Tito Caco Curimbaba Spadini

Comparação de Técnicas para Detecção de Eventos Sonoros para Sistemas de Segurança

Santo André, SP, Brasil

2017

Tito Caco Curimbaba Spadini

Comparação de Técnicas para Detecção de Eventos Sonoros para Sistemas de Segurança

Monografia apresentada ao curso de Engenharia de Informação da Universidade Federal do ABC como parte dos requisitos para obtenção do grau de Engenheiro em Eletrônica.

Universidade Federal do ABC – UFABC

Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas — CECS

Graduação em Engenharia de Informação

Orientador: Ricardo Suyama

Santo André, SP, Brasil

2017

Universidade Federal do ABC

ATA DE DEFESA DE TRABALHO DE GRADUAÇÃO EM ENGENHARIA DE INFORMAÇÃO

Ata de Defesa do Trabalho de Graduação em Engenharia de Informação da Universidade Federal do ABC

No dia **23 de novembro de 2017** reuniu-se a banca examinadora do trabalho apresentado como Trabalho de Graduação em Engenharia de Informação de **Tito Caco Curimbaba Spadini**, intitulado: **"Comparação de Técnicas para Deteção de Eventos Sonoros para Sistemas de Segurança"**. Após a exposição oral, o aluno foi arguido pelos componentes da banca que se reuniram reservadamente, e decidiram atribuir o conceito final A.

Orientador
Prof. Dr. Ricardo Suyama

Avaliador 1
Prof. Dr. Filipe Ieda Fazanaro

Avaliador 2
Prof. Dr. Kenji Nose Filho

Em memória daqueles que contribuíram para que este trabalho fosse desenvolvido, mas que não puderam continuar conosco para testemunhar sua conclusão.

Agradecimentos

Os agradecimentos principais são direcionados a Ricardo Suyama, Kenji Nose Filho, Filipe Ieda Fazanaro, Mário Minami, Murilo Bellezoni Loiola, André Kazuo Takahata, Marco Aurélio Cazarotto Gomes, Marcelo Grave, Ademir Ferreira da Silva, Pedro Ivo da Cruz, Jeferson Rodrigues Cotrim, Estefânia Angelico Pianoski Arata, Guilherme Tavares, Godofredo Quispe Mamani, Valério Cardoso e Igor Martins Genuino, por, direta ou indiretamente, terem contribuído para que a realização deste trabalho fosse possível.

Agradecimentos especiais são direcionados a Suely Benedita Curimbaba Spadini, América Benedita de Paiva Curimbaba, Bruno Teilor, Felipe Luccas Miranda Pinto, Lucas Leandro da Rocha, Rafael Pereira Pinto de Carvalho, Pedro Henrique Melo Franco Viviani e Rodolfo Kohei Soken.

Resumo

O avanço em técnicas baseadas em *Machine Learning* provê diversas possibilidades para a criação de um sistema capaz de detectar e classificar, automaticamente, sons captados em um dado ambiente. Uma das aplicações de interesse dessas técnicas seria em sistemas de segurança pública, nos quais o monitoramento de determinada área poderia ser realizado por meio de microfones, detectando automaticamente eventos sonoros associados a situações de perigo — como colisão de veículos, disparo por arma de fogo, grito de multidões etc —. O monitoramento baseado no som pode levar a soluções de baixo custo e alta confiabilidade. Por esse motivo, o objetivo do presente trabalho consiste em analisar e comparar técnicas de detecção e classificação de eventos sonoros, tendo em vista sua possível implementação em um sistema embarcado de baixo custo, com limitado poder de processamento.

Neste trabalho foram explorados especificamente cenários de explosões, disparos por armas de fogo e alarmes/sirenes, que foram classificados utilizando LDA, QDA, *Decision Tree* e KNN. O classificador KNN mostrou uma precisão de mais de 90% e demandou, em média, menos de 10 ms para efetuar cada classificação, o que sugere que o KNN é um classificador capaz de atender aos objetivos deste trabalho.

Palavras-chave: aprendizado de máquinas, reconhecimento de padrões, processamento de sinais, eventos sonoros, segurança.

Abstract

The advancement in techniques based on Machine Learning provides several possibilities for the creation of a system capable of automatically detecting and classifying sounds that were captured in a given environment. One of the applications of interest in these techniques would be public surveillance systems, in which monitoring of a certain area could be performed by microphones, detecting sound events associated with dangerous situations, such as vehicle collisions, gunshots, crowd shouts etc. The audio based surveillance can lead to low cost and high confiability solutions. For this reason, the goal of the present work is to analyze and compare sound detection and classification techniques, considering its possible implementation in a low cost embedded system with limited processing power.

In this work were explored scenarios of explosions, gunshots and alarms, which were classified using LDA, QDA, *Decision Tree* and KNN. The KNN classifier showed an accuracy of more than 90% and demanded, on average, less than 10ms to perform each classification, which suggests that the KNN is a classifier capable of meeting the objectives of this work.

Key-words: machine learning, pattern recognition, signal processing, sound events, surveillance.

Lista de abreviaturas e siglas

CART	Classification and Regression Trees
FPGA	Field Programmable Gate Array
GDI	Gini's Diversity Index
GFS	Greedy Forward Selection
GMM	Gaussian Mixture Models
HMM	Hidden Markov Model
KNN	k-Nearest Neighbors
LDA	Linear Discriminant Analysis
LED	Light Emitting Diode
MFCC	Mel-Frequency Cepstral Coefficients
PCA	Principal Component Analysis
QDA	Quadratic Discriminant Analysis
RMS	SFM Spectral Flatness Measure
SVM	Support Vector Machine
ZCR	Zero-Crossing Rate

Sumário

1 Introdução

1.1 Problemática

Uma parcela considerável dos sistemas de segurança pública depende exclusivamente da ação de pessoas em diversas etapas de seu funcionamento. O monitoramento de áreas públicas, por exemplo, costuma ser feito com o uso de câmeras de vídeo, geralmente dependentes de um operador (humano) que as direcione ou, no mínimo, que fique atento às telas onde são exibidas as imagens.

Considerando a quantidade de informação à qual o operador tem acesso — em geral, a central de controle conta com dezenas de monitores, apresentando uma grande quantidade de imagens das diversas localidades vigiadas — a tarefa de realizar um monitoramento minucioso de todas as áreas cobertas pelo sistema de vigilância é bastante árdua. Nesse contexto, observa-se uma grande possibilidade de falhas — por exemplo, eventos importantes podem deixar de ser notados pelo operador sem que as devidas medidas sejam tomadas — e, tratando-se de segurança pública, isso significa que a população está mais propensa a correr riscos desnecessários.

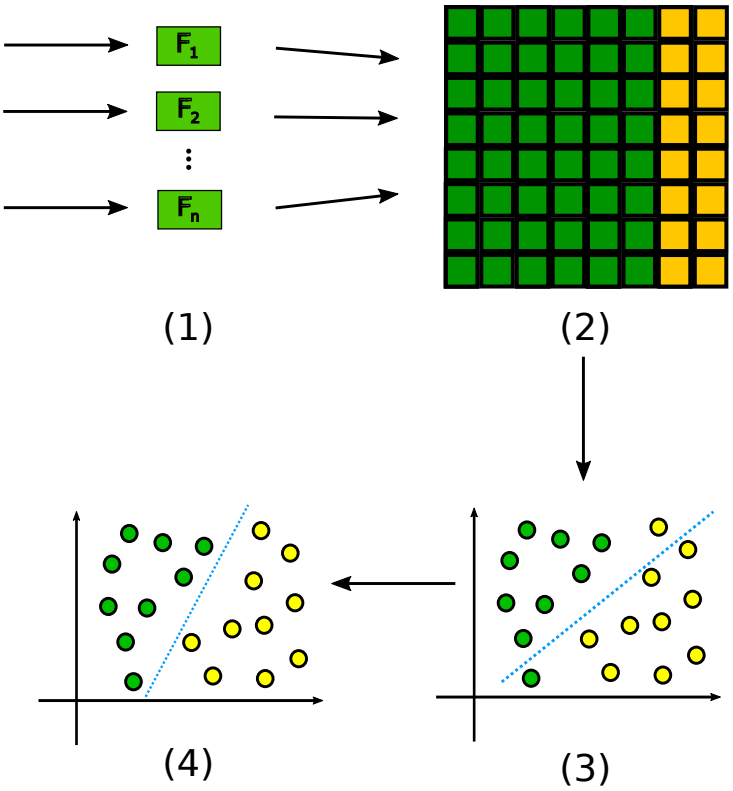
1.2 Visão Geral da Proposta

É possível encontrar diferentes propostas na literatura para realizar a detecção e classificação desses eventos sonoros que exploram métodos comumente encontrados na área de reconhecimento de padrões. Reconhecimento de disparos por armas de fogo e gritos de pessoas (VALENZISE et al., 2007), detecção acústica de situações atípicas sob diferentes níveis de ruído (NTALAMPIRAS; POTAMITIS; FAKOTAKIS, 2009b), detecção de eventos sonoros em *Health Care* à distância (MONTALVãO et al., 2010) e detecção de sons anormais com microfones de vigilância (ITO et al., 2009) são exemplos que exploram o *Gaussian Mixture Model* (GMM) (DAY, 1969) para modelagem do sinal observado.

Outro método de classificação é o *Hidden Markov Model* (HMM), que, por exemplo, foi utilizado para vigilância acústica de situações perigosas (NTALAMPIRAS; POTAMITIS; FAKOTAKIS, 2009a), rastreamento de eventos sonoros móveis (RODà; MICHELONI, 2011), detecção de eventos anormais baseados em acústica (SASOU et al., 2011), reconhecimento do tipo de arma com base em gravações de disparos por armas de fogo (KIKTOVA et al., 2015) e aumento de robustez de um sistema de reconhecimento automático de sons com adaptação a ruídos de fundo em situações reais (RABAOUI; LACHIRI; ELLOUZE, 2006).

De modo geral, sistemas de classificação não operam diretamente sobre os sinais, mas realizam a classificação baseada em características extraídas de dados colhidos. Essas características, chamadas de *Features*, podem exigir algum tipo de tratamento matemático para serem obtidas, tais como estatísticas ou o espectro do sinal.

Figura 1 – Modelo esquemático de um sistema de classificação genérico. Em (1) ocorre a extração das *Features*; em (2), a elaboração e separação da base de dados; em (3), o cálculo de uma função que separa os grupos a serem classificados; e em (4), o ajuste da função.



Como mostra a Figura 1, geralmente a primeira etapa da classificação é representada pela extração das *Features*. Após extraídas, as *Features* devem ser apropriadamente organizadas, formando uma base de dados que será dividida entre uma base para treinamento do classificador e uma base para testes. Existem diferentes possibilidade para o classificador, dentre as quais podem ser citados os classificadores lineares (baseados, por exemplo, na análise discriminante linear - LDA), classificadores quadráticos (ex: análise discriminante quadrática - QDA) e técnicas mais robustas, como as máquinas de vetores suporte (SVM) (DUDA; HART; STORK, 2000).

Ao longo deste trabalho serão explorados os métodos LDA, QDA, *Decision Tree* e KNN, que serão melhor explicados no Capítulo 2, e serão utilizados para classificar os eventos sonoros de potencial risco á segurança.

1.3 Justificativa

No presente projeto foi considerado o monitoramento das áreas por meio do som. A opção por esse tipo de monitoramento é baseada em duas características favoráveis: primeiramente, o som consome menor largura de banda na transmissão de informação, reduzindo a necessidade de altas taxas de transmissão, como no caso de imagens de alta definição; além disso, as técnicas de processamento de som exigem, em geral, menor poder de processamento do que as de vídeo, o que possibilitaria a implementação em sistemas embarcados mais simples, e.g. *Raspberry Pi* (RÖCKER et al., 2017), e, consequentemente, de menor custo, ou mesmo em um FPGA (WOODS et al., 2017).

Uma forma de minimizar tais erros seria diminuir a necessidade da participação humana no processo de vigilância. Para isso, é possível fazer uso de uma solução que chame a atenção do operador toda vez que alguma informação importante seja coletada na área monitorada. Por exemplo, uma vez que o sistema reconheça que ocorreu algum evento associado a um potencial risco à segurança, a atenção do vigilante é requerida de forma mais notória, com o disparo de um alarme ou simplesmente pelo acionamento de um LED.

O tipo desejado de solução será projetado a partir de técnicas de *Machine Learning* que serão utilizadas para o reconhecimento de padrões sonoros. Com o uso de um algoritmo apropriado, após coletar o som da área monitorada, é possível identificar eventos como disparos por armas de fogo, alarmes sonoros, explosões, acidentes automobilísticos, vidro sendo quebrado etc. e, assim, alertar o operador de que algo de errado ocorreu. Isso reduzirá as chances de eventos de risco não serem detectados, o que conferirá maior confiabilidade ao sistema de vigilância adotado.

Todo o desenvolvimento deste projeto foi realizado utilizando uma plataforma computacional que será detalhada na seção a seguir.

1.4 Plataforma Computacional

Considerando que este trabalho aborda especificamente aspectos de desempenho, é importante que seja declarada a plataforma computacional utilizada para realizar todas as simulações. A Tabela 1 lista todos os principais componentes do computador utilizado.

Tabela 1 – Configuração da plataforma computacional utilizada.

