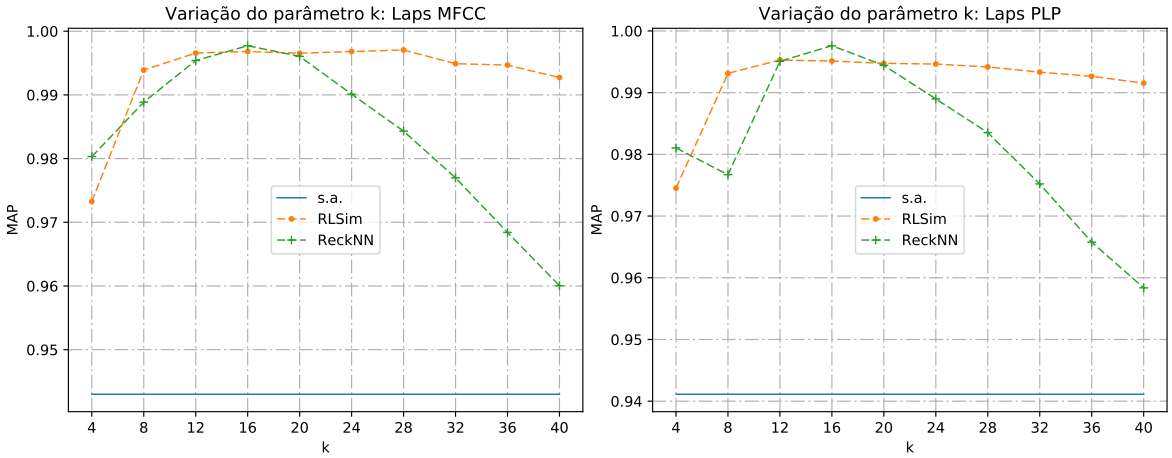
O estudo de variação do parâmetro k foi realizado no intervalo entre $k = 4$ a $k = 40$, contornando o valor de 20 gravações relevantes por locutor. Os gráficos de variação do parâmetro k estão dispostos nas Figuras 17 e 18.

Os resultados de variação do parâmetro k unidos às tabelas de precisões mostram que com k variando no intervalo $[12, 24]$ os dois algoritmos apresentam resultados aproximadamente equivalentes, com a exceção da modelagem MFCC+GMM. Observa-se nesta coleção que os resultados dos RL-Sim foram mais estáveis na maioria dos cenários, no sentido de apresentaram grandes variações em função da variação do parâmetro k.



(a) MFCC+GMM (b) PLP+GMM

Figura 17 – Gráficos da eficácia dos algoritmos de pós-processamento sobre o conjunto de dados Laps com modelagem GMM em função do parâmetro k.



(a) MFCC+VQ (b) PLP+VQ

Figura 18 – Gráficos da eficácia dos algoritmos de pós-processamento sobre o conjunto de dados Laps com modelagem VQ em função do parâmetro k.

4.3.1.3 YouTube

A Tabela 4 e a Figura 19 apresentam os resultados de recuperação para o conjunto de YouTube. Para este conjunto de maior porte foram realizadas 10 consultas randômicas para cada um dos 995 locutores únicos. Os resultados apresentados são a média de 5 tentativas, nas quais foram selecionadas randomicamente as 10 consultas de cada locutor. A origem das gravações dos locutores únicos são de mesmo vídeo, portanto não há diferenças de canal ou sessão entre gravações de mesmo locutor.

Os resultados positivos de recuperação são esperados pela similaridade entre gravações de mesmo locutor, mesmo entre a numerosa base de locutores. Dentre os métodos clássicos de reconhecimento de locutor, melhores resultados foram obtidos pela modelagem GMM, de forma equivalente entre os dois tipos de características de locutor consideradas. O desempenho superior da modelagem GMM neste conjuntos de dados pode ser atribuído

Tabela 4 – Resultados de recuperação para o conjunto de dados YouTube.

(a) MFCC

	YouTube – MFCC+GMM		
	s.a.	RL-Sim	ReckNN
MAP [%]	89.16	92.83	92.62
P@5 [%]	96.48	95.67	97.00
P@10 [%]	86.83	91.30	89.79
P@15 [%]	59.35	62.05	61.66
P@20 [%]	44.98	46.85	46.87
P@25 [%]	36.22	37.63	37.74
P@30 [%]	30.32	31.44	31.57
P@35 [%]	26.08	27.02	27.15
P@40 [%]	22.88	23.70	23.81
P@45 [%]	20.39	21.11	21.20
P@50 [%]	18.39	19.04	19.11
P@55 [%]	16.75	17.34	17.39

(b) PLP

	YouTube – PLP+GMM		
	s.a.	RL-Sim	ReckNN
MAP [%]	90.55	92.68	92.69
P@5 [%]	96.67	95.53	96.97
P@10 [%]	88.64	91.14	90.04
P@15 [%]	60.26	61.98	61.71
P@20 [%]	45.58	46.79	46.86
P@25 [%]	36.65	37.58	37.70
P@30 [%]	30.67	31.41	31.54
P@35 [%]	26.37	27.00	27.11
P@40 [%]	23.13	23.69	23.78
P@45 [%]	20.60	21.11	21.17
P@50 [%]	18.58	19.03	19.08
P@55 [%]	16.91	17.34	17.37

(c) MFCC

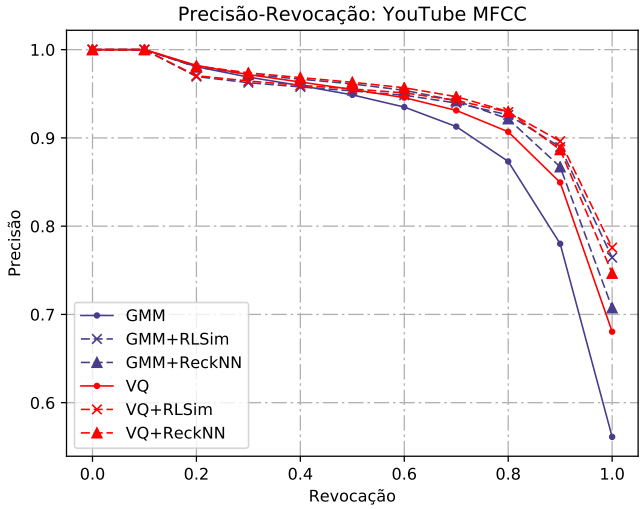
	YouTube – MFCC+VQ		
	s.a.	RL-Sim	ReckNN
MAP [%]	88.48	88.88	90.02
P@5 [%]	95.68	93.17	96.00
P@10 [%]	86.50	86.95	87.70
P@15 [%]	58.88	59.72	59.92
P@20 [%]	44.57	45.19	45.42
P@25 [%]	35.84	36.34	36.54
P@30 [%]	29.99	30.40	30.57
P@35 [%]	25.79	26.14	26.29
P@40 [%]	22.63	22.94	23.06
P@45 [%]	20.16	20.44	20.54
P@50 [%]	18.18	18.44	18.52
P@55 [%]	16.56	16.80	16.86

(d) PLP

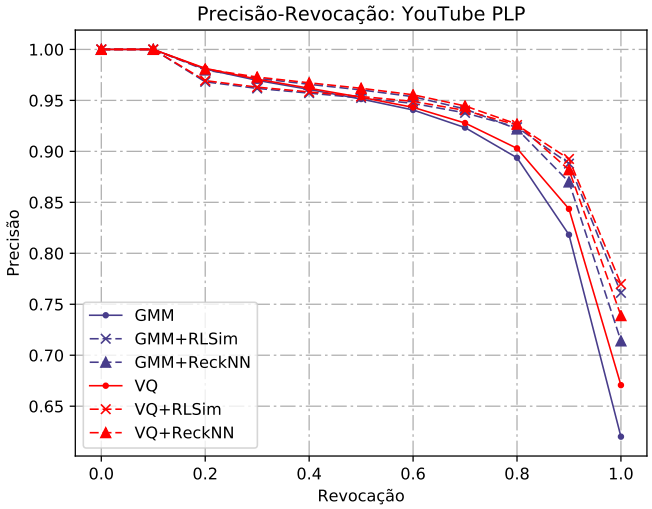
	YouTube – PLP+VQ		
	s.a.	RL-Sim	ReckNN
MAP [%]	88.17	88.53	89.75
P@5 [%]	95.53	92.97	95.84
P@10 [%]	86.17	86.58	87.41
P@15 [%]	58.70	59.50	59.77
P@20 [%]	44.43	45.04	45.30
P@25 [%]	35.74	36.22	36.44
P@30 [%]	29.91	30.30	30.49
P@35 [%]	25.72	26.06	26.23
P@40 [%]	22.56	22.87	23.00
P@45 [%]	20.10	20.38	20.49
P@50 [%]	18.13	18.39	18.46
P@55 [%]	16.51	16.75	16.81

à maior quantidade de amostras por gravação. Segundo o protocolo experimental, 10 segundos de duração por gravação equivalem a 1000 vetores de características acústicas de treinamento, em contraste com os outros conjuntos de dados que forneceram em torno de 300 vetores de características acústicas por gravação.

