
Curso de Ciência da Computação
Universidade Estadual de Mato Grosso do Sul

Análise de Sanidade de Madeira Utilizando o Extrator de Características MARSYAS e a Plataforma WEKA

Juliano Pereira Lima
Jullierme Rodrigues Gomes

Dr. Rubens Barbosa Filho (Orientador)

Curso de Ciência da Computação
Universidade Estadual de Mato Grosso do Sul

Dourados - MS
2014

ANÁLISE DE SANIDADE DE MADEIRA UTILIZANDO A PLATAFORMA WEKA

Juliano Pereira Lima e Jullierme Rodrigues Gomes

Monografia da disciplina Projeto Final de Curso, Coordenadoria do Curso Ciência da Computação, Universidade Estadual do Mato Grosso do Sul, como requisito para obtenção do título de Bacharel em Ciência da Computação.

Rubens Barbosa Filho

(Orientador)

Resumo

O reconhecimento de padrões é extremamente importante nas ferramentas automatizadas de classificação na atualidade. Na classificação de madeira não é diferente, a utilização de seus conceitos propicia análises com alta precisão.

Este trabalho tem como objetivo fornecer uma alternativa a forma de análise de sanidade de madeira convencional, utilizando as ferramentas automatizadas de extração de características e testes MARSYAS E WEKA respectivamente.

Inicialmente foram disponibilizados pelo orientador Dr Rubens Barbosa Filho, 150 arquivos de áudio, captados através da batida de um martelo normal ,com sanidade estimada por um especialista, e outros 150 captados através da batida de um martelo automático.

Os 150 sons captados através da batida de martelo manual e 150 através do martelo automático são divididos em 3 categorias, sendo 50 arquivos de sons respectivos a cada categoria: 1 Ano , onde a madeira tem pelo menos 1 ano de vida útil. Emergencial, onde a troca é necessária, mas não urgente. E Urgencial, onde a troca da peça de madeira é necessária com urgência.

Foram implementados e aplicados algoritmos de normalização de dados por padronização sobre os arquivos de áudio. Após o pré-processamento foi aplicado o janelamento de sinal. Posteriormente, no MARSYAS foram extraídos os coeficientes LPCC e MFCC e armazenados em arquivos de extensão ".arff", para aplicação posterior na plataforma WEKA.

Por fim, este material irá servir como apoio a futuras atividades de pesquisa na área de Mineração de dados.

Palavras-Chave: Análise, Classificação, Madeira, Pré-Processamento, Sanidade.

Abstract

The pattern recognition is extremely important in automated classification tools at the present time. On the classification of wood is no different, the use of their concepts provides high-precision analyses.

This study aims to provide an alternative form of conventional wood health analysis, using the automated tools and features extraction tests MARSYAS and WEKA respectively.

Were initially provided by the Advisor Dr Rubens Barbosa Filho, 150 audio files, obtained through the beat of a normal hammer, with estimated health by a specialist, and other 150 captured through the beat of an automatic hammer.

The 150 sounds picked up by tapping hammer manual and 150 through automatic hammer are divided into 3 categories, being 50 sound files corresponding to each category: 1 Year, where wood has at least 1 year of life. Emergency, where the Exchange is necessary, but not urgent. And Urgencial, where the urgent change is required.

Have been implemented and applied algorithms of data standardization for Standardization on the audio files. After the preprocessing has been applied the windowing of signal. Subsequently, the MARSYAS were extracted the coefficients LPCC and MFCC and stored in files of extension ".arff", for subsequent application in WEKA platform.

Finally, this material will serve as support to future research activities in the area of data mining.

Keyword: Analysis, Classification, Wood, Preprocessing, Sanity.

Agradecimentos

Em primeiro lugar meu agradecimento vai a Deus, pois, sem ele jamais chegaria até aqui.

Agradeço imensamente ao professor Rubens, pela orientação ao longo de todo o trabalho, sendo exemplo de paciência e dedicação.

Agradeço imensamente a minha mãe Adelita Aparecida Gutierrez Pereira, pelo apoio, empenho, carinho e paciência, por nunca ter deixado de me guiar, te amo muito mãe.

Agradeço muito a minha namorada Luara, por todo o carinho, apoio e compreensão ao longo do desenvolvimento desse trabalho.

E a todos os demais colegas e Professores que contribuíram para o sucesso deste trabalho.

Juliano Pereira Lima.

Agradeço muito ao apoio recebido da minha família, principalmente aos meus pais Pedro e Luiza, por todo o incentivo, carinho, paciência e suporte nos momentos difíceis.

Ao professor Rubens, por ter nos orientado da melhor forma possível com muita paciência e dedicação.

Aos colegas de turma e amigos que sempre contribuíram de forma direta ou indireta para a conclusão desta etapa.

Jullierme Gomes Rodrigues.

Sumário

Introdução	11
Fundamentação Teórica.....	15
2.1 Propriedades da Madeira	15
2.1.2 Aplicações da Madeira.....	15
2.1.1 Vida Útil	16
2.2 Som e suas Características.....	17
2.2.1 Arquivo WAV (<i>Waveform Audio Format</i>).....	18
Pré Processamento de Sinal	19
3.1 Conceitos	19
3.2 Janelamento de Sinal	19
3.3 Normalização	20
3.3.1 Reescala	20
3.3.2 Padronização.....	20
MARSYAS.....	23
4.1 Conceitos	23
4.2 Montando Listas de Arquivos de Áudio.....	23
4.3 Extração de Características	24
4.3.1 Extraindo Coeficientes LPCC (<i>Linear Predictive Coding Ceptrum</i>)	25
4.3.2 Extraindo Coeficientes MFCC(<i>Mel-Frequency Cepstral Coefficients</i>)	27
WEKA	29
5.1 Conceitos	29
5.2 K-Vizinhos mais Próximo (KNN)	32
5.3 K-Means	33
Metodologia	35
6.1 Ambiente de desenvolvimento	35
6.2 Pré Processamento de Sinal.....	35
6.3 Aplicação do MARSYAS	35
6.4 Testes na plataforma WEKA	36
Resultados.....	37
7.1 Classificação com o KNN.....	37
7.2 Classificação com o <i>K-Means</i>	40
Conclusão.....	45
Referências Bibliográficas.....	47

Exemplo de Base de Dados do WEKA Bibliográficas 51

Lista de Siglas

KNN - k-Nearest Neighbors Algorithm

LPC - *Linear Predictive Coding*

LPCC - *Linear Predictive Coding Ceptrum*

MARSYAS - *Music Analysis, Retrieval and Synthesis for Audio Signals*

MIR - *Music Information Retrieval*

MFCC - *Mel-Frequency Cepstrum Coefficients*

WAVE - *Waveform Audio Format*

WEKA - *Waikato Environment for Knowledge Analyses*

Lista de Tabelas

Tabela 1. Resultados dos treinamentos do KNN para pasta Manual.....	37
Tabela 2. Resultados dos treinamentos do KNN para pasta Automática.....	39
Tabela 3. Resultados dos treinamentos do K-Means para pasta Manual.....	40
Tabela 4. Resultados dos treinamentos do K-Means para pasta Automática..	42

Lista de Figuras

Figura 1.1: Etapas de reconhecimento de Som [2].....	11
Figura 2.1: Ilustração da propagação do som ([7]).....	17
Figura 2.2: Representação da senóide do som sem adição de ruído([7]).....	18
Figura 2.3: Representação da senóide do som com adição de ruído([7]).....	18
Figura 4.1: Exemplo de comando com <i>mkcollection</i>	23
Figura 4.2: Arquivo gerado após a utilização do executável <i>mkcollection</i>	24
Figura 4.3: Exemplo da linha de comando com <i>bextract</i>	25
Figura 4.4: Arquivo gerado após a utilização do executável <i>bextract</i>	25
Figura 4.5 : Ilustração do Processo LPC.	26
Figura 4.6 : Exemplo de comando para extração de coeficientes LPCC.....	26
Figura 4.7: Exemplo de comando para extração de coeficientes MFCC.....	27
Figura 5.1: Tela inicial do WEKA para seleção de interface.....	30
Figura 5.2: GUI <i>Explorer</i> do WEKA.....	30
Figura 5.3: GUI <i>Experimenter</i> do WEKA.....	31
Figura 5.4: GUI <i>Knowledgeflow</i> do WEKA.....	31
Figura 5.5: GUI <i>SimpleCLI</i> do WEKA.....	32
Figura 5.6: Classificação feita pelo KNN para um padrão desconhecido[17]..	33
Figura 5.7: Exemplo de Execução do algoritmo K-Means [18].....	34
Figura 7.1: Matriz de confusão LPCC(a) e MFCC(b) para pasta manual.....	37
Figura 7.2: Comparação da classificação de instâncias extraídas com LPCC e MFCC para a pasta Manual.....	38
Figura 7.3: Tempo de execução da classificação das instâncias extraídas com LPCC e MFCC para a pasta manual.....	38
Figura 7.4: Matriz de confusão LPCC(a) e MFCC(b) para pasta automática..	39
Figura 7.5: Comparação da classificação de instâncias extraídas com LPCC e MFCC para a pasta Automática	39
Figura 7.6: Tempo de execução da classificação das instâncias extraídas com LPCC e MFCC para a pasta Automática	40
Figura 7.7: Clusterização das instâncias de LPCC(a) e MFCC(b) para pasta Manual.....	41
Figura 7.8: Atribuição de instâncias extraídas com LPCC às classes	41
Figura 7.9: Atribuição de instâncias extraídas com MFCC às classes.....	42

Figura 7.10: Clusterização das instâncias de LPCC(a) e MFCC(b) para pasta Automática	43
Figura 7.11: Atribuição de instâncias extraídas com LPCC às classes	43
Figura 7.12: Atribuição de instâncias extraídas com MFCC às classes.....	44

Lista de Algoritmos

KNN.....	30
K-Means.....	31

Capítulo 1

1 Introdução

O avanço da tecnologia propiciou a utilização de ferramentas automatizadas voltadas para a classificação e reconhecimento de padrões. Porém, grande parte das classificações de materiais ainda são feitas manualmente, o que requer presença de especialistas, tornando o processo demorado e sujeito a erros de inconsistência nos dados.

Atualmente, o reconhecimento de padrões tem sido parte integrante de classificadores automatizados, por propiciar maior exatidão nos resultados. O Reconhecimento de Padrões trata da classificação e da descrição de uma estrutura de dados através de um conjunto de propriedades ou características[20].

A verificação de sanidade de madeira pode ser verificada através do som captado da batida na madeira. A Figura 1.1 a seguir, exemplifica os passos do processo de classificação de determinado som, tal qual um classificador de sanidade de madeira:

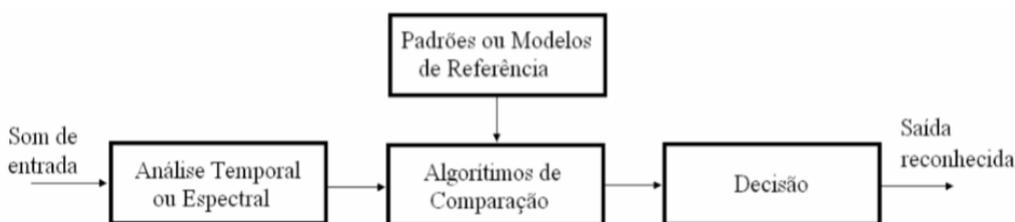


Figura 1.1: Etapas de reconhecimento de Som [2]

Um sistema que realiza o reconhecimento de áudio tem como sinal de entrada o som que será analisado pelo sistema. Na análise temporal ou espectral é feita a extração de parâmetros do áudio. Esta etapa resulta na transferência dos parâmetros encontrados aos algoritmos de comparação[2].

Estes algoritmos desenvolvem toda a comparação do áudio analisado com os padrões de referência. A partir desta comparação é tomada a decisão se houve ou não reconhecimento do áudio analisado, e assim é exteriorizada a resposta do sistema[2].

A estimativa de vida útil da madeira se faz extremamente necessária para garantir a segurança no seu manuseio. O uso da madeira, além de ser ambientalmente correta,

apresenta inúmeras vantagens inerentes ao próprio material[21]. Entre elas podem ser citadas a alta relação resistência mecânica por unidade de massa e as propriedades de resistência tanto à tração quanto à compressão e flexão[21].

Nesse contexto, a inteligência artificial se mostra como um grande meio de aperfeiçoamento de técnicas de classificação. Cada vez mais a inteligência artificial tem estado presente em nossas vidas, ela está em todos os lugares, na automatização dos carros e das indústrias, nos computadores e na medicina[1].

Este trabalho tem como objetivo realizar uma análise de sanidade de madeira utilizando a plataforma WEKA a fim de disponibilizar conhecimentos sobre o mesmo.

Os Objetivos específicos são:

- Extrair dados válidos de arquivos de som.
- Estudar os conceitos matemáticos necessários para o entendimento das técnicas de classificação de dados.
- Estudar os conceitos básicos e implementar os algoritmos que façam o pré-processamento de Dados.
- Estudar e aplicar os algoritmos LPC e MFCC de extração de características de dados através do *framework* MARSYAS.
- Testar os resultados obtidos na plataforma WEKA.

A justificativa da realização do trabalho é oferecer uma alternativa á análise de sanidade de madeira convencional, visando aumentar a segurança no manuseio da madeira e redução de custos financeiros.

Para o alcance dos objetivos, o trabalho foi distribuído nas seguintes etapas:

1. Estudo dos conceitos estatísticos, conceitos de som digital e de classificação de dados, através da revisão bibliográfica.
2. Implementação de algoritmos para pré-processamento de dados obtidos dos arquivos de som.
3. Utilização de algoritmos de extração de características LPC e MFCC através do *framework* MARSYAS.
4. Aplicação dos dados extraídos no MARSYAS na plataforma WEKA.

No capítulo 2, são relatados os conceitos teóricos sobre propriedades, aplicação e tempo de vida útil da madeira, além de descrever propriedades do som digital, necessário para o melhor entendimento do processo de classificação. No capítulo 3 são

relatados os principais conceitos em relação a pré-processamento de sinais, necessários para a melhor aplicação de algoritmos de extração de características.