Componente	**Descrição**
Processador	Intel Core i7 7700 3.6 GHz 8 MB
Memória RAM	2 x Kingston HyperX 8GB DDR4 2133 MHz
SSD	Kingston UV400 240 GB SATA III
HD	Seagate Barracuda 1 TB 7200 RPM SATA III
Placa de Vídeo	NVidia GeForce GTX 1060 6GB
Sistema Operacional	Linux Mint 18 (Cinnamon)

Ressalta-se o fato de que o sistema operacional e os programas utilizados estavam todos instalados no SSD.

Como os objetivos do trabalho e a plataforma em que será realizado foram expostos, pode ser melhor detalhada a metodologia, que será abordada no Capítulo 2, onde será feita uma explicação sobre as técnicas e as *Features* utilizadas no trabalho.

2 Metodologia

2.1 O que é reconhecimento de padrões

Na natureza é possível encontrar diversos tipos de dados que podem ser coletados por uma série de instrumentos, no entanto, podem não oferecer informação suficiente se considerados isoladamente. Para que seja possível obter maiores informações sobre o problema trabalhado, é necessário analisar mais elementos, que são conhecidos como *Features*.

2.2 *Features*

Features são características que podem permitir diferenciar os objetos estudados. Podem ser dados mensuráveis (diretos), como distância, ou dados calculáveis (indiretos), como a densidade espectral de potência. Para auxiliar na explicação, será utilizado um exemplo mais casual primeiro.

Se fosse de interesse classificar frutas, talvez fosse razoável dizer que a cor, a altura e a massa do objeto são *Features* aceitáveis. Por outro lado, para classificar modelos de carros, essas mesmas *Features* já poderiam não ser mais tão interessantes. Isso ocorre porque, no caso das frutas, simplesmente por saber a cor, já seria possível excluir uma considerável variedade de frutas; sabendo sua altura, outra parcela bastante considerável poderia ser excluída; e, por fim, sabendo a massa, seria consideravelmente alta a probabilidade de acerto de qual fruta se trate. Porém, no caso do modelo de veículo, isso não ocorreria da mesma forma, pois saber a cor não faria diferença, visto que qualquer veículo poderia ser pintado de qualquer cor; a altura não ajudaria muito também, pois ajudaria mais para encontrar a categoria de carros, mas não há tanta variação entre os modelos de uma mesma categoria; e, quanto à massa, apesar de haver variação, ela nem sempre seria grande para diferentes modelos de uma mesma categoria. Assim, este é um exemplo de que as *Features* razoáveis para uma determinada tarefa não necessariamente são boas para outras aplicações.

Nesse contexto, percebe-se que quando as combinações de possíveis valores das *Features* selecionadas probabilisticamente não contribuem para que seja possível separar com um bom nível de confiabilidade os objetos a serem classificados, as chances de ocorrerem erros de classificação (*Misclassifications*) são elevadas. Uma forma de consertar isso é utilizando *Features* que sejam mais adequadas e que permitam constatar discrepâncias maiores entre seus valores.

Uma boa escolha de *Features* pode influenciar drasticamente os resultados finais da classificação desejada e impacta também no tempo de processamento, assim como impactaria na complexidade da decisão se fossem seres humanos os responsáveis pela análise e classificação.

Por exemplo, suponha que o objetivo seja identificar qual a melhor compra de imóvel a ser feita. Imagine que se tenha as seguintes *Features* para avaliar casas: dormitórios, suítes, banheiros, área, tempo, preço, distância até o trabalho, vagas no estacionamento.

Tabela 2 – *Features* de casas a serem analisadas com o propósito da compra de uma delas em um exemplo hipotético para ilustrar uma boa seleção de *Features*.

	Dorm. [#]	Suítes [#]	Banh. [#]	Área [m^2]	Tempo [min]	Preço [$]	Dist. [km]	Estac. [#]
Casa 1	2	1	1	60	4	350k	10	2
Casa 2	2	1	3	115	14	750k	3	2
Casa 3	2	1	2	163	31	900k	5	2
Casa 4	1	1	2	58	1	570k	22	2
Casa 5	2	1	1	100	7	850k	1	2

Analisando a Tabela 2, pode-se dizer que, por apresentarem pouca ou nenhuma variação, o número de dormitórios, o número de suítes e o número de vagas no estacionamento são pouco relevantes para a tarefa de classificação.

Após reduzir o número de variáveis, ocorre também uma redução do grau de complexidade do problema analisado. Confira na Tabela 3.

Tabela 3 – *Features* remanescentes da Tabela 2 após eliminação dos menos influentes na decisão.

	Banh. [#]	Área [m^2]	Tempo [min]	Preço [$]	Dist. [km]
Casa 1	1	60	4	350k	10
Casa 2	3	115	14	750k	3
Casa 3	2	163	31	900k	5
Casa 4	2	58	1	570k	22
Casa 5	1	100	7	850k	1

Com isso, fica-se com um número menor de *Features* a serem consideradas, o que torna a complexidade da escolha um tanto menor. O mesmo ocorre com o uso de algoritmos de classificação. Quanto menor o número de *Features* a serem consideradas na classificação e menor for a complexidade de cada uma dessas *Features*, menor será o tempo demandado para que a resposta seja encontrada.

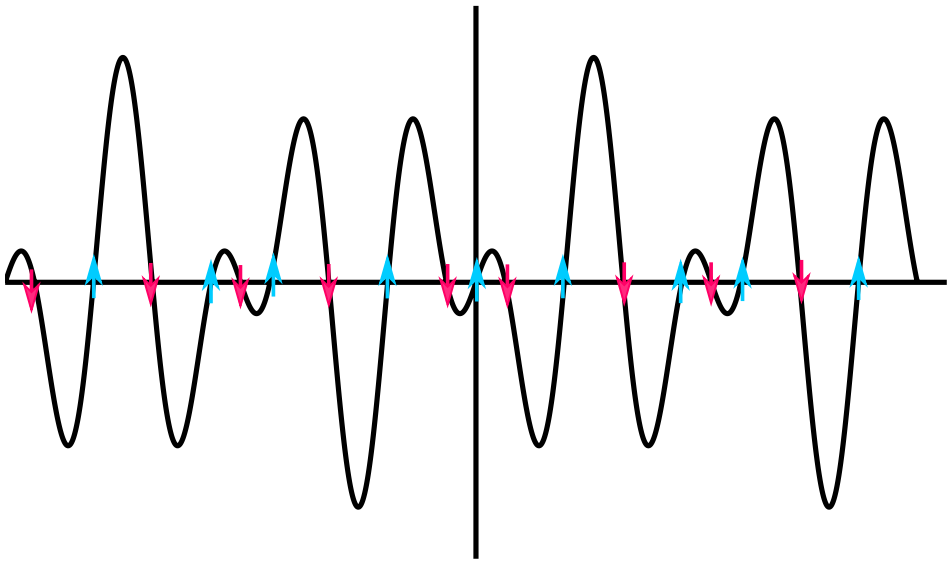
2.3 Extração de *Features*

No caso particular estudado neste trabalho, as *Features* extraídas a partir da base de áudios (explicada mais à frente) devem permitir a correta classificação dos eventos sonoros, e.g., sons de sirenes, disparos de arma de fogo etc. Nesse contexto, diferentes métricas já definidas na literatura podem ser utilizadas. Nesta seção são descritas as *Features* consideradas no presente trabalho.

2.3.1 *Zero-Crossing Rate* (ZCR)

Uma das *Features* mais comuns é a *Zero-Crossing Rate* (ZCR) (BACHU et al., 2008), que consiste na taxa de cruzamentos do eixo das abiscissas, também podendo ser interpretada como uma taxa de mudança de sinal (positivo para negativo ou vice-versa). Assim, podemos calcular o ZCR dividindo o número de vezes que ocorre mudança de sinal pelo número total de amostras.

Figura 2 – Exemplo de identificação de cruzamentos do eixo para o cálculo da ZCR.



A Figura 2 permite que seja melhor compreendida a ZCR. Pode-se perceber que, nesse caso, foram contabilizadas 18 cruzamentos.

Considerando a função $sinal[x(n)]$ como sendo apenas um verificador sobre o valor ser positivo ou não positivo, para valores positivos a saída da função será sempre igual a 1; para não positivos, sempre igual a zero. Fazendo-se uma simples subtração, é possível identificar quando ocorre a variação, mas deve-se utilizar o módulo dessa diferença para que não haja valores positivos e negativos sendo somados, assim sempre a verificação retornará valores 0 ou 1. Efetuando-se a soma de todos esses valores, obtém-se o número de vezes que o ocorre o cruzamento do eixo x ($y = 0$). Então, a fim de se obter a taxa desses cruzamentos, precisa-se dividir o total de cruzamentos pelo número total de amostras (N).

Assim, obtem-se:

$$ZCR = \frac{1}{N}\sum_{i=0}^{N-1} |sinal\,[x(i)] - sinal\,[x(i-1)]|, \tag{2.1}$$

mas como, na maior parte dos casos, os sistemas acabam tendo segmentos de um mesmo tamanho fixo, onde os valores de ZCR são comparados entre os segmentos, essa divisão acaba sendo desnecessária, então é possível efetuar apenas a contagem de cruzamentos do eixo sem precisar dividir pelo número de transições entre amostras (CHEN, 1988), mantendo, porém, a relevância desta *Feature*.

O valor da ZCR está relacionado à frequência do sinal analisado e, dessa forma, provê informação importante a respeito da natureza do som gravado.

2.3.2 Energia

Outra *Feature* bastante utilizada é a potência média do sinal pode ser calculada a partir da raiz quadrada da média quadrática da amplitude, também conhecida como energia RMS:

$$E_{rms} = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1} x_i^2}, \tag{2.2}$$

onde x é o sinal e N é o número de amostras do sinal.

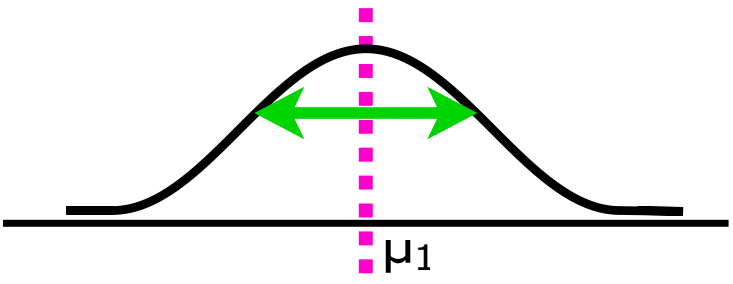
2.3.3 *Spread*

O desvio padrão das amostras é o segundo momento central, que também é interpretado como a variação (*Spread*) do sinal, pode ser encontrado a partir da raiz quadrada da variância, que é calculada por

$$\sigma^2 = \mu_2 = \int (x - \mu_1)^2 f(x)\; dx, \tag{2.3}$$

em que x é a amostra, μ_1 é a média do sinal e $f(x)$ é a função que descreve o sinal. A Figura 3 mostra bem o comportamento dessa *Feature*.

Figura 3 – Representação esquemática do comportamento referente à *Spread* (LARTILLOT; EEROLA, 2014).

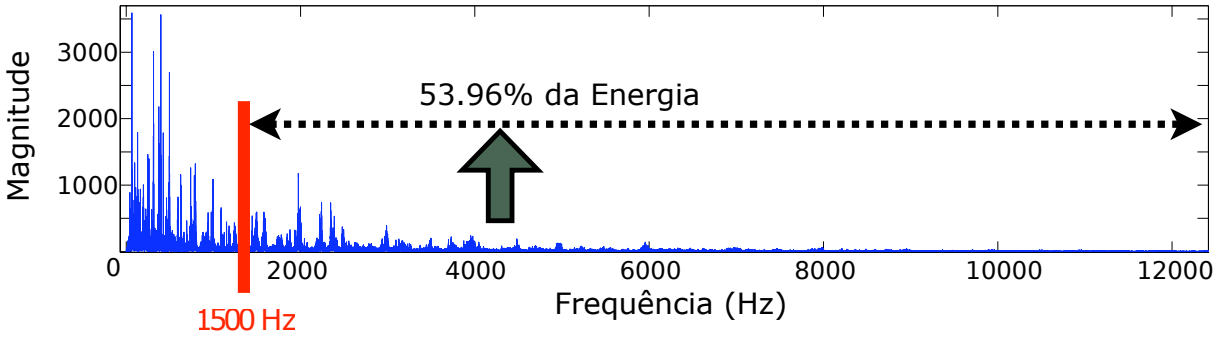


2.3.4 *Brightness*

O *Brightness* se refere à concentração de energia na região das altas frequências (relativas), visto que o que determina a faixa correspondente às altas frequências é a frequência de corte, que foi definida arbitrariamente. Essa Feature costuma ser expressada com valores entre zero e um.

Pela Figura 4, podemos compreender melhor o *Brightness* do áudio.

Figura 4 – Exemplo de identificação da frequência de corte e reconhecimento da razão de distribuição da energia nas frequências mais altas para o cálculo do *Brightness* (LARTILLOT; EEROLA, 2014).



