ESCOLA POLITÉCNICA
DE PERNAMBUCO

Departamento de
Sistemas
Computacionais

Classificação Automática de Gêneros de Áudio Digital

Trabalho de Conclusão de Curso

Engenharia da Computação

Nome do Aluno: Moacir da Cruz Souza Filho
Orientador: Prof. Dr. Carlos Alexandre Barros de Mello

Recife, Novembro de 2006

UNIVERSIDADE
DE PERNAMBUCO

ESCOLA POLITÉCNICA
DE PERNAMBUCO

Departamento de
Sistemas
Computacionais

Classificação Automática de Gêneros de Áudio Digital

Trabalho de Conclusão de Curso

Engenharia da Computação

Este Projeto é apresentado como requisito parcial para obtenção do diploma de Bacharel em Engenharia da Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Nome do Aluno: Moacir da Cruz Souza Filho
Orientador: Prof. Dr. Carlos Alexandre Barros de Mello

Recife, Novembro de 2006

UNIVERSIDADE
DE PERNAMBUCO

Moacir da Cruz Souza Filho

Classificação Automática de Gêneros de Áudio Digital

Resumo

Diversos esforços têm sido realizados, nos dias de hoje, no sentido de encontrar uma forma confiável e eficiente para classificar automaticamente o crescente volume de dados de áudio digital. É provável que, no futuro, a maior parte, ou todo o conteúdo de áudio produzido pelo homem, esteja disponível na Internet. A classificação automática desses dados estará cada vez mais presente na rotina da maioria dos usuários de computadores, já que o áudio digital é um dos grandes meios de entretenimento atual. Além disso, os sistemas MIR (*Music Information Retrieval*) atuais encontram-se em constante evolução e uma das grandes preocupações dos pesquisadores está em descobrir uma maneira ótima de classificar automaticamente seus dados.

No entanto, a maior parte dessas classificações ainda é feita manualmente, necessitando da presença de um especialista. Esse processo é lento e extremamente custoso, além de gerar uma diversidade de inconsistências e problemas relacionados a padronização.

Dentre as inúmeras maneiras de classificar um sinal de áudio (de acordo com a época, a área geográfica na qual é apreciado, etc.), o *Gênero Musical* ao qual ele pertence é de peculiar importância para os estudos de MIR. No presente trabalho, esse indicador será usado como base para a classificação dos sinais de entrada. Gêneros musicais são rótulos utilizados para organizar peças musicais que compartilham alguns critérios pré-definidos, estilos, ou “linguagens musicais” similares [1].

O estudo propõe-se a estudar uma combinação entre duas técnicas existentes de classificação automática de gêneros musicais e gerar uma nova abordagem, na intenção de obter resultados mais significativos e gerais que as primeiras. Para esses fins são utilizados majoritariamente dois *softwares*, o MARSYAS [2], para extração dos dados para análise, e o WEKA [3], para simulação das classificações.

Abstract

Several efforts have been made trying to obtain a reliable and efficient way to automatically classify the raising volume of digital audio data in the world. It's probably true that in a near future, almost all, or even all the music produced by humans will be available on the Internet. Thus, automatic audio classifiers will be more and more present in users' life, since digital music is already one of the most popular entertaining media. Besides that, current MIR (Music Information Retrieval) systems are constantly evolving and one of the researcher's worries relies, exactly, in how to automatically classify these data.

However, most part of these classifications are still made manually and they often require a specialist assistance. This process is extremely time expensive and usually creates a lot of inconsistencies and problems with standards.

Amongst the various ways to classify music audio signals (whether according to the time it is/was played, the geographical area where it is frequently listened, etc), its Genre have a great importance to the MIR studies. In the present work, these indicators are used as a base to classify the input audio signals. Musical genres are labels used to organize audio pieces that share some similar pre-defined patterns, styles or "musical languages" [1].

This work studies a combination of two existing techniques for automatic audio classification and generates a new approach, aiming to obtain more significant and general results than the previous ones. To accomplish this, two support softwares are used: MARYSAS [2], to extract and analyze the data, and WEKA [3], to help in the simulations and classifications.

Sumário

Índice de Figuras

Índice de Tabelas

Tabela de Símbolos e Siglas

MIR – *Music Information Retrieval*.
PDS – Processamento Digital de Sinais.
ROC – *Receiver Operating Characteristic*
MP3 – MPEG Layer III
MPEG – Moving Pictures Experts Group

Agradecimentos

Agradeço principalmente aos meus pais, Moacir Souza e Ana-Néry Souza, pelo amor, apoio, suporte, compreensão e imprescindível ajuda durante todas as épocas de minha vida. Aos meus irmãos, Filipe Augusto e Márlon Vinícius, pela amizade incondicional que venceu obstáculos severos e por todos os bons momentos proporcionados em sua companhia.

A todos os colegas e professores da faculdade pelos grandes ensinamentos e informações compartilhadas, pelo companheirismo, pelas risadas e por tudo que me proporcionaram vivenciar enquanto colega de classe, profissional e aprendiz.

Para que a significância devida seja dada a algumas pessoas também muito importantes em minha vida, preciso agora citar nomes e respectivos méritos.

Agradeço a Ágda Moreno, minha namorada, por todo amor e carinho, presentes em cada momento decisivo. Por acreditar em minha capacidade acima de tudo e por sempre elevar a patamares mais altos o ânimo para continuar, ainda que sob as pressões e frustrações sofridas ao longo caminho. Agradeço por estar sempre de braços abertos para me acalentar quando precisei. Que nossa relação seja ainda mais terna e completa do que sempre foi.

Agradeço a Clarice Melo pelo amor, pela grande dedicação à nossa amizade, pelo carinho e conforto incondicionais, cultivados ao longo de nossa jornada na faculdade e fora dela. Por fazer os dias mais divertidos e as risadas sempre mais significativas. Esses méritos decerto durarão desse ponto de nossa vida em diante.

Agradeço ao meu orientador, Carlos Alexandre Barros de Mello, pelo incentivo e encorajamento em épocas complicadas do curso (especialmente nesses passos finais), pela paciência com alguns de meus deslizes e pela orientação em si, uma das melhores que já tive ao longo da carreira acadêmica e profissional. Mas acima de tudo, agradeço pela presença, pelo senso de justiça e a impecabilidade no cumprimento de todas as suas responsabilidades (adquiridas e atribuídas). Espero poder manter, daqui em diante, o relacionamento de fácil convivência que sempre tivemos e, dessa forma, poder algum dia retribuir a altura por todas as contribuições favoráveis para minha formação como Engenheiro e para a vida.

Uma lista singela de pessoas igualmente importantes em minha vida também se faz necessária citar. Agradeço aos colegas Adélia Barros, André Câmara, André Henrique, Bruno César Quental (BC), Carlos Eduardo Alencar (Cadu), César Carvalho (Yke), Clóvis Santos, Daniel Gomes, Eduardo Zoby, Flávio Costa, Frederico Duarte, Rivaldo Filho (Dinho), Rodrigo Brayner (Zazá) e Rodrigo Gomes (Raça). Por fazerem a vida de estudante de Engenharia da Computação mais fácil de digerir, ainda que nem todas as azias pudessem ser tratadas.

Alguns agradecimentos especiais a José Eduardo Belarmino Alcoforado, a quem devo as grandes experiências vividas e absorvidas ao longo do tempo em que trabalhamos juntos na CORISCO Tecnologia, além dos impagáveis favores por toda a confiança depositada em minha pessoa durante essa época. Igualmente em especial, agradeço a Murilo Pontes, Caio Sabino e Lucas Torreão, também parceiros da empreitada CORISCO por quase dois anos.

