http://journal.unoeste.br/index.php/ca/index
DOI: 10.5747/ce.2019.v11.n2.e277
ISSN on-line 2178-8332

Colloquium Exactarum

Submetido: 02/12/2018 Revisado: 03/03/2019 Aceito: 16/06/2019

MÉTODOS COMPUTACIONAIS APLICADOS EM RECONHECIMENTO SONORO E BIOMETRIA POR VOZ

COMPUTATIONAL METHODS APPLIED IN SOUND RECOGNITION AND VOICE BIOMETRICS

Sabrina Cristina da Silva[1]; Danillo Roberto Pereira[1]; Francisco Assis da Silva[1]; Helton Molina Sapia[1]

[1]Universidade do Oeste Paulista – UNOESTE
Faculdade de Informática de Presidente Prudente – FIPP
E-mail: fipp@fipp.unoeste.br

RESUMO – Este trabalho apresenta o projeto, construção, desenvolvimento e análise da submissão de bases de dados à métodos computacionais utilizados no reconhecimento sonoro. Estão descritos todos os detalhes da metodologia utilizada no desenvolvimento e na coleta de dados, bem como as especificações de cada base de dados utilizada. Ponderações e resultados obtidos nas fases de tratamento dos sinais de voz, resultados das aplicações dos métodos extratores de características utilizados e dos métodos de aprendizado de máquina. Por fim, é avaliado o potencial da autenticação de cada método extrator de características combinados entre os diferentes métodos de aprendizado de máquina utilizado nas determinadas bases de dados.
Palavras-chave: Biometria por voz; extração de características; processamento digital de sinais; reconhecimento sonoro.

ABSTRACT – This work presents the design, construction, development and analysis of the submission of databases to computational methods used in speech recognition. There are all the details of the methodology used, without development and data collection, as well as specifications of each database used. Weights and results obtained in the phases of treatment of voice signals, results of the applications of the extraction methods of characteristics used and the methods of machine learning. Finally, the potential of the authentication of each combined characteristic extraction method between the different machine learning methods used in the databases is evaluated.
Keywords: Biometric voice; extract features; digital signal processing; speech recognition.

1. INTRODUÇÃO

Identidade sonora é uma característica composta por idioma, timbre de voz, sotaque, dicção e todos aspectos notáveis que definem a maneira com que um indivíduo fala, o que compõe um fator biométrico (COSTA, 2013).

Atualmente, este fator biométrico tem sido muito utilizado em ferramentas avançadas de identificação e autenticação de pessoas, a fim de incrementar mecanismos de segurança, mecanismos de inclusão e acessibilidade tecnológica, projetos de assistentes pessoais que atendem a comandos de voz, projetos de conversão interlinguística em tempo real, dentre outros. O uso da voz como recurso biométrico se destaca pela simplicidade, naturalidade e facilidade com que as amostras podem ser coletadas, sem exigir o uso de dispositivos de captura especializados. Além disso, a utilização de documentos para a identificação de indivíduos não se mostra mais eficiente e adequada aos cenários atuais, onde encontramos alto nível de mobilidade e conectividade. Dessa forma, o desenvolvimento de sistemas computacionais capazes de processar sinais de voz de diversos modos é um desafio importante para a evolução das tecnologias biométricas e aos conceitos de processamento de sinais (FIGUEIREDO, 1999).

Sistemas de processamento de sinais de voz normalmente utilizam-se de diversos modelos de representação como Modelos Baseados em Voz (Vocal Tract Length Normalization (VTLN), Frequências Formantes e Modelos de Estimação de Fluxo Global); Modelos Baseados em Voz/Sinal (Linear Predictive Coding (LPC), Line Spectral Frequencies (LSF) e Cepstral Coefficients).

Modelos baseados em voz fundamentam-se nos mecanismos de produção da voz humana, como o conceito de pulso glotal - sinal produzido pelas cordas vocais - e trato vocal. É o trato vocal que compreende a ação dos ressoadores, tais como: as cavidades oral e nasal, palatos, língua, dentes e lábios. Todos estes elementos são os responsáveis por caracterizar a maior parte da identidade sonora de um falante. Modelos mistos voz/sinal são modelos de sinais que fornecem representações compactas dos sinais de voz. São modelos amplamente utilizados para o processo de reconhecimento de voz, pois incorporam muitas interpretações relacionadas à fala, como os coeficientes Linear Predictive Coding, por exemplo, que podem ser associados aos modelos do trato vocal e pulso glotal em alguns casos (MOORE, 1979). No desenvolvimento deste trabalho, foram abordados os métodos referentes aos modelos baseados em voz e sinal utilizando o método Linear Predictive Coding e um modelo aproximado ao Cepstral Coefficients com o uso de Mel-Frequency Cepstral Coefficients.

Os sistemas de processamento de sinais de voz normalmente aplicados a sistema de reconhecimento biométrico compreendem fases de treinamento com amostras de dados reais. Nesta fase de treinamento, o sistema consome um grande número de amostras reais para que ele possa aprender e distinguir as características da população por meio do aprendizado de máquina. Uma forma alternativa de utilizar a base de dados é aplicar as informações nela contidas a testes de medição de desempenho de sistemas de reconhecimento, a fim de testar e comparar métodos distintos. Estas tarefas compreendem a abordagem e principal objetivo deste projeto, no qual, durante seu desenvolvimento foram executados e testados métodos extratores de características e métodos de aprendizado de máquina de diferentes universos, alguns comumente utilizados em projetos de reconhecimento sonoro (LPC e MFCC) e outros que são populares em diferentes aplicações (CDT) ou até mesmo pouco explorados como métodos computacionais (Aritmética Intervalar).