No capítulo 4, são descritos conceitos sobre o *framework* MARSYAS, e como através dele, aplicar algoritmos de extração de características em arquivos de áudio. No Capítulo 5 são descritos os conceitos sobre a plataforma WEKA, e sobre o funcionamento e aplicação dos algoritmos KNN e *K-Means*.

O capítulo 6 trata da metodologia do trabalho, conceituando sobre o ambiente de desenvolvimento das ferramentas utilizadas no trabalho, os passos da implementação de algoritmos de pré-processamento de sinal, a extração de características no MARSYAS, e testes na plataforma WEKA. Os resultados da precisão da classificação da sanidade da madeira são descritos no capítulo 7.

Capítulo 2

Fundamentação Teórica

2.1 *Propriedades da Madeira*

A madeira é um material composto de células produzidas por uma árvore viva para suportar a copa, conduzir água e nutrientes dissolvidos do solo à copa, armazenar materiais de reserva (principalmente carboidratos), sendo um tecido complexo devido a sua formação por diferentes tipos de células com funções diferenciadas[12].

Sua natureza biológica, submete-a aos diversos mecanismos de deterioração existentes na natureza. A essas características acrescenta-se a sua susceptibilidade ao fogo[10].

Essas desvantagens da madeira podem ser eliminadas ou, ao menos minimizadas, bastando para tal o emprego de tecnologias já disponíveis e de uso consagrado nos países desenvolvidos [10].

O aspecto, no entanto, que distingue a madeira dos demais materiais é a possibilidade de produção sustentada nas florestas nativas e plantadas e nas modernas técnicas siveiculturais¹ empregadas nos reflorestamentos, que permitem alterar a qualidade da matéria-prima de acordo com o uso final desejado [10].

No entanto, o desconhecimento das propriedades da madeira por muitos de seus usuários e a insistência em métodos de construção antiquados, são as maiores causas de desempenho insatisfatório da madeira frente a outros materiais [10].

Para a madeira, material caracteristicamente heterogêneo, a determinação de suas propriedades mecânicas é condição fundamental para sua utilização [14].

2.1.2 *Aplicações da Madeira*

Dentre as propriedades da madeira que justificam a preferência de sua utilização a outros materiais, são comumente citados o baixo consumo de energia para seu processamento, a alta resistência específica, as boas características de isolamento

¹ Técnicas que visam regenerar e melhorar povoamentos florestais

² *Music Information Retrieval*(Recuperação de informação de música)

térmico e elétrico, além de ser um material muito fácil de ser trabalhado manualmente ou por máquinas [10].

A alta resistência da madeira em relação ao seu baixo peso e o baixo consumo energético necessário para sua produção, são propriedades essenciais de materiais estruturais, principalmente para utilização em construções rurais[5].

A madeira é um excelente material para a construção de pontes em estradas vicinais no meio rural, para pequenos e médios vãos, não só pela frequente disponibilidade como, também, pelo seu potencial de resistência e durabilidade, o que a torna economicamente interessante[5].

A madeira é empregada na construção civil, de forma temporária, na instalação do canteiro de obras, nos andaimes, nos escoramentos e nas formas. De forma definitiva, é utilizada nas esquadrias, nas estruturas de cobertura, nos forros e nos pisos[10].

Hoje, a madeira ainda continua participando da matriz energética mundial, com maior ou menor intensidade, dependendo da região considerada[13].

A madeira e seus resíduos são muito utilizados para gerar energia devido à sua capacidade calorífica. A geração de energia por madeira é bastante vantajosa, pois economiza outras fontes de energia[13].

Seu uso é afetado por variáveis como: nível de desenvolvimento do país, disponibilidade de florestas, questões ambientais e sua competição econômica com outras fontes energéticas, como petróleo, gás natural, hidroeleticidade e energia nuclear [13].

2.1.3 Vida Útil

A vida útil de uma peça de madeira é o tempo necessário para que esta seja substituída. Além de sofrer a ação das intempéries ambientais, a madeira também é passível de biodeterioração por organismos que de uma forma ou de outra atuam sobre os compostos orgânicos da madeira[22].

As principais causas da deterioração da madeira são[23]:

- Desgaste Mecânico
- Degradação Física
- Degradação Química
- Degradação Biológica

2.2 Som e suas características

O som é a vibração do ar, ou seja, variações na pressão do ar, percebida pelo sistema auditivo. Se essa pressão varia de acordo com um padrão repetitivo diz-se que o som tem uma forma de onda periódica. Se não há um padrão perceptível no som este pode ser chamado de ruído [6].

O som apresenta as seguintes características básicas:

1. Amplitude: valor instantâneo do nível de sinal.
2. Frequência: quantidade de vezes na unidade de tempo em que a forma de onda do sinal se repete. Para os sinais compostos por mais de uma frequência, normalmente é levado em consideração dois valores distintos de frequência, um deles é o menor valor de frequência com alta amplitude (frequência fundamental), e o outro é a maior frequência do sinal (utilizado para delimitar a filtragem).
3. Fase: ângulo em que o sinal se apresenta[2]

A propagação do som pode ser representada por uma senóide. A representação pode ser observada na Figura 2.1:

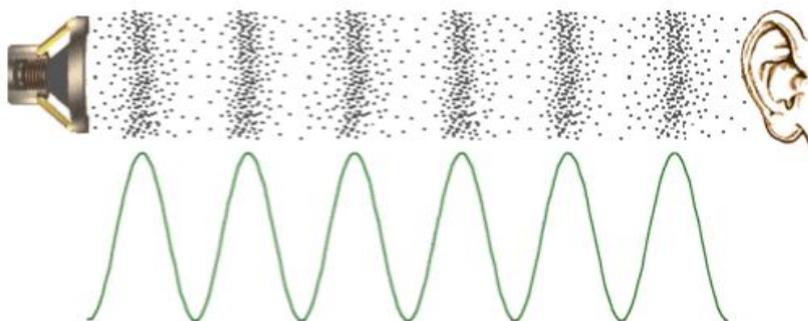


Figura 2.1: Ilustração da propagação do som ([7]).

Sistemas analógicos estão sujeitos a sofrerem interferência de fontes aleatórias na captação [7]. Estas interferências fazem com que o ruído seja adicionado ao sinal,

provocando distorções em suas amostragens. As Figuras 3 e 4 mostram respectivamente a representação gráfica das ondas sonoras sem e com adição do ruído ao sinal.

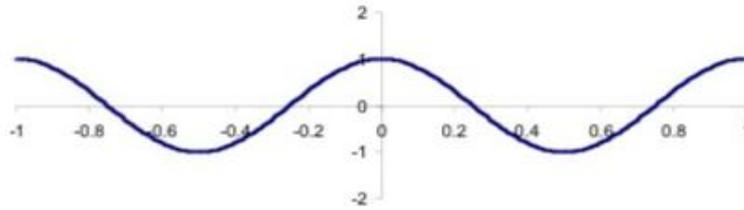


Figura 2.2: Representação da senoide do som sem adição de ruído([7]).

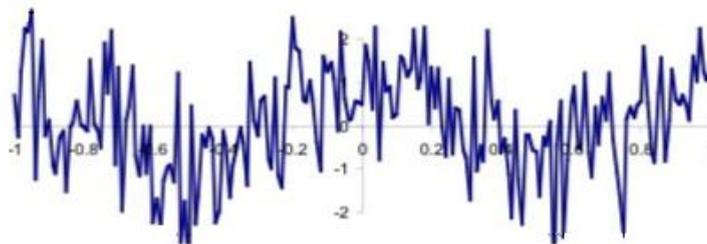


Figura 2.3: Representação da senoide do som com adição de ruído([7])

2.2.1 Arquivo WAV (*Waveform audio format*)

Wave(Wav) é um tipo de formato onde o som é armazenado através de sequencias numéricas. É uma forma de codificação em que o áudio é convertido em dados podendo ser gravado “fielmente” bit por bit [8]. O formato *Wave* foi criado pela IBM e pela Microsoft, nos anos 80 e suporta uma série de resoluções de bit, taxas de amostragens e canais de áudio[8]. É também muito utilizado em programas profissionais que processam áudio digital [8].

Um arquivo no formato WAVE possui um cabeçalho, o qual fornece algumas informações importantes, sendo as principais[20]:

- taxa (frequência) de amostragem;
- número de canais;
- quantização – resolução em *bits*;
- tamanho do arquivo em *bytes*.

Capítulo 3

Pré-Processamento de Sinal

3.1 Conceitos

Podem existir diversos objetivos na fase de pré-processamento. Um deles é solucionar problemas nos dados, tais como identificar e tratar dados corrompidos, atributos irrelevantes e valores desconhecidos [9].

Características do sinal de entrada que refletem o ambiente de gravação e o canal de comunicação, normalmente, atrapalham a tarefa do reconhecimento de padrões. As ações realizadas na fase de pré-processamento visam preparar os dados para que a fase seguinte, a fase de extração de conhecimento, seja mais efetiva[9].

3.2 Janelamento de Sinal

O janelamento de sinais é uma técnica simples que pode aumentar as características espectrais do sinal amostrado[21].

Em aplicações práticas envolvendo a amostragem de sinais pode-se obter somente uma gravação finita do sinal. Isso resulta em uma forma de onda truncada que possui características espectrais diferentes do sinal original. Tal descontinuidade produz a perda da informação espectral original[21].

Ao analisar uma sequência de dados finita através de Fourier ou outro método de análise espectral, o janelamento minimiza as margens de transição em formas de ondas truncadas, reduzindo dessa forma a perda espectral[21].

Existem várias razões para a utilização do janelamento de sinais. Algumas delas são[21]:

- Definição da duração do período de observação do sinal.
- Redução da perda espectral.
- Separar de um sinal de pequena amplitude de um sinal de grande amplitude

3.3 Normalização

Consiste em transformar os valores dos atributos de seus intervalos originais para um intervalo específico, como, por exemplo, [-1,1] ou [0,1] [18]. Esse tipo de transformação é especialmente valiosa para os métodos que calculam distâncias entre atributos [18]. Por exemplo, um método como o k-vizinhos mais próximos tende a dar mais importância para os atributos que possuem um intervalo maior de valores [18].

A ideia da normalização é fazer com que o sinal amostrado tenha a sua amplitude em um determinado intervalo de valores.

3.3.1 Reescala

Uma das formas de normalização é por reescala. Dentro de determinada amostra de valores, para cada valor do conjunto, é aplicada a seguinte equação:

$$Valor_{Novo} = Min + \left(\frac{X - Menor}{Maior - Menor} \right) * (Max - Min) \quad (1)$$

Onde Min e Max são respectivamente os valores mínimo e máximo para o intervalo desejado, Menor é e Maior são respectivamente o menor e maior valor da amostra.

3.3.2 Padronização

A normalização pode ser feita também por padronização, onde são utilizados dados estatísticos. É necessário o cálculo da média aritmética simples e variância dos valores da amostra. O cálculo da média é dado pela seguinte equação:

$$Media = \frac{\sum x}{N} \quad (2)$$

Onde $\sum x$ é a soma de todos os valores do conjunto e N o número de elementos do mesmo. Para se calcular a variância é necessário o cálculo dos desvios do conjunto. O cálculo do desvio é dado pela seguinte equação:

$$Desvio = X - Media \quad (3)$$

Onde X é um valor da amostra. Define-se a variância, como sendo a medida que se obtém somando os quadrados dos desvios das observações da amostra, relativamente à sua média, e dividindo pelo número de observações da amostra menos um. O cálculo da variância é dado pela seguinte equação:

$$\text{Variância} = \frac{(\sum X)^2}{N-1} \quad (4)$$

Onde a parte superior da equação é a somatória dos desvios ao quadrado, e N o número de elementos da amostra. A padronização em determinado conjunto é dada pela seguinte equação:

$$\text{Valor}_{\text{Novo}} = \frac{(Y - \text{Média})}{\text{Variância}} \quad (5)$$

Onde Y é o valor da amostra sem a aplicação da fórmula, a Média é a soma de todos os valores do conjunto dividido pelo número de elementos do mesmo, e a variância.

Capítulo 4

MARSYAS

4.1 Conceitos

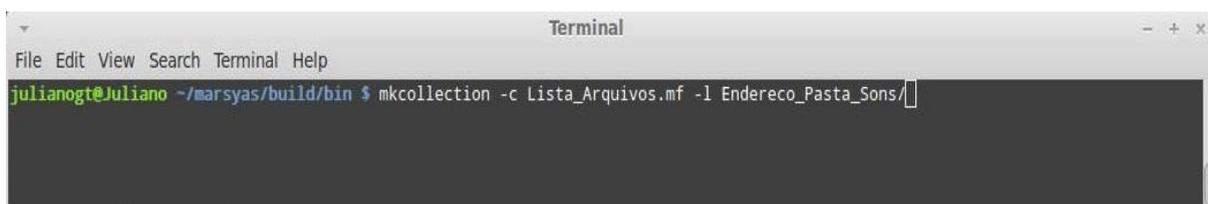
MARSYAS(**M**usic **A**nalysis **R**etrieval and **S**ynthesis for **A**udio **S**ignals), é um *framework* livre para análise, síntese e recuperação de áudio digital[14]. Escrito inicialmente por George Tzanetaki.

Atualmente é trabalhado por uma comunidade de desenvolvedores no mundo todo. Ele está sendo desenvolvido desde 1998 e tem sido usado para uma variedade de projetos, tanto na academia como na indústria[14].

A intenção do MARSYAS é ser um software suficientemente extensível que permita a construção de aplicativos para análise, síntese ou extração de características (ou vetores de características) obtendo alto desempenho e grande eficiência no processamento dos dados. Ele é composto de diversos módulos, cada um capaz de realizar, sozinho, algumas das tarefas mais comuns na análise e síntese de áudio, com ênfase em sinais musicais e MIR²[25].