E a todos os outros que fizeram e fazem parte de minha vida de uma forma importante, mas que não foram citados aqui, peço que perdoem minha pobreza de memória e o pequeno espaço ao qual estou limitado a escrever.

Capítulo 1

Introdução

De acordo com Shannon [4] em seus estudos sobre Teoria da Informação, elementos de áudio (voz, ou música) podem ser representados de três maneiras diferentes: em termos do seu *conteúdo*, da *informação* ou da *mensagem* que carregam. Uma maneira alternativa para se classificar áudio baseia-se no estudo do *sinal* que compõe a mensagem. Essa abordagem é tratada na área de Processamento de Sinais e os passos geralmente utilizados para executá-la estão descritos na Figura 1.

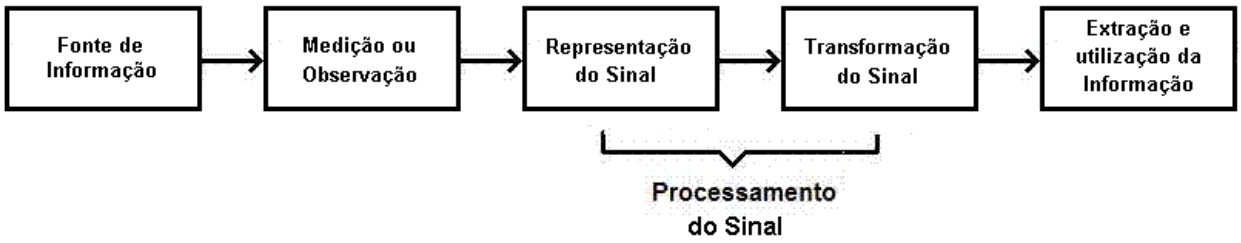


Figura 1. Visão geral da manipulação de informação e processamento de dados.

A fonte de informação, no caso deste estudo, é o próprio sinal de entrada. O processo de medição e observação leva em conta a sua forma de onda. A representação do sinal consiste em obter os dados a partir dessa forma de onda, baseando-se em algum modelo pré-definido, para então aplicar algum tipo de transformação a fim de deixá-lo em um formato mais facilmente tratável. O último passo do processo refere-se à coleta de informações sobre o sinal. Isso pode ser feito manualmente, ou utilizando processos computacionais e algoritmos complexos de propósito específico (por exemplo, podem-se usar algoritmos diferentes para cada tipo de característica do sinal que se deseja observar) [5].

Este trabalho estuda um tópico que vem ganhando grande espaço e renome no meio científico e comercial: classificação automática de áudio digital. Atualmente, empresas da área do entretenimento como a *Amazon* [6] e a *Barles & Noble* [7], aplicativos reprodutores de áudio, como o *Winamp* [8], o *Musicmatch Jukebox* [9] e o *XMMS* [10], além de diversos nomes da indústria do entretenimento, como o *Yahoo! Music* [11] e *AOL Music* [12] se utilizam de classificadores automáticos de áudio para aumentar as suas vendas e a comodidade do usuário na hora de comprar mídia digital ou virtual. Esse tipo de serviço geralmente oferece uma lista de artistas e/ou bandas ordenadas por gênero musical, facilitando o acesso às informações realmente relevantes para o cliente. Essa técnica evita buscas cansativas e improdutivas, além de aumentar a

satisfação dos usuários em relação ao serviço. À medida que a classificação é mais precisa e abrangente, *i.e.*, quanto mais gêneros puderem ser classificados de maneira correta, maiores serão as chances de um usuário encontrar algo que o agrade e, dessa forma, ao invés de apenas navegar a esmo, tornar-se comprador.

Isso é possível por causa do apelo direto aos gostos específicos de cada cliente. Se um histórico puder ser montado a partir de perguntas feitas em um cadastro, ou baseado nas últimas compras realizadas, é possível usar essas informações para sugerir itens que se encaixem ao perfil de preferências do usuário, aumentando assim, a chance mais produtos serem consumidos em visitas futuras.

Na comunidade científica, a crescente evolução das pesquisas com MIR e suas áreas relacionadas colocou a classificação automática de áudio como um dos focos principais de estudos onde se relacionam profissionais de muitos outros campos do conhecimento. Em especial, estudiosos e entusiastas das técnicas de Redes Neurais, aprendizado de máquinas e reconhecimento de padrões, encontraram na classificação automática de áudio digital, além de um grande desafio, uma enorme fonte de possibilidades para realização dos mais diversos experimentos e testes. A facilidade da obtenção de dados, devida à expansão de meios de compartilhamento de informação como a Internet, é um dos grandes facilitadores do trabalho desses profissionais. A popularização de outros meios de comunicação, especialmente na área de equipamentos portáteis, como *palm tops, hand helds* e similares também auxilia o progresso da área de pesquisa.

Há, no entanto, uma série de problemas que ainda dificultam a evolução dessa área de pesquisa. Entre eles, a falta de padronização sobre como classificar os diferentes gêneros musicais existentes, a escassez de documentação adequada sobre o assunto e a grande diversidade de formatos de armazenamento de áudio e mídia digital são alguns dos mais comuns. Ademais, a crescente diversificação da música no mundo, com suas incontáveis novas faces e desmembramentos, suas recombinações, recriações e modificações constantes transformam os limites entre os diferentes gêneros cada vez mais difíceis de verificar sem o auxílio de um especialista [13]. Contudo, apesar de essas questões serem as mais relevantes e as que tornam a classificação de áudio um problema de difícil solução computacional, deve-se considerar, igualmente, que a separação manual de áudio digital não é absolutamente precisa. Estudos determinam que esse tipo de classificação apresenta uma taxa de acerto de apenas 60% [14].

Atualmente, com a capacidade da Internet para modificar comportamentos e padrões sociais, influenciar opiniões e auxiliar na multiplicação e pluralização de novas tendências culturais e comportamentais, a classificação de áudio tornou-se alvo de estudos cada vez mais sérios e profundos. Ela é uma área que movimenta milhões de dólares no comércio *on-line*, onde os gigantes do entretenimento, valendo-se de sistemas cada vez mais avançados, podem oferecer ainda mais liberdade e personalização no fornecimento de serviços multimídia.

Como exemplificado brevemente em parágrafos anteriores, grandes portais utilizam-se da classificação automática de áudio para gerar listas personalizadas, ou sugerir itens de compra para seus usuários mais fiéis (processo também conhecido com criação de perfis). Além disso, bancos de dados com os produtos favoritos de um comprador assíduo, ou os itens relacionados aos que ele geralmente consome, podem ser usados para criar um serviço de sugestão automática para novos (e antigos) clientes. Valendo-se da capacidade de filtrar o indesejado, ou o que não está dentro do perfil do usuário, o serviço de sugestão cria uma comodidade extra para o cliente que, desse modo, pode se tornar um consumidor ainda mais fiel. Para usuários que costumam consumir música Clássica, por exemplo, dificilmente serão sugeridos artistas ou bandas de *Folk* ou *Heavy Metal*.

No entanto, apesar de serem usadas relativamente em larga escala, a maioria das técnicas existentes para classificação automática de áudio leva em consideração aspectos muito

específicos do sinal, do sistema, ou do arquivo que está sendo tratado. Dessa forma, elas perdem a capacidade de generalização, diminuindo a possibilidade de serem utilizadas para uma variedade maior de propósitos. Em outros casos, o tempo que se perde na procura por técnicas mais adequadas para solucionar cada problema específico é longo e quase sempre custa muito caro tanto para pesquisadores (tempo), quanto para investidores (dinheiro).