2. BASE DE DADOS

Inicialmente, os sinais usados nas etapas de processamento deste trabalho, foram coletados por meio do smartphone de participantes que se dispuseram a colaborar com a construção da base. A proposta era compor uma base inédita, com capturas de voz de trinta candidatos pronunciando trinta comandos em condições adversas de ambiente e com diversidade de dispositivos, para realizar um tratamento dos sinais ruidosos, aplicar os métodos extratores de características, realizar o treinamento, classificação para, por fim, analisar quais métodos se mostram mais eficientes mediante sinais consideravelmente ruidosos com base nos resultados obtidos.

Após a realização do pré-processamento e extração de características, a base construída foi descartada, pois apresentou relativa instabilidade e inadequação nas informações contidas em cada sinal de voz, mostrando-se insatisfatória. A imensa variação entre os sinais comprometeu a etapa classificatória, apresentando resultados desfavoráveis a métodos cuja garantia de alto desempenho é inquestionável, como, por exemplo, o LPC (COSTA, 2013). Desta forma, surgiu a necessidade de adquirir bases alternativas para que fosse possível garantir a realização de todos os testes e obter resultados confiáveis por meio dos métodos LPC, MFCC, CDT e Aritmética Intervalar.

A primeira base alternativa adquirida, é a base disponibilizada pelos pesquisadores do Google na biblioteca de código-aberto, TensorFlow(www.tensorflow.org/versions/master/tutorials/audio_recognition). A base é constituída por 64.721 arquivos no formato WAV, divididos entre trinta comandos distintos. A quantidade de arquivos de áudio varia de acordo com o comando; nem todos apresentam a mesma quantidade de arquivos, e os falantes repetem cada comando de uma a nove vezes.

Outra base utilizada neste trabalho foi adquirida através do “The CMU Audio Databases”, disponibilizado pelo grupo de estudos robustos de reconhecimento de fala de Carnegie Mellon University (www.speech.cs.cmu.edu/databases/). Esta, por sua vez, é composta por 4.180 amostra de sinais de áudio no formato WAV, as quais são distribuídas entre 16 comandos distintos. Os comandos são repetidos cinco vezes por cada falante, resultando numa mesma quantidade de arquivos de áudio para cada comando.

Para todas as bases de dados, os sinais são previamente processados a fim de eliminar intervalos de silêncio presentes no início ou no fim de cada elocução, evitando, assim, que os resultados das fases de processamento e testes sejam comprometidos.

3. TRATAMENTO DOS SINAIS DE VOZ E PRÉ-PROCESSAMENTO

É importante que os sinais sejam pré-processados e tratados antes de serem submetidos aos métodos extratores de características. Neste trabalho, a primeira operação nos sinais brutos é a remoção da faixa silenciosa normalmente presente no início e fim de cada elocução. Esses trechos silenciosos potencializam a quantidade de ruídos no sinal, o que prejudica a extração de características e classificação, acarretando em resultados insatisfatórios.

Outra função de suma importância é a conversão dos sinais em arquivo texto; os dados registrados na forma de arquivo texto facilitam os processamentos posteriores. Convertidos em arquivo texto, um script de contagem é disparado para obter o arquivo com menor número de informações a fim de realizar um truncamento nos dados do sinal para que possa ser processado pelo método de janelamento da Aritmética Intervalar, que será esclarecida mais adiante.

4. EXTRAÇÃO DE CARACTERÍSTICAS

Extratores de características são métodos computacionais que usam funções

matemáticas para transformar dados complexos que costumam ser relativamente volumosos, ocupando bastante espaço em disco. A transformação feita por extratores de características resulta em uma compactação dos dados, representando-os de uma maneira menos volumosa em termos de quantidade de informação e ainda assim preservando as principais características do dado original. Para realizar esta etapa, utilizamos os métodos: *Linear Predictive Coding* (LPC), *Mel-Frequency Cepstral Coefficients* (MFCC), *Cumulative Distribution Transform* (CDT) e a Aritmética Intervalar. Os métodos LPC e MFCC são métodos comumente aplicados no contexto de reconhecimento sonoro; o LPC é o método mais popular na realização da análise do sinal de voz e é muito utilizado nas etapas de reconhecimento de comandos, enquanto o MFCC faz a medição de similaridade entre os sinais, sendo muito utilizado na recuperação de informações e na distinção e classificação de gêneros musicais. Em contrapartida, o CDT e a Aritmética Intervalar são métodos pouco explorados no contexto de reconhecimento sonoro, mas que apresentam vantagens em relação à performance, simplicidade e robustez. O CDT é um método robusto que transforma dados linearmente separáveis lidando com as variações das intensidades do sinal, além disso, possui baixo custo computacional e garante que informações não serão perdidas. Já a Aritmética Intervalar é uma ferramenta poderosa, de simples entendimento e implementação que permite realizar análise intervalar por meio de conceitos estatísticos.

Ademais, foi adotada uma estratégia de combinação entre os métodos CDT e LPC, e CDT e MFCC com o intuito de potencializar a qualidade dos vetores de características obtidos por meio destes métodos, e também para comparar os resultados obtidos nas combinações com os resultados obtidos por cada método isolado.

4.1. Linear Predictive Coding

Linear Predictive Coding (LPC) é o principal método para extrair coeficientes de sinais de voz humana. O princípio deste método se baseia na análise de amostras anteriores contidas no sinal; a predição linear aponta que uma amostra pode ser predita (valor aproximado da amostra) por uma combinação linear dos valores das amostras anteriores, considerando a correlação entre elas (COSTA, 2013). A estimação de cada amostra se dá por meio de uma combinação linear de *p* amostras anteriores, na qual é possível afirmar que um valor de *p* maior fornece um modelo mais preciso. Como o método LPC se utiliza dos parâmetros presentes no sinal de voz (fala) que representam o trato vocal, qualquer mudança na anatomia do trato vocal compromete os coeficientes LPC obtidos. Os coeficientes LPC que compõem o vetor de características podem ser extraídos por meio da seguinte equação (1).

$$\tilde{s}(n) = \sum_{k=1}^{K} c_k s(n-k) \qquad (1)$$

Neste trabalho, foi utilizada a biblioteca *Audiolazy* para submeter as amostras ao LPC. *Audiolazy* consiste, basicamente, em um pacote para tratamento de áudio em *Python* e processamento de sinais digitais. Os segmentos do sinal de áudio de cada amostra foram extraídos por meio da biblioteca *Pydub*, através do *AudioSegment*; o resultado obtido foi processado pelo método LPC e os coeficientes resultantes desta operação foram armazenados em arquivos no formato texto e também convertidos em imagens.