Os resultados de pós-processamento indicam uma eficácia equivalente entre RL-Sim e ReckNN, ambos sem ganho significativo de precisão em comparação com outras coleções de dados. Os maiores ganhos de eficácia foram obtidos para a modelagem VQ. Entretanto o estudo de variação do parâmetro k realizado sobre um dos subconjuntos aleatórios de consulta aponta que, para vizinhanças menores que $k = 10$, o ReckNN aperfeiçoa a modelagem clássica de locutor e supera o RL-Sim com maior margem de precisão, conforme representado nos gráficos das Figuras 20 e 21.



(a) MFCC

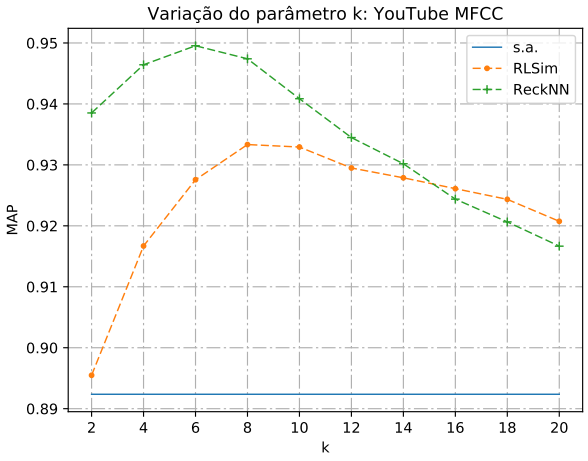


(b) PLP

Figura 19 – Gráfico de precisão versus revocação para conjunto de dados YouTube.

Figura 20 – Gráficos da eficácia dos algoritmos de pós-processamento sobre o conjunto de dados YouTube com modelagem GMM em função do parâmetro k.

(a) MFCC+GMM



(b) PLP+GMM

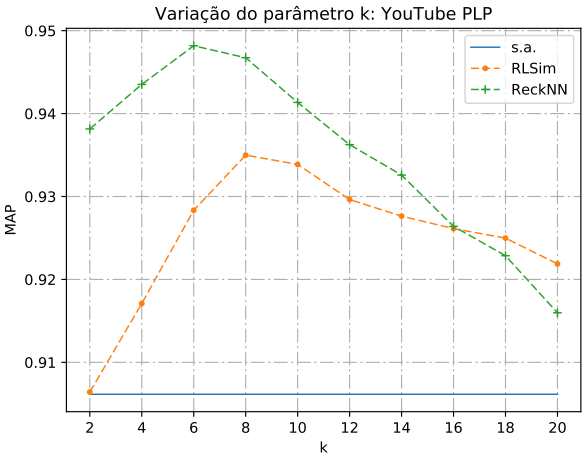
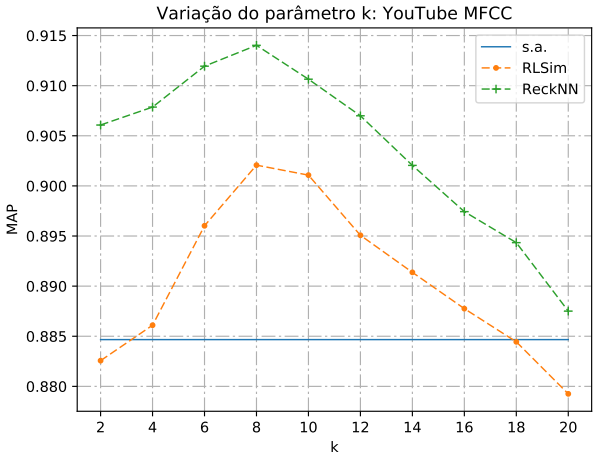
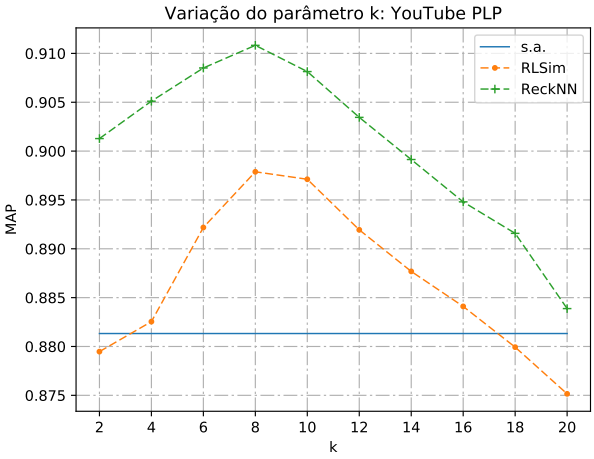


Figura 21 – Gráficos da eficácia dos algoritmos de pós-processamento sobre o conjunto de dados YouTube com modelagem VQ em função do parâmetro k.

(a) MFCC+VQ



(b) PLP+VQ



4.3.1.4 Estudo de eficácia dos algoritmos não supervisionados

Em resumo, os resultados de recuperação tiveram ganhos expressivos pela aplicação do pós-processamento por RL-Sim ou ReckNN. Estes dois algoritmos, porém, apresentaram diferentes comportamentos, variando entre as abordagens e conjuntos de dados considerados.

Relembrando o algoritmo ReckNN descrito no Capítulo 3, as distâncias por este calculadas consideram com maior peso as gravações iniciais das listas de classificação, analisando suas vizinhanças de forma crescente, porém atribuindo ênfase maior nas similaridades das posições iniciais. Em perspectiva, conjectura-se que este algoritmo tenha eficácia reduzida em meio a muitos não-relevantes espalhados pelo conjunto de dados: objetos não relevantes à consulta, mas que se encontram dentro dos limites de vizinhança da mesma. Por consequência, não-relevantes entre posições iniciais de vizinhança podem ou reduzir a pontuação de autoridade das listas de classificação e estagnar o avanço do algoritmo sobre relevantes ou privilegiar objetos não relevantes, diminuindo a eficácia do algoritmo frente a listas de ranqueamento com baixa eficácia nas primeiras posições.

Em contra-partida, o RL-Sim analisa as listas de ranqueamento de cada consulta em sua profundidade, comparando estas às listas de ranqueamento de todas as gravações da coleção. Dessa forma, a informação contextual entre uma consulta X_i e modelo m_j é calculada mesmo que $j \notin \mathcal{N}(i,k)$. Pela constatação que a diferença principal entre os dois algoritmos se encontra no caráter completista do RL-Sim, supõe-se que o motivo de sua eficácia superior também se encontra neste quesito. A análise dos resultados de precisão das listas de classificação nas Tabelas 2a e 2b, nas quais o algoritmo ReckNN teve eficácia inferior ao RLSim, revela que as precisões iniciais P@10 e P@15 destas tabelas, sem pós-processamento, apresentam taxas muito baixas de precisão. Entretanto, as posições

de lista que correspondem aos valores de precisão mencionados se encaixam dentro do parâmetro limítrofe de vizinhança definido ao ReckNN, fato que, portanto, pode inibir o potencial de eficácia do algoritmo.

Com o objetivo de aprofundar esse estudo e permitir a visualização de não-relevantes foram geradas representações de listas de classificação na forma de matriz de cores. A Figura 22 é a visualização das lista de classificação do conjunto de dados Laps com MFCCs e GMM. Entre as abordagens testadas no conjunto Laps, esta foi a que apresentou maior discrepância entre resultados do RL-Sim e ReckNN, conforme constata-se na Tabela 3. Na figura, as consultas estão dispostas na horizontal e suas respectivas listas de classificação na vertical, com começo na parte superior da imagem. *Pixeis* azuis representam gravações de locutores relevantes a consulta, a linha vermelha demarca o nível de vizinhança $k = 15$ e os *pixeis* em cinza representam gravações não relevantes porém de locutor idêntico ao da posição anterior ou posterior na respectiva lista de classificação. Os *pixeis* em cinza representam o efeito de agrupamento de locutor realizado pelos algoritmos de aprendizado não supervisionado sobre locutores não relevantes à consulta e a correlação de falsos positivos nos começos das listas.

O número de *pixeis* cinza dentro do limite de vizinhança $k = 15$ foi contabilizado por conjunto de dados e abordagem. Os valores correspondentes aos conjuntos de dados unidos aos seus respectivos MAPs por abordagem, como já descritos nas tabelas de precisão anteriores, são apresentados nas Tabelas 5, 6 e 7. Nas tabelas, "Pixeis N.R." apresenta a contagem de resultados não-relevantes nas primeiras posições das listas de ranqueamento.