Apesar de todas as dificuldades encontradas, diversos pesquisadores conseguiram grandes êxitos na classificação de áudio, valendo-se de diferentes técnicas e abordagens. Dentre elas, podem-se citar estudos que utilizaram Transformadas Wavelet [15], Classificadores Taxonômicos [13], Redes Neurais [13][5][16], Aprendizado Estatístico [15] e Máquinas de Vetor de Suporte [17][18].

1.1 Motivação

A classificação de áudio, como já enfatizado, é assunto de grande importância para diversas áreas. Ultimamente, muitas discussões e especulações, embora ainda bastante informais, têm surgido a respeito da grande proliferação da música através dos meios de comunicação em massa, notadamente a Internet. Dentre as diversas polêmicas a respeito de assuntos relacionados à distribuição de mídia digital, como direitos autorais e compartilhamento ilegal de arquivos, diversas pesquisas apontam para a música, ou mais especificamente, para o áudio digital, como um dos maiores atrativos para os que desejam fornecer serviços na área de entretenimento pessoal. Empresas como a *Apple* [19], que alcançou uma distinta posição no mercado de venda de mídia digital, têm enfrentado uma severa luta contra rivais de mercado, instituições que preservam os direitos autorais de artistas e até mesmo contra algumas entidades governamentais. Como explicitado anteriormente, a Internet tornou-se o meio mais eficiente de espalhamento de informação e a música já está incorporada como um dos seus principais pontos de apoio e atração.

Além dos aspectos mais estritamente mercadológicos, que envolvem compra e venda de áudio e/ou mídia digital, onde estão atuando grandes empresas como *Apple* [19], *Yahoo! Music* [11], *Amazon* [6], *Barnes & Noble* [7] e *Microsoft* [20], a classificação automática de áudio também vem aumentando sua expressividade dentro das redes sociais e comunidades virtuais. Dois exemplos contemporâneios são o *Last.fm* [21] e o *MySpace* [22]. Nesses cenários, recomendações e troca de arquivos de áudio digital entre os usuários tornou-se um dos meios mais rápidos e eficientes para se proliferar algum estilo, artista ou grupo musical.

O *Last.fm* trata-se de uma estação de rádio *on-line* e um sistema de recomendação de músicas. Uma vez cadastrado, o usuário utiliza alguns *softwares* (*plug-ins*) que são capazes de coletar informações sobre as músicas que estão sendo ouvidas, diretamente do reprodutor de áudio. Além de características básicas, como nome do artista e nome da música, os *plug-ins* do *Last.fm* são capazes de obter informações mais complexas como duração da faixa de áudio, álbum e gênero musical ao qual ela pertence. Com esses dados em mãos, o *site* monta uma lista das músicas favoritas do usuário e apresenta-lhe recomendações e outros usuários do sistema que possuem um gosto musical similar ao dele. É a partir daí que, ao entrarem em contato com outros estilos similares aos que já conhecem e aprovam, os usuários do sistema podem se tornar compradores. Usando uma estratégia de *marketing* simples, que se vale da comodidade fornecida pela Internet, o *Last.fm* mostra diversas informações sobre o artista ou banda, como por exemplo, em que álbum uma música específica se encontra e se o álbum está ou não disponível para aquisição em lojas *on-line* de venda de entretenimento, como a *Amazon*.

O *MySpace* não é uma rede social estritamente relacionada a áudio como o *Last.fm*, mas ela também possibilita aos usuários fazer recomendações das músicas de sua preferência, além de

permitir que um reprodutor de áudio seja embutido na página principal, recebida na hora do cadastro. Novamente, a classificação automática dos gêneros musicais dessas listas de músicas fornece informações suficientes para que recomendações de outros estilos similares sejam mais facilmente aceitas pelos usuários, que cada vez mais podem vir a se tornar potenciais compradores do que lhes é apresentado.

No âmbito científico, muitos trabalhos importantes, como [13], [14] e [15], têm apresentado uma série de contribuições extremamente valorosas para o crescimento e maturação dos estudos e técnicas utilizadas em MIR. Conferências como a ISMIR (*International Symposium on Music Information Retrieval*) e a ICMC (*Internationl Computer Music Conference*), frequentemente reúnem pesquisadores que trabalham colaborativamente no intuito de aprimorar as aplicações já existentes na área, além de aumentar os esforços a favor das novas pesquisas. Dentre elas, alguns métodos mais populares utilizam combinações de outras técnicas já bem fundamentadas como Redes Neurais e Aprendizado de Máquinas, bem como novos algoritmos computacionais e aperfeiçoamento de formulações fornecidas por áreas irmãs como a Matemática, a Física e PDS (Processamento Digital de Sinais).

Embora a evolução de MIR e suas áreas relacionadas já possa ser considerada bastante expressiva, a maioria dos trabalhos ainda não é suficientemente genérica para possibilitar uma abordagem padronizada, que resolva a maioria dos problemas de classificação automática sem necessidade de empregar um grande esforço no tratamento das bases de dados, treinamento dos classificadores e grande disponibilidade de tempo em pesquisa.

A maioria das técnicas trata o problema da classificação automática de áudio digital de maneira muito específica, atendo-se, geralmente, a formatos ou padrões escolhidos por critérios pouco parcimoniosos, que dificultam a reprodutibilidade dos experimentos para futuros estudos, melhorias e análise.

Uma vez que o método científico, em sua essência, também é uma preocupação que o presente trabalho leva em consideração, procurou-se abordar o problema utilizando técnicas de grande reconhecimento e métodos que se permitam ser facilmente reproduzidos em laboratório.

Ao longo da história de MIR, uma série de autores preocupou-se fortemente com os aspectos de reprodutibilidade de seus experimentos, documentando bem o uso das técnicas de Redes Neurais, Transfomadas Wavelet, Aprendizado Estatístico, Classificação Taxonômica, etc. No entanto, as primeiras propostas para sistemas de reconhecimento de áudio foram criadas a partir de sistemas mais simples, para reconhecimento de voz. Os professores Rabiner, Schafer e Huan, são autores de uma produção acadêmica bastante vasta sobre esse tópico. Escreveram diversos livros, ensaios e projetos sobre o assunto; alguns dos quais, tornaram-se leitura obrigatória para estudantes e entusiastas da área de processamento digital de áudio [5], [23]. Seus trabalhos abordam de maneira exaustiva todos os aspectos mais profundos da área de reconhecimento de voz, desde os detalhes a respeito da morfologia do som (partindo de uma análise detalhada do trato vocal) até as nuances matemáticas em que podem ser decompostas os sinais.

Pouco depois dessa fase inicial, alguns trabalhos que incluíam distinção ente sinais falados e não-falados foram propostos. A maioria deles esteve relacionada à classificação de notícias (de telejornais ou rádio), música e sons do ambiente (também conhecido na área como ruído de fundo). Trabalhos como o de Saunders [24] trataram do problema de distinção entre música e voz de maneira bastante abrangente. E também importante na área de distinção entre sinais musicais e não-musicais está o trabalho de Berenzweig e Ellis [25], que se propõe localizar porções de uma peça musical que contenham uma voz cantando. No trabalho deles, uma saída de ativação de fonemas de um sistema de reconhecimento automático de voz é usado como o vetor características para classificar os segmentos com canto.

Em trabalhos ainda mais recentes, e similares ao presente, podem ser citados autores como George Tzanetakis e Perry Cook. Eles possuem uma vasta literatura sobre classificação de áudio digital, utilizando técnicas de aprendizado de máquina como MLP (*Multi-Layer Perceptron*), kNN (*k-Nearest Neighbors*), SVM (*Support Vector Machine*) e outros. Alguns de seus trabalhos mais notáveis [14] valem-se da extração de texturas de timbre com janelas de tempo para alimentar classificadores automáticos como o WEKA e treinar redes neurais para o reconhecimento de gêneros musicais distintos.