2.3.5 *Skewness*

Uma *Feature* que analisa a simetria da distribuição do sinal é a *Skewness* (MARDIA, 1970), que também é compreendida como o terceiro momento central e é calculada fazendo-se

$$\mu_3 = \int (x - \mu_1)^3 f(x)\ dx, \tag{2.4}$$

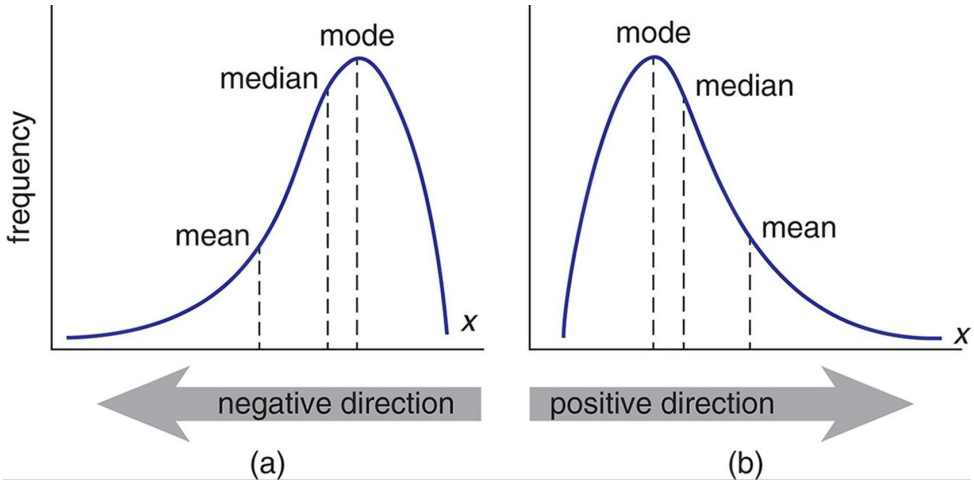
e, então, calculando-se a taxa entre μ_3 e o cubo do desvio padrão, ou seja

$$\frac{\mu_3}{\sigma^3}.$$

Note que valores negativos desse coeficiente indicam um aspecto mais concentrado à esquerda da média e que decai suavemente à direita da média; e, para valores positivos desse coeficiente, o aspecto se mostra suavemente crescente à esquerda da média e decai bruscamente após a média, finalmente, quando o coeficiente é nulo, isso indica uma distribuição simétrica dos dados.

Por uma questão de conveniência, essa *Feature* costuma utilizar um intervalo que vai de -3 a +3. É possível observar o comportamento da *Skewness* a partir da Figura 5.

Figura 5 – Representação esquemática do comportamento referente à *Skewness* (SIMPLIFIED, ; LARTILLOT; EEROLA, 2014).

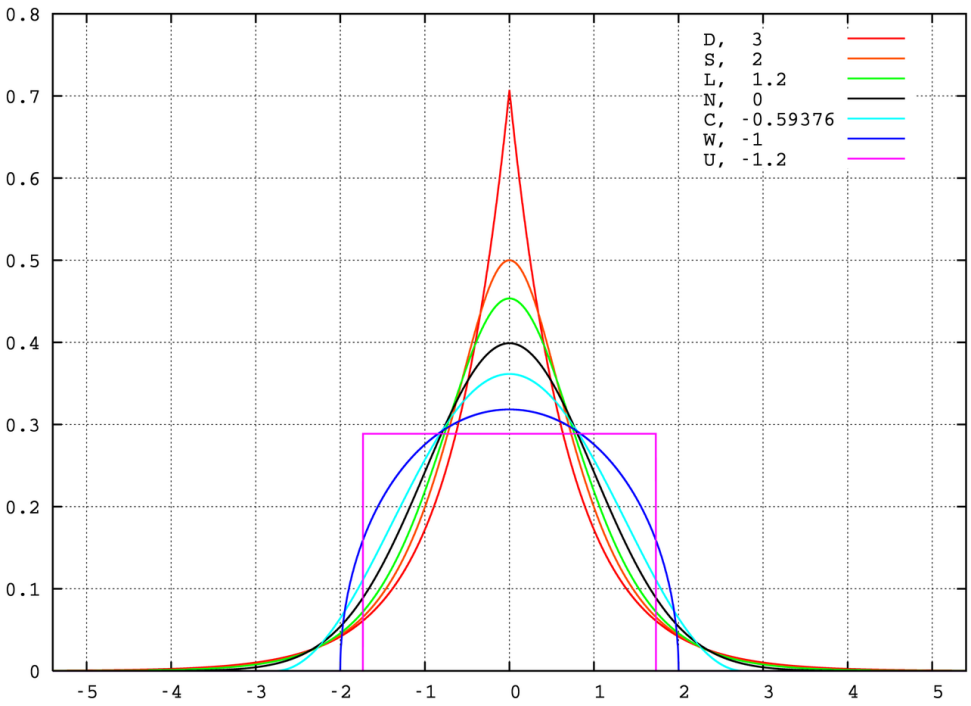


2.3.6 *Kurtosis*

A *Kurtosis* (MARDIA, 1970) mede a dispersão que caracteriza o pico da curva da função de densidade de probabilidade. Pode-se, também, definir essa *Feature* matematicamente como o quarto momento central dividido pelo quadrado do segundo momento central (CASELLA; BERGER, 2010). Assim:

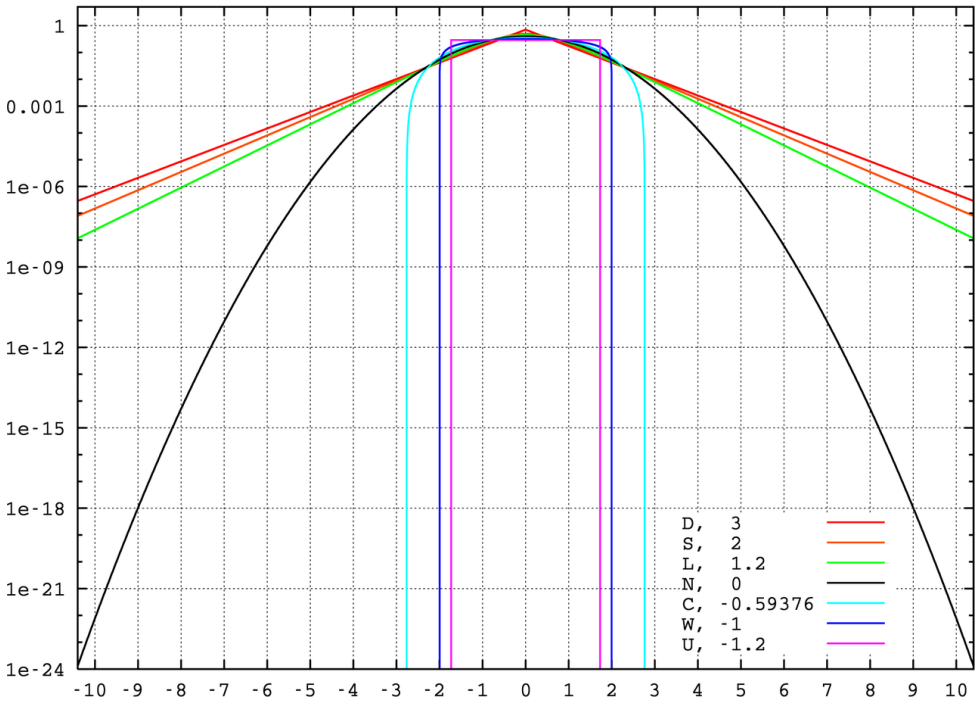
$$\frac{\mu_4}{\mu_2^2} = \frac{E\left(X - E\left(X\right)\right)^4}{\left[E\left(X - E\left(X\right)\right)^2\right]^2} \tag{2.5}$$

Figura 6 – Representação da *Kurtosis* (LARTILLOT; EEROLA, 2014) para múltiplas distribuições distintas (MARKSWEEP, 2006b).



O comportamento resultante da *Kurtosis* pode ser melhor compreendido pelas Figuras 6 e 7 (em escala logarítmica). Observando a Figura 6, nota-se um comportamento de distribuição Gaussiana (normal) para o caso em que a *Kurtosis* é nula. Para valores positivos mais altos, o comportamento observado é leptocúrtico, assemelhando-se mais a uma distribuição Laplaciana. E, para valores negativos, o comportamento se aproxima mais de uma distribuição uniforme.

Figura 7 – Múltiplas representações, em escala logarítmica, de *Kurtosis* para diferentes distribuições (MARKSWEEP, 2006a).



2.3.7 Entropia

A Entropia relativa de Shannon, definida por

$$H(p) = -\sum_{i=0}^{N-1} p\left(x_i\right) log_2\left(x_i\right), \tag{2.6}$$

oferece uma descrição geral da curva de entrada $\mathbf{p}$ e indica em que regiões particulares há picos predominantes. Por exemplo, quando tem-se uma curva extremamente plana, isso indica uma elevada incerteza quanto aos valores de saída da variável aleatória $\mathbf{X}$, cuja função de massa de probabilidade é $\mathbf{p}(\mathbf{x_i})$, então pode-se dizer que a entropia é máxima.

2.3.8 *Spectral Flatness Measure* (SFM)

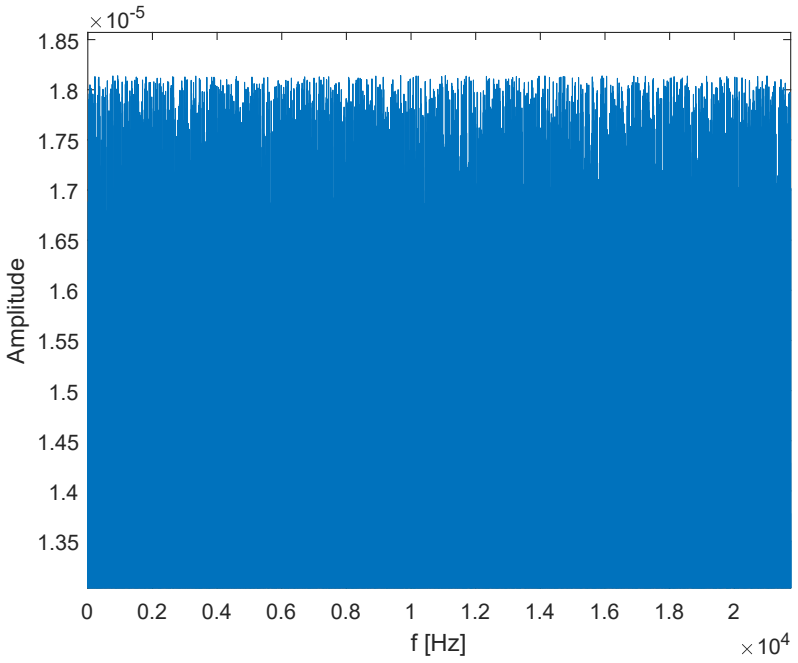
A *Spectral Flatness Measure* (SFM) (JOHNSTON, 1988; DUBNOV, 2004) é também conhecida por outros nomes, tais como Coeficiente de Tonalidade e Entropia de Wierner (SATAPATHY et al., 2016). Trata-se de uma medida utilizada para caracterizar um

espectro de áudio, permitindo que seja quantificado o quão similar a ruídos o som é. O cálculo da SFM pode ser feito por

$$SFM = \frac{\sqrt[N]{\prod_{i=0}^{N-1} x(i)}}{\left(\frac{\sum_{i=0}^{N-1} x(i)}{N} \right)} \tag{2.7}$$

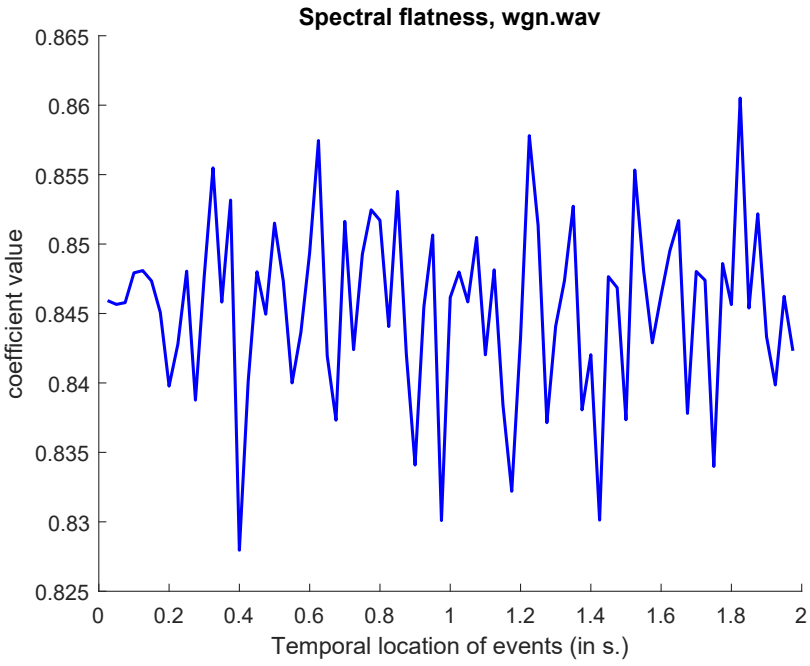
É interessante observar que, matematicamente, trata-se simplesmente da divisão da média geométrica pela média aritmética.

Figura 8 – Espectro de um sinal altamente ruidoso (branco).



Nas Figuras 8 e 9 é possível observer, respectivamente, o espectro de um ruído branco e o *plot* e sua SFM.

Figura 9 – Gráfico de exemplo de SFM para um sinal altamente ruidoso (com SFM próximo de 1).



2.3.9 *Mel-Frequency Cepstral Coefficients* (MFCC)

A convolução de dois sinais no domínio do tempo equivale à multiplicação das transformadas de Fourier de cada um desses sinais. *A priori*, não é possível distinguir qual parte da resultante equivale a cada um dos sinais convoluídos. No entanto, ao se aplicar a função logaritmica, será obtida uma soma (sobreposição), então, ao aplicar-se a transformada inversa de Fourier, obter-se-á o *Cepstrum* (PETRY; ZANUZ; BARONE, 1999).