4.2 Montando listas de Arquivos de Áudio

Para melhor aplicação de algoritmos, o MARSYAS permite criar uma coleção de arquivos de áudio a serem processados. A lista gerada tem extensão ".mf", e é construída através do executável *mkcollection*. A Figura 5 a seguir mostra um exemplo de linha de comando no terminal do Linux *Mint* 13, considerando estar no diretório que contém o executável *mkcollection*:

A screenshot of a Linux terminal window titled "Terminal". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal prompt shows the user "julianogt@Juliano" in the directory "~/marsyas/build/bin". The command entered is "mkcollection -c Lista_Arquivos.mf -l Endereco_Pasta_Sons/". The cursor is at the end of the command line.

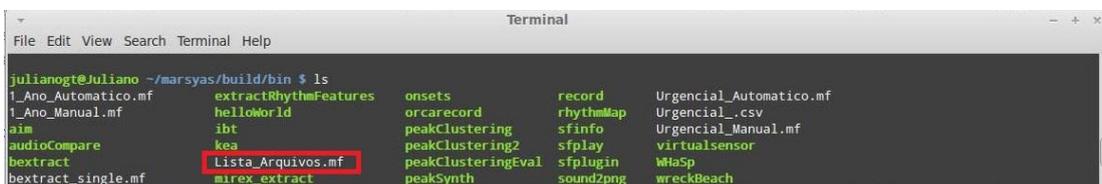
```
Terminal
File Edit View Search Terminal Help
julianogt@Juliano ~/marsyas/build/bin $ mkcollection -c Lista_Arquivos.mf -l Endereco_Pasta_Sons/
```

Figura 4.1: Exemplo de comando com *mkcollection*

² *Music Information Retrieval*(Recuperação de informação de música)

O parâmetro `-c` significa *collection*, que deve ser precedido do nome da lista de arquivos a ser criada. No exemplo foi utilizado como nome da lista de arquivos de áudio a ser criada "Lista_Arquivos.mf". O parâmetro `-l`, significa *linking*. Após o `-l` se especifica o endereço que contém os arquivos de áudio. No exemplo foi utilizado "Endereco_Pastas_Som".

Após a execução do comando demonstrado na Figura 4.1, a lista de arquivos é gerada no diretório em que está contido o *mkcollection*, como demonstrado na Figura 4.2 a seguir.



```

juliannot@Juliano ~/marsyas/build/bin $ ls
1_Ano_Automatico.mf  extractRhythmFeatures  onsets                record                Urgencial_Automatico.mf
1_Ano_Manual.mf     helloWorld              orcarecord            rhythmMap             Urgencial_csv
aim                 ibt                     peakClustering        sfinfo                Urgencial_Manual.mf
audioCompare        kea                     peakClustering2      sfplay                virtualsensor
bextract            Lista_Arquivos.mf      peakClusteringEval   sfplugin              WlaSp
bextract_single.mf  mirex_extract          peakSynth            soundzpng             wreckBeach
  
```

Figura 4.2: Arquivo gerado após a execução do executável *mkcollection*

Ao executar essa linha de comando, dado determinado endereço, todos os subdiretórios que contêm arquivos de extensão ".wav" ou ".au" serão inseridos na coleção de arquivos. É importante verificar a permissão de administrador, é recomendado a utilização de *sudo* no começo da linha de comando, que dá privilégio de administrador do sistema.

4.3 Extração de Características

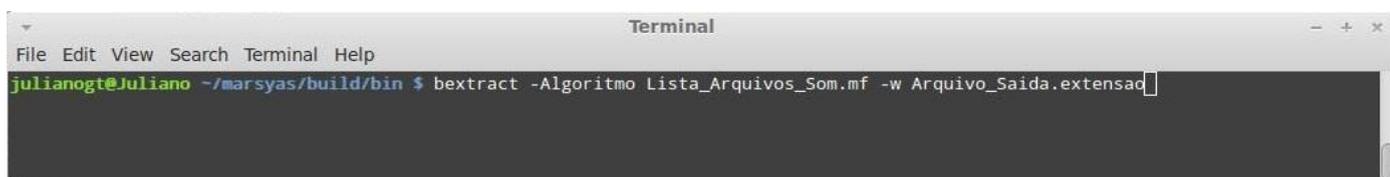
A extração de características no MARSYAS é feita através dos algoritmos disponibilizados nesse *framework*. A escolha das características é importante pois elas têm grande impacto na eficiência do processo de classificação[20].

Através da extração de características, o espaço de dados é transformado em um espaço de características que possui a mesma dimensão do espaço de dados original, porém é representado por um número reduzido de características efetivas [16].

A extração de características como representações de segmentos do sinal de áudio é uma forma de representação com quantidade de informação reduzida. Esta informação é armazenada em vetores de características, que são formados por valores numéricos.[20]

Para ser executada a extração de dados no MARSYAS é utilizado o executável *bextract*. Ele pode ser usado para experimentos completos de extração de características e classificação com vários arquivos[15].

Na execução do comando, é possível anexar na linha de execução, diferentes coleções de arquivos de áudio, definir qual algoritmo de extração de características será aplicado, e qual nome e extensão do arquivos onde os valores serão armazenados .A Figura 4.3 a seguir mostra o exemplo da linha de execução utilizando *bextract* no terminal do Linux Mint 13 "Maya":



```

Terminal
File Edit View Search Terminal Help
julianogt@Juliano ~/marsyas/build/bin $ bextract -Algoritmo Lista_Arquivos_Som.mf -w Arquivo_Saida.extensao

```

Figura 4.3: Exemplo da linha de comando com *bextract* .

Após a execução do comando citado na Figura 4.3, o arquivo é gerado no diretório que contém o executável *bextract*. O arquivo gerado, caso não seja especificada sua extensão, tem extensão ".arff". A Figura 4.4 exemplifica o arquivo gerado após a execução do comando citado anteriormente.



```

Terminal
File Edit View Search Terminal Help
1_Ano_Manual.mf          extractMoodFeatures      mkcollection              peakSynth                 sfplay                    Urgencial_.csv
aim                     extractRhythmFeatures    mudbox                    phasevocoder             sfplugin                  Urgencial_Manual.mf
audioCompare            helloWorld               music.mf                  pitchdtw                 sound2png                 Valores_Gerados.arff
bextract                ibt                      onsets                   pitchdtw_pair            sound2sound               virtualsensor
bextract_single.mf     kea                     orcarecord                pitchextract              speakerSeg                 WHaSp
Emergencial_Automatico.mf Lista_Arquivos.mf       peakClustering           record                    speakerSeg2               wreckBeach
Emergencial_Manual.mf  mirex_extract            peakClustering2          rhythmMap                 tempo
julianogt@Juliano ~/marsyas/build/bin $

```

Figura 4.4: Arquivo gerado após a utilização do executável *bextract*

4.3.1 Extraíndo Coeficientes LPCC (*Linear Predictive Coding Ceptrum*)

Coeficientes LPCC são extraídos através da execução da codificação preditiva linear (LPC).

A codificação preditiva linear (LPC) é definida como um método de se predizer a nova resposta do sinal de saída de um sistema, a partir das amostras passadas do sinal, ou seja, é a tendência de se obter o comportamento de fenômenos futuros em função de amostras passadas.

A Figura 4.5 a seguir exemplifica os passos da execução do LPC.

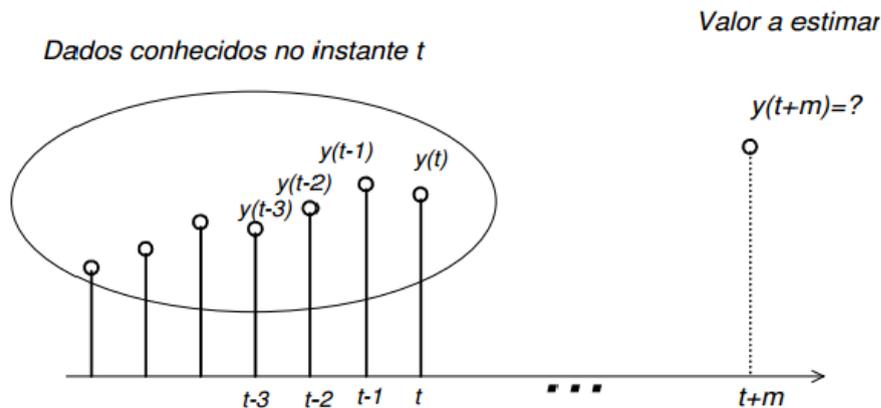


Figura 4.5: Ilustração do Processo LPC

Com base nas amostras anteriores: $y(t-3)$, $y(t-2)$, $y(t-1)$, $y(t)$ aonde t é uma faixa de tempo qualquer e $m > 1$, estima-se $y(t+m)$.

A equação básica de predição linear é dada por [2] :

$$\tilde{s}(n) = \sum_{k=1}^p \alpha_k s(n-k) \quad (6)$$

Onde $\tilde{s}(n)$ é a amostra calculada da amostra real $s(n)$ a partir de uma combinação linear de p amostras com α_k como os coeficientes e $k = 1, 2, 3, \dots, p$. Tem-se ainda a equação 6:

$$E = \sum_n e^2(n) \quad (6)$$

Onde E é o erro médio de predição de curto tempo, e $e(n) = s(n) - \tilde{s}(n)$ é o erro individual.

A extração dos coeficientes LPCC de uma lista de arquivos de áudio no MARSYAS é feita pela seguinte linha de execução ilustrada na Figura 8:

```

Terminal
File Edit View Search Terminal Help
juliano@Juliano ~/marsyas/build/bin $ bextract -lpcc Lista_Arquivos_Som.mf -w Arquivo_Saida.extensao

```

Figura 4.6: Exemplo de comando para extração de coeficientes lpcc

O executável *bextract* extrai dados de determinado arquivo de áudio ou lista de arquivos de áudio. O parâmetro `-lpcc` indica que serão extraídos coeficientes `lpcc`. A lista de arquivos de áudio no exemplo tem nome de "Lista_Arquivos_Som.mf". O parâmetro `-w` (*write*) indica escrita em arquivo, deve ser precedido do arquivo de saída que serão armazenados os valores extraídos. No exemplo citado na Figura 4.6, o arquivo de saída é "Arquivo_Saida.extensao".

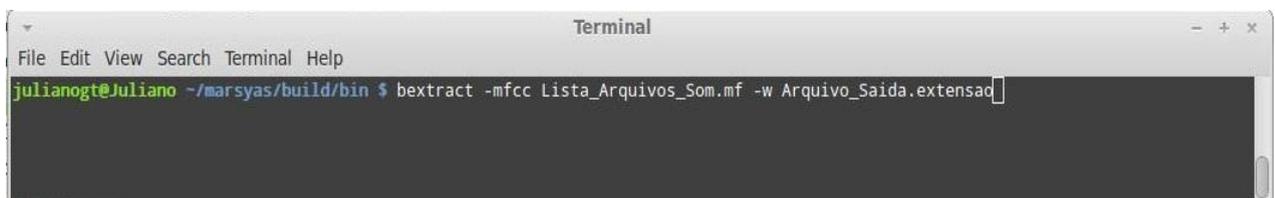
4.3.2 MFCC (*Mel-Frequency Cepstral Coefficients*)

Especialmente MFCC, ou *Mel Frequency Cepstral Coefficients*, têm sido usados mais comumente para aplicações de tratamento de voz, mas também apresentaram grande utilidade na área musical, auxiliando aplicações de compressão de dados [24].

MFCC é uma escala perceptual de *pitches* (tons) baseada na escala Mel[25]. A escala Mel destina-se a realizar medidas de som. No entanto, diferentemente do Decibel, que mensura taxas como força (potência) ou intensidade do sinal, a escala Mel baseia-se na correlação psicológica entre a frequência da nota ouvida e sua frequência fundamental, ou seja, o *pitch*[25]. De maneira simples, o *pitch*, é a medida da resposta humana ao estímulo recebido por um som, representando a relação entre a faixa de frequência em que esse som se encontra e a sua frequência fundamental. A medida dessa distância é que define exatamente o *pitch* do sinal[25].

A ideia do MFCC é distribuir os coeficientes cepstrais de acordo com as faixas críticas, em vez da tradicional distribuição linear[26].

A extração dos coeficientes MFCC de uma lista de arquivos de áudio através do MARSYAS é feita pela seguinte linha de execução demonstrada na Figura 9:



```

Terminal
File Edit View Search Terminal Help
julianogt@Juliano ~/marsyas/build/bin $ bextract -mfcc Lista_Arquivos_Som.mf -w Arquivo_Saida.extensao

```

Figura 4.7 Exemplo de linha comando para extração de coeficientes MFCC

Similar à extração de coeficientes explicadas anteriormente, o executável *bextract* extrai dados de determinado arquivo de áudio ou lista de arquivos de áudio. O parâmetro `-mfcc` indica que serão extraídos coeficientes `mfcc`. A lista de arquivos de áudio no exemplo tem nome de "Lista_Arquivos_Som.mf". O parâmetro `-w` (*write*)

indica escrita em arquivo, deve ser precedido do arquivo de saída que serão armazenados os valores extraídos. No exemplo citado na Figura 4.6, o arquivo de saída é "Arquivo_Saida.extensao".

Capítulo 5

WEKA (*Waikato Environment for Knowledge Analyses*)

5.1 Conceitos

O WEKA é um software criado pela Universidade de Waikato, na Nova Zelândia. O software foi desenvolvido utilizando a plataforma Java. O WEKA é um programa de distribuição gratuita e de código aberto, que possui uma coleção de algoritmos para mineração de dados. Seu propósito inicial é trabalhar com mineração de dados, valendo-se de diversos algoritmos e ferramentas de aprendizagem de máquina para resolução de problemas de associação, classificação, regressão e *clustering*³[25].

Os dois grandes atrativos do WEKA são a sua facilidade de uso, por conta de um conjunto de interfaces gráficas intuitivas e amigáveis, e a possibilidade de o pesquisador aprimorar a ferramenta para servir a propósitos mais específicos (por se tratar de um software de código aberto, permite a realização de alterações por quem esteja disposto a levá-las adiante)[25].

O WEKA utiliza um formato específico de arquivo, que recebe a extensão ".arff", que armazena os dados de uma maneira específica [25].