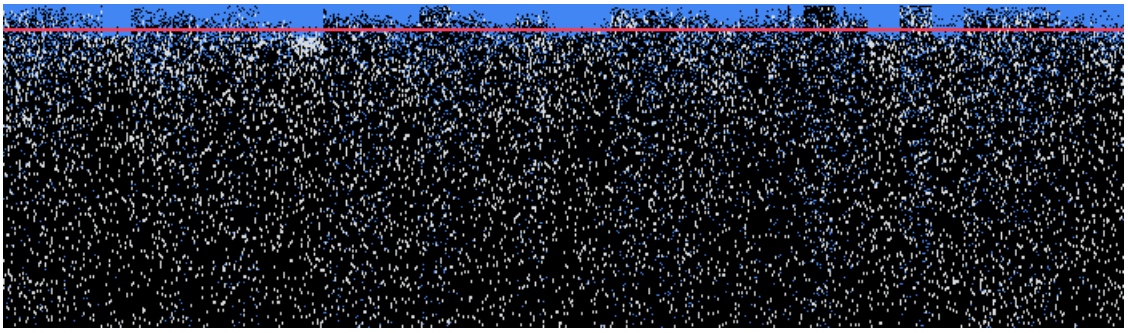
Tabela 5 – Contagem de não-relevantes nas vizinhanças com $k = 15$ para a coleção CHAINS.

CHAINS				
Modelagem	Característica	Pós-processamento	Pixeis N.R.	MAP[%]
GMM	MFCC	s.a.	12907	9.08
GMM	MFCC	RL-Sim	146	37.99
GMM	MFCC	ReckNN	15655	10.13
GMM	PLP	s.a.	12841	12.08
GMM	PLP	RL-Sim	289	36.88
GMM	PLP	ReckNN	13766	14.33
VQ	MFCC	s.a.	1152	31.61
VQ	MFCC	RL-Sim	80	38.01
VQ	MFCC	ReckNN	352	34.42
VQ	PLP	s.a.	1345	30.42
VQ	PLP	RL-Sim	150	37.28
VQ	PLP	ReckNN	484	33.54

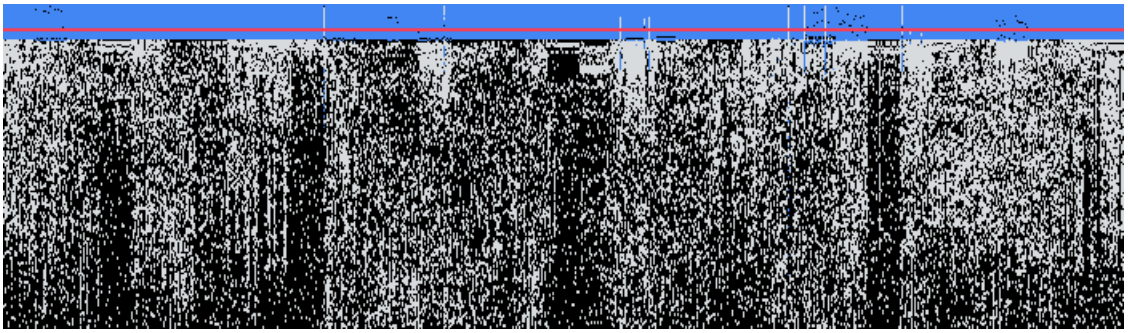
A comparação das Tabelas 5 e 6 traz evidências que corroboram com a conjectura que o algoritmo ReckNN teve eficiência degradada por correlações entre não relevantes nos topos de lista de classificação. Destacam-se as contagens de *pixeis* cinza nas abordagens GMM do conjunto CHAINS, nas quais o ReckNN teve pior desempenho. Nestas observa-se

Figura 22 – Representação das listas de classificação sobre o conjunto de dados Laps.

(a) MFCC+GMM



(b) MFCC+GMM+RLSim



(c) MFCC+GMM+ReckNN

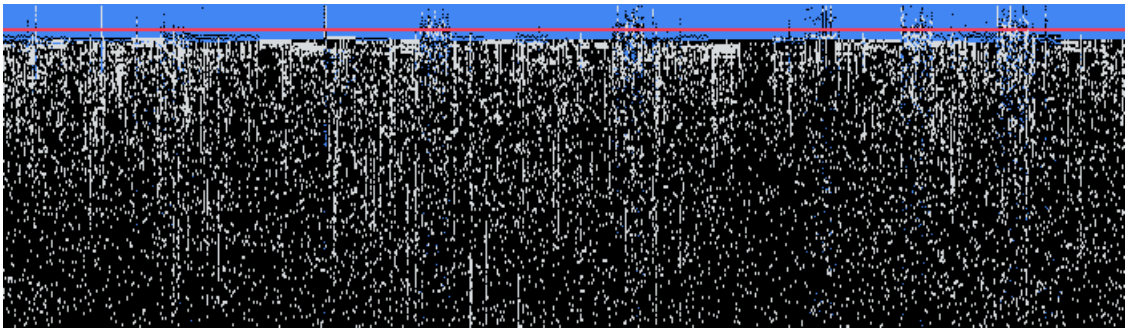


Tabela 6 – Contagem de não-relevantes nas vizinhanças com $k = 15$ para a coleção Laps.

Laps				
Modelagem	Característica	Pós-processamento	Pixeis N.R.	MAP[%]
GMM	MFCC	s.a.	469	69.28
GMM	MFCC	RL-Sim	92	98.93
GMM	MFCC	ReckNN	180	94.21
GMM	PLP	s.a.	152	89.20
GMM	PLP	RL-Sim	54	99.45
GMM	PLP	ReckNN	12	99.53
VQ	MFCC	s.a.	124	94.30
VQ	MFCC	RL-Sim	28	99.67
VQ	MFCC	ReckNN	2	99.90
VQ	PLP	s.a.	147	94.11
VQ	PLP	RL-Sim	42	99.51
VQ	PLP	ReckNN	6	99.80

Tabela 7 – Contagem de não-relevantes nas vizinhanças com $k = 15$ para a coleção YouTube.

YouTube				
Modelagem	Característica	Pós-processamento	Pixeis N.R.	MAP[%]
GMM	MFCC	s.a.	24837	89.16
GMM	MFCC	RL-Sim	46126	92.83
GMM	MFCC	ReckNN	34942	92.62
GMM	PLP	s.a.	26328	90.55
GMM	PLP	RL-Sim	47489	92.68
GMM	PLP	ReckNN	35108	92.96
VQ	MFCC	s.a.	28138	88.48
VQ	MFCC	RL-Sim	47976	88.88
VQ	MFCC	ReckNN	38774	90.02
VQ	PLP	s.a.	27502	88.17
VQ	PLP	RL-Sim	47800	88.53
VQ	PLP	ReckNN	38394	89.75

o aumento expressivo da contagem de resultados não-relevantes. Também observado na Tabela 7, relativa ao conjunto de dados YouTube, que demonstrou menor ganho de precisão frente aos algoritmos de pós-processamento, há grande correlação entre gravações não relevantes nas posições iniciais das listas, fator que compromete da eficácia dos algoritmos considerados.

4.3.2 Resultados para identificação de locutor

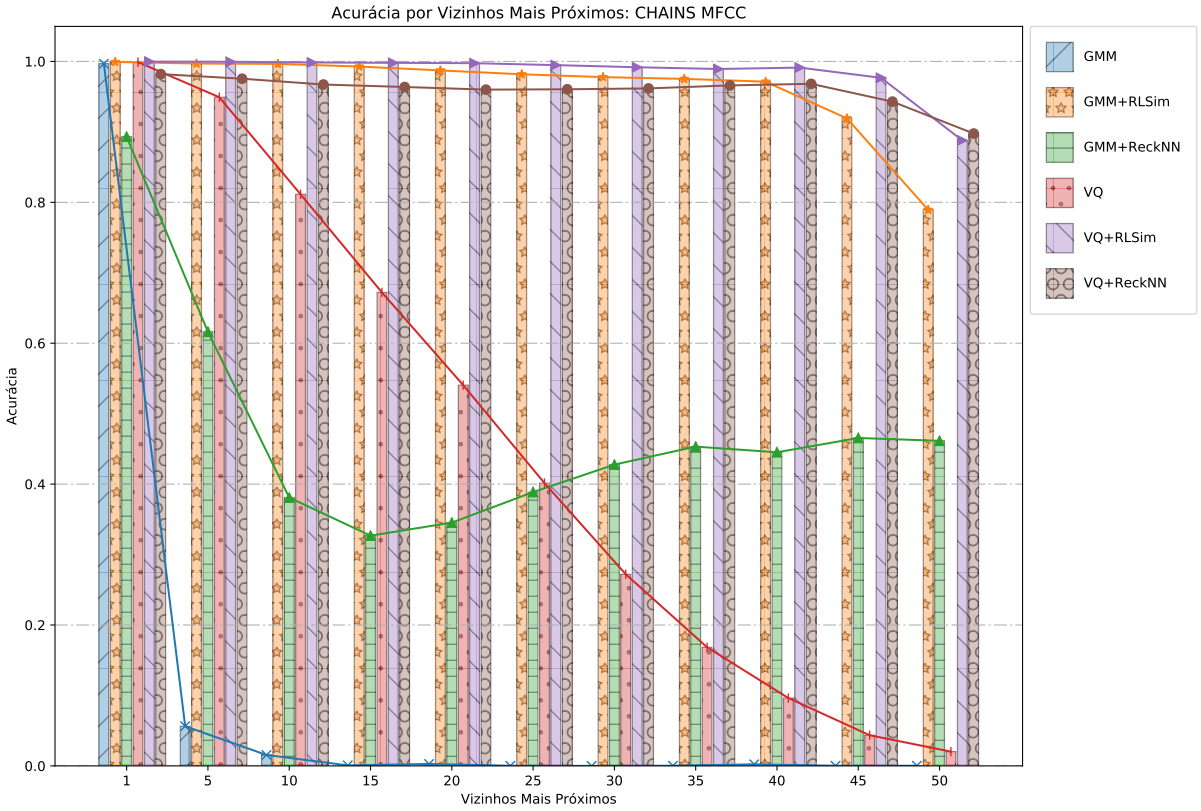
A identificação de locutor, realizada com base nas listas de ranqueamento geradas no processo de recuperação de locutor, ocorreu pela aplicação do classificador por vizinhos mais próximos, descrito na seção 3.4. A média de acurácia foi calculada com base na decisão do classificador para cada consulta. As Figuras de 23 a 28 apresentam os resultados de média de acurácia em função de valores crescentes de vizinhança mais próxima, usados como parâmetro do classificador. Cada figura considera uma característica e coleção de dados diferentes. Os parâmetros de vizinhança máxima k dos algoritmos RL-Sim e ReckNN foram definidas com os valores que atingiram melhor resultados de recuperação, segundo os experimentos conduzidos na primeira modalidade de experimento. A Tabela 8 apresenta os respectivos valores de k para cada algoritmo não supervisionado e coleção de dados.