Se, antes mesmo de se aplicar a função logarítmica, aplicar-se ao espectro real do sinal um banco de filtros que busca mimetizar a percepção humana de altura e intensidade do som, obteremos ao final os coeficientes mel-cepstrais (MFCC, do inglês *Mel-Frequency Cepstral Coefficients*). A conversão das frequências para a escala Mel pode ser feita pela Equação (2.8)

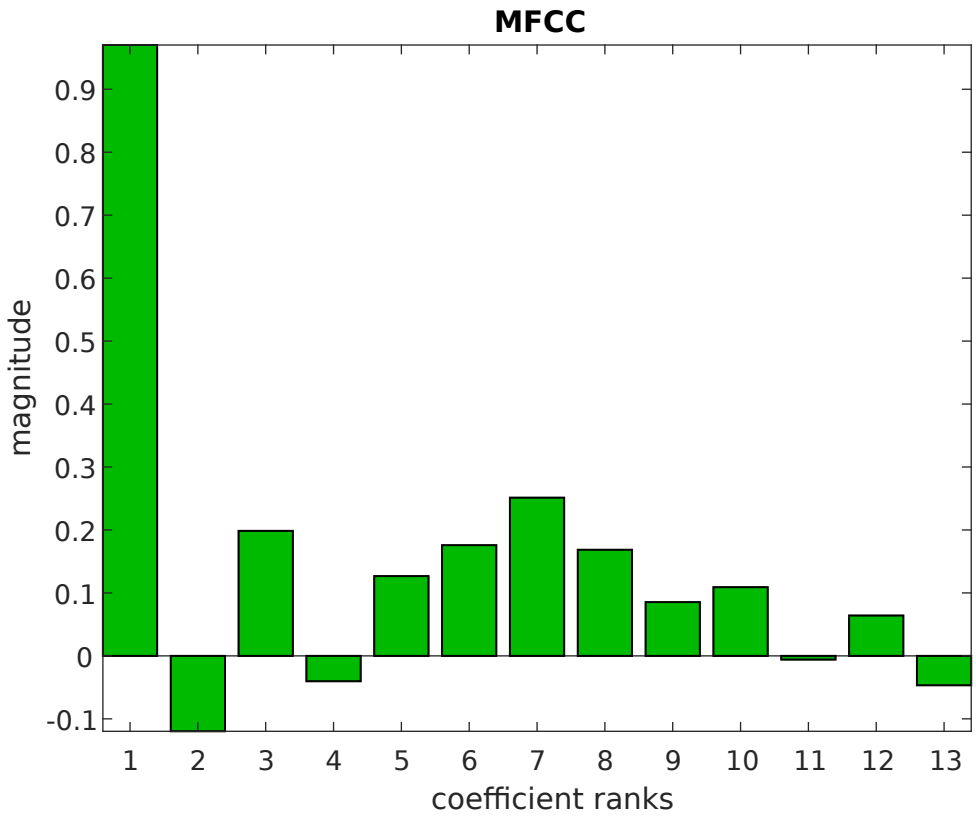
$$Mel = 2595 \cdot log\left(1 + \frac{f}{700}\right), \tag{2.8}$$

e o cálculo dos coeficientes mel-cepstrais (MFCC) pode ser feito pela Equação (2.9)

$$c(n) = \sum_{i=0}^{N-1} log|S\,(i)| \cdot \cos\left[n\left(i - \frac{1}{2}\right)\frac{\pi}{N}\right], \tag{2.9}$$

com $0 \leq n < P$, sendo que $c(n)$ é o n-ésimo coeficiente, N é o número de filtros, $S(i)$ é o sinal de saída do banco de filtros e P é o número de coeficientes.

Figura 10 – Exemplo do MFCC de um disparo por arma de fogo.



Agora, que todas as *Features* abordadas neste trabalho foram devidamente expostas e explicadas, é possível passar à etapa seguinte, que se refere à sua seleção.

2.4 Seleção de *Features*

Ter as *Features* não basta para que seja projetado um bom classificador. Quanto maior o número de *Features*, maior tende a ser o grau de complexidade com o qual o problema precisará ser trabalhado para que uma solução seja encontrada.

Computacionalmente, é muito custoso trabalhar com funções que dependam de muitas variáveis para serem calculadas. Assim, uma das mais importantes etapas a serem cumpridas é a de redução da dimensão da função, ou seja, redução do custo computacional do problema.

Mas a etapa de seleção das *Features* não visa somente a questão do custo computacional. Sem uma boa seleção, é possível que o classificador não atinja um desempenho tão bom quanto poderia.

Utilizar um maior número de *Features* não necessariamente resultará em um melhor classificador. A partir do momento que uma determinada *Feature* apresenta valores extremamente próximos para objetos de diferentes categorias de classificação, por mais

numerosas que sejam as bases de dados utilizadas na etapa de treinamento do classificador, pior tenderá a ser a acurácia e a robustez que ele apresenta.

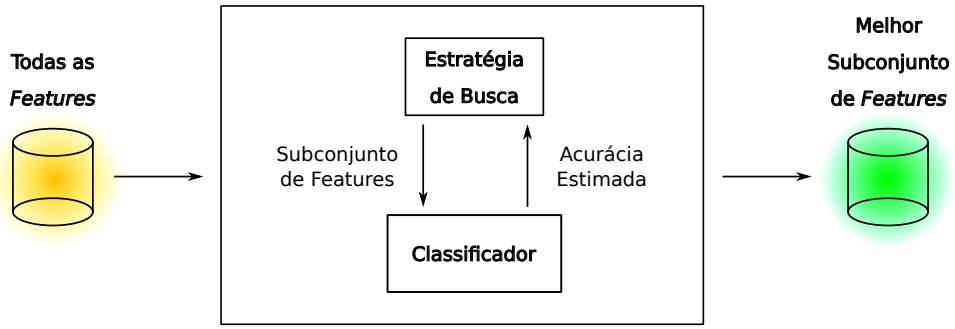
Felizmente, existem diversas técnicas para que tais problemas sejam contornados, minimizados ou até mesmo solucionados. Um dos métodos existentes para efetuar uma melhor seleção das *Features* a serem trabalhadas é conhecido como ***Wrapper*** (RAZA; QAMAR, 2017; SHERWOOD; DERAKHSHANI, 2011). O *Wrapper* é uma das formas que mais sofre no quesito desempenho (no sentido de tempo demandado para concluir sua tarefa), mas destaca-se na questão da eficácia observada em sua seleção final das *Features*.

Esse método consiste em utilizar o que pode ser conhecido como "força bruta", pois, de modo geral, trata-se de uma longa sequência de testes de desempenho do classificador, passando por subconjuntos (candidatos) elaborados a partir do grande conjunto de *Features* disponíveis. Ou seja, o método *Wrapper* testa as diferentes combinações de *Features* e encontra uma que seja a que proporciona os melhores resultados dentro dos casos comparados.

Vale destacar o fato de o *Wrapper* ser um método, não um algoritmo[1]. Existem diversos algoritmos que foram desenvolvidos seguindo o método *Wrapper*, não necessariamente havendo um que seja "o melhor", mas sim vários que, dependendo do que se tem como objetivo, podem ser considerados muito bons.

A Figura 11 ilustra o processo esquemático de um método *Wrapper* em sua busca pela melhor combinação de *Features*.

Figura 11 – Processo esquemático e generalista de um *Wrapper*.



É importante observar que o *Wrapper* não encontrará um conjunto de *Features* que seja necessariamente o melhor para todos os classificadores ao mesmo tempo. O conjunto de *Features* que ele encontra, na verdade, é o melhor candidato de todos os que foram calculados, mas apenas para o classificador utilizado em seu próprio processo de seleção. O classificador utilizado no processo do *Wrapper* influencia diretamente nos resultados

[1] Neste contexto, considera-se que um método é algo mais amplo e genérico, podendo ser interpretado meramente como uma ideia do que será feito. Dependendo do caso, pode-se até mesmo considerar "vago". Um algoritmo, por sua vez, trata-se de um conjunto bem ordenado de passos bem definidos para se atingir um dado objetivo.

que o processo do *Wrapper* poderá apresentar no final. Isso faz com que esse seja um método altamente dependente da boa funcionalidade do classificador e trata-se de um método que faz com que o resultado da saída do *Wrapper* não possa ser considerado como necessariamente correto para outros classificadores, impondo a obrigatoriedade de o *Wrapper* ser utilizado uma vez para cada classificador.

A abordagem de *Wrapper* utilizada é a *Greedy Forward Selection* — GFS —, que consiste em analisar as possibilidades a cada iteração e sempre optar pela que oferece o maior ganho sobre o elemento que se almeja otimizar (SUN; YAO, 2006).

Com a melhor combinação de *Features* escolhida para cada classificador, foram feitas as análises comparativas entre os diferentes classificadores para comparar questões como custo computacional, tempo de classificação, acurácia, robustez e, futuramente, complexidade de implementação.

2.5 Ruído

O ruído envolvido nos sinais de áudio é um fator de elevada influência no desempenho do classificador e, portanto, não pode ser desconsiderado. Além de outras formas de ruído, as duas principais fontes de ruídos dependem do equipamento de gravação e dos demais sons presentes no ambiente de gravação.

Para tentar considerar esses ruídos na metodologia utilizada, os áudios coletados utilizaram uma ampla variedade de equipamentos e ambientes (situações) de gravação diferentes, havendo, portanto, diversas combinações resultantes de ruído de equipamento e ruído ambiente. Essa variedade de combinações, considerando que se trata de uma base de dados consideravelmente grande, faz com que os classificadores passem a ser capazes de se ajustar, de modo a considerar diferentes situações reais em que esses sistemas trabalhariam e a evitar efeitos de *Overfitting* (i.e. um ajuste muito "rígido" e específico, que não permite uma generalização flexível), atingindo um grau de generalização confiável (GUYON; YAO, 1999; AALST et al., 2010).

2.6 Classificadores

O classificador é um algoritmo que cumprirá com um processo para identificar categorias a que pertencem as amostras do estudo (PEREIRA; MITCHELL; BOTVINICK, 2009). Existem diversas abordagens diferentes para que isso seja feito.

Os tipos de classificação existentes são: supervisionada, não supervisionada e semi supervisionada. Alguns classificadores permitem múltiplas abordagens. Alguns classificadores são bastante populares, como *Decision Trees*, KNN (ALTMAN, 1992) e SVM (CORTES; VAPNIK, 1995). O *k-means* (HARTIGAN, 1975) utiliza uma abordagem de

clusterização. Há também abordagens que podem envolver redução de dimensionalidade, como ocorre com o PCA (ABDI; WILLIAMS, 2010) e o LDA (MCLACHLAN, 2004).

A próxima etapa é, então, a da seleção de *Features*, visto que 21 *Features* é considerado um número muito grande, lembrando que equivaleria a dizer que teríamos uma função com 21 *inputs* que seriam computados por uma mesma função. Para isso, foi utilizado um método *Wrapper* de variação para encontrar algumas das melhores possíveis combinações de *Features*, considerando 4 tipos de classificadores: LDA, QDA, *Tree* e KNN.

2.6.1 *Linear Discriminant Analysis* (LDA)

O classificador LDA (FISHER, 1936) trabalha com o uso de misturas de distribuições normais multivariáveis. Tratando-se de um caso linear, todas as classes possuem a mesma matriz de covariância, havendo variações exclusivamente na média. Com isso, é calculada a média amostral de cada classe e, então, a covariância amostral, subtraindo a média amostral de cada classe a partir de observações da classe e adotando a matriz de covariância empírica do resultado.

A ideia desse classificador é a de atribuir às novas amostras uma classe, que será atribuída com base nas *Features* apresentadas pela amostra. Pode-se imaginar que cada *Feature*, i.e. x, corresponde a um valor de entrada, i.e. cada entrada, e que y corresponde à classe, i.e. a saída. Pode-se ter múltiplas entradas diferentes, sendo ao menos uma para cada *Feature*, então pode-se dizer que a entrada corresponderá a um vetor de *Features*, portanto, será representado por $\mathbf{x}$, que é definido conforme a Equação 2.10.

$$\mathbf{x} = [x_0, x_1, \ldots, x_{N-1}], \tag{2.10}$$

em que $\mathbf{x}$ é o vetor de *Features*, cada x_n corresponde a uma *Feature* diferente e N é o número total de *Features*.

De modo geral, os valores de entrada, $\mathbf{x}$, podem conter valores em diversas escalas de medição, como números inteiros, números reais, intervalos, taxas e afins, mas os valores de saída, y, podem apenas ser representados por um mesmo tipo de valor, que pode ser uma escala numérica discreta com os valores possíveis definidos e associados a cada classe, e.g. {0,1,2,3}, ou os nomes das classes efetivamente, e.g. {“casual”, “explosion”, “gunshot”, “siren”}.

Para uma boa aplicação do LDA, é preciso que as amostras possuam *Features* cujos valores sejam linearmente separáveis. Em uma área (2 dimensões), será criada uma fronteira na forma de retas; no espaço (3 dimensões), haverá planos separando as classes; e no hiperespaço (n dimensões, em que n corresponde ao número de *Features*), haverá hiperplanos (dimensão $n-1$). O número de fronteiras, de modo geral, é igual a $n-1$.