No WEKA é possível escolher entre quatro interfaces gráficas (*GUI – Graphical User Interface*), que são:

- *Explorer*: Ambiente para exploração de dados (fig. 5.2).
- *Experimenter*: Ambiente para experimentos e testes estatísticos (fig. 5.3).
- *KnowledgeFlow*: Possui as mesmas funções do Explorer, porém com uma interface de arrastar e soltar (fig. 5.4).
- *Simple CLI*: Fornece uma interface de linha de comando simples, que permite a execução direta de todas as funções do WEKA, e com um menor custo de memória do que o Explorer (fig. 5.5).

³ É uma técnica de mineração de dados para fazer agrupamentos automáticos de dados segundo seu grau de semelhança



Figura 5.1: Tela inicial do WEKA para seleção de interface

Neste trabalho a interface utilizada para mineração de dados foi a *Explorer*. Ela foi escolhida por ser uma interface fácil e intuitiva, que reúne os principais pacotes de filtros, classificações, clusterizações, associações do WEKA.

Nesta GUI, o WEKA possui seis abas diferentes: *Preprocess*, *Classify*, *Cluster*, *Associate*, *Select Attributes*, *Vizualize*.

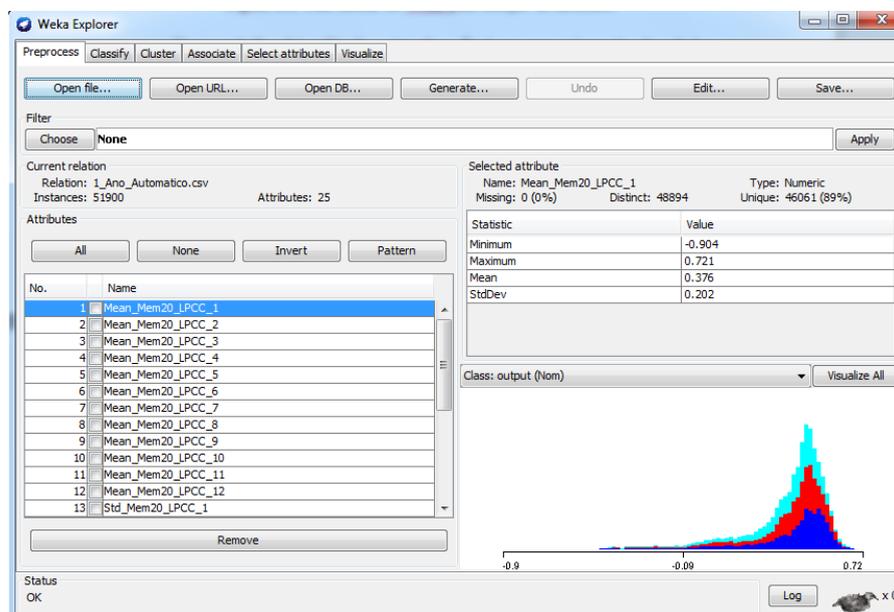


Figura 5.2: GUI *Explorer* do WEKA

As abas relevantes a este trabalho são a *Classify* para a utilização do método *K*-vizinhos mais próximo (*K-Nearest Neighbors – KNN*), que no WEKA é conhecido como

IBk, e a aba *Cluster* para o método de clusterização K-Médias (*K-Means*) que no WEKA é chamado de *SimpleKMeans*.

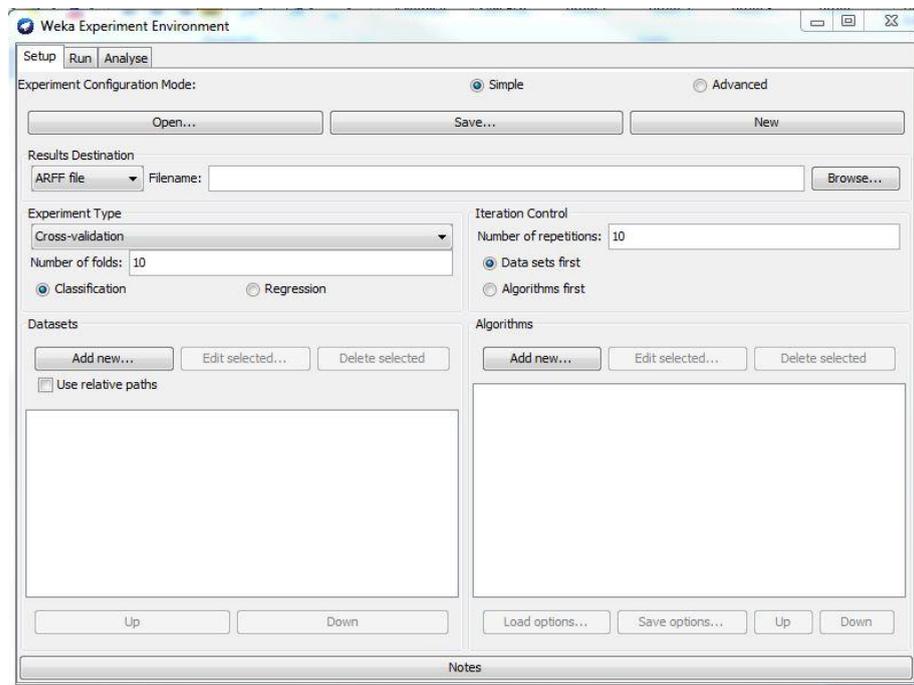


Figura 5.3: GUI *Experimenter* do WEKA

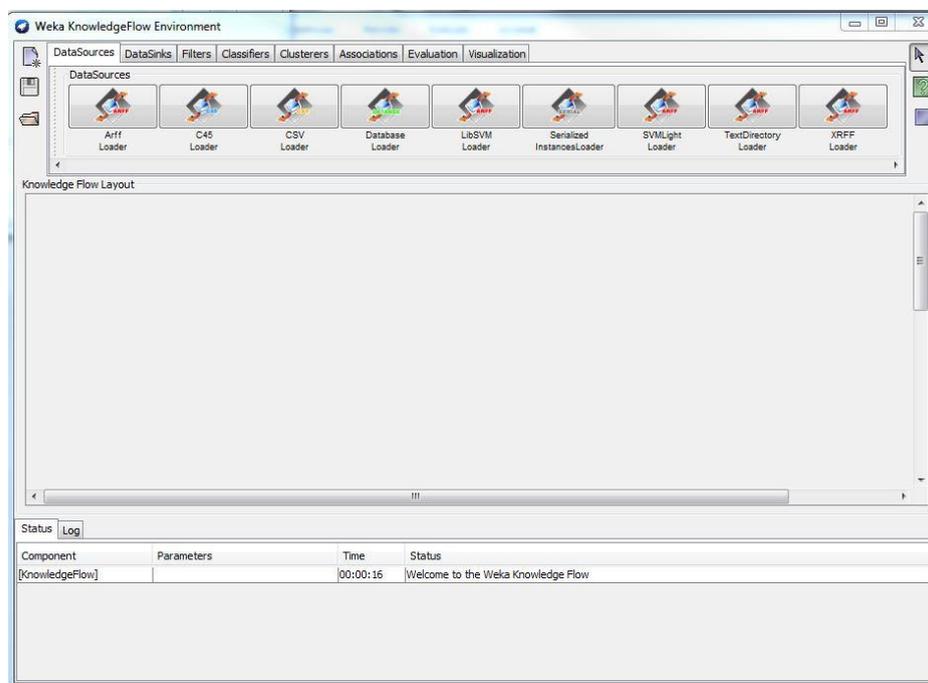


Figura 5.4: GUI *KnowledgeFlow* do WEKA

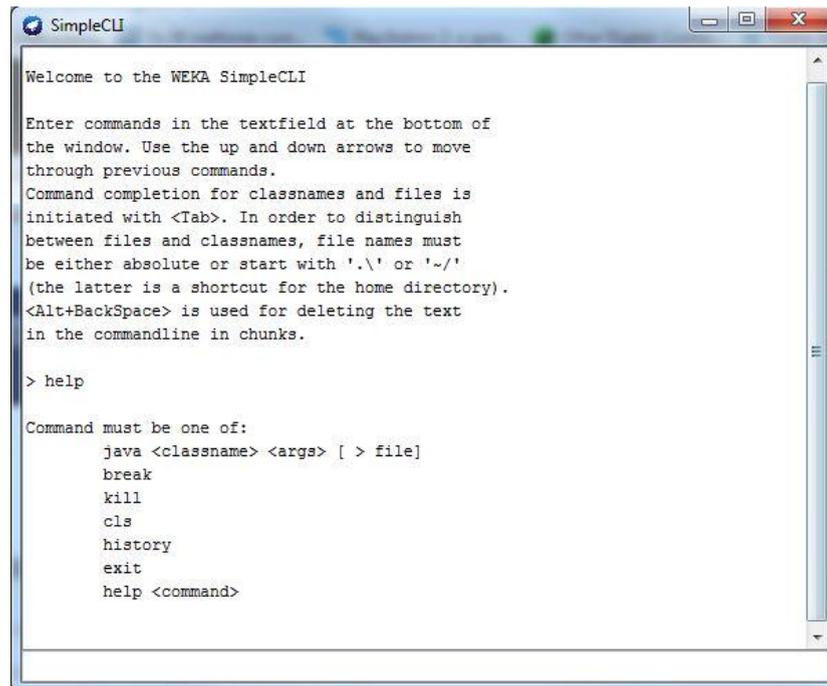


Figura 5.5: GUI *SimpleCLI* do WEKA

5.2 K-Vizinhos mais Próximo (KNN)

O KNN é um dos algoritmos de classificação mais simples, usado para classificar objetos com base em sua proximidade no espaço de características [16].

“O KNN possui um parâmetro chamado K, que indica o número de vizinhos que serão usados pelo algoritmo durante a fase de teste. O parâmetro K faz com que o algoritmo consiga uma classificação mais refinada, porém o valor ótimo de K varia de um problema para o outro, o que faz com que, para cada base de dados, sejam testados vários valores diferentes de forma a descobrir qual o melhor valor de K para determinado problema.” [17]

O funcionamento do algoritmo consiste em calcular a distância entre o elemento desconhecido e os demais elementos da amostra. A forma mais comum do cálculo da distância é pelo método Euclidiano.

A distância Euclidiana é dada por:

$$d(x, y) = \sqrt{\sum_{j=1}^n (x_j - y_j)^2} \quad (7)$$

Depois de obter a distância são identificados os k vizinhos mais próximos. O valor de k é sempre um número ímpar, para que não haja empate na determinação da classe a que o elemento pertence.

Então o elemento desconhecido é rotulado à classe de acordo com a maioria de seus k vizinhos. A Figura 5.6 mostra o funcionamento do algoritmo para classificar o elemento x_d entre as classes “+” e “*”, k assume os valores 1, 3 e 5.

Se $k=1$, então o elemento será classificado como “+”. Para $k=3$, o elemento será classificado “+”, e se $k=5$ será classificado como “*”.

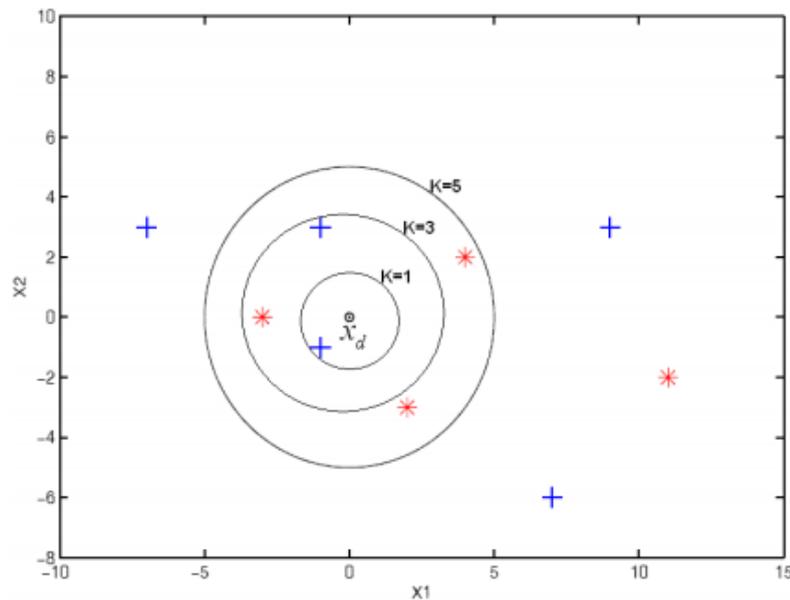


Figura 5.6: Classificação feita pelo KNN para um padrão desconhecido x_d [17].

5.3 K-Means

O *K-Means* (ou K-Médias) é um algoritmo simples e popular que tem como objetivo a criação de agrupamentos (clusters) de objetos de modo que os objetos de um grupo sejam similares entre si, utilizando a ideia de centroides como representantes de grupos [18].