Tabela 8 – Parâmetro k de vizinhança máxima.

	RL-Sim	ReckNN
CHAINS	10	40
Laps	16	16
YouTube	6	8

Os resultados de acurácia de identificação produzidos pela aplicação da Equação 3.25 indicam que as novas distâncias calculadas pelos algoritmos de aprendizado não

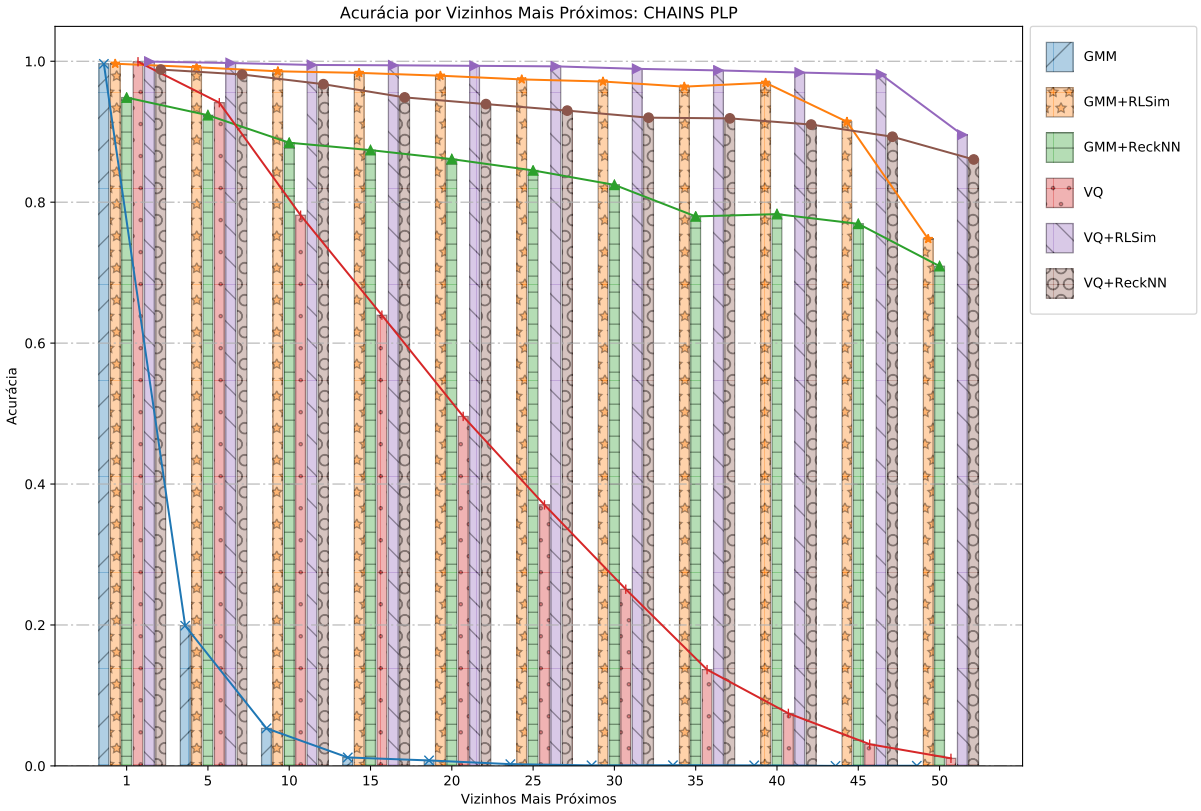
Figura 23 – Acurácia de identificação no conjunto de dados CHAINS usando MFCCs.



supervisionado foram eficazes em aproximar gravações relevantes entre si e afastar demais gravações não relevantes. Isto é observado na média das distâncias ponderada por aparição em $\mathcal{N}(c, k_{cl})$, na qual gravações relevantes por consulta c foram inferiores às de gravações pertencentes a outros locutores até níveis de vizinhança que ultrapassaram consideravelmente o número total de segmentos relevantes por locutor. Nesta questão, destaca-se a Figura 26 na qual, com o classificador kNN considerando as 50 primeiras posições, obteve-se taxas de acurácia de 99.71% para a modelagem VQ+ReckNN, 99.42% para a modelagem VQ+RL-Sim, 0.14% para a modelagem VQ sem pós-processamento, 98.71% para a modelagem GMM+ReckNN, 99.57% para a modelagem GMM+RL-Sim e 0% para a modelagem GMM sem pós-processamento. Portanto, os algoritmos de aprendizado não supervisionado aperfeiçoaram a identificação para taxas próximas de 100%, mesmo quando, dentre as 20 gravações relevantes por locutor do conjunto Laps, o classificador considerou os 50 vizinhos mais próximos.

Entre as acurácias do classificador com $k_{cl} = 1$ observa-se leve queda nos resultados com base nas distâncias calculadas pelo RL-Sim e ReckNN, o que demonstra um limite do aperfeiçoamento não supervisionado. Porém nota-se a exceção na Tabela 25, na qual o resultado inicial por GMM de 93.85% é aperfeiçoado pelo RL-Sim para 99.00% e 97.71% pelo ReckNN, com ganho relativo de +5.48% e +4.11% respectivamente.

Figura 24 – Acurácia de identificação no conjunto de dados CHAINS usando PLPs.