Para tentar melhorar o desempenho do classificador, i.e. diminuir a taxa de classificações erroneamente realizadas (*Misclassifications*), é utilizado um critério chamado *Total Error of Classification* (TEC). A regra de classificação é simplesmente atribuir a amostra ao grupo com a maior probabilidade condicional, que é a regra de Bayes, e isso minimiza a TEC. Desta forma, para que o classificador atribua a amostra à classe i, é preciso que seja válida a condição da inequação (2.11).

$$P(i|\mathbf{x}) > P(j|\mathbf{x}) \qquad \forall \;\; j \neq i. \tag{2.11}$$

Deseja-se saber, então, qual a probabilidade de uma dada amostra (representada por um vetor de *Features* $\mathbf{x}$) pertencer à classe i, ou seja, o que se quer é obter $P(i|\mathbf{x})$, o que, na prática, não é fácil de se conseguir. O caso contrário da probabilidade condicional, no entanto, não é difícil de se conseguir, pois trata-se de um caso em que, dado que a amostra pertence à classe i, deseja-se saber qual é a probabilidade de essa amostra possuir o conjunto de *Features* $\mathbf{x}$. Com isso, pode-se aplicar a Regra de Bayes, vista na Equação (2.12), que relaciona ambas as probabilidades:

$$P(i|\mathbf{x}) = \frac{P(\mathbf{x}|i)P(i)}{\sum\limits_{\forall j} P(\mathbf{x}|j)P(j)}. \tag{2.12}$$

A probabilidade *a priori* com a qual a Equação (2.12) trabalha, $P(i)$, é conhecida sem a necessidade de que sejam realizadas medições. É comum, inicialmente, considerar que as probabilidades dos grupos são todas iguais ou, em vez disso, considerar o número de amostras em cada grupo como base para definir essas probabilidades. Pode-se, por exemplo, assumir que $P(i)$ equivale ao número de amostras da classe i dividido pelo número total de amostras disponíveis, independentemente de suas respectivas classes.

Para tornar todo o processo mais prático, assume-se que todas as classes são compostas por amostras cujos dados seguem uma distribuição multivariada Gaussiana (normal) e que todas as classes possuem uma matriz de covariância igual.

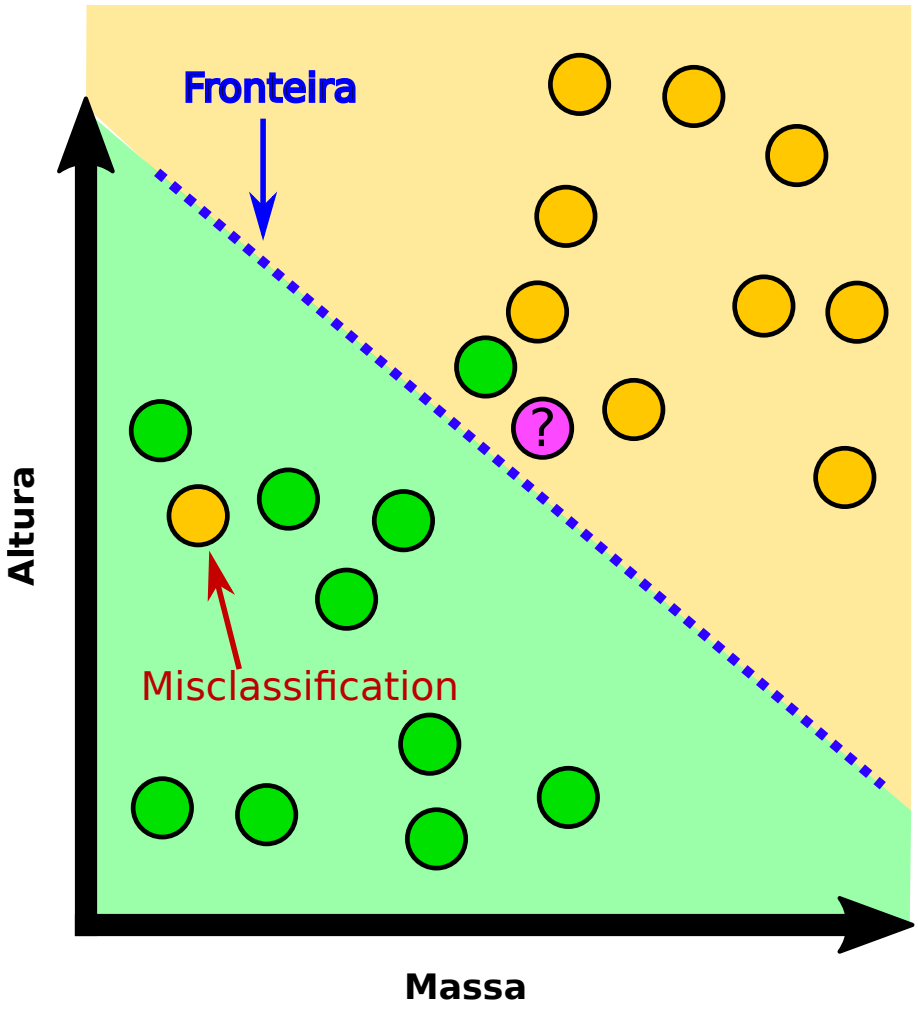
A Equação (2.13) define a fórmula principal do classificador LDA (TEKNOMO, 2015):

$$f_i = \mu_\mathbf{i}\mathbf{C^{-1}}\mathbf{x_k^T} - \frac{1}{2}\mu_\mathbf{i}\mathbf{C^{-1}}\mu_\mathbf{i}^\mathbf{T} + \ln\left(P(i)\right), \tag{2.13}$$

sendo k o número da amostra a ser analisada, i é a classe à qual deseja-se associar a amostra, f_i é a "intensidade" de associação à classe i e $P(i)$ é a probabilidade *a priori* da classe i.

Assim, possuindo-se os valores de f para cada uma das classes, pode-se afirmar que a amostra será associada à classe que lhe conferir maior valor de f.

Figura 12 – Exemplo ilustrativo de LDA.



A fim de tornar mais confortável a compreensão de um classificador LDA, é conveniente observar a Figura 12, que ilustra o comportamento de tal classificador em um exemplo hipotético de classificação com duas classes e duas *Features*. Cada eixo da figura é referente a uma *Feature* diferente e cada cor está associada a uma classe.

Suponha, por exemplo, que as amostras de cor verde são limões, i.e., estão associadas à classe "limão", e que as amostras de cor amarela são laranjas. Como se sabe, laranjas tendem a ser maiores e mais pesadas que limões, mas isso não necessariamente será observado em todas as amostras. Note que há casos em que a laranja é mais leve e menor do que alguns limões.

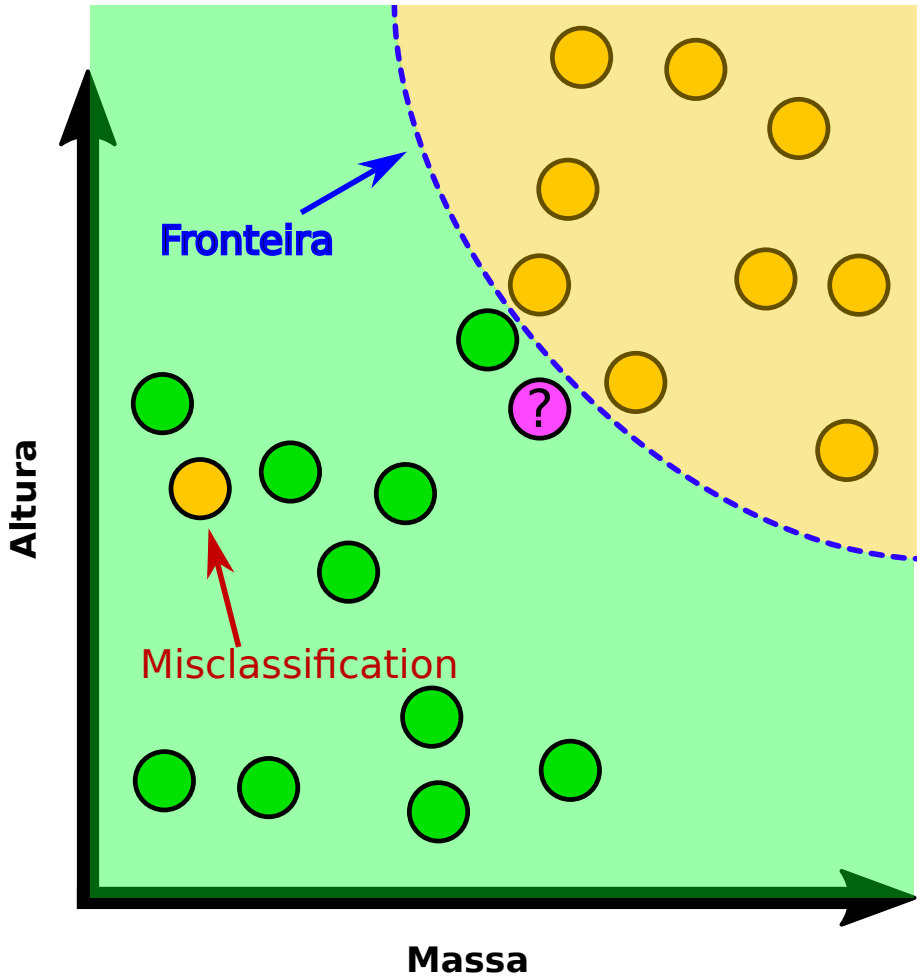
Diferente de todas as demais amostras da figura, a amostra rosada não estava presente na estapa de treinamento. Trata-se de uma nova amostra a ser classificada. Considerando as regiões delimitadas pela fronteira, a nova amostra será classificada como uma laranja.

Na próxima seção será abordada uma versão mais flexível de análise de discriminante, permitindo que sejam traçadas fronteiras baseadas em regiões cônicas.

2.6.2 *Quadratic Discriminant Analysis* (QDA)

O QDA é um tipo de classificador que trabalha de forma muito similar ao LDA, porém, em vez de utilizar hiperplanos para separar as regiões do hiperespaço, são utilizadas nuvens de fronteiras cônicas (parábolas, elipses ou hipérboles). Isso implica em regiões que são separadas por limites mais flexíveis, o que, apesar haver a possibilidade de elevar o tempo demandado para realizar a classificação, confere a capacidade de criar regiões que têm maiores chances de separar de forma mais precisa as amostras que pertencerem a diferentes categorias, aumentando, assim, a acurácia do classificador.

Figura 13 – Exemplo ilustrativo de QDA.



As amostras utilizadas no exemplo das Figuras 12 e 13 são as mesmas, ou seja, os valores de ambas as *Features* explodaras no exemplo são os mesmos. O que muda é apenas o classificador, que, como se pode observar, é mais flexível no caso do QDA.

Observa-se que, diferentemente de como ocorre com o LDA, vide Figura 12, devido à maior flexibilidade do QDA, Figura 13, a amostra rosada será classificada como "limão". Entretanto, é importante ressaltar que a maior flexibilidade do QDA não necessariamente conferirá um desempenho superior ao LDA, apesar de isso ser o esperado.

Na seção seguinte, será apresentado um novo tipo de classificador, que é mais explicitamente apoiado na Regra de *Bayes*, sendo, em geral, um tipo bastante rápido de classificador.

2.6.3 *Decision Tree*

Utilizando-se uma abordagem preditiva, a ideia do classificador baseado em *Decision Tree* utilizará todo um grande conjunto de amostras adequadamente rotuladas (classificadas) para que, a partir de uma etapa indutiva, seja possível generalizar e criar, então, um modelo que permita, por fim, que novas amostras (ainda não classificadas) possam passar por uma etapa dedutiva de classificação.

Pode-se interpretar que este classificador trabalha com base em regras binárias, o que oferece a possibilidade de representar toda a classificação na forma de árvores binárias de profundidades variadas. Para chegar à resposta desejada (classe), "pergunta-se" algo que possa ser traduzido em "sim" ou "não". Com base na resposta dada, novas "perguntas" são feitas até que seja possível concluir a qual classe pertence a amostra em questão.

O algoritmo parte da raiz da árvore e cada nó corresponde a uma etapa de decisão, em que se obtém o resultado de um teste. Assim que o resultado de um nó é obtido, parte-se para o nó seguinte, sempre procurando distinguir o atributo mais informativo (ZUBEN; ATTUX, 2010), apesar de que diferentes algoritmos podem resultar em diferentes árvores. Isso significa que não há apenas uma única árvore possível para classificar um mesmo conjunto de amostras, mas sim diversas árvores, dependendo sempre de qual é o algoritmo utilizado.

Neste trabalho foi utilizado o algoritmo *Classification and Regression Trees* (CART) (BREIMAN et al., 1984), que é capaz de pesquisar relações entre os dados até mesmo quando não são evidentes, assim como pode produzir árvores bastante simples, sendo sempre binárias. Este algoritmo não utiliza pré-poda, mas sim pós-poda pela redução do fator de custo de complexidade, expandindo-se a árvore exaustivamente (ZUBEN; ATTUX, 2010).