4.2. Mel-Frequency Cepstral Coefficients

Os coeficientes resultantes do método MFCC se baseiam na escala *Mel*, cuja ideia principal é alterar o espectro do sinal de voz perante uma escala com características mais perceptíveis ao ouvido humano, que são muito melhores em discernir as pequenas mudanças entre as baixas e altas frequências (CUADROS et al. 2007). Desta forma, esta

escala colabora com a melhoria dos recursos perceptivos do âmbito computacional, aproximando os resultados atingidos aos resultados que podem ser obtidos naturalmente pela percepção humana e a possibilidade de traçar uma comparação entre a frequência real que é medida em *Hertz* (Hz) e a frequência percebida que é medida sob a escala *Mel* (mels) (MARANA; CHIACHIA; PAPA, 2014).

A conversão de uma frequência f para a escala *Mel* se dá através da equação (2).

$$M(f) = 1125 \ln(1 + \frac{f}{700}) \quad (2)$$

Neste trabalho, o método MFCC utilizado pertence à biblioteca *Librosa* do *Python*. As informações do sinal são carregadas e submetidas ao MFCC; em seguida, os coeficientes são escalados e retornados a uma lista. Por convenção, são extraídos 20 coeficientes, pois diversas documentações concluem que aumentar o número de coeficientes não melhoram o conjunto de coeficientes resultante; a melhora é gradual até a faixa dos vigésimos coeficientes.

4.3. Aritmética Intervalar

A Aritmética Intervalar é uma ferramenta numérica para manipulação e operação com intervalos. Definida por Moore, durante muito tempo, foi um dos principais modelos de computação numérica com intervalos (EINSENCRAFT, 2007).

A principal finalidade da Aritmética Intervalar é obter limitantes inferiores e superiores do contradomínio de funções reais, o qual torna importante os conceitos da propriedade de inclusão, variação de uma função e a extensão intervalar (LUXBURG; SCHOLKOPF, 2008). A faixa de valores reais é ilustrada em um intervalo *[a, b]*, onde *a* é tido como limite inferior e *b* é tido como limite superior; e um número real simples, *a*, pode ser representado por um intervalo [*a, a*], denotado como intervalo degenerado (EINSENCRAFT, 2007).

Neste trabalho, a Aritmética Intervalar foi aplicada em sua forma mais simplista, a fim de obter o limite inferior e superior de cada faixa do sinal de voz. Para realizar esta tarefa, foi necessário que os arquivos com os sinais de voz estivessem truncados para definir uma janela de tamanho adequado para que todos arquivos pudessem ser processados utilizando o mesmo tamanho de janela. Na fase de pré-processamento, a aplicação obtém qual arquivo de voz, dentre todos os arquivos da base, possui o menor número de dados. Esse valor mínimo é usado para truncar os demais arquivos, igualando todos eles em quantidade de informações. Obtendo um conjunto de arquivos truncados, o valor mínimo é utilizado para calcular o tamanho da janela, a fim de obter cerca de 20 coeficientes por arquivo: 20 coeficientes correspondentes aos limites inferiores dos intervalos, e 20 coeficientes correspondentes aos limites superiores dos intervalos; ou seja, na Aritmética Intervalar, cada sinal de voz é representado por um vetor de 40 características. Neste contexto, a janela definida representa um intervalo, do qual é extraído o valor mínimo e máximo - limite inferior e limite superior. Estes valores extraídos são obtidos como coeficientes e compõem o vetor de características resultante.

5. MÉTODOS DE APRENDIZADO DE MÁQUINA

O aprendizado de máquina é responsável pela classificação dos áudios; por meio de funções de captação, organização e processamento dos dados, é possível avaliar e interpretar os resultados, apoiando-se em padrões identificados no conjunto de dados. Para isso, são usados os vetores de características resultantes dos métodos executados na etapa de extração de características. Cada um dos métodos aplicados registra em um arquivo texto o vetor de características obtido para cada sinal de voz que foi processado. Ao fim desta etapa, cada conjunto de amostras possui uma vasta quantidade de arquivos com características; estes arquivos são agrupados

pela identificação de cada método extrator de características, contendo os vetores de cada sinal que pertence ao conjunto de amostras.

Neste contexto, são utilizados métodos supervisionados, pois os conjuntos para treino e teste são separados mediante uma gama de informações previamente fornecida aos métodos classificadores. Desta forma, todos os arquivos com vetores de características precisam estar reunidos em um único arquivo. Para isso, para cada vetor de características, formamos um par com a classe correspondente e os coeficientes do vetor; seguidamente, esse par é concatenado em um arquivo que resultará em um *dataset* de vetores de características para cada respectivo método extrator. Os *datasets* obtidos são utilizados pelos métodos da *scikit-learn* para realizar o treinamento e testes a fim de obter a taxa de classificação. No desenvolvimento deste projeto, utilizamos os métodos classificadores *Support Vector Machine* (SVM), *K-Nearest Neighbors* (KNN), *Naive Bayes*, *Optimun-Path Forest* (OPF) e CNN.

6. CLASSIFICADORES

Os métodos classificadores SVM, KNN e Naive Bayes aplicados neste trabalho pertencem à biblioteca *scikit-learn*, uma biblioteca de código-aberto dedicada ao aprendizado de máquina para o *Pyhton* que interage com as bibliotecas numéricas *NumPy* e *SciPy*. A *scikit-learn* possui uma extensa documentação e uma vasta quantidade de algoritmos para classificação, regressão e agrupamento, dentre outros. Já o OPF foi executado por meio da biblioteca de funções *LibOPF*.