Por fim, trabalhos como o de Lambrou e Kudumakis [15], que também utilizam extração de vetores de características, como texturas de timbre, para classificar gêneros musicais usando Transformadas Wavelet, completam o cenário de fundo em que se desenvolvem e evoluem, atualmente, os trabalhos da área de PDS e MIR.

1.2 Objetivo

O objetivo deste trabalho é realizar uma comparação entre algumas das técnicas de classificação de áudio, utilizando duas bases de dados de arquivos. A primeira trata-se de uma base já conhecida e utilizada no trabalho de Li e Ogihara [17] e Tzanetakis e Cook [14]. A segunda é composta por um conjunto de arquivos que fazem parte da coleção particular deste autor. Neste estudo, são realizadas extração de características e tratamento dos dados em conjunto com a aplicação de algumas técnicas de redes neurais e aprendizado de máquina para classificá-los automaticamente. Dois softwares de suporte são usados para esses fins: a extração dos dados é realizada com o MARSYAS [2] e, para treinamento das redes, foi utilizado o WEKA [3], ambos descritos com mais detalhes em Capítulos subseqüentes.

Duas das técnicas estudadas, [13] e [14], são combinadas com o objetivo de gerar uma nova abordagem, mais geral e de melhor desempenho, para a classificação automática de áudio. A comparação entre os resultados é feita através da criação de curvas ROC (*Receiver Operating Characteristics*) [14], analisando os valores encontrados de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos. Curvas ROC são uma plotagem gráfica da sensitividade *versus* especificidade de um sistema binário de classificação. O termo sensitividade expressa a proporção de itens tomados de um conjunto de testes, que foram classificados corretamente, ou seja, são os Verdadeiros Positivos (TP - *True Positives*). A literatura a respeito do assunto trata os dois termos de maneira intercambiável. Por sua vez, o termo especificidade expressa a proporção de itens que não pertencem à classe em questão, ou seja, são os Verdadeiros Negativos (TN - *True Negatives*). A literatura também trata de maneira intercambiável os termos especificidade e Verdadeiros Negativos. Por fim, os termos Falsos Positivos (FP - *False Positives*), e Falsos Negativos (FN - *False Negatives*), são, respectivamente, os duais de TP e TN. Ou seja, os FP's representam os termos classificados erroneamente para a base de testes e os FN's, os que foram classificados como não pertencentes a certa classe da qual, na realidade, são integrantes [26].

1.3 Estrutura do documento

O Capítulo 2 trata dos conceitos básicos para auxiliar o entendimento mais profundo das técnicas utilizadas nos experimentos do trabalho, bem como algumas definições e princípios importantes das áreas relacionadas com o estudo, como sinais discretos, contínuos, quantização, amostragem, etc.

O Capítulo 3 relaciona e explicita os *softwares* de apoio utilizados nos experimentos de extração de dados e classificação das bases de áudio. Encontram-se nesse Capítulo um pouco do histórico de cada programa, alguns exemplos de utilização e a coleção de funcionalidades mais importantes para o presente projeto.

Os detalhes a respeito da metodologia utilizada na separação e pré-processamento das bases de dados, o preparo e a escolha dos classificadores, extração de características e os experimentos realizados encontram-se elucidados no Capítulo 4.

O quinto Capítulo elenca as conclusões alcançadas com o projeto e lista alguns trabalhos futuros que o autor pretende considerar e levar adiante para o aprimoramento dos resultados obtidos.

Capítulo 2

Conceitos Básicos

Este Capítulo trata de alguns conceitos básicos necessários para o melhor entendimento das técnicas utilizadas neste trabalho. Aqui são delineados um breve histórico e os elementos mais indispensáveis para a compreensão de PDS e algumas técnicas largamente usadas nessa área.

2.1 Processamento Digital de Sinais

Uma das grandes preocupações dos estudos de PDS [5], [27] está em obter representações discretas do sinal para, dessa forma, poder tratá-lo convenientemente. O uso das técnicas digitais atuais fornece subsídios suficientes para a criação de funções e algoritmos extremamente sofisticados para estudos de extração de dados e sua utilização no processamento de sinais.

Uma das primeiras aplicações de técnicas de sistemas digitais deu-se na área de processamento de voz, com o intuito de simular o comportamento de sistemas analógicos complexos. A relativa facilidade que os pesquisadores encontraram em simular esses sistemas no computador evitava a necessidade de construí-los para realizar seus experimentos. Além disso, dados os altos custos envolvidos na construção de sistemas analógicos, a sua utilização em larga escala era deveras proibitiva, mesmo para a crescente comunidade de estudos da área.

Em meados dos anos 60, os avanços na área de processamento digital foram grandemente acelerados e um dos principais fatores que contribuiu para isso foi a rápida evolução das indústrias de *hardware*. Essa evolução proporcionou um aumento significativo na produção de equipamentos mais baratos, robustos e rápidos. A partir de então, começou a ficar claro que aplicações com sistemas digitais não eram úteis apenas para simulações laboratoriais de sistemas analógicos complexos, extrapolando os limites das implementações iniciais, voltadas para processamento de voz. Na mesma época que a revolução na indústria de *hardware* alavancou o crescimento e o aumento da utilização de sistemas digitais, os avanços teóricos da própria área levaram ao fortalecimento das técnicas de processamento digital sobre as utilizadas com sistemas analógicos.

Dentre as vantagens existentes no trabalho com sistemas digitais, destacam-se o seu alto grau de confiabilidade e sua natureza intrinsecamente mais compacta que a dos sistemas analógicos. Ademais, a miniaturização de componentes de *hardware*, como micro-controladores, permitiu que a implementação de sistemas completos e independentes pudesse ser idealizada em apenas uma pastilha (*chip*), aumentando sua flexibilidade, portabilidade, o barateamento do custo de fabricação e a conseqüente facilidade de aquisição por um público não técnico-científico cada

vez maior. Essa popularização também auxiliou o crescimento da área, pois, à medida que mais e mais pessoas se interessavam por práticas com sistemas de áudio digital, a própria indústria começou a fornecer mais subsídios a preços bastante moderados e um melhor suporte para seus produtos e sistemas. Com isso a comunidade científica ganhou ainda mais entusiasmo para avançar em suas pesquisas e, dessa forma, melhorar o desenvolvimento de novas tecnologias.

Sistemas digitais de áudio relativamente complexos encontram-se hoje em MP3 *players*, *hand helds*, *palm tops*, telefones móveis e outros tipos de equipamentos portáteis, como relógios, chaveiros e similares. Se comparados às máquinas utilizadas para processamento de sinais de áudio da década de 60, os equipamentos supracitados possuem dezenas e até centenas de vezes mais recursos de armazenamento de informações e realizam as mesmas tarefas com uma eficiência extremamente superior. De fato, aplicações simples como codificação e decodificação de sinais de voz com apenas alguns segundos de duração levavam, pelo menos, uma hora para fornecerem resultados (muitas vezes imprecisos e carregados de ruídos indesejáveis). Atualmente esse tempo tem decaído bruscamente para alguns milisegundos.

Outros motivos para preferir utilizar sistemas digitais de áudio estão relacionados à segurança que eles são capazes de fornecer. Na transferência de informação através de um canal de comunicação (redes de computadores, por exemplo), os dados podem ser cifrados para manter sua confidencialidade. Além disso, essas informações podem ser enviadas através de canais ruidosos (se uma codificação suficientemente robusta for utilizada) e, como estão representados em uma forma discreta, não diferem em essência de qualquer outro tipo de dado, podendo vir a ser transmitidos em conjunto com códigos de outros formatos. Ao receptor é necessário, apenas, utilizar a forma correta de decodificação para distinguir os dados de áudio daqueles de outros tipos quaisquer [5].