Para realizar a escolha dos atributos preditivos a serem explorados nos nós da árevore, o algoritmo CART utiliza uma medida conhecida como *Gini's Diversity Index* (GDI) (Gini, 1912), que, para K classes, é matematicamente definida de acordo com a Equação (2.14):

$$\text{gini}_{\text{index}}(\text{nó}) = 1 - \sum_{i=0}^{K-1} P(i|\text{nó}). \tag{2.14}$$

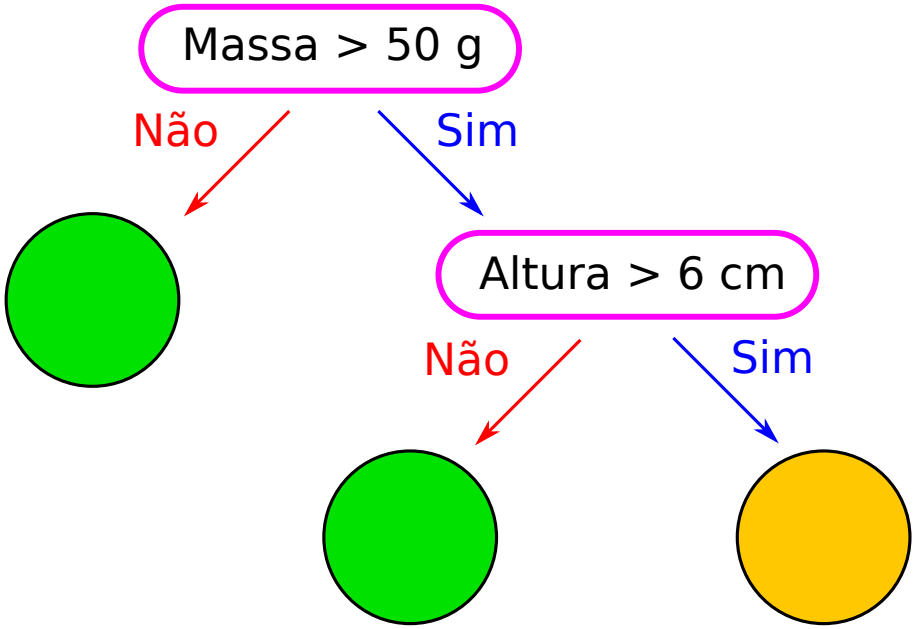
Calculando-se, então, a diferença entre o $\text{gini}_{\text{index}}$ antes e após o processo de divisão, obtém-se o *Gini*, que é, efetivamente, o que será utilizado como critério para esta seleção, selecionando-se o caso em que houver maior valor de *Gini* (ZUBEN; ATTUX, 2010), que

é definido pela Equação (2.15):

$$Gini = \text{gini}_{\text{index}}(\text{pai}) - \sum_{f=0}^{N} \left[\frac{n_f}{n_p} \, \text{gini}_{\text{index}}(f) \right], \tag{2.15}$$

em que N é o número de nós-filhos, n_p é o número total de objetos do nó-pai e n_f é o número de exemplos relacionados ao nó-filho f.

Figura 14 – Ilustração hipotética de uma *Decision Tree*.



No exemplo da Figura 14, observa-se que a primeira análise a ser realizada é se a massa da amostra é maior que 50 gramas. Caso não seja, o classificador já atribuirá a amostra à classe "limão". Por outro lado, caso a massa seja, sim, superior a 50 gramas, é necessário passar por mais uma etapa, em que é averiguado se a amostra possui altura superior a 6 centímetros. Se não for, então, segundo o classificador, trata-se de um limão; caso contrário, uma laranja.

Claramente, o exemplo abordado é bastante simples e intuitivo, visto que utiliza-se apenas de duas classes, as amostras estão convenientemente localizadas no espaço (de apenas 2 *Features*) e foram escolhidos critérios simples para elaborar a árvore, mas isso não significa que este tipo de classificador seja ruim; apenas significa que trata-se de um classificador que pode, sim, ser utilizado para tarefas mais simples sem que, para isso, seja necessariamente demandado um grande poder computacional ou elevados níveis de sofisticação matemática.

O próximo classificador se utiliza de um sistema diferente, pois, essencialmente, considera as amostras vizinhas para concluir à que classe pertence a amostra a ser classificada.

2.6.4 *k-Nearest Neighbors* (KNN)

O algoritmo *k-Nearest Neighbors* (KNN) (BIAU; DEVROYE, 2015) utilizado é preparado para funcionar ainda que haja observações com dados faltando. Se algum valor de Y ou algum peso, que são dados essenciais, estiver faltando, toda a linha correspondente à observação em questão será removida. Os pesos serão normalizados de modo que a soma de todos eles resulte em 1.

A média ponderada do preditor j é dada pela Equação (2.16)

$$\overline{x}_j = \sum_{B_j} w_k x_{jk}, \tag{2.16}$$

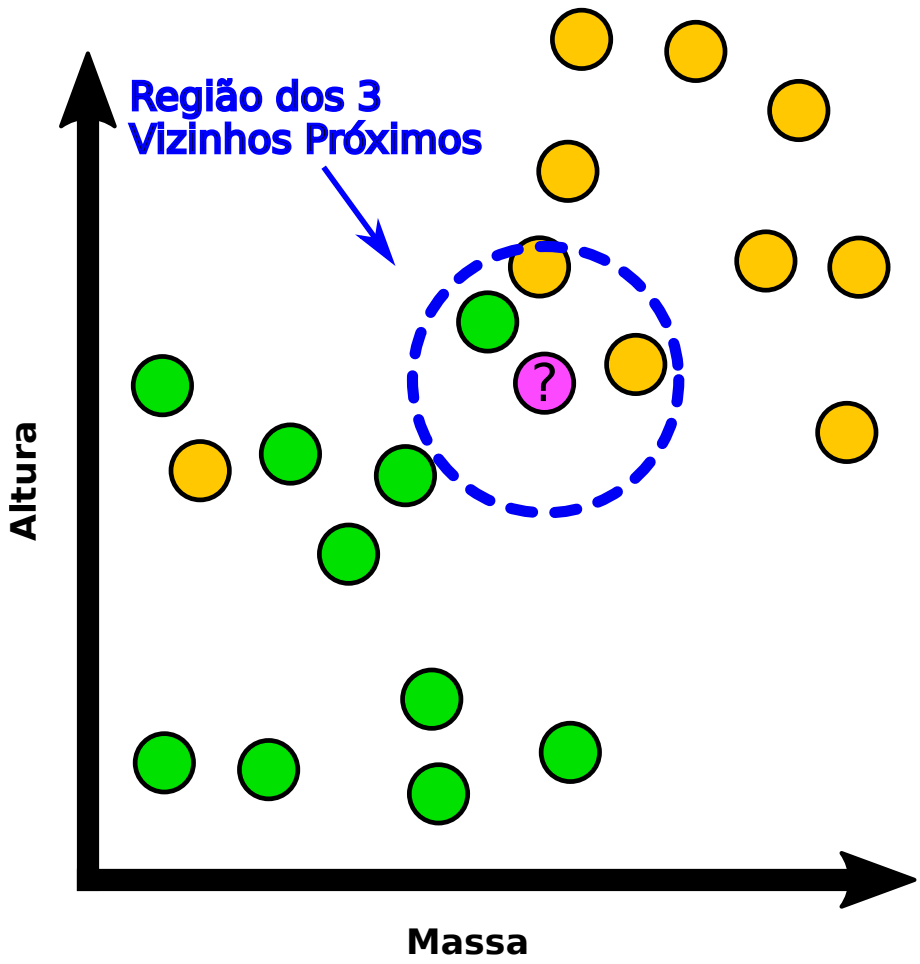
e o desvio padrão ponderado é dado pela Equação (2.17)

$$s_j = \sum_{B_j} w_k \left(x_{jk} - \overline{x}_j\right), \tag{2.17}$$

onde B_j é o conjunto de índices de k para os quais x_{jk} e w_k não estão faltando.

Para os fins desta pesquisa, foram utilizadas distâncias métricas baseadas na distância Euclidiana nos cálculos utilizando distâncias no algoritmo do KNN.

Figura 15 – Exemplo ilustrativo de KNN.



O exemplo de KNN exposto pela Figura 15 tem como objetivo ilustrar seu funcionamento de forma simples. Os eixos representam *Features* diferentes e as cores verde e amarela indicam amostras já rotuladas (conhecidas). Ao inserir uma nova amostra (cor de rosa), o algoritmo toma como base seus k vizinhos mais próximos, que no caso do exemplo da figura é igual a 3, para encontrar qual a classe "mais presente" nessa região delimitada.

Conforme foi feito nos exemplos anteriores, as amostras verdes são limões, as amarelas são laranjas e a rosa é a nova amostra a ser classificada. Assim sendo, pelo KNN com $k = 3$, a amostra nova é uma laranja.

3 Resultados e Discussão

A fim de avaliar as soluções elaboradas para a classificação dos sinais, foi realizado um conjunto de simulações com um banco de dados de sons com os diferentes tipos de eventos. Nessa seção é apresentado o procedimento adotado, bem como os resultados das simulações realizadas.

3.1 Cenário de Simulação

A base de dados utilizada nas simulações foi elaborada a partir de outras bases de dados disponíveis gratuitamente pela Internet[1] (SALAMON et al., 2017; FOGGIA et al., 2015), e, também, a partir de gravações de áudios colhidos de filmes, jogos e até mesmo vídeos gratuitamente disponibilizados via Internet. Foram considerados mais de 1000 arquivos de áudio, em 4 classes distintas a serem identificadas:

- Classe *Casual*: arquivos de áudio contendo gravações em ambientes com situação normal, sem intercorrências;
- Classe *Explosão*: arquivos de áudio nos quais se identifica a presença de sons de explosão;
- Classe *Disparos por Arma de Fogo*: arquivos de áudio contendo sons de disparos de arma de fogo;
- Classe *Alarmes*: arquivos de áudio contendo sons de sirenes e alarme (tanto de veículos como de residências).

Todos os arquivos de áudio foram pré-processados utilizando os programas *Audacity* e *SoX*. A configuração foi feita de modo que todos os áudios seguissem necessariamente um determinado padrão, com 16kHz, 8 bits, canal Mono e em formato WAV.

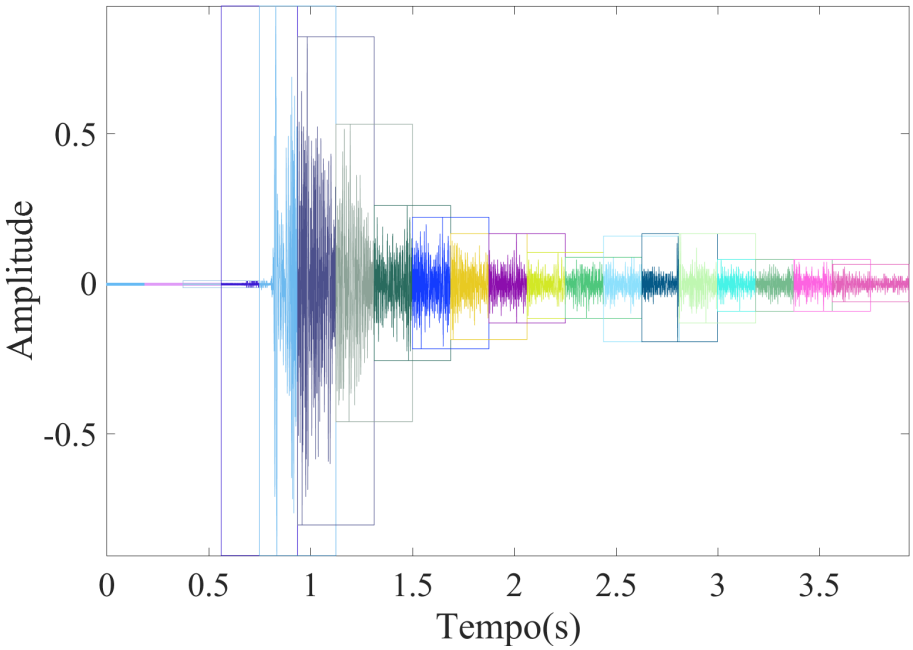
A base de áudios foi dividida em duas partes, sendo uma primeira parte (cerca de 75%) utilizada para compor a base de treinamento e uma parte menor (cerca de 25%) para compor a base de testes. Nenhum dos arquivos da base de treinamento foi utilizado durante a etapa de testes.

A etapa de extração das *Features* foi finalizada com a criação de uma base de dezenas de milhares de trechos de áudio em decorrência da segmentação adotada com trechos que seguiram o padrão de 4000 amostras cada e passos (*Overlap*) de 50%, ou seja,

[1] Muitos dos áudios também podem ser encontrados em https://freesound.org/.

2000 amostras, que foi seguida pela aplicação de um janelamento de *Hamming* por conta de possíveis efeitos de borda. Um exemplo hipotético desse tipo de segmentação pode ser observado na Figura 16, em que cada segmento está cercado por um retângulo com cor diferente.

Figura 16 – Exemplo hipotético de como foi feita a segmentação do áudio.



Com a segmentação realizada, o número final de trechos distintos foi suficiente para que pudessem ter sido realizados quase 1 milhão de testes com combinações distintas de conjuntos de trechos (não utilizados no treinamento e nem na validação) aleatoriamente selecionados para compor a base de testes a cada teste.

Foram extraídas as seguites *Features*: ZCR, Potência RMS, *Brightness*, *Spread*, *Skewness*, *Kurtosis*, Entropia, SFM e MFCC, totalizando 9 *Features* ou ainda, de outro ponto de vista, visto que o MFCC foi considerado apenas para os 13 primeiros coeficientes, podendo compreender cada coeficiente como uma diferente *Feature*, o total de *Features* aumenta para 21.

3.2 Resultados

Uma etapa essencial no processo de classificação dos trechos de áudio se refere à seleção das *Features* a serem utilizadas. Conforme mencionado no Capítulo 2, neste trabalho foi utilizada a metodologia de *Greedy Forward Selection*, através da qual as *features* eram acrescentadas uma a uma ao vetor a ser classificado. Após o pré-processamento, o

total de 21 *Features* foi reduzido para *8*, o que reduziu consideravelmente a complexidade computacional do classificador.

Apesar de haver variações para diferentes simulações envolvendo o *Wrapper*, algumas *Features* se destacaram por terem sido selecionadas em grande parte das simulações. Essas *Features* foram: ZCR, MFCC[2], *Skewness* e *Kurtosis*.