6.1. Support Vector Machine

A Máquina de Vetores de Suporte (do inglês *Support Vector Machine - SVM*) mapeia o espaço de entrada para um espaço de maior dimensão a fim de encontrar a maior margem para separar as diferentes classes. A maior margem calculada é definida como hiperplano de separação ótimo, pois a distância é maximizada entre as classes, permitindo afirmar que duas classes são linearmente separáveis se existe esse hiperplano ótimo entre as amostras de classes diferentes (ARAÚJO e SILVA, 2015). Aplicar este método proporciona resultados consideravelmente melhores que resultados obtidos por outros métodos de aprendizado de máquina, tornando-o ainda mais atrativo a estudantes e pesquisadores interessados pelo Aprendizado de Máquina.

O método SVM utiliza uma função de *kernel k(xi, xj)* para conseguir separar dados linearmente separáveis ou não-linearmente separáveis. Há três funções *kernel* principais que podem ser utilizadas no SVM: linear, função de base radial (RBF) e polinomial (ALBERTO; ALMEIDA, 2013).

Neste trabalho, a função *kernel* utilizada foi a função de base radial (RBF) pois se forem feitas escolhas corretas de seus parâmetros, os resultados obtidos podem se assemelhar com a função *kernel* polinomial; a função RBF é encontrada na *sklearn.svm* do pacote SVC, da biblioteca *scikit-learn*.

A função RBF é dada pela equação (3) (ALBERTO e ALMEIDA, 2013).

$$k(x_i, x_j) = \exp\left(-\frac{1}{2\gamma^2} \| x_i - x_j \|^2\right), \gamma > 0 \quad (3)$$

Em todas as funções *kernel* que podem ser aplicadas, os parâmetros *C* e γ devem ser definidos pelo usuário; contudo, existem técnicas para testar e definir o melhor valor para os parâmetros. Para definir os parâmetros foram testadas as sequências exponenciais $C=2^{-5}, 2^{-4}, 2^{-3} \ldots, 2^{15}$ e $\gamma=2^{-15}, 2^{-14}, 2^{-13} \ldots, 2^{3}$. Na RBF da *scikit*, o parâmetro de regularização *C* ficou definido com valor 100 e γ em 0.01 para processar todos os *datasets*.

6.2. K-Nearest Neighbors

O método *K-Nearest Neighbors* (KNN) é um dos métodos de aprendizado mais simples, de fácil entendimento, fácil

implementação e que pode obter resultados notáveis. KNN é um método que pode ser usado tanto para classificação, quanto para regressão, todavia, é amplamente usado na classificação de problemas industriais. Para qualquer técnica, três aspectos devem ser observados: facilidade de interpretação das saídas, tempo de cálculo e poder de predição. Fronte a isto, o KNN se destaca até mesmo de outros métodos robustos pelas vantagens de facilidade de entendimento do funcionamento do algoritmo e de sua velocidade de processamento.

A ideia principal é classificar um elemento de acordo com as classes dos *k* vizinhos mais próximos, onde *k* >= 1. Para isso, o algoritmo calcula a distância do elemento dado para cada elemento da base de treinamento e então ordena os elementos da base de treinamento do mais próximo ao de maior distância.

Neste trabalho, o pacote *neighbors* da biblioteca *scikit-learn* foi utilizado na aplicação do KNN. Para encontrar o melhor valor de *k* para o respectivo *dataset*, aplica-se a equação, onde *n* representa a quantidade de dados (4).

$$k = \frac{\sqrt{n}}{2} \tag{4}$$

6.3. Naive Bayes

Naive Bayes (NB) é um método muito aplicado na mineração de textos e classificação de documentos textuais, derivado da teoria da decisão *Bayesiana* (ALBERTO; ALMEIDA, 2013) que utiliza a determinação de probabilidade posterior para classificar um dado. O *Naive Bayes* é sempre citado como um método simples, rápido e que, em várias situações, apresenta uma considerável precisão nos resultados, mesmo usando uma pequena quantidade de dados no conjunto de teste para realizar a classificação. Outra principal característica deste algoritmo é a forma de tratar cada característica de forma independente. Por exemplo, se uma fruta é considerada maçã por ser vermelha, redonda e com um diâmetro de aproximadamente 10 cm, o algoritmo não classifica a fruta como maçã por todas estas associações, ele despreza a correlação entre os atributos que normalmente é utilizada em outros métodos para melhorar os resultados. Sem a associação de atributos, o *Naive Bayes* conta com a probabilidade total do teorema de *Bayes*, onde é possível afirmar que a probabilidade de uma mensagem $\vec{x} = \langle x_i, \dots, x_n \rangle$ pertencer a categoria $c_i \in \{ c_s, c_l \}$ de acordo com a equação (5) .

$$P(c_i|\vec{x}) = \frac{P(c_i).P(\vec{x}|c_i)}{P(\vec{x})} \tag{5}$$

O filtro NB classifica cada mensagem na categoria que maximiza $P(c_i).P(\vec{x}|c_i)$.

6.4. Optimum-Path Forest

O classificador *Optimum-Path Forest* (OPF) é um método supervisionado que se assemelha às SVMs, porém, em desempenho, demonstra-se muito mais rápido (LATHI, 2007).

Este método reduz o problema de classificação de padrões a um problema de particionamento em um grafo baseado no espaço de atributos dos dados. Inicialmente, o algoritmo faz a escolha de alguns elementos chaves que podem ser chamados de sementes. Essas sementes iniciam um processo de conquista no grafo, oferecendo caminhos de custo ótimo para as demais amostras. No grafo, cada caminho está associado a uma função de custo. Esta função de custo considera a distância entre os objetos e seus vetores de características, considerando que todos os objetos ao longo do caminho pertencem à mesma classe. O caminho de valores ótimos parte do conjunto de sementes no grafo. Essas sementes competem entre si, sendo que cada uma delas define uma árvore de caminhos ótimos, de tal maneira que a união desses caminhos ótimos forme uma floresta orientada, estendendo-se pelos objetos no espaço de atributos (MUNIZ, 2009).