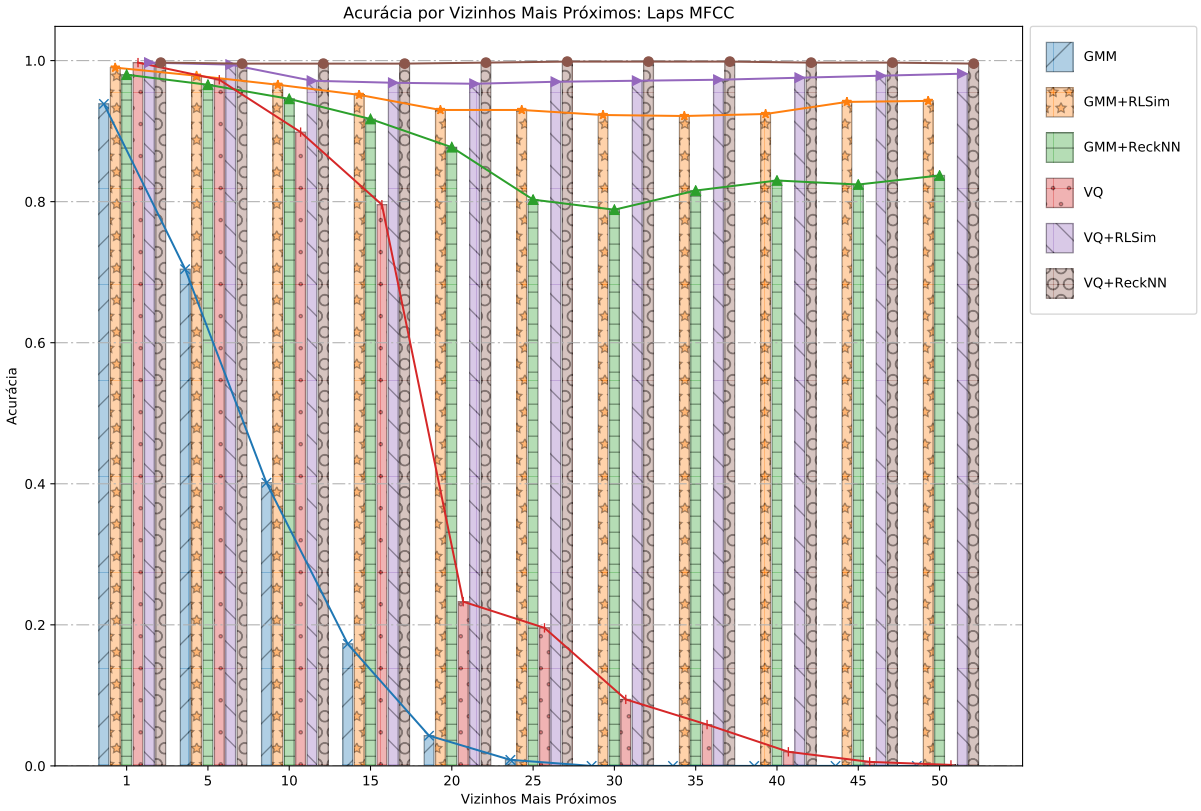
4.3.3 Experimento *holdout*

O experimento *holdout* foi conduzido com base no experimento nomeado *holdout10* em (SCHMIDT; SHARIFI; MORENO, 2014). Entretanto a versão do experimento realizada neste trabalho foi de proporção menor, por conta da complexidade de aplicação dos algoritmos de aprendizado não supervisionado sobre bases de locutores com mais de 50000 elementos. Aqui o tamanho do conjunto de consulta $\mathcal{C}_c$ foi de 730 gravações e o banco de locutores $\mathcal{C}_b$ foi formado por aproximadamente 26000 gravações.

Conforme descrito na seção 4.2, o protocolo experimental do conjunto *holdout* utilizou o aprendizado não supervisionado alterado para cumprir a condição do experimento *holdout10* de (SCHMIDT; SHARIFI; MORENO, 2014).

Em princípio, o mesmo protocolo experimental utilizado nos primeiros experimentos podia ser efetuado, dado que cada consulta fosse realizada e processada em separado das demais. Entretanto esse processo não foi uma alternativa viável nas condições de implementação do arcabouço disponíveis. O RL-Sim e o ReckNN aplicado sobre o conjunto de locutores completo, adicionado de uma consulta por vez, demonstra um custo computacional elevado sem a otimização do código que realiza tal tarefa. Portanto utilizou-se as propostas de alteração aos algoritmos de aprendizado não supervisionado, descritos na Seção 3.3.4, na condição em que não deseja-se recalcular as distâncias entre os objetos da

Figura 25 – Acurácia de identificação no conjunto de dados Laps usando MFCCs.



coleção de gravações $\mathcal{C}_b$.

Foram computadas métricas de avaliação de forma semelhante às outras modalidades de experimento, sendo essas comparadas com o desempenho sem e com pós-processamento. Também fez-se variação sobre o valor de k, em busca do melhor desempenho dos algoritmos. Os resultados apresentados nas Tabelas 9 e 10 são a média de 9 tentativas, nas quais foram selecionadas randomicamente as 10 consultas de cada um dos 73 locutores.

Tabela 9 – Recuperação com RL-Sim e ReckNN adaptados para conjunto de dados *holdout* do YouTube.

	YouTube - **holdout**					
	MAP [%]	P@5 [%]	P@10 [%]	P@15 [%]	P@20 [%]	P@60 [%]
i-vector	48.25	55.58	55.77	55.76	55.68	55.24
RL-Sim-alt-k15	49.37	56.98	56.86	56.58	56.32	55.55
ReckNN-alt-k15	53.67	58.05	57.96	57.81	57.69	57.40
RL-Sim-alt-k35	52.49	58.14	58.04	57.84	57.73	57.00
ReckNN-alt-k35	55.36	58.18	58.18	58.18	58.20	58.16
RL-Sim-alt-k50	54.30	58.65	58.58	58.40	58.26	57.65
ReckNN-alt-k50	**56.56**	**58.86**	**58.85**	**58.85**	**58.86**	**58.80**

Observa-se que a informação contextual das consultas $\mathcal{C}_c$ em relação as gravações de $\mathcal{C}_b$ unida à informação contextual entre as gravações de $\mathcal{C}_b$ pode ser usada para aperfeiçoar os resultados da recuperação e identificação de locutores. Ambos os algoritmos de aprendizado

Figura 26 – Acurácia de identificação no conjunto de dados Laps usando PLPs.