Os passos para execução do algoritmo são[19]:

1. Escolher k centroides de forma aleatória;
2. Atribuir cada elemento da amostra ao grupo que possua o centro mais próximo (assim como no KNN, o cálculo da distância mais utilizado é o Euclidiano, mas pode ser utilizado qualquer outro método);
3. Recalcular o valor do centroide (média) de cada grupo, dos objetos que se encontram no grupo;

4. Repetir os passos 2 e 3 até que os grupos se estabilizem.

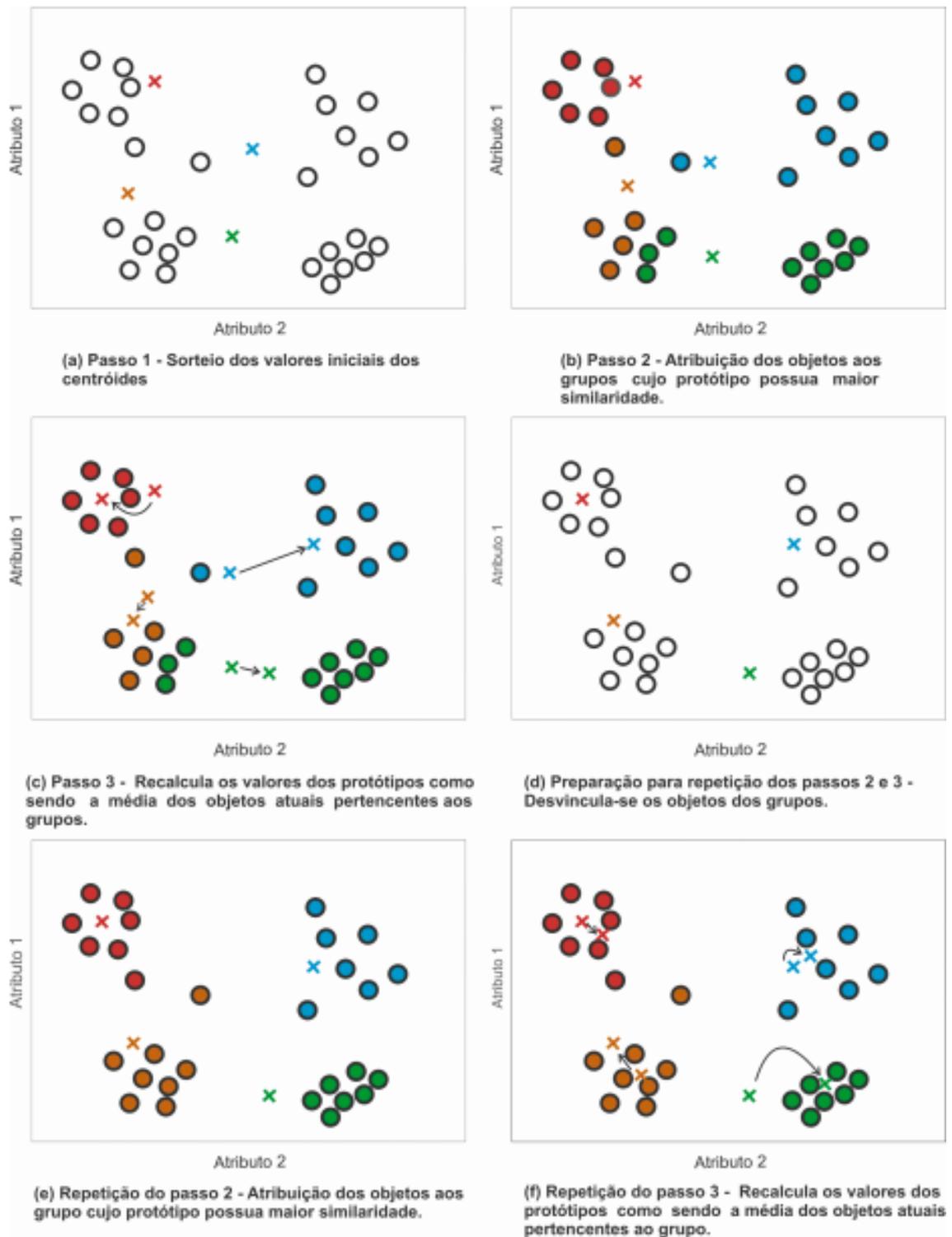


Figura 5.7: Exemplo de Execução do algoritmo K-Means [18]

Capítulo 6

Metodologia

6.1 Ambiente de desenvolvimento

Os algoritmos de pré-processamento de sinal foram implementados na linguagem C, utilizando o compilador GCC versão 4.7.2, no sistema operacional Windows 8.1.

A aplicação dos algoritmos de extração de características no MARSYAS e testes no WEKA foram realizadas no linux Mint 13 "Maya", em um computador com processador Core i7 de 1.8 GHz com picos de até 3 GHz, e 8 gb de memória RAM.

6.2 Pré Processamento de Sinal

Os algoritmos de pré-processamento foram aplicados aos diretórios que continham os arquivos de áudio de extensão ".wav". Inicialmente foram armazenados em arquivos de extensão ".txt" todos os valores correspondentes a determinada sanidade. Para unir os valores de determinada sanidade a um respectivo arquivo, foi utilizada a estrutura de dados Arvore AVL.

O motivo da utilização da Arvore AVL foi devido a sua complexidade na inserção de elementos ($O(\log n)$) lidos dos arquivos de som, visto que não poderia existir valores repetidos .

O processo de normalização por padronização foi aplicado a cada elemento da árvore AVL e posteriormente armazenada em um arquivo de extensão ".txt".

6.3 Aplicação do MARSYAS

A extração de coeficientes LPCC e MFCC foram aplicadas no MARSYAS, com comandos vistos no Capítulo 4. Inicialmente foram montadas as coleções da dados de extensão ".mf" de nome "Urgencial.mf". "Emergencial.mf" e "1_Ano.mf", respectivas a captação da batida da madeira através do martelo comum e automático. Após terminada a construção das listas de arquivo de áudio, foi aplicada a extração de coeficientes LPCC e MFCC.

Os coeficientes foram armazenados em arquivos de extensão ".arff" para posterior utilização na plataforma WEKA.

6.4 Testes na plataforma WEKA

O WEKA foi utilizado para classificação dos dados extraídos no MARSYAS, com os algoritmos KNN e *K-Means*.

A entrada é um arquivo de extensão “.arff” que contém as características extraídas (LPCC ou MFCC) das três classes de madeira (1 Ano, Emergencial, Urgencial).

Para testes de classificação das instâncias com o KNN, utilizou – se o método IBK, disponível na aba de classificação (*Classify*) da GUI Explorer, com a técnica de validação cruzada (*Cross-validation*) com 10 blocos, como opção de testes[25].

A validação cruzada particiona o conjunto de treinamento em partes iguais e faz com que o classificador utilize uma parte para treinamento e o restante para teste[27].

Já para os testes com o *K-Means* o método utilizado foi o SimpleKmeans, que pode ser encontrado na aba de *Cluster*. O tipo de *cluster* utilizado foi Classes para avaliação de clusters (*Classes to clusters evaluation*).

Capítulo 7

Resultados

7.1 Classificação com KNN

Os resultados obtidos na plataforma WEKA, através da execução dos algoritmos KNN e *K-means* nos dados extraídos do MARSYAS, mostram os seguintes resultados através das Tabelas 1, 2,3 e 4.

Os arquivos que contém as características extraídas com LPCC e MFCC utilizados nos testes, possuem 51900 instâncias cada, tanto para a pasta de arquivos automáticos quanto para a pasta de arquivos manuais.

A tabelas 1 mostra os resultados dos treinamentos do KNN para a pasta de arquivos de som obtidos através da batida do martelo comum.

Tabela 1. Resultados dos treinamentos do KNN para pasta Manual

	Instâncias Classificadas corretamente		Tempo de Execução	
	LPCC	MFCC	LPCC	MFCC
K = 3	46526 (89,65%)	51715 (99,64%)	6:12	14:47
K = 5	45613 (87,89%)	51659 (99,54%)	6:51	16:27
K = 7	44815 (86,35%)	51593 (99,41%)	6:55	17:42

Abaixo na Figura 7.1 é possível ver a matriz de confusão gerada para a classificação das instâncias extraídas através dos dois métodos LPCC (a) e MFCC (b) no melhor caso, onde $k=3$. Através da diagonal principal da matriz, é possível determinar o número de instâncias que foram classificadas de forma correta.

```

=== Confusion Matrix ===
      a   b   c  <-- classified as
15669  855  776 |  a = 1_Ano_Manual
  894 15428  978 |  b = Emergencial_Manual
  814  1057 15429 |  c = Urgencial_Manual
  
```

(a)

```

=== Confusion Matrix ===
      a   b   c  <-- classified as
17240   29   31 |  a = 1_Ano_Manual
   23 17237   40 |  b = Emergencial_Manual
   40   22 17238 |  c = Urgencial_Manual
  
```

(b)

Figura 7.1: Matriz de confusão LPCC (a) e MFCC (b) para pasta manual

O gráfico da Figura 7.2 nos mostra um comparativo entre a precisão no número de instâncias classificadas para cada uma das classes. Já o gráfico da Figura 7.3 traz a comparação do tempo gasto na execução da classificação das instâncias.

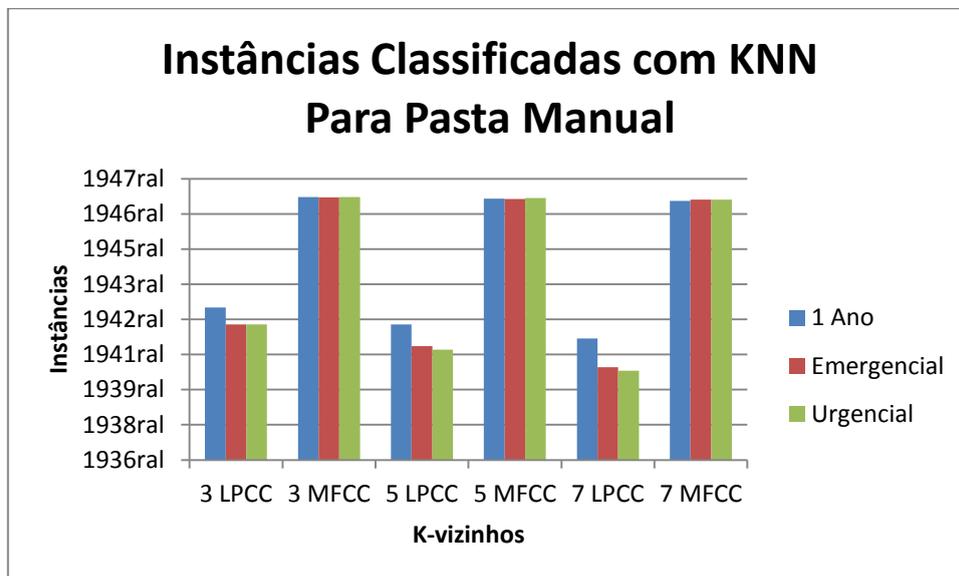


Figura 7.2: Comparação da classificação de instâncias extraídas com LPCC e MFCC para a pasta Manual

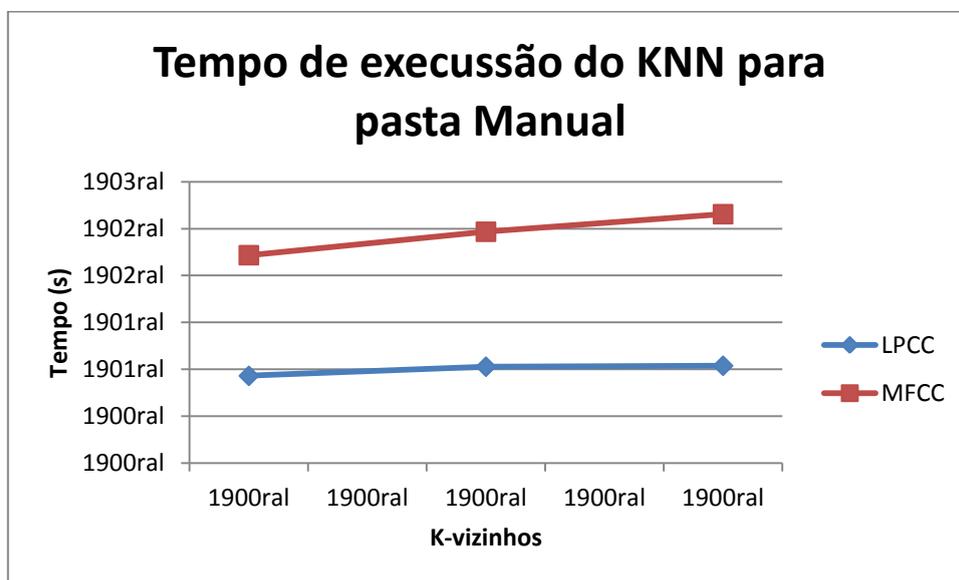


Figura 7.3 : Tempo de execução da classificação das instâncias extraídas com LPCC e MFCC para a pasta Manual.

A tabela 2 mostra os resultados dos treinamentos do KNN para a pasta de arquivos de som obtidos através da batida do martelo automático.

Tabela 2. Resultados dos treinamentos do KNN para pasta Automática

	Instâncias Classificadas corretamente		Tempo de Execução	
	LPCC	MFCC	LPCC	MFCC
K = 3	43396 (83,61%)	51655 (99,59%)	4:35	12:21
K = 5	42497 (81,88%)	51622 (99,46%)	5:08	14:52
K = 7	41590 (80,14%)	51535 (99,30%)	6:18	16:28

```

=== Confusion Matrix ===
      a      b      c  <-- classified as
14463  1474  1363 |   a = 1_Ano_Automatico
1548  14313  1439 |   b = Emergencial_Automatico
1286   1394 14620 |   c = Urgencial_Automatico
(a)

```

```

=== Confusion Matrix ===
      a      b      c  <-- classified as
17222   36    42 |   a = 1_Ano_Automatico
29  17235   36 |   b = Emergencial_Automatico
35    36 17229 |   c = Urgencial_Automatico
(b)

```

Figura 7.4: Matriz de confusão LPCC (a) e MFCC (b) para pasta automática

Os gráficos das Figuras 7.6 e 7.7 mostram respectivamente o comparativo entre o número de instâncias classificadas de forma correta e o tempo gasto para classificação destas instâncias.