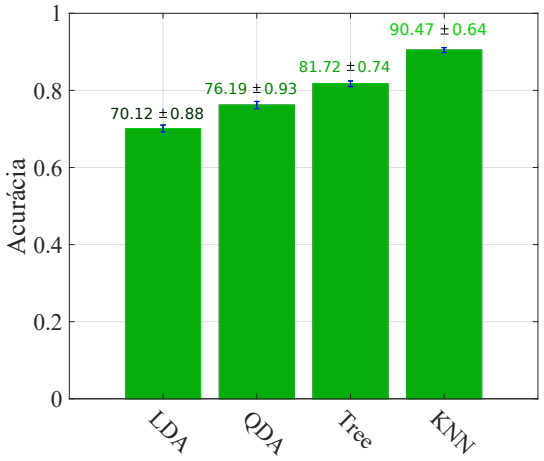
Após a etapa de seleção de atributos, foram realizadas as classificações dos trechos de áudio com 4 diferentes classificadores: LDA, QDA, KNN e *Decision Tree*. Os resultados exibidos na Tabela 4 e nas Figuras 17a e 17b mostram os valores de desempenho (acurácia e tempo demandado para concluir o processo de classificação) para cada classificador.

Com todas as classificações já efetuadas, faz-se a contagem de quantas classificações foram corretamente realizadas. Dividindo-se esse número pelo número total de classificações, obtém-se a acurácia do classificador.

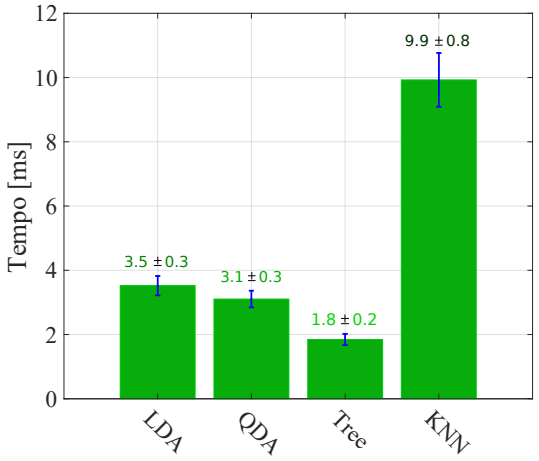
Tabela 4 – Resultado comparativo dos classificadores LDA, QDA, *Tree* e KNN.

Classificador	ΔT [ms]	Acurácia [%]
LDA	3.5 ± 0.3	70.12 ± 0.88
QDA	3.1 ± 0.3	76.19 ± 0.93
Tree	1.8 ± 0.2	81.72 ± 0.74
KNN	9.9 ± 0.8	90.47 ± 0.64

Figura 17 – Gráficos comparativos de (a) acurácia e (b) tempo de classificação dos classificadores LDA, QDA, *Tree* e KNN.



(a) Acurácia dos classificadores.



(b) Tempo de classificação dos classificadores.

Nas figuras 17a e 17b pode-se verificar que o classificador LDA demonstrou o pior desempenho no quesito acurácia, visto que mal conseguiu passar dos 70%, porém,

[2] Nem sempre os coeficientes que mais se destacavam eram os mesmos, mas o MFCC, de modo geral, teve uma frequência bastante elevada na presença após a etapa de seleção de *Features*.

diferentemente do esperado, seu tempo de classificação foi maior do que nos casos do QDA e, principalmente, do *Tree*, que chegou a ser praticamente a metade do demandado pelo LDA.

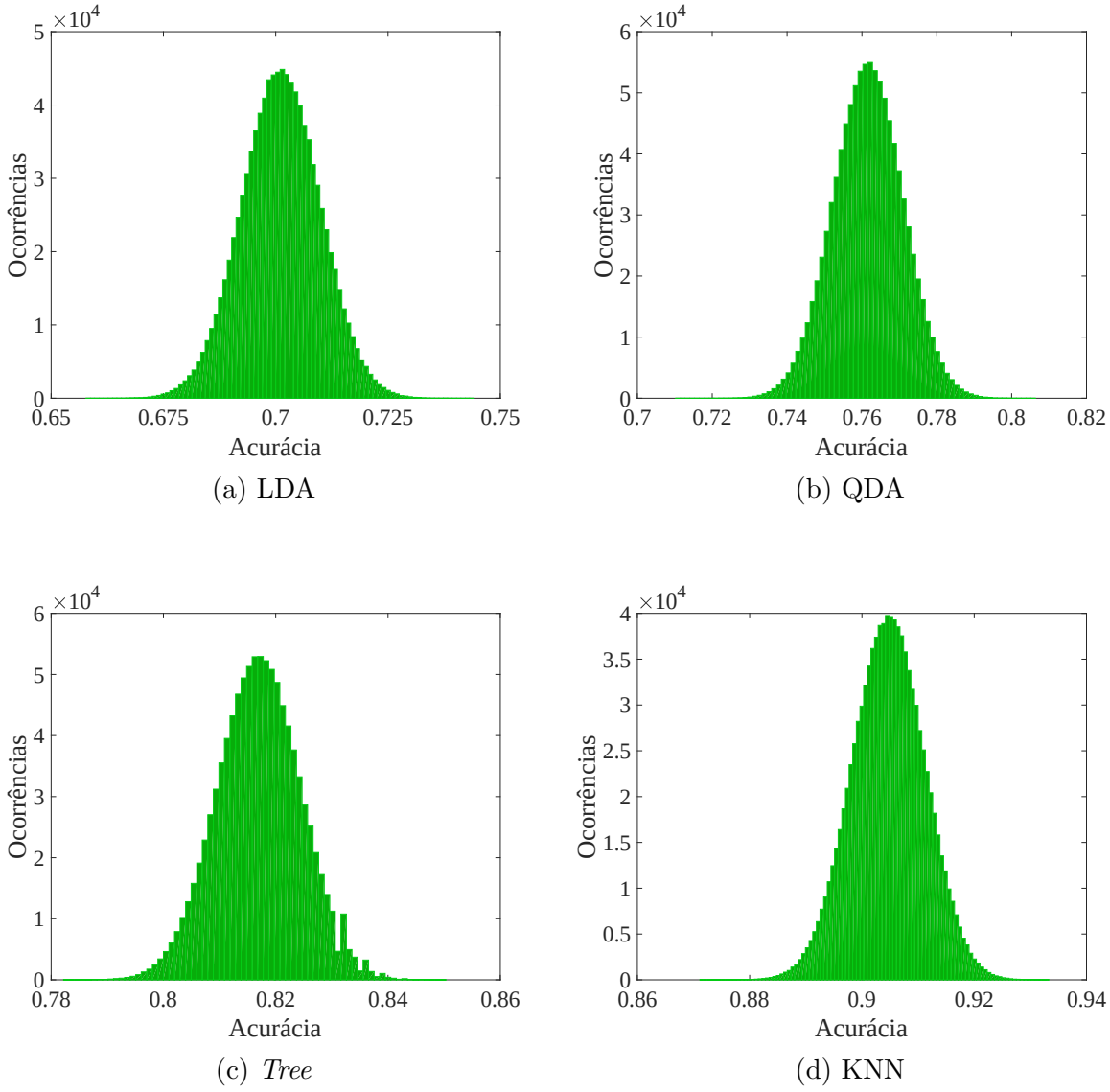
Surpreendentemente[3], o QDA demandou menos tempo que o LDA para classificar e, ainda assim, atingiu uma acurácia superior. O *Tree* foi o que demonstrou a melhor combinação de acurácia com tempo de classificação, pois conseguiu atingir mais de 81% de acurácia com menos de 2ms de tempo de classificação. O classificador KNN conseguiu atingir o mais alto nível de acurácia média com um baixo valor de desvio padrão, no entanto, o seu tempo de classificação é de quase o triplo do segundo classificador mais lento, que, no caso deste estudo, foi o LDA.

Os histogramas da Figura 18 foram elaborados a partir dos resultados de 10^6 simulações com subconjuntos aleatoriamente selecionados a partir do conjunto principal de entrada. Por eles, pode-se ver que todos os classificadores tiveram um bom comportamento em termos de desvio padrão, o que permite dizer que implica em uma alta confiabilidade. Os resultados expostos nesses histogramas foram obtidos após os testes com todas as combinações de trechos terem sido concluídos, permitindo que fosse possível obter aproximadamente a média e o desvio padrão em torno da média nas medidas de desempenho.

O histograma do classificador *Tree* exibiu uma leve excentricidade ao lado direito, na região que compreende a faixa entre cerca de $0.83 \sim 0.84$, mas isso se deve apenas ao fado de que algumas microfaixas de acurácia foram menos observadas pelos testes realizados, não implicando, portanto, em qualquer problema significativo que possa ser compreendido como um fator que comprometa o classificador.

[3] O que se espera, de modo geral, é que classificadores LDA sejam mais velozes que QDA.

Figura 18 – Histogramas de acurácia dos classificadores (a) LDA, (b) QDA, (c) *Tree* e (d) KNN.



(a) LDA (b) QDA

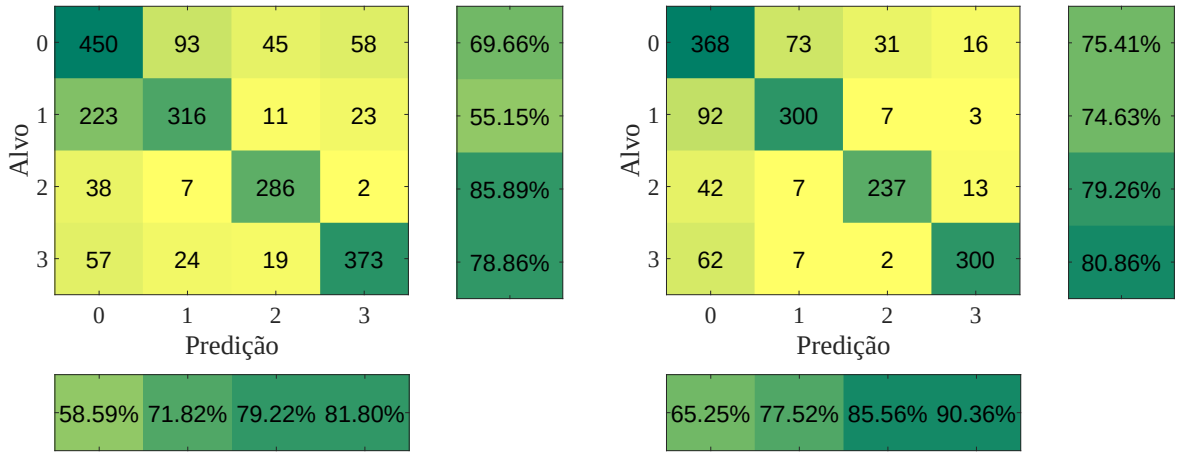
(c) *Tree* (d) KNN

Os histogramas da Figura 18 ajudam a visualizar a questão da confiabilidade de cada classificador. Quanto mais estreira a distribuição no histograma, ou seja, quanto menor é o espalhamento da distribuição, mais confiável é o valor médio de acurária do classificador, indicando que são baixas as chances de ocorrerem variações mais elevadas na acurácia do classificador.

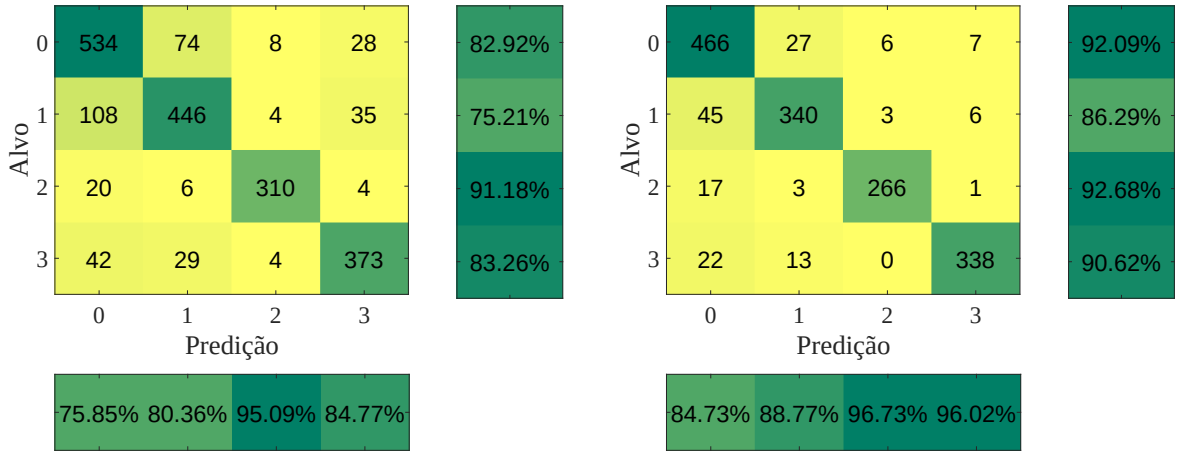
Na Figura 19 são apresentadas as matrizes de confusão obtidas com os diferentes classificadores. Por meio das matrizes é possível visualizar a quantidade de vezes em que o classificador não realizou corretamente a classificação, e qual o tipo de erro cometido. Por exemplo, na Figura 19a observa-se que o classificador LDA, em 93 ocorrências, classificou um trecho de áudio *Casual* como um áudio *Explosão* (considera-se um erro de *Falso Positivo*); e em 223 ocorrências classificou um trecho de áudio contendo explosões como sendo um trecho casual (um erro de *Falso Negativo*).