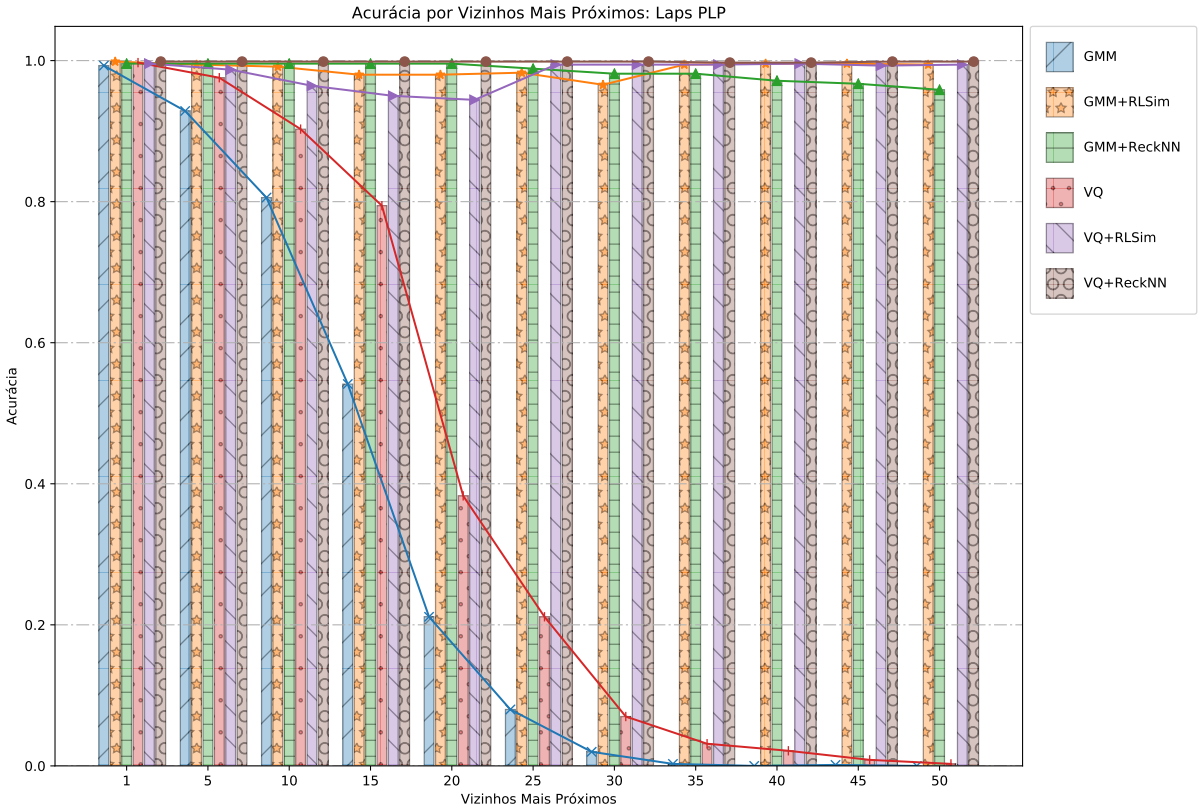


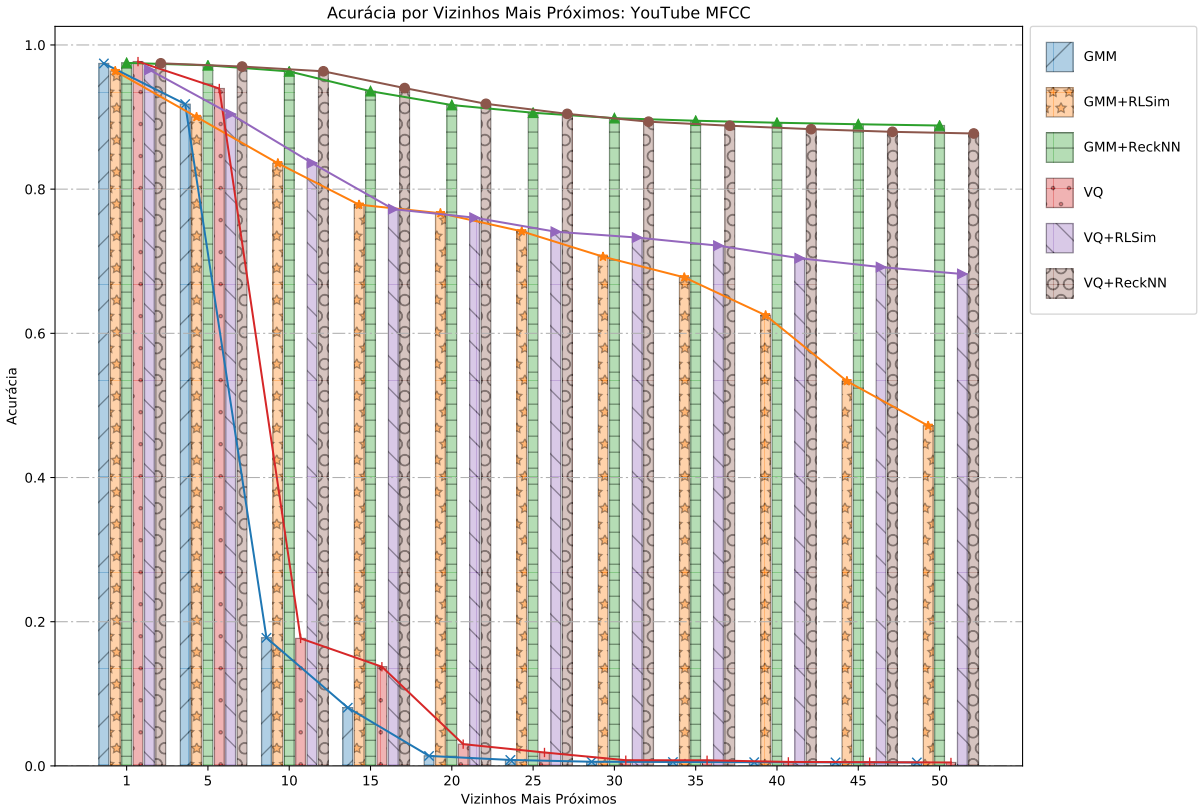
Tabela 10 – Acurácia de identificação com RL-Sim e ReckNN adaptados para conjunto de dados *holdout* do YouTube.

	k-Vizinhos mais Próximos										
	1	5	10	15	20	25	30	35	40	45	50
i-vector	54.99	50.35	48.20	46.31	44.62	43.16	42.67	41.90	41.61	41.06	40.59
RL-Sim-alt-k50	58.63	57.67	56.62	55.34	54.56	53.82	53.08	52.37	51.61	50.86	50.22
ReckNN-alt-k50	58.87	58.87	58.85	58.85	58.87	58.88	58.84	58.82	58.72	58.72	58.70

não supervisionado foram capazes de aperfeiçoar os resultados iniciais, atingindo, por exemplo, para o ReckNN alterado e com limite de vizinhança $k = 50$, ganhos relativos de até +17,22% em termos de MAP e +7,05% em acurácia de identificação kNN com $k_{cl} = 1$. O RL-Sim atingiu resultados levemente inferiores ao ReckNN, com ganhos relativos de até +12,53% em termos de MAP e +6,61% em acurácia de identificação kNN com $k_{cl} = 1$.

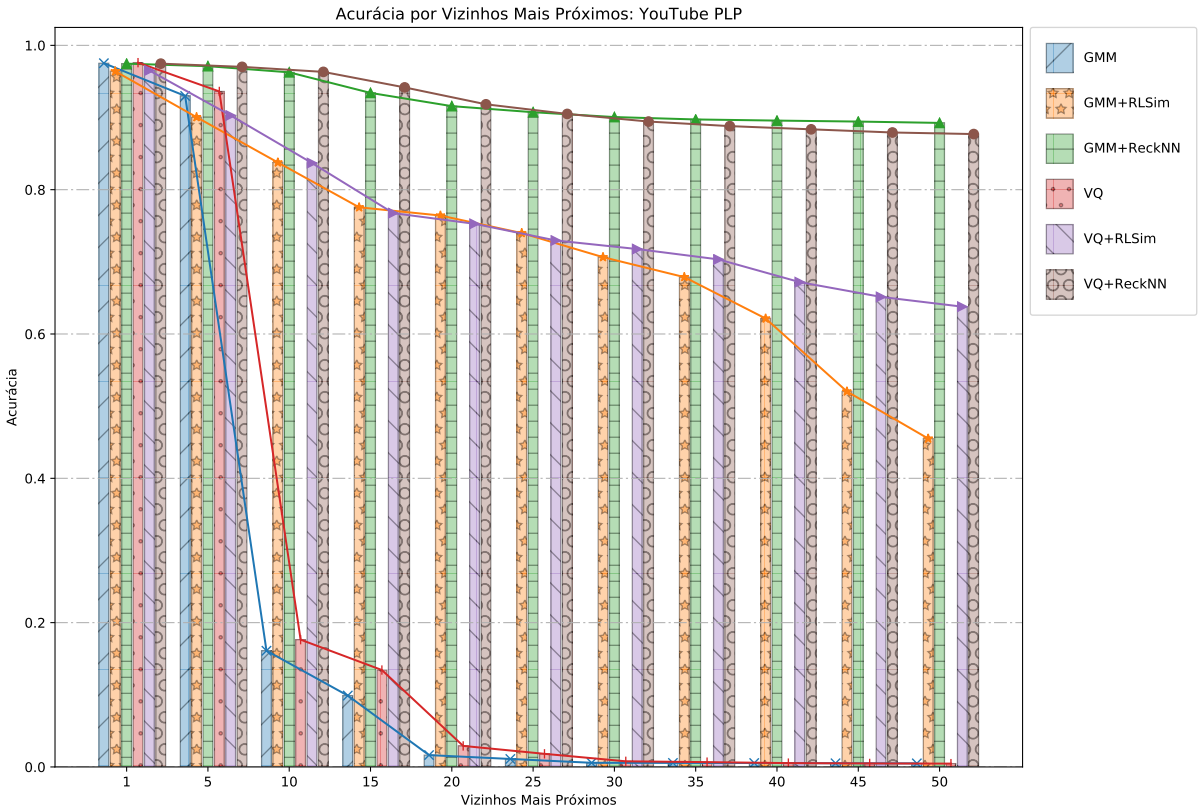
Em comparação com o experimento *holdout10*, a coleção reservada como banco de locutores C_b utilizada foi a metade da considerada em (SCHMIDT; SHARIFI; MORENO, 2014). Também quanto aos *i-vectors*, neste trabalho foram utilizados *i-vectors* de 30 dimensões, sem normalização por WCCN no processo de cálculo de distâncias. Em (SCHMIDT; SHARIFI; MORENO, 2014), foram usados *i-vectors* de 200 dimensões com normalização por LDA e WCCN no processo de cálculo de distâncias entre *i-vectors*. Os resultados apresentados na Tabela 10 são levemente superiores aos apresentados pela identificação em (SCHMIDT; SHARIFI; MORENO, 2014), por classificador kNN com $k_{cl} = 1$. O ganho

Figura 27 – Acurácia de identificação no conjunto de dados YouTube usando MFCCs.



relativo de acurácia de identificação é de +9.79% com o algoritmo RL-Sim alterado e +10.24% com o ReckNN alterado.

Figura 28 – Acurácia de identificação no conjunto de dados YouTube usando PLPs.



5 Conclusão e trabalhos futuros

Neste trabalho foi desenvolvido um arcabouço de reconhecimento de locutor que propõe unir técnicas clássicas e não supervisionadas de caracterização e modelagem de locutor à técnicas de pós-processamento por algoritmos de aprendizado não supervisionada. O arcabouço é desenvolvido como um sistema de recuperação de gravações por conteúdo, sendo o conteúdo explorado e alvo de recuperação as características de locutor de eficácia comprovada e utilizadas recentemente (KINNUNEN; LI, 2010; HANSEN; HASAN, 2015; SNYDER; GARCIA-ROMERO; POVEY, 2015; RICHARDSON; REYNOLDS; DEHAK, 2015).

A principal contribuição deste trabalho consiste na aplicação do pós-processamento com os algoritmos RL-Sim e ReckNN aos resultados de reconhecimento de locutor. O objetivo dessa fusão é recalcular distâncias/similaridades de forma a posicionar as gravações mais significativas nas primeiras posições das lista de classificação. Isto é realizado por meio da busca de informações contextuais codificadas em listas de classificação que relacionam tais gravações. Dessa forma, o pós-processamento baseado em algoritmos de aprendizado não supervisionado teve seu espectro de aplicação ampliado. Além de recuperação de imagens (PEDRONETTE; TORRES, 2014a), vídeos (ALMEIDA; PEDRONETTE; PENATTI, 2014) e aplicações biológicas (ALMEIDA et al., 2016), ganhos significativos também foram obtidos em reconhecimento de locutor.