2.1.1 Sistemas e Sinais Discretos

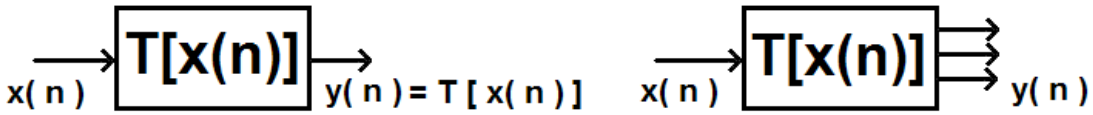
Como descrito na Seção anterior, os estudos da área de PDS estão intimamente relacionados com a maneira como os sinais são representados. A grande maioria dos sistemas físicos que envolvem processamento de informação, principalmente quando se trata de comunicação, utiliza, inicialmente, sinais contínuos, em geral com padrões variáveis no tempo [27].

Exemplos bastante próximos do cotidiano encontram-se nos equipamentos que tratam voz e áudio de maneira geral. A voz humana, a música e a quase totalidade dos outros sinais de áudio são sinais contínuos. Manipular esse tipo de dado é uma tarefa complexa e quase sempre impraticável para a maioria das aplicações. A natureza variável no tempo desses sinais não possibilita que sua representação seja precisa em sistemas computacionais. Dessa maneira, para que os estudos da área pudessem progredir, fez-se necessário encontrar uma maneira para utilizar esse tipo de sinal em sistemas digitais. Estudos como os de Shannon [4] mostraram que funções de uma variável poderiam ser usadas a fim de mapear tais sinais para um formato mais facilmente computável. Esse tipo de dado, conhecido como sinal discreto, *i.e.*, aquele que pode ser representado como um conjunto seqüencial de dados independentes, tornou o processamento digital de sinais uma tarefa mais simples e conveniente para pesquisadores e estudiosos da área.

Uma vez que o sinal contínuo é amostrado e transformado em uma série de números a partir das funções matemáticas apropriadas, isolar cada um deles e observar variações em seu comportamento torna-se uma tarefa mais facilmente realizável. Além de simplificar o próprio estudo do sinal, essa representação proporciona uma grande facilidade de reprodução do experimento se comparada a trabalhos que utilizam sinais contínuos. Replicar uma seqüência de números derivados de uma função é uma tarefa simples e direta, pois, se a função for recalculada com os mesmos parâmetros inicias, o sinal exato é novamente obtido. O mesmo não acontece

com sinais contínuos, que dependem de fatores nem sempre triviais de se controlar (geralmente, variáveis no tempo que não seguem padrões bem definidos).

No entanto, PDS não se preocupa apenas com a representação de um sinal analógico por uma função matemática, a fim de deixá-los mais facilmente tratáveis, mas também com a transformação de um formato do sinal em outro, ou seja, uma seqüência de entrada mapeada diretamente para uma seqüência de saída. Alguns desses sistemas recebem uma entrada e respondem com apenas uma saída, como mostrado na Figura 2 (a). De maneira similar, muitos sistemas de manipulação de áudio são concebidos para estimar múltiplos parâmetros do sinal que variam com o tempo (como é o caso do presente trabalho). Esses sistemas devem ser capazes de responder com mais de uma saída (um vetor de saídas) ao excitamento provocado por apenas uma entrada, como descrito no diagrama de blocos da Figura 2 (b).



(a) (b)

Figura 2. Representação de Diagramas de Blocos de Sistemas com (a) entrada única e saída única, e (b) entrada única e saída múltipla.

Com o auxílio de técnicas como quantização e amostragem, adicionados ao uso de filtros, a discretização e tratamento de sinais digitais começaram a ser amplamente utilizados para baratear e acelerar os estudos com sinais digitais. Nas Seções subseqüentes, são fornecidas breves explanações de algumas dessas técnicas, posto que seu entendimento mais profundo constitui obrigatoriedade para a compreensão apropriada deste estudo.

2.1.2 Amostragem

O trabalho de transportar um sinal do domínio contínuo para o domínio discreto é realizado efetivamente por técnicas de amostragem [28]. A amostragem é o processo que consiste em representar a forma de onda do sinal como uma série de números tomados a intervalos regulares, onde cada um indica a amplitude do sinal em um determinado instante. Esse processo é largamente utilizado em diversos sistemas que fazem parte do nosso cotidiano. Encontramos processos de amostragem, por exemplo, em aparelhos de televisão, telefones, redes de computadores e diversos outros.

Estritamente falando, o processo de amostragem não está exclusivamente ligado à digitalização de sinais analógicos. A amostragem de sinais pode resultar em diversos tipos de saídas diferentes, não necessariamente digitais. De fato, pode-se obter de uma amostragem uma seqüência de pulsos analógicos, conhecida como modulação por amplitude de pulso (PHM, *Pulse-height Modulation)* [5] ou de pulsos com amplitude fixa, conhecidos como modulação por posição de pulso (PPM, *Pulse Position Modulation*) [5] ou modulação por comprimento de pulso (PWM, *Pulse width Modulation*) [5]. No entanto, a representação mais utilizada para uma amostragem é uma seqüência de números binários, conhecida como modulação por código de pulso (PCM, *Pulse Code Modulation*) [5]. É compreensível que essa abordagem seja a mais popular e a mais utilizada, uma vez que os números binários são naturalmente mais fáceis de armazenar e processar em sistemas computacionais.

Existem algumas regras que devem ser seguidas quando se deseja realizar amostragens de sinais digitais. Uma das grandes preocupações dos estudiosos de PDS, Teoria da Informação e

áreas relacionadas é evitar um efeito negativo que ocorre na amostragem de sinais, chamado *aliasing*. Quando esse efeito acontece, dois sinais contínuos diferentes tornam-se indistinguíveis, não permitindo que o sinal original seja reconstruído unicamente. O que ocorre efetivamente para os sinais tornarem-se idênticos é um processo de superposição entre as réplicas deslocadas do espectro original. As altas freqüências assumem a identidade de freqüências mais baixas, e distorcem, dessa forma, o espectro do sinal contínuo original [28].

Para evitar esse tipo de problema, a taxa mínima na qual um sinal deve ser amostrado, precisa ser, no mínimo, igual a duas vezes o valor da sua freqüência máxima. Esse valor é definido como taxa de amostragem de Nyquist ou apenas taxa de Nyquist [28], em homenagem ao cientista sueco Harry Nyquist que, com seus estudos em conjunto com Claude Shannon, contribuiu grandemente para os avanços das áreas de PDS e Teoria da Informação.

No entanto, para fins práticos e laboratoriais, geralmente, são usados valores maiores que a taxa de Nyquist para evitar que ocorra *aliasing* na reconstrução do sinal original.

A seguir, na Figura 3, é mostrado um exemplo genérico da amostragem de um sinal contínuo. Como dito anteriormente, o conjunto de pontos, tomado por todas as interseções entre seu invólucro e as retas igualmente espaçadas, formam o conjunto de números capazes de representar o sinal, *i.e.*, a amostragem.