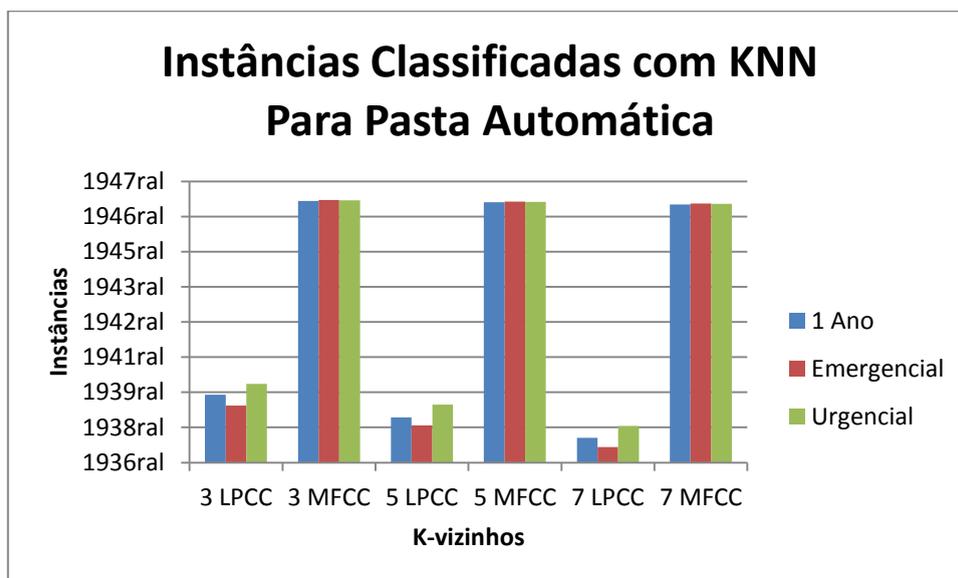


Figura 7.5: Comparação da classificação de instâncias extraídas com LPCC e MFCC para a pasta Automática

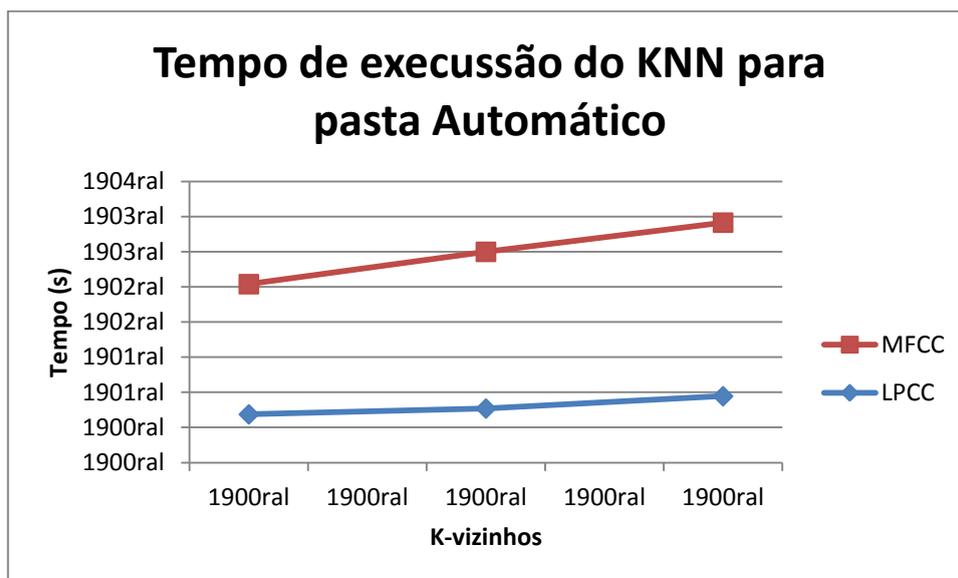


Figura 7.6: Tempo de execução da classificação das instâncias extraídas com LPCC e MFCC para a pasta Automática.

Ao classificarmos as características extraídas através das técnicas LPCC e MFCC com o método do KNN, observou - se que os melhores resultados foram obtidos com o menor número de vizinhos ($k = 3$).

Também é possível notar que embora o tempo de execução do KNN seja menor para a classificação das instâncias geradas com o LPCC, o percentual de instâncias classificadas corretamente é bem superior utilizando as características extraídas pelo MFCC.

7.2 Classificação com K-Means

Como temos 3 classes diferentes, para qualquer valor de $k \neq 3$ estaríamos aumentando as chances de classificar erradamente as instâncias. Por isso para o K-Means foi considerado apenas $k = 3.7$

Tabela 3. Resultados dos treinamentos do K-Means para pasta Manual

	Instâncias Classificadas corretamente		Tempo de Execução	
	LPCC	MFCC	LPCC	MFCC
K = 3	17869 (34,43%)	17578 (33,87%)	0:36	0:43

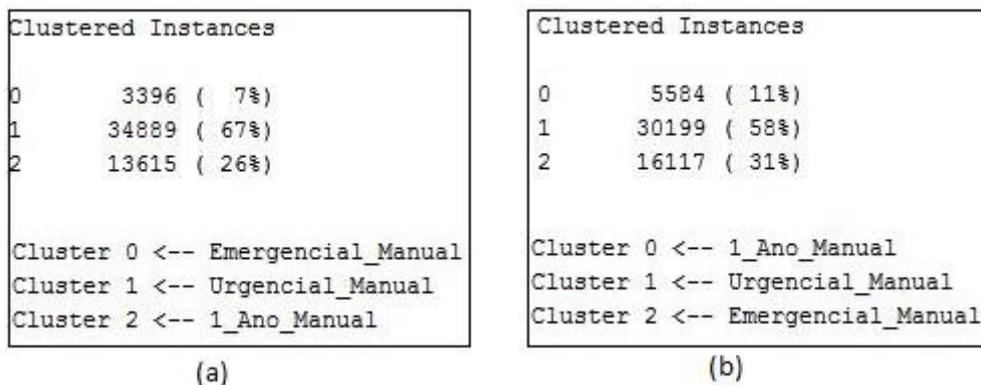


Figura 7.7: Clusterização das instâncias de LPCC (a) e MFCC (b) para pasta Manual

A Figura 7.7 nos mostra como as instâncias foram classificadas, e é possível perceber que a maioria delas, tanto das extraídas com LPCC quanto com MFCC, convergiram para a classe Urgencial.

As Figuras 7.8 (LPCC) e 7.9 (MFCC) mostram os 3 agrupamentos e as instâncias divididas entre eles. Podemos notar que a classe Urgencial (em vermelho) é a que apresenta forma mais densa, com menos espaços entre uma instância e outra, mostrando que possui mais pontos em seu espaço.

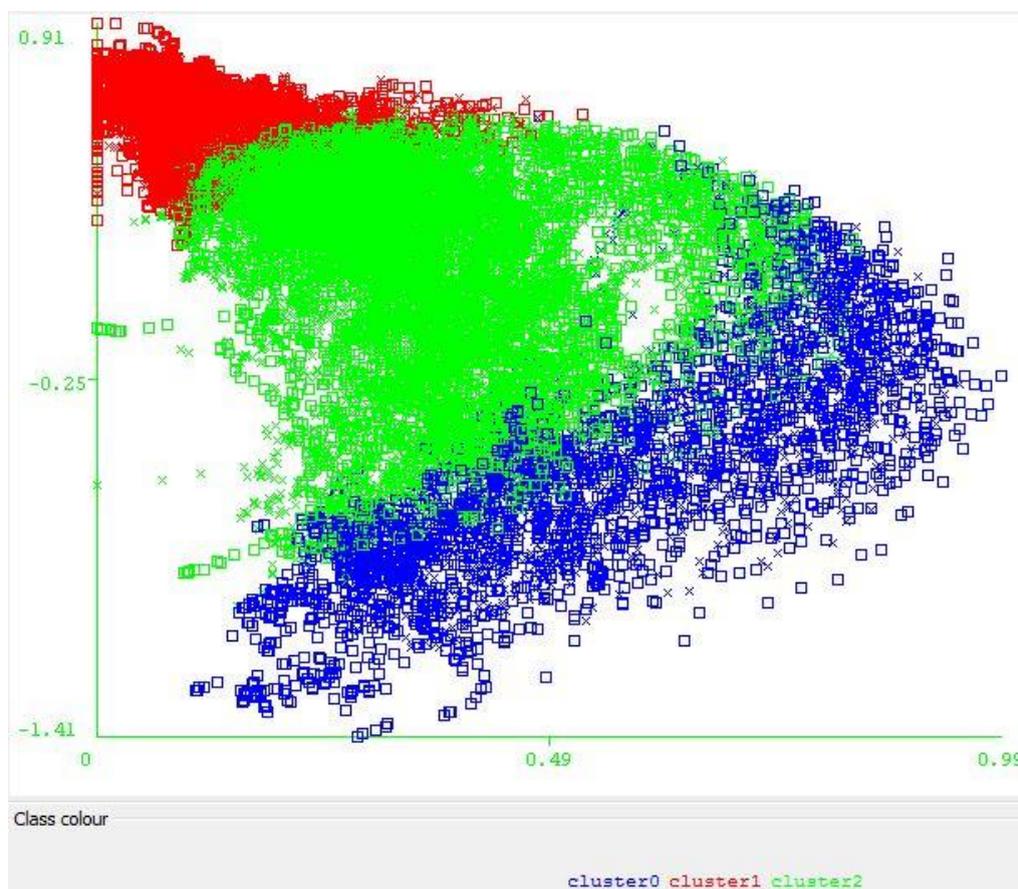


Figura 7.8: Figura : Atribuição de instâncias extraídas com LPCC às classes

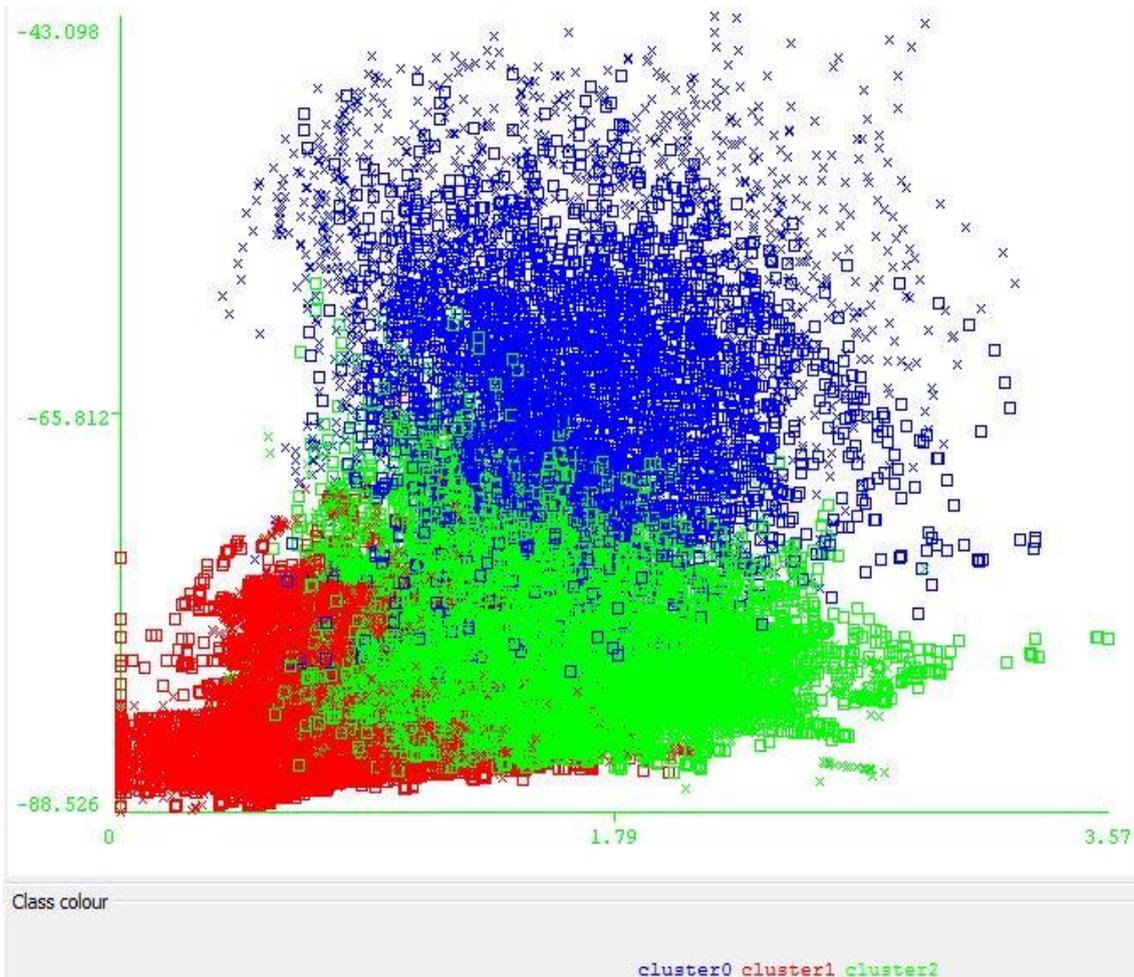


Figura 7.9: Atribuição de instâncias extraídas com MFCC às classes

7

Tabela 4. Resultados dos treinamentos do K-Means para pasta Automática

	Instâncias Classificadas corretamente		Tempo de Execução	
	LPCC	MFCC	LPCC	MFCC
K = 3	17417 (33,56%)	17884 (34,46%)	0:36	0:12

Clustered Instances		Clustered Instances	
0	6641 (13%)	0	8017 (15%)
1	685 (1%)	1	13284 (26%)
2	44574 (86%)	2	30599 (59%)
Cluster 0 <-- Urgencial_Automatico		Cluster 0 <-- 1_Ano_Automatico	
Cluster 1 <-- Emergencial_Automatico		Cluster 1 <-- Emergencial_Automatico	
Cluster 2 <-- 1_Ano_Automatico		Cluster 2 <-- Urgencial_Automatico	

(a)

(b)

Figura 7.10: Clusterização das instâncias de LPCC (a) e MFCC (b) para pasta Automática

Para a pasta automática houve divergência na classificação das características. Enquanto que as instâncias extraídas com LPCC convergiram para agrupamento da classe de 1 ano, as instâncias do MFCC seguiram em sua maioria para o grupo da classe Urgencial.

Nas Figuras 7.11 e 7.12 que representam as características do LPCC e MFCC respectivamente, o grupo dominante aparece na cor verde e podemos fazer a mesma observação feita na seção anterior para as Figuras 7.9 e 7.10. Onde as classes dominantes (1 ano e Urgencial) por possuírem mais instâncias estão menos espaçadas que as demais.