A etapa de treinamento consiste em construir uma floresta de caminhos ótimos onde os objetos em uma dada OPF não

possui o mesmo rótulo que a semente (raiz) da árvore na qual este objeto pertence.

Entre as vantagens do OPF podemos citar que ele não depende de ajustes de parâmetros, demonstra ser mais rápido que classificadores baseados em SVM, é naturalmente multi-classes e permite verificar a eficácia de um mesmo conjunto de recursos em diferentes espaços de distâncias.

Neste trabalho, para implementar o classificador OPF, utilizou-se a *LibOPF* que é uma biblioteca dedicada ao desenvolvimento de classificadores baseados em floresta de caminhos ótimos.

7. METODOLOGIA

Para o desenvolvimento das etapas e obtenção de *datasets* e resultados, foram construídos *scripts* para cada fase fundamental do processamento. O primeiro *script* a ser executado é o de pré-processamento, onde todos os sinais de áudio são submetidos a um tratamento; realizando o recorte da faixa silenciosa presente no áudio e adicionando as informações contidas no sinal a um arquivo texto. Esta etapa foi realizada com o auxílio das bibliotecas *AudioSegment* e *Audiolazy* do *Python*, para as funções de recorte do trecho silencioso, conversão para arquivo texto e obtenção da quantidade de dados contida no menor sinal de áudio para posteriores truncamentos.

O próximo *script* executado é o dos extratores de características. Para cada método extrator, executa-se um *script* que obtém todos os áudios, submetendo-os ao programa em *Python* de cada respectivo método extrator de características.

Neste trabalho, para aplicar o LPC, foi utilizado o método LPC implementado pela biblioteca *Audiolazy*. Por meio dele, o áudio bruto é submetido ao método, que por sua vez retorna uma lista contendo coeficientes e rótulos. Nesta aplicação, o conjunto resultante é composto por 20 coeficientes, os quais são armazenados em um arquivo texto.

Para extrair os coeficientes por meio do MFCC foi utilizada a biblioteca *librosa* do *Python*. *Librosa* é uma biblioteca dedicada à análise, criação e recuperação de arquivos de áudio. Com ela, o áudio bruto é submetido ao método como parâmetro junto ao número de coeficientes a serem extraídos. O método, por sua vez, retorna uma estrutura com diversas informações da qual é possível extrair os coeficientes por meio do método "*mean*". Neste trabalho, foi adotado o padrão de extrair 20 coeficientes, visto que a literatura aponta que um número superior a este não oferece melhorias aos resultados.

Para extrair características usando a Aritmética Intervalar, foi necessário realizar o truncamento dos arquivos de áudio para que fosse possível realizar o "janelamento" e definir os coeficientes para cada intervalo. Foi adotada uma janela de tamanho 20 para obter os valores mínimos e máximos de cada intervalo. O áudio truncado é dividido em 20 "gavetas" que armazenam o mínimo e máximo de cada intervalo, em seguida, estes valores são armazenados em arquivos texto para compor o vetor de características. Desta forma, ao fim da etapa, para cada sinal processado, o método da Aritmética Intervalar retorna um vetor com 40 elementos, representando os picos mínimos e máximos do sinal de voz.

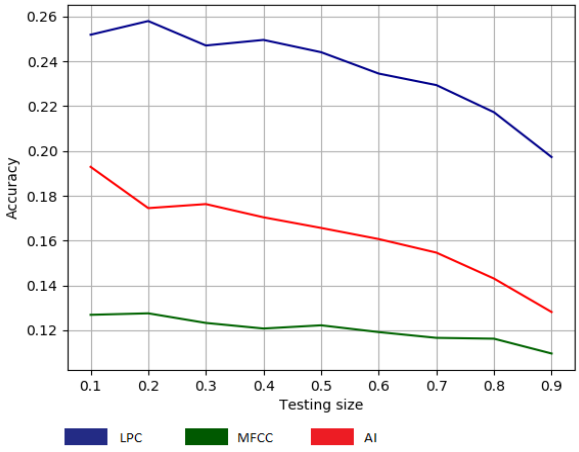
Dentre todos os métodos citados, o CDT é o que mais diferencia-se em sua implementação, pois não há um número reduzido de coeficientes como saída de seu processamento, a quantidade de dados contidos na entrada é a mesma para os dados que compõem a saída. Diferente dos demais, ele realiza o processamento do sinal baseando-se nos dados contidos no arquivo texto, e não no áudio bruto. Ele analisa as principais variações contidas em cada intervalo do sinal e armazena os resultados dos cálculos em um arquivo texto. A principal vantagem desta transformação é que não ocorre a perda de informações, pois é possível realizar a operação inversa e obter os valores reais.

Após processar cada arquivo de áudio na etapa de extração de características, todos os arquivos nos quais estão armazenados os coeficientes, são concatenados de acordo com cada método, a fim de compor um *dataset*. Estes *datasets* são utilizados na fase de treinamento e testes. Nesta etapa, a biblioteca *scikit* foi a principal responsável na colaboração da realização dos treinos e testes. Por meio dela, foi possível separar em cada *dataset* um conjunto de treino e testes, além executar os métodos SVM, KNN e *Naive Bayes* para realizar a classificação. Neste trabalho, foi desenvolvido um *script* para realizar testes com conjuntos de tamanhos variados. O tamanho dos conjuntos de treino e testes variam de 0.1 a 0.9. Para cada execução de cada método são armazenados em arquivos texto as respectivas matrizes de confusão e o valor da acurácia. Em seguida, as informações contidas nestes arquivos são submetidas à análises e geração de gráficos.