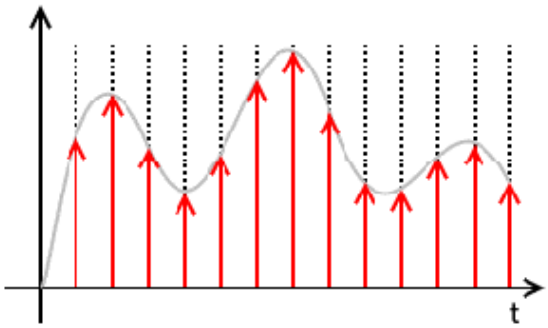


Figura 3. Amostragem de um sinal contínuo.

A Figura 4 ilustra os diversos espectros assumidos pelo sinal para taxas de amostragem diferentes (w_s). A Figura 4(a) mostra o espectro do sinal original de tempo contínuo. A Figura 4(b) exemplifica o espectro de um sinal amostrado com uma taxa maior que a de Nyquist. Na Figura 4(c) é delineado o comportamento do espectro de um sinal que foi amostrado com taxa igual à de Nyquist e, finalmente, a Figura 4(d) apresenta o espectro assumido por um sinal amostrado com taxa menor que a taxa mínima de duas vezes a freqüência máxima do sinal original. A freqüência máxima do sinal é representada, em todas as figuras, pelo parâmetro W.

Deve-se observar que, à medida que o valor da freqüência w_s vai diminuindo, as réplicas do sinal contínuo vão se aproximando uma da outra até que se superpõem e se somam, distorcendo as características originais do sinal contínuo e invalidando seu uso para fins de reconstrução.

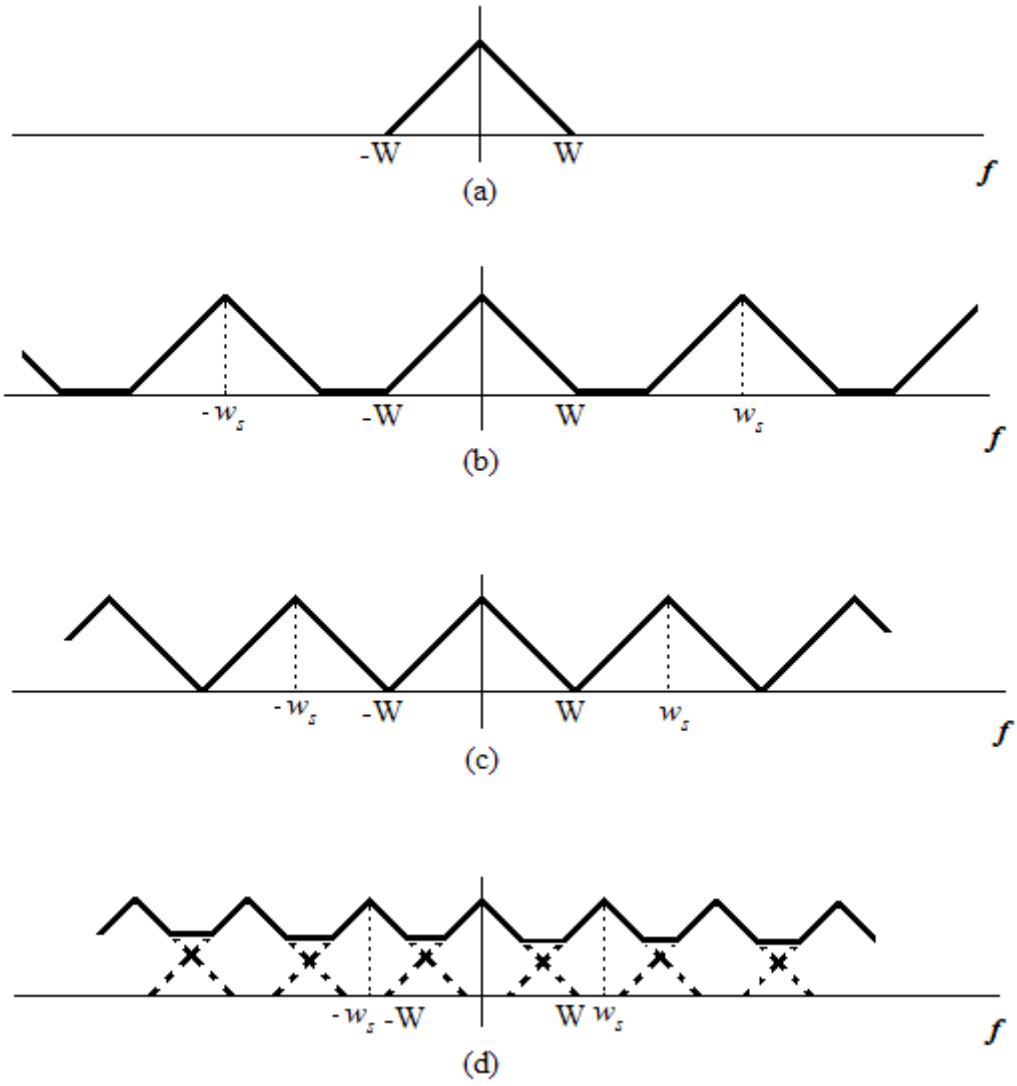


Figura 4. Diversos espectros do sinal para taxas de amostragens diferentes. (a) Espectro do sinal original de tempo contínuo. (b) Espectro do sinal amostrado com uma taxa maior que a de Nyquist. (c) Espectro do sinal amostrado com taxa igual à taxa de Nyquist. (d) Espectro do sinal amostrado com taxa menor que a taxa de Nyquist.

Por sua vez, a reconstrução do sinal é um problema que envolve uma combinação entre sinais de tempo contínuo e discreto. É, em essência, o processo inverso da amostragem, no qual, a partir de um conjunto de dados discreto, *i.e.*, cadeias de números capazes de representar um sinal contínuo, pode-se recuperar o sinal original. A Figura 5 apresenta um diagrama de blocos que exemplifica essa idéia.

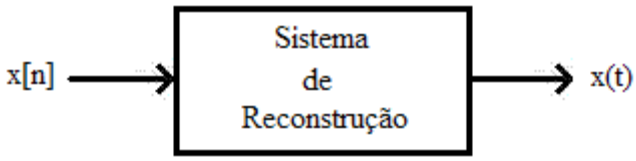


Figura 5. Diagrama de blocos ilustrando o processo de recuperação de um sinal discreto a partir de um sinal de tempo contínuo.

Nos parágrafos anteriores desta Seção, constatou-se que as amostras de um sinal nem sempre são capazes de fornecer uma maneira única de retomar o sinal contínuo correspondente. Por exemplo, se tomarmos um sinal senoidal e realizarmos amostras com espaçamento de um período, o sinal amostrado terá aparência de uma constante e não poderemos determinar se o sinal contínuo era uma constante ou uma senóide [28].

O problema pode ser visualizado de maneira mais simples a partir da Figura 6 a seguir, onde dois sinais distintos possuem o mesmo conjunto de amostras.