A avaliação experimental conduzida nos moldes do arcabouço foi feita sobre três coleções de dados de características únicas, apresentando cenários variados à tarefa de reconhecimento de locutor: mistura entre gravações de qualidade e gravações ruidosas, gravações em inglês ou gravações em português, conjuntos de dados de pequena escala ou larga escala e completa separação de conteúdo e condições de gravação entre consultas de gravações do banco de dados. Os experimentos ratificaram a conjectura inicial, sobre a possibilidade de aperfeiçoamento de sistemas de reconhecimento de locutor de forma não supervisionada e eficiente pela utilização dos algoritmos de pós-processamento considerados.

O arcabouço como apresentado ainda pode ser aplicado na tarefa de agrupamento de locutores, na qual busca-se separar uma gravação de diálogo por interlocutores. Experimentos podem ser conduzidos nesse sentido com a divisão da gravação completa de diálogo em segmentos de curta duração e assumindo que cada segmento contém a fala de somente um interlocutor.

Outras possibilidades para trabalhos futuros incluem utilizar a abordagem com redes neurais profundas para extrair *bottleneck features* como características acústicas, assim como analisar o efeito da fusão de características acústicas. Também podem ser

realizados experimentos com cascateamento de modelagens e a fusão de abordagens de reconhecimentos de locutor. Além da possibilidade de fusão dos métodos discutidos por meio de abordagens convencionais de *ensemble learning*, o método não supervisionado de agregação por ranqueamento *rank aggregation* (PEDRONETTE; TORRES, 2013b; PEDRONETTE; PENATTI; TORRES, 2014) pode ser aplicado pelo RL-Sim ou ReckNN para fundir os resultados da recuperação de locutor realizada por diferentes tipos de características ou modelagens de locutor.

Referências

ABDEL-HAMID, O.; JIANG, H. Fast speaker adaptation of hybrid nn/hmm model for speech recognition based on discriminative learning of speaker code. In: IEEE. *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. [S.l.], 2013. p. 7942–7946. Citado na página 13.

ALMEIDA, J.; PEDRONETTE, D. C.; PENATTI, O. A. Unsupervised manifold learning for video genre retrieval. In: BAYRO-CORROCHANO, E.; HANCOCK, E. (Ed.). *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. [S.l.: s.n.], 2014, (Lecture Notes in Computer Science, v. 8827). p. 604–612. Citado 2 vezes nas páginas 30 e 78.

ALMEIDA, J. et al. Unsupervised distance learning for plant species identification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, v. 9, n. 12, p. 5325–5338, Dec 2016. Citado 2 vezes nas páginas 30 e 78.

ALMEIDA, J.; PEDRONETTE, D. C. G.; PENATTI, O. A. B. Unsupervised manifold learning for video genre retrieval. In: *CIARP*. [S.l.: s.n.], 2014. p. 604–612. Citado na página 29.

ALSULAIMAN, M.; MAHMOOD, A.; MUHAMMAD, G. Speaker recognition based on arabic phonemes. *Speech Communication*, v. 86, p. 42 – 51, 2017. ISSN 0167-6393. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167639315300649>. Citado na página 13.

ALVES, C. *FalaBrasil Project*. s.d. Disponível em: <http://www.laps.ufpa.br/falabrasil>. Acesso em: 20 jul. 2017. Citado 2 vezes nas páginas 32 e 54.

ARTHUR, D.; VASSILVITSKII, S. k-means++: The advantages of careful seeding. In: SOCIETY FOR INDUSTRIAL AND APPLIED MATHEMATICS. *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. [S.l.], 2007. p. 1027–1035. Citado na página 39.

BAI, X.; BAI, S.; WANG, X. Beyond diffusion process: Neighbor set similarity for fast re-ranking. *Information Sciences*, v. 325, p. 342 – 354, 2015. Citado na página 30.

BEIGI, H. *Fundamentals of speaker recognition*. [S.l.]: Springer Science & Business Media, 2011. 153-155 p. Citado 2 vezes nas páginas 36 e 38.

BISHOP, C. M. *Pattern Recognition and Machine Learning*. 1st ed. 2006. corr. 2nd printing. ed. [S.l.]: Springer, 2006. 431–438 p. (Information science and statistics). ISBN 9780387310732,0387310738. Citado na página 41.

BONASTRE, J.-F.; WILS, F.; MEIGNIER, S. Alize, a free toolkit for speaker recognition. In: IEEE. *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*. [S.l.], 2005. v. 1, p. I–737. Citado 2 vezes nas páginas 43 e 58.

BROOKES, M. *VOICEBOX: Speech Processing Toolbox for MATLAB*. s.d. Disponível em: <http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html>. Acesso em: 20 jul. 2017. Citado na página 43.

BURTON, D. Text-dependent speaker verification using vector quantization source coding. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, IEEE, v. 35, n. 2, p. 133–143, 1987. Citado na página 13.

CAMPBELL, W. et al. Support vector machines for speaker and language recognition. *Computer Speech & Language*, v. 20, n. 2–3, p. 210 – 229, 2006. ISSN 0885-2308. Odyssey 2004: The speaker and Language Recognition WorkshopOdyssey-04Odyssey 2004: The speaker and Language Recognition Workshop. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0885230805000318>. Citado 2 vezes nas páginas 24 e 25.

CAMPOS, V. d. A.; PEDRONETTE, D. C. G. Effective speaker retrieval and recognition through vector quantization and unsupervised distance learning. In: ACM. *Proceedings of the 1st International Workshop on Multimedia Analysis and Retrieval for Multimodal Interaction*. [S.l.], 2016. p. 27–32. Citado na página 54.

CHEN, Y. et al. Ranking consistency for image matching and object retrieval. *Pattern Recognition*. On-Line. Citado na página 30.

CIERI, C. et al. *The mixer corpus of multilingual, multichannel speaker recognition data*. [S.l.], 2004. Citado na página 32.

DEHAK, N. et al. Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification. In: *Tenth Annual conference of the international speech communication association*. [S.l.: s.n.], 2009. Citado 3 vezes nas páginas 25, 42 e 43.

DEHAK, N. et al. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, IEEE, v. 19, n. 4, p. 788–798, 2011. Citado 3 vezes nas páginas 27, 42 e 43.

DELIYSKI STEVE AN XUE, D. Effects of aging on selected acoustic voice parameters: Preliminary normative data and educational implications. *Educational Gerontology*, Taylor & Francis, v. 27, n. 2, p. 159–168, 2001. Citado na página 20.

DODDIPATLA, R.; HASAN, M.; HAIN, T. Speaker dependent bottleneck layer training for speaker adaptation in automatic speech recognition. In: *Fifteenth Annual Conference of the International Speech Communication Association*. [S.l.: s.n.], 2014. Citado na página 13.

DONOSER, M.; BISCHOF, H. Diffusion Processes for Retrieval Revisited. In: *CVPR*. [S.l.: s.n.], 2013. p. 1320–1327. Citado na página 29.

ELLIS, D. P. W. *PLP and RASTA (and MFCC, and inversion) in Matlab*. 2005. Disponível em: <http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/>. Acesso em: 20 jul. 2017. Citado na página 38.

GARCIA-ROMERO, D.; ESPY-WILSON, C. Y. Analysis of i-vector length normalization in speaker recognition systems. In: *Interspeech*. [S.l.: s.n.], 2011. v. 2011, p. 249–252. Citado na página 42.

GODFREY, J. J.; HOLLIMAN, E. C.; MCDANIEL, J. Switchboard: Telephone speech corpus for research and development. In: IEEE. *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*. [S.l.], 1992. v. 1, p. 517–520. Citado na página 31.

GRIMALDI, M.; CUMMINS, F. Speaker identification using instantaneous frequencies. *IEEE Transactions on Audio, Speech, and Language Processing*, IEEE, v. 16, n. 6, p. 1097–1111, 2008. Citado 2 vezes nas páginas 32 e 54.

HANSEN, J. H. L.; HASAN, T. Speaker recognition by machines and humans: A tutorial review. *IEEE Signal Processing Magazine*, v. 32, n. 6, p. 74–99, 2015. Citado 4 vezes nas páginas 14, 20, 21 e 78.

HAUTAMÄKI, V.; KINNUNEN, T.; FRÄNTI, P. Text-independent speaker recognition using graph matching. *Pattern Recognition Letters*, Elsevier, v. 29, n. 9, p. 1427–1432, 2008. Citado 3 vezes nas páginas 22, 31 e 32.

HIGGINS, A.; VERMILYEA, D. *KING Speaker Verification LDC95S22*. 1995. Disponível em: <https://catalog.ldc.upenn.edu/LDC95S22>. Acesso em: 20 jul. 2017. Citado na página 32.

HUANG, X.; LEE, K.-F. On speaker-independent, speaker-dependent, and speaker-adaptive speech recognition. *IEEE Transactions on Speech and Audio processing*, IEEE, v. 1, n. 2, p. 150–157, 1993. Citado na página 13.

JIANG, J.; WANG, B.; TU, Z. Unsupervised metric learning by self-smoothing operator. In: *ICCV*. [S.l.: s.n.], 2011. p. 794–801. Citado na página 29.