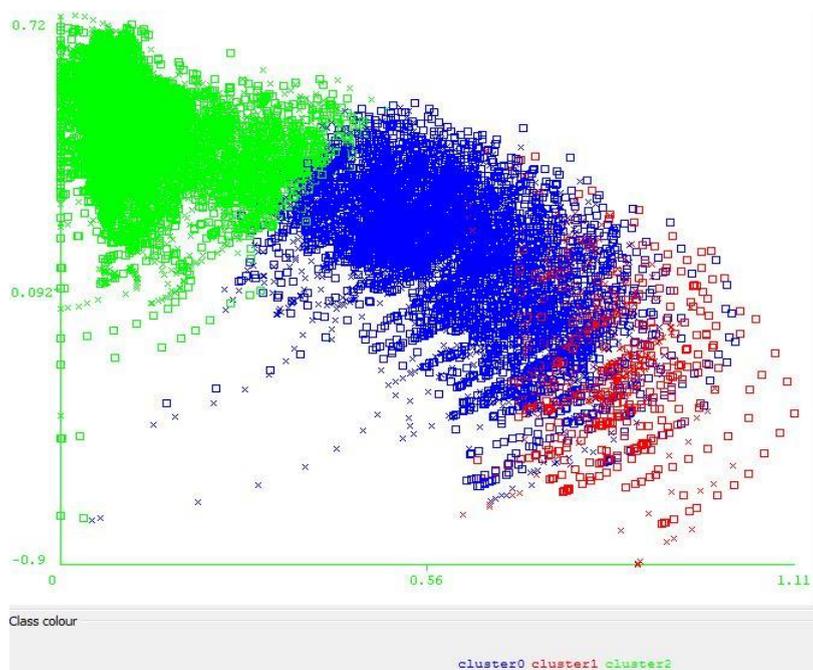


Figura 7.11: Atribuição de instâncias extraídas com LPCC às classes

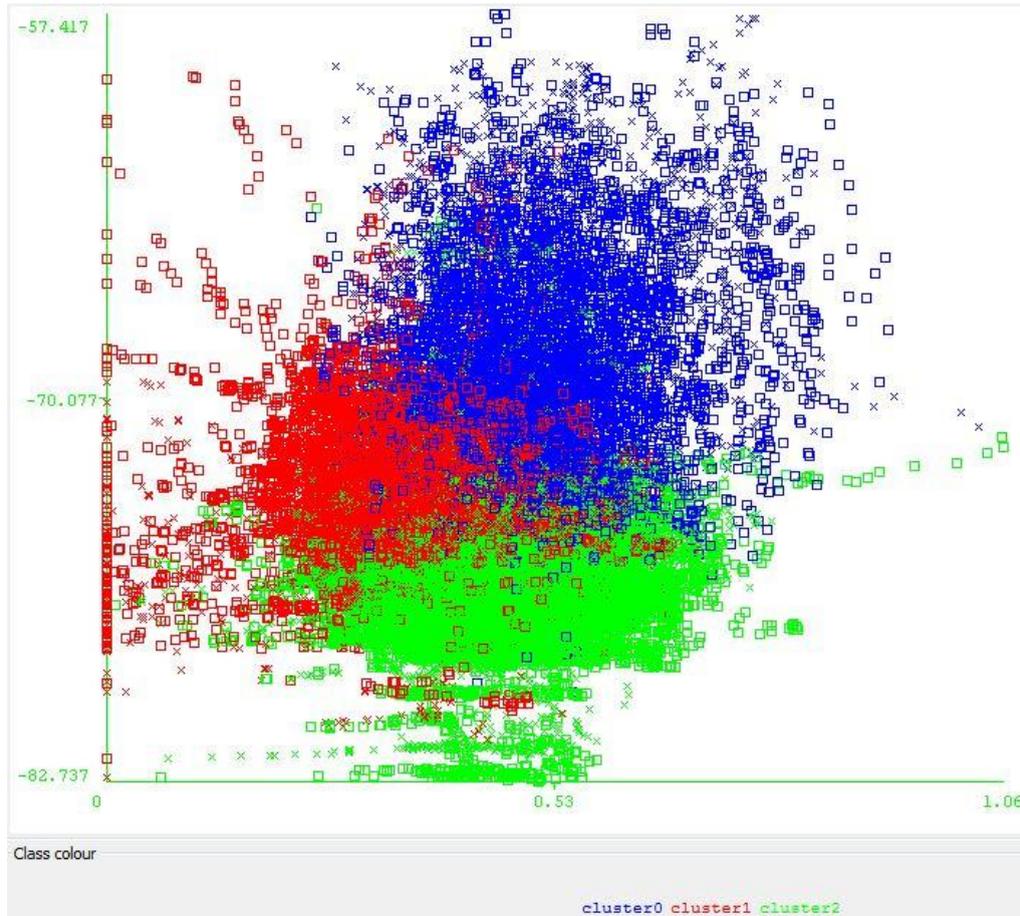


Figura 7.12: Atribuição de instâncias extraídas com MFCC às classes

Embora não haja uma grande diferença no tempo de execução do K-Means para as técnicas do LPCC e MFCC como no KNN, seu desempenho no que diz respeito à classificação das instâncias de maneira correta, é muito inferior à apresentada pelo algoritmo do KNN.

Acreditamos que o baixo desempenho de classificação na utilização do K-Means para este trabalho se deu pela proximidade das instâncias, uma vez estão dentro de um intervalo muito pequeno.

Capítulo 8

Conclusão

A classificação de materiais é imprescindível para o desenvolvimento da humanidade. Nesse sentido existe uma busca por automatização no processo de classificação

Este trabalho apresentou uma alternativa a análise de madeira convencional, descrevendo e aplicando os conceitos de normalização de dados, extração de características, e por fim reconhecimento de padrões através de testes na plataforma WEKA , utilizando os algoritmos KNN e *K-means*.

Para trabalhos futuros recomenda-se:

- Estudo mais aprofundados dos algoritmos estudados e aplicados
- Implementação de uma Rede Neural baseada nos métodos de extração de características do MARSYAS
- Implementação de um software móvel para classificação de madeira utilizando os conceitos estudados nesse trabalho

Referências Bibliográficas

- [1]. Silva, B. M. e Vanderlinde, M.. *Inteligência Artificial, Aprendizado de Máquina*. Artigo, Centro de Educação Superior do Alto Vale do Itajaí.
- [2]. Muniz, D. N.. *Estudo Sobre Reconhecimento de Áudio Repetitivo: Desenvolvimento de um Protótipo*. Monografia, Tecnologia Em Sistemas de Informação, Instituto Federal de Santa Catarina, 2009.
- [3]. Carvalho, R. T. S.. *Estudo Comparativo de Técnicas de Extração de Características para Reconhecimento de Fonemas*. Projeto de Conclusão de Curso, Curso de Graduação em Engenharia de Teleinformática, Departamento de Engenharia de Teleinformática, Universidade Federal do Ceará, 2009 [Acesso em 5 jun 2014]. Disponível em:
http://www.cgeti.ufc.br/monografias/RAPHAEL_TORRES_SANTOS_CARVALHO.pdf
- [4]. Piedade, M. A.. *Introdução à teoria da classificação*. Rio de Janeiro: Interciência, 1977.
- [5]. Júnior, C. C. e Dias, A.A.. *Utilização da Madeira Em Construções Rurais*, Artigo, Agriambi, Revista Brasileira de Engenharia Agrícola e Ambiental, 1997 [Acesso em 10 set 2014]. Disponível em:
<http://www.agriambi.com.br/revista/v1n1/071.pdf>
- [6]. Vicente, N. M. D.. *Sistema de Reconhecimento Sonoro Para Instrumentos De Corda*, Monografia, Engenharia da Computação, Núcleo De Ciências Exatas E Tecnológicas, Centro Universitário Positivo, 2007.
- [7]. Pedrosa, D. P. F.. *Conceitos Básicos de Áudio Digital*. Ciência da Computação, Departamento de Ciências Exatas e Naturais, Universidade Federal do Rio Grande do Norte, 2007 [Acesso em 20 jun 2014]. Disponível em:
http://www2.ufersa.edu.br/portal/view/uploads/setores/164/arquivos/SistemaMultimedia/aula07_audio_digital.pdf
- [8]. Pereira, I. E. D. R. e Campos, R. S.. *Seminário sobre o formato WAVE*, artigo, Escola de Ciência da Informação, Curso de Biblioteconomia, Disciplina de Introdução à Informática, Universidade Federal de Minas Gerais, 2005.
- [9]. Batista, G. E. A.P. A.. *Pré-Processamento de Dados em Aprendizado de Máquina Supervisionado*, Tese de Doutorado, Ciência da Computação, Instituto de Ciências

Matemáticas e de Computação, Universidade de São Paulo, 2003 [Acesso em 22 jun 2014]. Disponível em:

<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-06102003-160219/publico/TeseDoutorado.pdf>

[10]. Zenid, G. J.. *Madeira Na Construção Civil*, artigo, Divisão de Produtos Florestais, Instituto de Pesquisas Tecnológicas do Estado de São Paulo S.A, 2003.

[11]. Costa, R. C. S. *Inspecção Automática de Laranjas Destinadas à Produção de Suco, Utilizando Técnicas de Processamento Digital de Imagens*. Monografia, Curso Superior de Tecnologia em Mecatrônica Industrial, Centro Federal de Educação Tecnológica do Ceará, 2006.

[12] Klock, U..*Características gerais da madeira*. Artigo,Universidade Federal do Paraná, 2006.

[13] Brito, J. O.. *O uso energético da madeira*. Artigo, Portal de Revistas, Sistema Integrado de Bibliotecas, Universidade de São Paulo, 2007.

[14] Site do MARSYAS [Acesso em 20 jul 2014]. Disponível em <http://MARSYAS.info/>

[15] Tzanetakis G., Cook, P..*MARSYAS: a framework for audio analysis. Organised Sound*, 2000 [Acesso em 20 jul 2014].

[16]Chagas, F..*Variações do Método kNN e suas Aplicações na Classificação Automática de Textos*, UEG,2009 [Acesso em 20 jul 2014]. Disponível em :

<http://www.inf.ufg.br/mestrado/sites/www.inf.ufg.br.mestrado/files/uploads/Dissertacoes/Fernando%20Chagas.pdf>

[17] Bezerra, M. E. R. Bez. *Métodos Baseados na Regra do Vizinho mais Próximo para Reconhecimento de Imagens*, 2006. [Acesso em 10 jul 2014]. Disponível em:

<http://tcc.ecomp.poli.br/20052/MiguelEugenio.pdf>

[18] Fontana, A, Naldi. M. C.. *Estudo de Comparação de Métodos para Estimação de Números de Grupos em Problemas de Agrupamento de Dados*. Universidade de São Paulo[Acesso em 7 jul 2014]. Disponível em:

http://www.icmc.usp.br/CMS/Arquivos/arquivos_enviados/BIBLIOTECA_113_RT_340.pdf

[19] Borges, V. R. P.. *Comparação entre as Técnicas de Agrupamento K-Means e Fuzzy C-Means para Segmentação de Imagens Coloridas*. Faculdade de Computação, Universidade Federal de Uberlândia [Acesso em 7 jul 2014]. Disponível em:

http://www.enacomp.com.br/2010/cd/artigos/completos/enacomp2010_42.pdf

[20] Viana, A.. *APOSTILA*, Disciplina Edição de Áudio I, Universidade Federal Do Rio Grande do Norte, 2006. Disponível em:

<http://www.ibam-concursos.org.br/documento/Edicao.pdf>

[21] Andrade, A. O., Soares, A. B. *Técnicas de Janelamento de Sinais*. Artigo, Faculdade de Engenharia Elétrica, Universidade Federal de Uberlândia, 2002 [Acesso em 3 jul 2014]. Disponível em:

<http://www.laps.ufpa.br/valquiria/pds-graduacao/Material%20de%20apoio/Janelamento.pdf>

[22] Nascimento, A. M. D.. *Uso da Madeira Preservada É Vantajoso Economicamente* ?, Artigo, Universidade Federal Rural do Rio de Janeiro, 1996 [Acesso em 19 jul 2014]. Disponível em:

<http://www.floram.org/files/v3n%C3%BAnico/v3nunicoa2.pdf>

[23] Mendes, A.D. S, Alves, M. V. D. S.. *A Degradação Da Madeira e Sua Preservação*, Livro, Instituto Brasileiro de Desenvolvimento Florestal, 1988.

[24] ANAGNOSTOPOULOU, C.; FERRAND, M. e SMAILL, A., *Music and Artificial Intelligence*, Springer, 2002.

[25] Filho, M. D. C. Z.. *Classificação automática de Gêneros de Áudio Digital*, Trabalho de Conclusão de Curso, Engenharia da Computação, Departamento de Sistemas Computacionais, Escola Politécnica de Pernambuco, 2006 [Acesso em 10 set 2014]. Disponível em:

<http://tcc.ecomp.poli.br/20062/MoacirFilho.pdf>

[26] Dufax, A.. *Detection and Recognition of Impulsive Sound Signals*, Tese de Doutorado, Institut de Microtechnique Neuchatel, 2001 [Acesso em 10 set 2014]. Disponível em:

http://lpm.epfl.ch/webdav/site/lpm/users/175547/public/Sound_recognition.pdf

[27] Santos, L. D. M. dos, Mikami, R., Vendramin, A. C. B. K., Kaestner, C. A. A. *Procedimentos de Validação Cruzada em Mineração de Dados para ambiente de Computação Paralela*. Universidade Tecnológica Federal do Paraná (UTFPR), 2009.