8. RESULTADOS E DISCUSSÕES

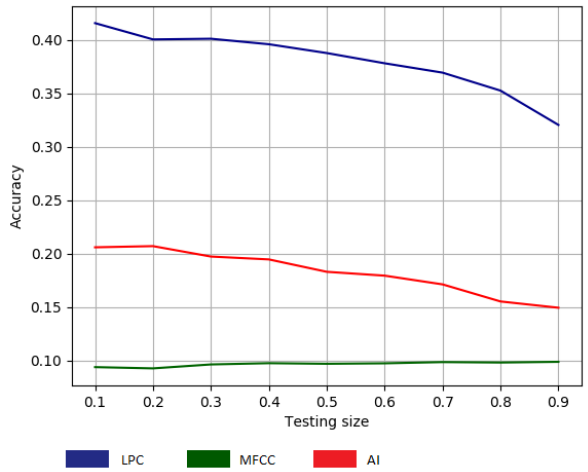
Diversos casos de testes foram elaborados e realizados com os comandos presentes na base de dados disponibilizada pelo Google. Casos como testes com comandos numéricos, testes com comandos não numéricos, comandos cuja pronúncia é semelhante e teste com toda a base de dados. Um caso de teste em particular é composto por 10 comandos selecionados: *"yes"*, *"no"*, *"up"*, *"down"*, *"left"*, *"right"*, *"on"*, *"off"*, *"stop"* e *"go"*. A seleção destes comandos foi feita de acordo com uma lista de comandos referentes a um exemplo de uso da base com *Tensorflow* disponível na plataforma. A lista apresentada era composta por 12 comandos, contendo "*silence"* e "*unknow"*, contudo, estes dois comandos não estão presentes no pacote disponibilizado, portanto, foi realizado um teste com os dez comandos restantes. Com 12 comandos e utilizando *Tensorflow*, é possível obter uma acurárica de 26.3% (www.tensorflow.org/versions/master/tutorials/audio_recognition), entretanto, como é possível observar na Fig.2, utilizando SVM o método LPC obteve um aumento na acurácia, chegando a 42% (0.42). Neste contexto, toda metodologia previamente descrita foi aplicada a este caso de teste a fim de comparar os resultados e a performance dos métodos utilizados neste trabalho. De modo geral, foi possível observar que a Aritmética Intervalar (AI), manteve sua taxa de acertos superior à taxa de acertos do MFCC, como é possível observar nas figuras Figura 1 e Figura 2 e observar os melhores resultados listados na Tabela 1.

Figura 1. Acurácia em função da taxa de teste referente ao método de classificação KNN retratando o LPC, AI e MFCC; onde a taxa de acertos da AI se manteve superior ao tradicional método MFCC.



Fonte: (Autores, 2017).

Figura 2. Acurácia em função da taxa de teste referente ao método de classificação SVM retratando o LPC, a AI e MFCC; no qual a taxa de acertos atingida pelo LPC se mantém superior a 26.3%.



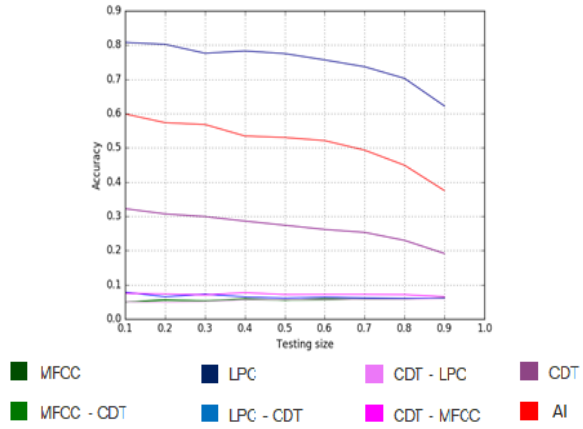
Fonte: (Autores, 2017).

Tabela 1. Taxa de acertos por método extrator e classificador

Método	SVM	KNN	Bayes
LPC	0.42 teste = 0.1	0.26 teste = 0.2	0.15 teste = 0.6
MFCC	0.10 teste = 0.9	0.12 teste = 0.2	0.14 teste = 0.2
AI	0.21 teste = 0.2	0.19 teste = 0.1	0.11 teste = 0.6

No caso de teste realizado sob a base de dados construída em um ambiente controlado, o MFCC não conseguiu apresentar melhores resultados mantendo-se na faixa de 0.1 da taxa de acertos, sendo ultrapassado pela AI e pelo CDT, como é possível observar na Figura 3.

Figura 3. Acurácia em função da taxa de teste referente ao método de classificação SVM retratando o LPC, AI e MFCC; onde a taxa de acertos da AI e do CDT mantiveram-se superior ao tradicional método MFCC.



Fonte: (Autores, 2017).

9. CONCLUSÕES

Este trabalho apresentou a proposta de utilizar a AI e o CDT como métodos a serem considerados nas etapas do processamento digital de sinais, seja por sua robustez (CDT) ou significativa eficácia e simplicidade (AI). Também foram realizadas análises da taxa de acertos dos métodos AI, CDT, LPC e MFCC combinados aos classificadores SVM, KNN, *Naive Bayes* e OPF, todos aplicados em bases de dados distintas, de ambiente controlado e ambiente não controlado.

Com as bases obtidas e os resultados dos testes realizados, é possível reforçar a viabilidade do uso de tais métodos no contexto de reconhecimento sonoro. Contudo, ainda se faz necessária a análise e aprofundamento de resultados referentes à performance e poder de processamento requisitado por cada método extrator de característica combinado aos métodos de classificação utilizados.

REFERÊNCIAS

ALBERTO, T.C; ALMEIDA, T.A. **Aprendizado de Máquina Aplicado na Detecção Automática de Comentários Indesejados**. Departamento de Computação (DComp). Campinas: Unicamp, 2013. Disponivel em:

http://www.dt.fee.unicamp.br/~tiago/papers/ENIAC13.pdf.