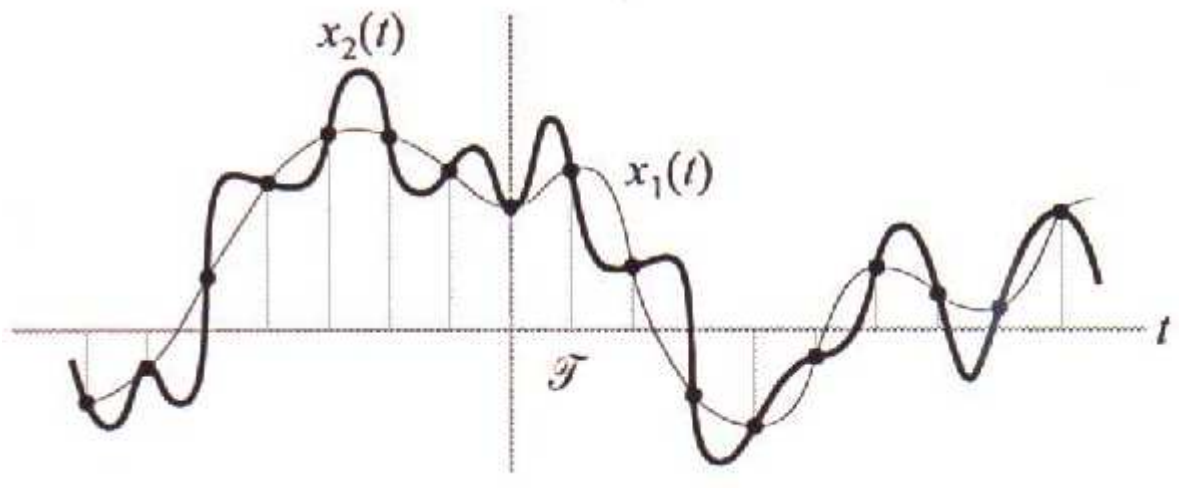


Figura 6. Sinais distintos com o mesmo conjunto de amostras (Figura retirada do livro Haykin, S. e Veen B. V., “Sinais e Sistemas”, Ed. Bookman, 2001).

Em casos como o da Figura 6 não é possível para o sistema de reconstrução obter unicamente o sinal original e, desse modo, conclui-se que apenas o conjunto de amostras não é capaz de indicar seguramente o comportamento do sinal. Para que se possa obter uma resposta mais precisa nesse caso, faz-se necessária a adição de algumas restrições extra para o sinal de tempo contínuo, como por exemplo, exigir que as transições de uma amostra para a outra sejam tão suaves quanto possível. Essa suavidade, que reflete a taxa com a qual o sinal contínuo muda, está diretamente relacionada com a sua freqüência máxima.

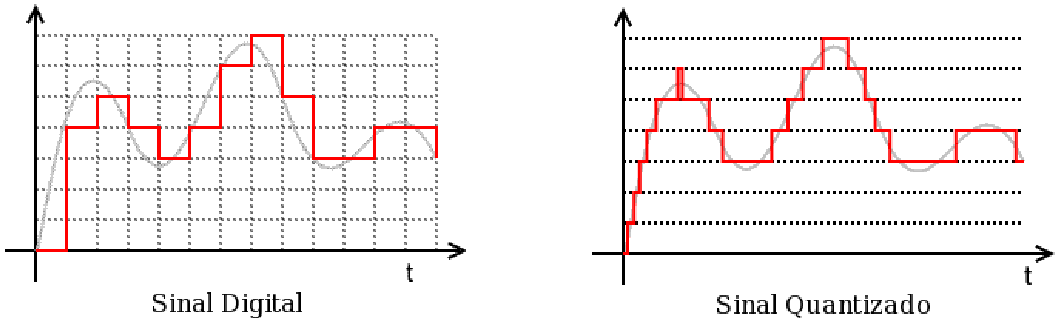
A partir dessas constatações, Nyquist foi capaz de definir o valor ótimo para a taxa de amostragem sinal contínuo, que, como já mencionado, deve ser maior ou igual a duas vezes a freqüência máxima do sinal.

2.1.3 Quantização

Para PDS, quantização é o processo de aproximação de uma faixa contínua de valores (ou um grande conjunto de valores discretos possíveis), para um conjunto relativamente pequeno de símbolos discretos ou valores inteiros [28]. Algumas técnicas especiais, como Amostragem Ideal [5][28] dispensam o uso de quantização, mas uma vez que este tipo de artifício não foi utilizado nesse trabalho, seus pormenores não serão explicitados.

Quantização é, em linhas gerais, o processo que transforma um sinal discreto (resultado da amostragem), nos sinais digitais que são filtrados pelo processador. Tanto a amostragem quanto a quantização são realizadas por um conversor analógico digital, sendo a quantização um processo descrito em nível de bits, *e.g.*, um CD de áudio é amostrado a uma freqüência de 44.100 Hz e quantizado com 16 bits. A Figura 7(a) mostra um sinal digitalizado, e a Figura 7(b), o mesmo sinal quantizado. Em 7(a) o sinal contínuo é transformado em sinal digital, utilizando um

conjunto de possíveis valores discretos, através do processo de amostragem. Em 7(b) o sinal o sinal é quantizado para obter uma aproximação maior do sinal original. Embora ocorram perdas na representação do sinal, o processo de quantização gera uma aproximação bastante satisfatória e mais facilmente tratável em sistemas computacionais.



(a) (b)

Figura 7. Sinal analógico (a) digitalizado e (b) quantizado.

Como explicado mais adiante no Capítulo 4, os arquivos de áudio utilizados neste trabalho foram compactados como MP3, com taxa de 128 Kbps. Essa configuração tornou-se um padrão *de facto* simplesmente por causa de sua popularidade na Internet. Além da taxa de compressão, que permite armazenar aproximadamente 1 MB por minuto, a qualidade de áudio alcançada é aproximadamente igual àquela usada em CD's. A quantização utilizada para todos os arquivos utilizados neste trabalho foi de 16 bits.

2.1.4 Filtros Digitais

Filtros são usados em PDS para dois propósitos: 1) remover dados indesejáveis do sinal, como ruídos provenientes de alguma fonte de interferência, ou inseridos de maneira não controlada, e 2) extrair partes úteis do sinal, como espectros de freqüência dentro de algum limite determinado [5][28]. Sendo assim, eles se prestam a dois serviços principais: separação de sinais que foram combinados e restauração de sinais que sofreram algum tipo de distorção. Um filtro trabalha basicamente recebendo uma entrada e exibindo uma saída como resposta ao processamento realizado (filtragem), segundo explicitado na Figura 8, abaixo.

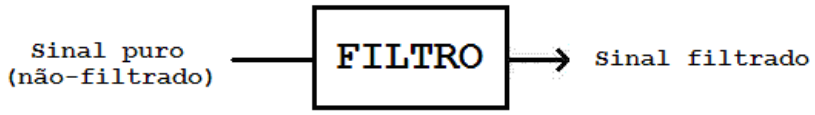


Figura 8. Representação em diagrama de blocos do processo de filtragem de sinal.

De um modo geral, filtros podem ser de dois tipos: analógicos e digitais. Essa é a principal e mais abrangente das classificações e, além dela, existem mais algumas subdivisões, como filtros lineares e não lineares; passivos ou ativos; invariantes no tempo ou adaptativos; causais ou não-causais; estáveis ou instáveis; Filtros de Resposta Finita (FRF) ou Filtros de Resposta Infinita (FRI) e alguns outros, cujas explicações pormenorizadas não estão no escopo deste trabalho.

Apesar de filtros analógicos e digitais serem capazes de realizar as mesmas tarefas, eles são essencialmente diferentes, tanto na sua construção física quanto na maneira como tratam os sinais.

Os filtros analógicos são construídos a partir de componentes eletrônicos discretos, como resistores, capacitores e amplificadores operacionais e são usados largamente na construção de eqüalizadores digitais em sistemas de alta fidelidade, sistemas para melhoramento de sinais de áudio e vídeo e em aplicações para realizar redução de ruído. Além disso, filtros analógicos sempre manipulam, em todos os estágios da filtragem, componentes elétricas (corrente ou tensão), que descrevem quantitativamente o sinal. Dessa forma, sua susceptibilidade a sofrer interferências de ruídos externos, como pontos irradiadores de energia eletromagnética, é maior que as dos filtros digitais.