[Acesso em 30 set 2014]. Disponível em:

<http://www.lbd.dcc.ufmg.br/colecoes/erad/2009/047.pdf>

Apêndice A

Exemplo de Base de dados do WEKA

O conjunto a seguir exemplifica a estrutura interna dos arquivos ARFF utilizados pelo WEKA durante esse trabalho

% Created by Marsyas

@relation 1_Ano_Automatico.arff

@attribute Mean_Mem20_LPCC_1 real
@attribute Mean_Mem20_LPCC_2 real
@attribute Mean_Mem20_LPCC_3 real
@attribute Mean_Mem20_LPCC_4 real
@attribute Mean_Mem20_LPCC_5 real
@attribute Mean_Mem20_LPCC_6 real
@attribute Mean_Mem20_LPCC_7 real
@attribute Mean_Mem20_LPCC_8 real
@attribute Mean_Mem20_LPCC_9 real
@attribute Mean_Mem20_LPCC_10 real
@attribute Mean_Mem20_LPCC_11 real
@attribute Mean_Mem20_LPCC_12 real
@attribute Std_Mem20_LPCC_1 real
@attribute Std_Mem20_LPCC_2 real
@attribute Std_Mem20_LPCC_3 real
@attribute Std_Mem20_LPCC_4 real
@attribute Std_Mem20_LPCC_5 real
@attribute Std_Mem20_LPCC_6 real
@attribute Std_Mem20_LPCC_7 real
@attribute Std_Mem20_LPCC_8 real
@attribute Std_Mem20_LPCC_9 real
@attribute Std_Mem20_LPCC_10 real
@attribute Std_Mem20_LPCC_11 real
@attribute Std_Mem20_LPCC_12 real

@attribute output

{1_Ano_Automatico,Emergencial_Automatico,Urgencial_Automatico}

@data

% filename

/home/julianogt/Documentos/sanidades/1ano/automatico/a32.wav,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

% srate 86.1328,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

0.553954,0.439875,0.270770,0.115481,-0.030540,-0.118943,-0.203079,-0.330418,-
 0.473079,-0.483606,-0.518018,-
 0.505070,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000
 0,0.000000,0.000000,0.000000,0.000000,1_Ano_Automatico
 0.553843,0.435781,0.268603,0.119583,-0.027104,-0.115995,-0.199709,-0.329872,-
 0.466780,-0.482091,-0.521452,-
 0.512661,0.000483,0.017847,0.009446,0.017879,0.014977,0.012850,0.014689,0.00238
 0,0.027456,0.006607,0.014968,0.033092,1_Ano_Automatico
 0.551141,0.443051,0.276414,0.126271,-0.021143,-0.111644,-0.199920,-0.330194,-
 0.474663,-0.493660,-0.535450,-
 0.525466,0.011764,0.037182,0.035809,0.033386,0.029301,0.022344,0.014765,0.00282
 7,0.045097,0.051204,0.062054,0.063371,1_Ano_Automatico
 0.546423,0.438289,0.274454,0.123796,-0.021206,-0.106164,-0.193147,-0.321621,-
 0.465866,-0.483666,-0.517083,-
 0.504657,0.023126,0.042939,0.037113,0.035838,0.029323,0.031462,0.032351,0.03742
 3,0.059432,0.068706,0.104406,0.114420,1_Ano_Automatico
 0.549184,0.446558,0.285577,0.135995,-0.007288,-0.092432,-0.184008,-0.314333,-
 0.456455,-0.477984,-0.517069,-
 0.509379,0.026856,0.056297,0.060383,0.062522,0.065427,0.064972,0.049519,0.04776
 6,0.071268,0.073038,0.104406,0.116274,1_Ano_Automatico
 0.547715,0.444222,0.282751,0.136410,-0.008106,-0.098129,-0.187623,-0.313146,-
 0.449300,-0.470930,-0.509019,-
 0.498757,0.027354,0.057483,0.062302,0.062412,0.065814,0.071694,0.053276,0.04764
 7,0.076248,0.078746,0.110075,0.125519,1_Ano_Automatico
 0.552098,0.449174,0.293336,0.155612,0.019917,-0.068776,-0.159142,-0.278358,-
 0.407907,-0.431205,-0.471452,-
 0.464154,0.034173,0.061049,0.075873,0.100484,0.134145,0.142440,0.131796,0.15512
 1,0.190786,0.187554,0.195588,0.195108,1_Ano_Automatico
 0.553367,0.452828,0.299590,0.160752,0.027022,-0.062863,-0.157126,-0.267216,-
 0.396188,-0.423129,-0.465707,-
 0.460486,0.034686,0.062553,0.078851,0.100928,0.135045,0.142689,0.131417,0.15893
 8,0.193601,0.188599,0.195823,0.194994,1_Ano_Automatico
 0.550550,0.448719,0.301221,0.167625,0.039739,-0.050327,-0.145764,-0.250054,-
 0.376046,-0.400471,-0.445340,-
 0.440886,0.036750,0.065880,0.078575,0.102282,0.140875,0.148121,0.136672,0.16937
 5,0.205164,0.206357,0.209994,0.208744,1_Ano_Automatico
 0.547346,0.446874,0.297668,0.171099,0.046288,-0.043557,-0.144264,-0.242506,-
 0.359403,-0.387611,-0.434544,-
 0.431736,0.039035,0.066614,0.081426,0.101630,0.140499,0.147925,0.136198,0.16898
 9,0.210060,0.208775,0.211525,0.209739,1_Ano_Automatico
 0.545663,0.448503,0.300206,0.172448,0.047331,-0.044391,-0.144878,-0.235173,-
 0.348079,-0.374939,-0.422616,-
 0.416942,0.039437,0.066821,0.081339,0.101060,0.140001,0.148393,0.136489,0.16819
 5,0.209731,0.210250,0.213201,0.214427,1_Ano_Automatico
 0.541301,0.443987,0.300825,0.178082,0.051273,-0.034640,-0.130767,-0.215967,-
 0.322001,-0.344829,-0.393886,-
 0.389322,0.042947,0.070218,0.081160,0.100868,0.138859,0.149577,0.144119,0.17787
 4,0.224474,0.234280,0.235914,0.235808,1_Ano_Automatico
 0.544006,0.452176,0.305927,0.184700,0.055252,-0.033911,-0.133360,-0.218986,-
 0.330511,-0.363272,-0.416986,-

0.415365,0.045298,0.078342,0.082309,0.100886,0.137592,0.149200,0.145853,0.180287,0.233100,0.257815,0.267450,0.272987,1_Ano_Automatico
0.514765,0.409617,0.267572,0.168941,0.066601,-0.008722,-0.116047,-0.209227,-0.321142,-0.345319,-0.383637,-
0.375764,0.133103,0.203956,0.193452,0.130681,0.139397,0.173297,0.156697,0.179271,0.230937,0.261289,0.293125,0.311792,1_Ano_Automatico
0.486543,0.374249,0.235343,0.150359,0.063053,0.005136,-0.093595,-0.182711,-0.298898,-0.331688,-0.366531,-
0.351368,0.175034,0.251446,0.238648,0.160078,0.142688,0.175003,0.173849,0.197662,0.236587,0.260829,0.294762,0.319708,1_Ano_Automatico
0.473731,0.345073,0.206447,0.126871,0.047652,0.003924,-0.078071,-0.159513,-0.271933,-0.305278,-0.333557,-
0.323902,0.178965,0.274898,0.266026,0.194281,0.166582,0.175939,0.177209,0.206014,0.245754,0.270665,0.312334,0.328791,1_Ano_Automatico
0.461882,0.321991,0.177558,0.108399,0.040317,0.002693,-0.075281,-0.148003,-0.248961,-0.279215,-0.305484,-
0.294786,0.181094,0.285159,0.287941,0.211304,0.172971,0.176879,0.175652,0.202545,0.247346,0.277256,0.319640,0.337136,1_Ano_Automatico
0.454251,0.309416,0.155819,0.088846,0.021895,-0.010975,-0.082947,-0.137925,-0.226251,-0.249269,-0.270650,-
0.258674,0.180267,0.285228,0.296373,0.227237,0.197428,0.195347,0.184200,0.198186,0.246576,0.285776,0.332292,0.351071,1_Ano_Automatico
0.442906,0.296174,0.141562,0.070577,0.005817,-0.021316,-0.086973,-0.133048,-0.206244,-0.221162,-0.242260,-
0.223036,0.180774,0.285012,0.297357,0.238756,0.213485,0.205974,0.187629,0.194556,0.241927,0.288971,0.334194,0.360305,1_Ano_Automatico
0.439945,0.290621,0.135398,0.063181,-0.001018,-0.031556,-0.098498,-0.135335,-0.204848,-0.216421,-0.231434,-
0.207881,0.179411,0.283234,0.295889,0.239540,0.216703,0.215446,0.201009,0.197115,0.240459,0.285382,0.328545,0.354450,1_Ano_Automatico
0.442243,0.295358,0.138781,0.062297,-0.007179,-0.045393,-0.122599,-0.155435,-0.222396,-0.234895,-0.241305,-
0.210206,0.181143,0.286465,0.297798,0.239378,0.219192,0.229070,0.237660,0.233178,0.270343,0.312729,0.339780,0.356539,1_Ano_Automatico
0.436907,0.290245,0.131278,0.053268,-0.014056,-0.050611,-0.125242,-0.140997,-0.207452,-0.208494,-0.210492,-
0.175612,0.179402,0.286214,0.297360,0.237506,0.219818,0.230527,0.238085,0.231506,0.271296,0.315453,0.334971,0.344894,1_Ano_Automatico
0.442893,0.289836,0.129137,0.053151,-0.015776,-0.049049,-0.122870,-0.137588,-0.196741,-0.202011,-0.204672,-
0.177566,0.183358,0.285798,0.295370,0.237409,0.219141,0.230754,0.237491,0.229086,0.258290,0.306173,0.325593,0.348299,1_Ano_Automatico
0.445620,0.288218,0.117730,0.050353,-0.019514,-0.057553,-0.136609,-0.141048,-0.194287,-0.199058,-0.213329,-
0.182732,0.183991,0.285574,0.295599,0.237571,0.220018,0.232263,0.241807,0.229904,0.257556,0.305655,0.326344,0.347711,1_Ano_Automatico
0.441863,0.275810,0.100518,0.029698,-0.040915,-0.076460,-0.148551,-0.143542,-0.199361,-0.200587,-0.205456,-
0.165984,0.181361,0.276785,0.282986,0.227546,0.213699,0.229509,0.241666,0.230632,0.260276,0.306581,0.320756,0.335070,1_Ano_Automatico

0.443345,0.274672,0.093849,0.018478,-0.048092,-0.077134,-0.151833,-0.145043,-
0.200501,-0.203519,-0.203281,-
0.161276,0.182150,0.276347,0.281796,0.228161,0.216176,0.229987,0.243802,0.23178
4,0.260895,0.308201,0.319867,0.333917,1_Ano_Automatico
0.434013,0.264614,0.079639,0.001392,-0.074720,-0.103239,-0.179250,-0.178433,-
0.239034,-0.245474,-0.237172,-
0.194682,0.176446,0.270138,0.268703,0.202894,0.171525,0.193323,0.212831,0.20204
8,0.231266,0.292007,0.307481,0.330812,1_Ano_Automatico
0.428550,0.258951,0.071216,-0.004369,-0.080905,-0.106809,-0.181758,-0.187812,-
0.242451,-0.241727,-0.230498,-
0.185580,0.173529,0.266028,0.261194,0.198236,0.166863,0.192051,0.212917,0.20289
3,0.231740,0.291481,0.305247,0.326643,1_Ano_Automatico
0.431431,0.261169,0.063592,-0.010882,-0.091632,-0.120939,-0.201295,-0.217064,-
0.269130,-0.276866,-0.259489,-
0.208426,0.175123,0.267025,0.256489,0.191708,0.153273,0.184214,0.211043,0.21374
2,0.240912,0.305896,0.319683,0.336600,1_Ano_Automatico
0.432537,0.266122,0.074784,-0.003416,-0.087199,-0.117405,-0.196269,-0.217026,-
0.274246,-0.280885,-0.254927,-
0.208521,0.175558,0.270505,0.266858,0.201830,0.159903,0.187466,0.212844,0.21374
8,0.239200,0.305735,0.319693,0.336632,1_Ano_Automatico
0.433718,0.255199,0.061109,-0.013735,-0.094848,-0.120413,-0.197839,-0.220573,-
0.273176,-0.281621,-0.256868,-
0.208691,0.176223,0.266332,0.260803,0.199369,0.159670,0.188215,0.213095,0.21375
5,0.239369,0.305630,0.319954,0.336633,1_Ano_Automatico
0.429588,0.244464,0.040049,-0.035469,-0.103178,-0.133754,-0.208166,-0.225655,-
0.277878,-0.292409,-0.268533,-
0.222113,0.176369,0.266640,0.259022,0.195469,0.156295,0.183204,0.204242,0.20831
2,0.233865,0.294933,0.312479,0.331469,1_Ano_Automatico
0.429082,0.240343,0.043183,-0.028248,-0.093731,-0.123595,-0.198655,-0.215372,-
0.265879,-0.278342,-0.251963,-
0.207894,0.175870,0.261647,0.263372,0.208063,0.170292,0.190057,0.206262,0.20495
5,0.220590,0.273841,0.281569,0.301420,1_Ano_Automatico
0.459806,0.287138,0.081347,-0.007169,-0.100435,-0.145751,-0.223463,-0.233543,-
0.281646,-0.305586,-0.297425,-
0.257670,0.143534,0.221535,0.236343,0.210962,0.161130,0.151391,0.193071,0.21299
7,0.232397,0.284096,0.286523,0.297769,1_Ano_Automatico
0.488708,0.316301,0.107737,0.006456,-0.101694,-0.166671,-0.251458,-0.261293,-
0.297981,-0.312148,-0.299021,-
0.264878,0.096381,0.181349,0.212290,0.203094,0.161231,0.136074,0.160887,0.18958
4,0.225386,0.283350,0.285930,0.293575,1_Ano_Automatico
0.503255,0.345760,0.134772,0.026818,-0.092035,-0.170976,-0.274526,-0.286551,-
0.319127,-0.331628,-0.328187,-
0.287560,0.088050,0.149257,0.191104,0.185565,0.152290,0.136624,0.139423,0.16768
6,0.209503,0.271291,0.268745,0.286248,1_Ano_Automatico
0.516963,0.371205,0.166282,0.043976,-0.083496,-0.166304,-0.272235,-0.279504,-
0.320373,-0.339846,-0.341981,-
0.300102,0.078843,0.125996,0.165968,0.174335,0.152061,0.139057,0.141850,0.17801
2,0.207749,0.262321,0.256099,0.275274,1_Ano_Automatico
0.516289,0.379767,0.179891,0.063721,-0.064756,-0.146652,-0.262117,-0.280275,-
0.335909,-0.362719,-0.369380,-0.328484,0.079880,0.118901,0.148573,0.158684