ARAÚJO, K. M.; SILVA, E. M. Utilização do Algoritmo de Máquina de Vetores de Suporte (SVM) para Predição de Dados Climáticos. *In*: CONGRESSO DE COMPUTAÇÃO E SISTEMAS. 18., , 2015. p. 149

BLANZ, V. et al.. Comparision of view-based object recognition algorithms using realistic 3d models. 1996. *In*: C. VON DER MALSBURG, W.; VON SEELEN, J.C.; VORBRUGGEN, and B. Sendhoff, (ed)., **Artificial Neural Networks – ICANN**, , Berlin, 1996.Sringer Lecture Notes in Computer Science, v. 1112. pages 251 – 256https://doi.org/10.1007/3-540-61510-5_45

BRESOLIN, A.A. **Estudo do reconhecimento de Voz para Acionamento de Equipamentos Elétricos via Comandos em Português**. 2003. Tese (Doutorado) – Universidade do Estado de Santa Catarina – UDESC, Centro de Ciências Tecnológicas – CCT. Joinville, 2003.

CAMPOS, W.; MACIEL, A.; CARVALHO, E. Investigação de uma Arquitetura para Verificação e Reconhecimento de Locutor. In: SIMPÓSIO BRASILEIRO DE SISTEMAS MULTIMÍDIA E WEB, 2008, Vitória, ES. **Anais** [...]. Vitória., 2008, p. 17-20.

CARDOSO, S.A.; CASTANHO, J.E.C.; FRANCHIN, M.N.; FONTES, I.R. SESAME: Sistema de Reconhecimento de Comandos de Voz Utilizando PDS e RNA. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA, 18. 2010, Bonito, MS. Anais [...] Bonito – MS, 2010.

CORTES, S.; VAPNIK, V. **"Support Vector Machines"**. 1995. Machine Learning, 20:273-297. https://doi.org/10.1007/BF00994018

CUADROS, C. D.; CATALDO, E., DA SILVA, D. G., ALCAIM, A., & APOLINÁRIO Jr, J. A. (2007**). "Comparação entre as técnicas de MFCC e ZCPA para reconhecimento robusto de locutor em ambientes ruidosos"**. Rio de Janeiro, RJ, 2007.

CUSTÓDIO, R.F. **Análise Não-Linear no Reconhecimento de Padrões Sonoros: Estudo de Caso para Sons Pulmonares.** 1999. Tese (Doutorado) – Universidade Federal do Rio Grande do Sul, Instituto de informática. Porto Alegre, 1999.

COSTA, W. C. A. et al. Classificação de sinais de vozes saudáveis e patológicas por meio da combinação entre medidas da análise dinâmica não linear e codificação preditiva linear. **Revista Brasileira de Engenharia Biomédica**, v. 29, n. 1, 2013. https://doi.org/10.4322/rbeb.2013.010

DE LA VEGA, A. S. **"Apostila de Teoria para Processamento Digital de Sinais"**. 2016. 290. Apostila de Teoria – Graduação, Engenharia de Telecomunicações, UFF/TCE/TET 2016.

LIMA, C. A. M. **Comitê de Máquinas: uma abordagem unificada empregando máquinas de vetores-suporte**. 2004. 342. Tese (Doutorado) - Universidade Estadual de Campinas. Campinas, 2004.

DIAS, M. F. R.; PASCUTTI, P. G.; DA SILVA, M. L. Aprendizado de Máquina e Suas Aplicações em Bioinformática. **Semioses**, v. 10, n. 1, p. 23-37, 2016. https://doi.org/10.15202/10.15202/1981-996X.2016v10n1p23

DOS SANTOS, C. N. **"Aprendizado de máquina na identificação de sintagmas nominais: o caso do português brasileiro"**. 2005. Tese (Doutorado) - Instituto Militar de Engenharia. Rio de Janeiro, 2005.

EINSENCRAFT, M**. Processamento Digital de Sinais** São Paulo: Universidade Presbiteriana Mackenzie, 2007.

FIGUEIREDO, L.H. **Raycasting intervalar de superficies implícitas cp, aritmética afim**. 1999. Dissertação (Mestrado) – Pontifícia

universidade Católica do Rio de Janeiro, Departamento de Informática. Rio de Janeiro, RJ, 1999.

HAYKIN, S.; VAN VEEN, B., **"Sinais e Sistemas"**. Porto Alegre, RS: Bookman, 2001.

HAYKN, S**. Redes Neurais:** princípios e prática. Porto Alegre: Bookman, 2001. 900 p.

HÖLBIG, C. A.; PAVAN, W.; VENDRUSCULO, T.; CLAUDIO, D. M. Análise de Ferramentas Intervalares para Computação Gráfica. In: SIMPÓSIO DE INFORMÁTICA DO PLANALTO MÉDIO, 2., 2000, Passo Fundo. **Anais** [...]. Passo Fundo: UPF, 2000.

KOLOURI, S; PARK, S. R; ROHDE, G. K. The Radon cumulative distribution transform and its application to image classification. **IEEE transactions on image processing**, v. 25, n. 2, p. 920-934. 2016. https://doi.org/10.1109/TIP.2015.2509419

LATHI, B. P. **Sinais e Sistemas Lineares**. 2.ed. Porto Alegre, RS: Bookman, 2007.

LORENA, A.C.; DE CARVALHO, A.C.P.L.F. Uma introdução às support vector machines. **Revista de Informática Teórica e Aplicada**, v. 14, n. 2, p. 43-67, 2007.

LUXBURG, U.; SCHOLKOPF, B. **"Statical Learning Theory: Models, Concepts, and Results"**. 2008. E-print arXiv preprint arXiv:0810.4752.

MACHADO, A. F. **Conversão de Voz Inter-Linguística**. 2013. Tese (Doutorado) -. Instituto de Matemática e Estatística - Universidade de São Paulo. São Paulo, 2013.