JR, J. P. C. Testing with the yoho cd-rom voice verification corpus. In: IEEE. *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*. [S.l.], 1995. v. 1, p. 341–344. Citado na página 31.

JR, J. P. C.; REYNOLDS, D. et al. Corpora for the evaluation of speaker recognition systems. In: IEEE. *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*. [S.l.], 1999. v. 2, p. 829–832. Citado na página 31.

KAIN, A. B. *High resolution voice transformation*. Tese (Doutorado) — Oregon Health & Science University, 2001. Citado na página 32.

KENNY, P. et al. 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. Citado na página 42.

KINNUNEN, T.; KILPELAINEN, T.; FRANTI, P. Comparison of clustering algorithms in speaker identification. *dim*, v. 1, p. 2, 2011. Citado na página 23.

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. Citado 9 vezes nas páginas 13, 20, 21, 22, 24, 25, 26, 39 e 78.

LANE, H.; TRANEL, B. The lombard sign and the role of hearing in speech. *Journal of Speech, Language, and Hearing Research*, v. 14, n. 4, p. 677–709, 1971. Disponível em: <+ http://dx.doi.org/10.1044/jshr.1404.677>. Citado na página 20.

LDC. *Linguistic Data Consortium*. 1992. Disponível em: <https://www.ldc.upenn.edu/>. Acesso em: 20 jul. 2017. Citado na página 32.

LEE, H. et al. Unsupervised feature learning for audio classification using convolutional deep belief networks. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2009. p. 1096–1104. Citado na página 13.

LEI, Y. et al. 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. Citado 3 vezes nas páginas 13, 25 e 27.

LIU, Y. et al. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, v. 40, n. 1, p. 262 – 282, 2007. Citado na página 29.

MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. *Introduction to Information Retrieval*. [S.l.]: Cambridge University Press, 2008. ISBN 0521865719, 9780521865715. Citado 3 vezes nas páginas 7, 26 e 47.

MILNER, B. A comparison of front-end configurations for robust speech recognition. In: IEEE. *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*. [S.l.], 2002. v. 1, p. I–797. Citado na página 37.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Citado na página 43.

PEDRONETTE, D. C. G.; ALMEIDA, J.; TORRES, R. da S. A scalable re-ranking method for content-based image retrieval. *Information Sciences*, v. 265, n. 1, p. 91–104, 2014. Citado na página 30.

PEDRONETTE, D. C. G.; PENATTI, O. A.; TORRES, R. d. S. Unsupervised manifold learning using reciprocal knn graphs in image re-ranking and rank aggregation tasks. *Image and Vision Computing*, Elsevier, v. 32, n. 2, p. 120–130, 2014. Citado 6 vezes nas páginas 30, 31, 47, 48, 58 e 79.

PEDRONETTE, D. C. G.; TORRES, R. da S. Image re-ranking and rank aggregation based on similarity of ranked lists. *Pattern Recognition*, v. 46, n. 8, p. 2350–2360, 2013. Citado 4 vezes nas páginas 30, 45, 46 e 58.

PEDRONETTE, D. C. G.; TORRES, R. da S. Image re-ranking and rank aggregation based on similarity of ranked lists. *Pattern Recognition*, v. 46, n. 8, p. 2350–2360, 2013. Citado na página 79.

PEDRONETTE, D. C. G.; TORRES, R. da S. Unsupervised distance learning by reciprocal knn distance for image retrieval. In: *International Conference on Multimedia Retrieval (ICMR'14)*. [S.l.: s.n.], 2014. Citado na página 78.

PEDRONETTE, D. C. G.; TORRES, R. da S. Unsupervised manifold learning using reciprocal knn graphs in image re-ranking and rank aggregation tasks. *Image and Vision Computing*, v. 32, n. 2, p. 120–130, 2014. Citado na página 30.

REYNOLDS, D. A.; QUATIERI, T. F.; DUNN, R. B. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, v. 10, n. 1, p. 19 – 41, 2000. ISSN 1051-2004. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1051200499903615>. Citado 2 vezes nas páginas 24 e 31.

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. Citado 4 vezes nas páginas 21, 23, 41 e 57.

RICHARDSON, F.; REYNOLDS, D.; DEHAK, N. Deep neural network approaches to speaker and language recognition. *IEEE Signal Processing Letters*, IEEE, v. 22, n. 10, p. 1671–1675, 2015. Citado 7 vezes nas páginas 13, 24, 27, 28, 31, 32 e 78.

SCHMIDT, L.; SHARIFI, M.; MORENO, I. L. Large-scale speaker identification. In: IEEE. *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. [S.l.], 2014. p. 1650–1654. Citado 7 vezes nas páginas 32, 54, 55, 56, 57, 73 e 75.

SENOUSSAOUI, M. et al. An i-vector extractor suitable for speaker recognition with both microphone and telephone speech. In: *Odyssey*. [S.l.: s.n.], 2010. p. 6. Citado 2 vezes nas páginas 31 e 32.

SENOUSSAOUI, M. et al. A study of the cosine distance-based mean shift for telephone speech diarization. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, IEEE Press, v. 22, n. 1, p. 217–227, 2014. ISSN 2329-9290. Citado na página 13.

SHANNON, C. E. Communication in the presence of noise. *Proceedings of the IRE*, IEEE, v. 37, n. 1, p. 10–21, 1949. Citado na página 19.

SHEN, X. et al. Object retrieval and localization with spatially-constrained similarity measure and k-nn re-ranking. In: *CVPR*. [S.l.: s.n.], 2012. p. 3013 –3020. Citado na página 30.

SHUM, S. H. et al. Unsupervised clustering approaches for domain adaptation in speaker recognition systems. Proc. IEEE Odyssey, p. 265–272, 2014. Citado 2 vezes nas páginas 17 e 56.

SMITH, S. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Pub., 1997. ISBN 9780966017632. Disponível em: <https://books.google.com.br/books?id=rp2VQgAACAAJ>. Citado na página 19.

SNYDER, D.; GARCIA-ROMERO, D.; POVEY, D. Time delay deep neural network-based universal background models for speaker recognition. In: IEEE. *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. [S.l.], 2015. p. 92–97. Citado 5 vezes nas páginas 13, 27, 31, 32 e 78.

STEVENS, S. S. On the psychophysical law. *Psychological review*, American Psychological Association, v. 64, n. 3, p. 153, 1957. Citado na página 37.

STEVENS, S. S.; VOLKMANN, J. The relation of pitch to frequency: A revised scale. *The American Journal of Psychology*, JSTOR, v. 53, n. 3, p. 329–353, 1940. Citado na página 37.

TEKTRONIX. Understanding fft overlap processing. 2009. Disponível em: <http://materias.fi.uba.ar/6644/info/anespec/avanzado/real time/ understanding FFT Overlap Processing.pdf>. Citado na página 21.

TZAGKARAKIS, C.; MOUCHTARIS, A. Robust text-independent speaker identification using short test and training sessions. In: *Proc. European Signal Processing Conf.(EUSIPCO)*. [S.l.: s.n.], 2010. p. 586–590. Citado na página 32.

WANG, J. *Physiologically-motivated feature extraction methods for speaker recognition*. Tese (Doutorado) — Faculty of the Graduate School, Marquette University, 2013. Citado na página 31.

WANG, J.; JOHNSON, M. T. Physiologically-motivated feature extraction for speaker identification. In: IEEE. *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. [S.l.], 2014. p. 1690–1694. Citado na página 31.

WOLF, J. J. Efficient acoustic parameters for speaker recognition. *The Journal of the Acoustical Society of America*, ASA, v. 51, n. 6B, p. 2044–2056, 1972. Citado 2 vezes nas páginas 13 e 20.

YANG, X.; KOKNAR-TEZEL, S.; LATECKI, L. J. Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In: *CVPR*. [S.l.: s.n.], 2009. p. 357–364. Citado na página 29.

YANG, X.; PRASAD, L.; LATECKI, L. Affinity learning with diffusion on tensor product graph. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, v. 35, n. 1, p. 28–38, 2013. Citado na página 29.

YOUNG, S. J. et al. *The HTK Book, version 3.4*. Cambridge, UK: Cambridge University Engineering Department, 2006. Citado na página 38.

YOUTUBE. *Busca no YouTube por Google Tech Talks*. 2007. Disponível em: <https://www.youtube.com/results?q=GoogleTechTalks>. Acesso em: 20 jul. 2017. Citado 2 vezes nas páginas 33 e 54.

YOUTUBE. *YouTube by the numbers*. 2017. Disponível em: <https://www.youtube.com/yt/about/press/>. Acesso em: 20 jul. 2017. Citado na página 55.

ZHANG, Y.; CHUANGSUWANICH, E.; GLASS, J. Extracting deep neural network bottleneck features using low-rank matrix factorization. In: IEEE. *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. [S.l.], 2014. p. 185–189. Citado 2 vezes nas páginas 27 e 28.