Embora ambas as situações correspondam a erros de classificação, dependendo do contexto de aplicação, é necessário tratar os tipos de erro com pesos (importâncias) diferentes. No caso específico de sistemas de segurança, por exemplo, pode-se argumentar que é mais crítico que ocorra um erro de *Falso Negativo* (há uma situação de perigo, mas o sistema não sinaliza como tal) do que um erro de *Falso Positivo*.

Figura 19 – Matrizes de confusão dos classificadores (a) LDA, (b) QDA, (c) *Tree* e (d) KNN. As classes consideradas são: 0) casual, 1) explosão, 2) disparo por arma de fogo e 3) alarme/sirene. A classe 0 indica que nenhum evento foi detectado.



(a) LDA (b) QDA



(c) *Tree* (d) KNN

Dessa forma, observa-se que, mesmo os classificadores apresentando altas taxas de classificações corretas, a taxa de *Falsos Negativos* ainda é bastante significativa.

Considerando que este trabalho atua em uma área de segurança e que, exceto pela classe (0), que se refere aos casos em que nenhuma ocorrência das demais classes foi detectada, todas as classes de eventos são referentes a algum grande risco, portanto, toda e qualquer classificação que erroneamente foi considerada como classe (0) é considerado um falso negativo.

Mesmo para o classificador KNN, verifica-se, por exemplo, que a taxa de falso negativo supera 15%, valor que precisa ser melhorado para que se possa considerar a implementação prática do sistema.

4 Conclusão

No presente trabalho foram estudadas técnicas para classificação de sinais, no contexto de sistemas de segurança baseado em áudio. Nesse tipo de sistema, o monitoramento dos ambientes é feito por meio dos sons ambientes e o sistema deve reconhecer automaticamente as situações de risco, identificando eventos sonoros pré-definidos - situação *casual*, *explosão*, *disparo de arma de fogo* e *alarme*.

Foram consideradas diferentes *Features* para realizar a classificação dos sinais — compondo assim um vetor de atributos com 21 diferentes características —. Por meio de uma metodologia de seleção de atributos, foi possível reduzir consideravelmente o tamanho do vetor de atributos, utilizando, ao final, um conjunto de apenas 8 *Features* para a classificação dos eventos.

A escolha por uma técnica ou outra deve levar em consideração a aplicação a que se destina a técnica, visto que algumas destas podem exibir leves aumentos de acurácia em detrimento de carregarem consigo o peso de serem muito mais dependentes de um elevado poder computacional, fazendo com que demorem mais tempo para serem capazes de efetuar uma dada classificação. Isso mostra que não existe uma técnica que seja definitivamente a melhor de todas para todos os fins, mas sim que há técnicas mais indicadas para determinados fins. Para o caso particular estudado neste trabalho, foi possível observar que o KNN é foi técnica mais poderosa no quesito acurácia. Entretanto, em termos do custo computacional, a melhor técnica foi a *Decision Tree*. Sob um olhar da relação custo-benefício, o modelo de *Decision Tree* utilizado saiu-se relativamente melhor, visto que sua classificação foi efetuada em menos tempo que a do KNN, apesar de sua acurácia ter sido inferior à do KNN.

Em futuros trabalhos deseja-se incluir novas categorias de classificação, minimizar os falsos negativos e implementar o melhor classificador encontrado em um *Hardware* embarcado (e.g. *Raspberry Pi* ou mesmo um FPGA).

Referências

AALST, W. M. Van der et al. Process mining: a two-step approach to balance between underfitting and overfitting. *Software & Systems Modeling*, Springer, v. 9, n. 1, p. 87, 2010. Citado na página 24.

ABDI, H.; WILLIAMS, L. J. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, Wiley Online Library, v. 2, n. 4, p. 433–459, 8 2010. ISSN 1939-0068. Disponível em: <http:http://dx.doi.org/10.1002/wics.101>. Citado na página 25.

ALTMAN, N. S. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, v. 46, n. 3, p. 175–185, 1992. Disponível em: <http://www.tandfonline.com/doi/abs/10.1080/00031305.1992.10475879>. Citado na página 24.

BACHU, R. et al. Separation of voiced and unvoiced using zero crossing rate and energy of the speech signal. In: *American Society for Engineering Education (ASEE) Zone Conference Proceedings*. [S.l.: s.n.], 2008. p. 1–7. Citado na página 15.

BIAU, G.; DEVROYE, L. *Lectures on the Nearest Neighbor Method*. Springer International Publishing, 2015. (Springer Series in the Data Sciences). ISBN 9783319253886. Disponível em: <https://books.google.com.br/books?id=GhQpCwAAQBAJ>. Citado na página 31.

BREIMAN, L. et al. *Classification and regression trees*. [S.l.]: CRC press, 1984. Citado na página 29.

CASELLA, G.; BERGER, R. *Inferencia Estatistica*. CENGAGE DO BRASIL, 2010. ISBN 9788522108947. Disponível em: <https://books.google.com.br-/books?id=VVerYgEACAAJ>. Citado na página 18.

CHEN, C. *Signal Processing Handbook*. Taylor & Francis, 1988. (Electrical and Computer Engineering). ISBN 9780824779566. Disponível em: <https://books.google.com.br-/books?id=10Pi0MRbaOYC>. Citado na página 16.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine Learning*, v. 20, n. 3, p. 273–297, 1995. ISSN 1573-0565. Disponível em: <http://dx.doi.org/10.1007/BF00994018>. Citado na página 24.

DAY, N. E. Estimating the components of a mixture of normal distributions. *Biometrika*, v. 56, n. 3, p. 463–474, 1969. Disponível em: <http://dx.doi.org/10.1093/biomet/56.3.463>. Citado na página 9.

DUBNOV, S. Generalization of spectral flatness measure for non-gaussian linear processes. *IEEE Signal Processing Letters*, v. 11, n. 8, p. 698–701, Aug 2004. ISSN 1070-9908. Citado na página 19.

DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern Classification (2Nd Edition)*. [S.l.]: Wiley-Interscience, 2000. ISBN 0471056693. Citado na página 10.

FISHER, R. A. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, Blackwell Publishing Ltd, v. 7, n. 2, p. 179–188, 1936. ISSN 2050-1439. Disponível em: <http://dx.doi.org/10.1111/j.1469-1809.1936.tb02137.x>. Citado na página 25.

FOGGIA, P. et al. Reliable detection of audio events in highly noisy environments. *Pattern Recognition Letters*, v. 65, n. Supplement C, p. 22 – 28, 2015. ISSN 0167-8655. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167865515001981>. Citado na página 33.

Gini, C. *Variabilità e mutabilità*. [S.l.: s.n.], 1912. Citado na página 29.

GUYON, X.; YAO, J.-f. On the underfitting and overfitting sets of models chosen by order selection criteria. *Journal of Multivariate Analysis*, Elsevier, v. 70, n. 2, p. 221–249, 1999. Citado na página 24.

HARTIGAN, J. *Clustering Algorithms*. New York: John Wiley & Sons Inc., 1975. Citado na página 24.

ITO, A. et al. Detection of abnormal sound using multi-stage gmm for surveillance microphone. In: *Information Assurance and Security, 2009. IAS '09. Fifth International Conference on*. [S.l.: s.n.], 2009. v. 1, p. 733–736. Citado na página 9.

JOHNSTON, J. D. Transform coding of audio signals using perceptual noise criteria. *IEEE Journal on Selected Areas in Communications*, v. 6, n. 2, p. 314–323, Feb 1988. ISSN 0733-8716. Citado na página 19.

KIKTOVA, E. et al. Gun type recognition from gunshot audio recordings. In: *Biometrics and Forensics (IWBF), 2015 International Workshop on*. [S.l.: s.n.], 2015. p. 1–6. Citado na página 9.

LARTILLOT, P. T. O.; EEROLA, T. *MIRtoolbox 1.6.1 User's Manual*. [S.l.], 2014. Citado 3 vezes nas páginas 16, 17 e 18.

MARDIA, K. V. Measures of multivariate skewness and kurtosis with applications. *Biometrika*, JSTOR, p. 519–530, 1970. Citado 2 vezes nas páginas 17 e 18.

MARKSWEEP. *Logscale plot of several symmetric unimodal probability densities with unit variance*. 2006. Disponível em: <https://upload.wikimedia.org/wikipedia-/commons/0/0b/Standard_symmetric_pdfs_logscale.png>. Citado na página 19.

MARKSWEEP. *Plot of several symmetric unimodal probability densities with unit variance*. 2006. Disponível em: <https://upload.wikimedia.org/wikipedia/commons/e/e6-/Standard_symmetric_pdfs.png>. Citado na página 18.

MCLACHLAN, G. *Discriminant Analysis and Statistical Pattern Recognition*. Wiley, 2004. (Wiley Series in Probability and Statistics). ISBN 9780471691150. Disponível em: <http://books.google.com.br/books?id=O_qHDLaWpDUC>. Citado na página 25.

MONTALVãO, J. et al. Sound event detection in remote health care - small learning datasets and over constrained gaussian mixture models. In: *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*. [S.l.: s.n.], 2010. p. 1146–1149. ISSN 1557-170X. Citado na página 9.

NTALAMPIRAS, S.; POTAMITIS, I.; FAKOTAKIS, N. On acoustic surveillance of hazardous situations. In: *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. [S.l.: s.n.], 2009. p. 165–168. ISSN 1520-6149. Citado na página 9.

NTALAMPIRAS, S.; POTAMITIS, I.; FAKOTAKIS, N. A portable system for robust acoustic detection of atypical situations. In: *Signal Processing Conference, 2009 17th European*. [S.l.: s.n.], 2009. p. 1121–1125. Citado na página 9.

PEREIRA, F.; MITCHELL, T.; BOTVINICK, M. Machine learning classifiers and fmri: A tutorial overview. *NeuroImage*, v. 45, n. 1, Supplement 1, p. S199 – S209, 2009. ISSN 1053-8119. Mathematics in Brain Imaging. Disponível em: <http:/-/www.sciencedirect.com/science/article/pii/S1053811908012263>. Citado na página 24.

PETRY, A.; ZANUZ, A.; BARONE, D. A. C. Utilização de técnicas de processamento digital de sinais para a identificação automática de pessoas pela voz. *Simpósio sobre Segurança em Informática, São José dos Campos, SP*, 1999. Citado na página 21.

RABAOUI, A.; LACHIRI, Z.; ELLOUZE, N. Robustness improvement of an automatic sounds recognition system by hmm adaptation to real world background noise. In: *Information and Communication Technologies, 2006. ICTTA '06. 2nd*. [S.l.: s.n.], 2006. v. 1, p. 1298–1299. Citado na página 9.

RAZA, M.; QAMAR, U. *Understanding and Using Rough Set Based Feature Selection: Concepts, Techniques and Applications*. Springer Singapore, 2017. ISBN 9789811049651. Disponível em: <http://books.google.com.br/books?id=4DUqDwAAQBAJ>. Citado na página 23.

RÖCKER, C. et al. *Information and Communication Technologies for Ageing Well and e-Health: Second International Conference, ICT4AWE 2016, Rome, Italy, April 21-22, 2016, Revised Selected Papers*. Springer International Publishing, 2017. (Communications in Computer and Information Science). ISBN 9783319627045. Disponível em: <http://books.google.com.br/books?id=24otDwAAQBAJ>. Citado na página 11.

RODà, A.; MICHELONI, C. Tracking sound sources by means of hmm. In: *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*. [S.l.: s.n.], 2011. p. 83–88. Citado na página 9.

SALAMON, J. et al. Scaper: A library for soundscape synthesis and augmentation. In: *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA). New Paltz, NY, USA*. [S.l.: s.n.], 2017. Citado na página 33.

SASOU, A. et al. Acoustic based abnormal event detection using robust feature compensation. In: *TENCON 2011 - 2011 IEEE Region 10 Conference*. [S.l.: s.n.], 2011. p. 255–258. ISSN 2159-3442. Citado na página 9.

SATAPATHY, S. et al. *Information Systems Design and Intelligent Applications: Proceedings of Third International Conference INDIA 2016*. Springer India, 2016. (Advances in Intelligent Systems and Computing, v. 1). ISBN 9788132227557. Disponível em: <https://books.google.com.br/books?id=qFqLCwAAQBAJ>. Citado na página 19.

SHERWOOD, J.; DERAKHSHANI, R. *A Quality Metric to Improve Wrapper Feature Selection in Multiclass Subject Invariant Brain Computer Interfaces*. University of Missouri-Kansas City, 2011. Disponível em: <http://books.google.com.br-/books?id=sWtqAQAACAAJ>. Citado na página 23.

SIMPLIFIED, L. *Skewness and kurtosis | kullabs.com*. Disponível em: <http://www-.kullabs.com/classes/subjects/units/lessons/notes/note-detail/9135>. Citado na página 18.

SUN, P.; YAO, X. Greedy forward selection algorithms to sparse gaussian process regression. In: *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. [S.l.: s.n.], 2006. p. 159–165. ISSN 2161-4393. Citado na página 24.

TEKNOMO, K. *Discriminant Analysis Tutorial*. [S.l.], 2015. Citado na página 26.

VALENZISE, G. et al. Scream and gunshot detection and localization for audio-surveillance systems. In: *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*. [S.l.: s.n.], 2007. p. 21–26. Citado na página 9.

WOODS, R. et al. *FPGA-based Implementation of Signal Processing Systems*. Wiley, 2017. ISBN 9781119077954. Disponível em: <http://books.google.com.br-/books?id=77D4DQAAQBAJ>. Citado na página 11.

ZUBEN, F. J. V.; ATTUX, R. R. F. *Árvores de Decisão*: Notas de aula. 2010. Citado na página 29.