MARANA, A.N; CHIACHIA, G; PAPA, J.P. **Análise de Desempenho de Classificadores baseados em Redes Neurais, Máquinas de Vetores de Suporte e Floresta de Caminhos Ótimos Para o Reconhecimento de Dígitos Manuscritos**. Bauru: UNESP, Faculdade de Ciências, Departamento de Computação, 2014.

MELO, D.B. **Um Sistema de Reconhecimento de Comandos de Voz Utilizando a Rede Neural ELM**. 2011. Monografia (Graduação) - Departamento de Engenharia de Teleinformática, Centro de Tecnologia, Universidade Federal do Ceará. Fortaleza, 2011.

MONTANHER, T.M. **"Estimação de Modelos Markov Ocultos usando Aritmética intervalar"**. 2015. Tese (Doutorado) – Instituto de Matemática e Estatística, Universidade de São Paulo. São Paulo, 2015.

MOORE, R. E. **"Methods and Applications of Inteerval Analysis"**. SIAM, Phifadelphia, 1979. https://doi.org/10.1137/1.9781611970906

MUNIZ, D. N., **"Estudo Sobre Reconhecimento de Áudio Repetitivo: Desenvolvimento de um Protótipo"**, 2009, São José, Instituto Federal de Santa Catarina. 2009.

NUNES, R.A.A; ALBUQUERQUE, M.P.; ALBUQUERQUE, M.P.; SEIXAS, J.M. **"Introdução a Processadores de Sinais Digitais - DSP"**. CBPF-NT-001/06. Fev., 2006.

PARK, S. R; KOLOURI, S; KUNDU, S; ROHDE, G. K. The cumulative distribution transform and linear pattern classification**. Applied and Computational Harmonic Analysis**, v. 45, n. 3, p. 616-64, Nov. 2018. . https://doi.org/10.1016/j.acha.2017.02.002

PETRY, A; ZANUZ A.; BARONE, D.A.C. Utilização de Técnicas de Processamento Digital de Sinais para a Identificação Automática pela Voz. In: SIMPÓSIO SOBRE SEGURANÇA EM INFORMÁTICA. 1999. **Anais** [...]. São José dos Campos, SP, 1999.

PICON, C.T; ROSSI, I; Jr. P.P.M. **Análise da classificação de imagens por descritores de**

cor utilizando várias resoluções. São Carlos, SP.: Instituto de Ciências Matemáticas e de Computação (ICMC). Universidade de São Paulo (USP), [2011].

SABANAL, S.; NAKAGAWA, M. The Fractal Properties of Vocal Sounds and Their Application in the Speech Recognition Model. **Chaos, Solitons & Fractals**, [S.l.], v.7, n.11, p. 1825-1843, 1996. https://doi.org/10.1016/S0960-0779(96)00043-4

SANTANA, F.T.; DÓRIA NETO A.D.; SANTIAGO, R.H.N.. 2012. Sinais de Sistemas Definidos sobre Aritmética Intervalar Complexa. **TEMA Tend. Mat. Apl. Comput.,** v. 13, n. 1, p. 85-86, 2012. https://doi.org/10.5540/tema.2012.013.01.0085

SANTANA, F. T.; DÓRIA NETO, A. D.; SANTIAGO, R. H. N. Uma Proposta de Análise Intervalar para o Mapa Auto-Organizável de Kohonen. Learning and Nonlinear Models, **Revista da Sociedade Brasileira de Redes Neurais (SBRN),** v. 7, n. 1, p. 03-08, 2009. https://doi.org/10.21528/LNLM-vol7-no1-art1

SANTANA, F. T.; SANTIAGO, R. H. N., DÓRIA NETO, A.D**.Fundamentação Intervalar Complexa para Sinais e Sistemas"**. In: CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL, 33., 2010, Águas de Lindóia. Anais [...]. 2009, São Carlos. Anais [...]. . São Carlos: SBMAC, 2009,p. 1-7.

SILVA DE NOVAIS, D. P. **Um Modelo Intervalar para Reconhecimento de Fala por Computadores**. 2012. . Dissertação (Mestrado) – Universidade Estadual do Sudoeste da Bahia. Vitória da Conquista, BA, 2012.

SOARES, F.A. **"Aprendizado de Máquina"**. 2008. 18. LES/PUC-Rio.

SOUZA, B. F. S.; TEIXEIRA, A. S.; SILVA, de A.T.F. **Classificação de bioma caatinga usando Support Vector Machines (SVM)"**. Anais XIV Simpósio Brasileiro de Sensoriamento Remoto, p. 7917-7924. Natal/RN, Abril de 2009.

SUNAGA, T. Theory of na Interval Algebra and its Applications to Numerical Analysis**"**. **RAAG Memoirs** v. 2 (1958), 29-46.

TOZADORE, D. C. **Aplicação de um robô humanoide autônomo por meio de reconhecimento de imagem e voz em sessões pedagógicas interativas**. 2016. Tese (Doutorado) - Universidade de São Paulo. São Carlos, 2016.

TRINDADE, R.M.P. **Uma Fundamentação Matemática para Processamento Digital de Sinais Intervalares**. 2009. Tese - Universidade Federal do Rio Grande do Norte – Centro de Tecnologia. Natal, RN, 2009. https://doi.org/10.5540/tema.2009.010.01.0087

TRINDADE, R.M.P; BEDREGAL, B.R.C; NETO, A.D.D. Princípios de Processamento Digital de Sinais Intervalares. In: CNMAC, 31., 2008, Belém, PA. **Anais** [...]. Belém, 2008. . https://doi.org/10.5540/tema.2009.010.01.0087

ZAMPIERI, C.E.A. **Recuperação de Imagens Multiescala Intervalar**. 2010. Dissertação (Mestrado) – Universidade Estadual de Campinas, Instituto de Computação. Campinas, 2010.