Filtros digitais possuem uma série de vantagens em relação às suas contrapartes analógicas. Eles são programáveis e mais facilmente desenvolvidos, implementados e testados em computadores de propósito geral, ou estações de trabalho simples. São mais estáveis que filtros analógicos, pois não sofrem interferência de intemperismos climáticos (principalmente calor e umidade), ou interferências eletromagnéticas. E finalmente, conseguem alcançar um desempenho métrico bastante satisfatório, de modo que é relativamente simples construir um filtro de 1000 Hz capaz de alcançar freqüências quase perfeitas de até 999 Hz e bloquear completamente um sinal de 1001 Hz, tarefa difícil para os filtros analógicos executarem. Além disso, filtros digitais trabalham em baixas freqüências com maior precisão e, devido à evolução nos métodos utilizados na sua construção, eles já são capazes de oferecer melhores resultados em certas aplicações que requerem manipulação de dados em alta freqüência (trabalho antes realizado apenas por filtros analógicos). No entanto, diferentemente dos filtros digitais, os analógicos não perdem informação no processo de quantização.

Sua flexibilidade e adequabilidade a mudanças também é notável. Alguns filtros digitais são capazes de se adaptar a possíveis alterações nas características do sinal, e continuar o trabalho de filtragem sem necessitarem parar para realizar reconfigurações manuais. Por fim, processadores suficientemente rápidos podem lidar com combinações complexas de filtros em paralelo ou em cascata (em série), mantendo o *hardware* simples e compacto. Isso possibilita uma economia substancial para desenvolvedores e analistas, que não precisam modificar a estrutura completa dos circuitos para adaptar seus sistemas a alterações do sinal, ou para realizar estudos e testes que não foram previstos de antemão, como aconteceria se fossem usados filtros analógicos.

O funcionamento de filtros analógicos, suas técnicas e métodos de implementação são processos já bem fundamentados, mas encontram-se fora do escopo deste trabalho. Será subministrada, portanto, uma breve elucidação sobre o funcionamento básico de filtros digitais apenas, uma vez que o presente estudo vale-se da implementação de alguns desses para extrair porções dos arquivos de áudio que são manipulados.

O sinal analógico de entrada deve primeiro passar pelos processos de amostragem e quantização, feitos por um CAD (Conversor Analógico Digital) [5]. Os resultados são números binários que representam os valores amostrados da entrada. Esses números são, então, transferidos para o processador que realiza todos os cálculos numéricos necessários para a filtragem em si (esses cálculos geralmente consistem em multiplicar os valores da entrada por constantes e realizar um somatório desses produtos). Um CDA (Conversor Digital Analógico) [5] finaliza o processo, transformando a seqüência binária novamente em um sinal analógico. A metodologia inteira é ilustrada na Figura 9.

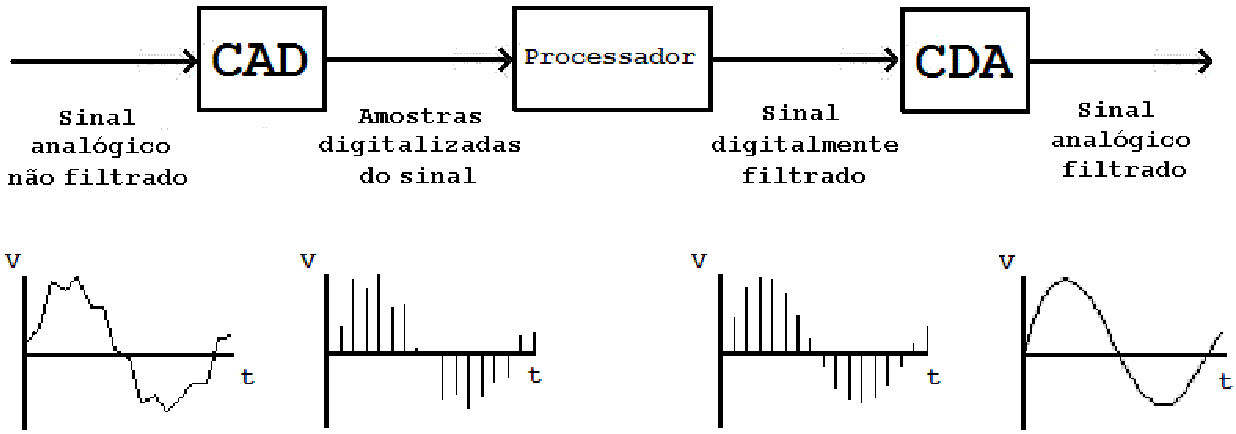


Figura 9. Processo de filtragem digital de um sinal analógico.

Neste trabalho, o uso de filtros digitais implementados via *software* [2], será de grande importância para extração de porções pré-definidas do sinal. Uma vez que o processamento de sinais de áudio ainda é uma tarefa computacionalmente cara e demorada, foram utilizadas algumas porções dos dados ao invés de tratá-los por completo. Nestes esforços, filtros são usados para coletar apenas as partes do sinal que possuam características suficientemente relevantes para auxiliar na corretude da classificação. De acordo com Desphande *et al* [13] e Tzanetakis *et al* [14], essa abordagem nos remete a bons resultados e evita um desperdício de tempo e processamento com dados não substanciais para os fins almejados.

2.2 Processamento de Sinais de Áudio

Alguns conceitos básicos de Processamento de Sinais de áudio precisam ser elucidados para que o entendimento de termos citados em Seções e Capítulos posteriores seja completo.

Em Processamento Digital de Sinais o uso de ferramentas matemáticas como Transformadas de Fourier diretas e inversas, Transformadas Wavelet, Transformadas Z e afins, é bastante comum, uma vez que a área de PDS deve grande parte de sua evolução ao aprimoramento de novos teoremas e paradigmas que surgem dentro da matemática e da física. O estudo do comportamento de sinais é uma área vasta e extremamente rica e este trabalho se vale de uma série de seus tópicos mais populares, tais como: Centróide Espectral, *Rollof*, MFCC (*Mel Frequency Cepstral Coefficients*), Energia, Magnitude, Molduras de Análises, STFT (*Short Time Fourier Transform*) e Taxa de Passagem pelo Zero. As Sub-Seções a seguir explanam brevemente cada um desses termos.

2.2.1 Centróide Espectral

O Centróide Espectral é uma das texturas de timbre do sinal e pode ser definido como o centro de gravidade do espectro da magnitude da STFT (*Short Time Fourier Transform*). O centróide é a medida do formato do espectro do sinal e quanto maior o seu valor, mais suaves as altas freqüências ficarão. Em PDS, o centróide espectral é usado com intuito de normalizar as magnitudes do sinal que possuem valores muito distintos [5], [14].

2.2.2 *Rollof*

Como apresentado na Seção 2.1.4, filtros são amplamente utilizados em diversos sistemas de processamento de sinais, desde os mais simples, de média ou baixa precisão, como aparelhos de televisão e rádios portáteis, até aqueles extremamente complexos usados em equipamentos médicos da área de fonoaudiologia que necessitam trabalhar com precisão e fornecer a maior confiabilidade possível [5]. Desse modo, a construção de filtros com respostas rápidas e exatas é uma preocupação conjunta das áreas de PDS (que os utiliza), Matemática, Física e Engenharia (que pesquisam novos teoremas e paradigmas para seu aperfeiçoamento) e da indústria de equipamentos de hardware (que os fabrica).

No entanto, apesar dos esforços conjuntos de todas as áreas relacionadas, não é possível construir um filtro ideal. O que se pode considerar como estado da arte nesse sentido são aproximações alcançadas por algumas teorias matemáticas relacionadas à área de comunicação. Para a idealização de filtros reais, usa-se uma técnica bastante popular envolvendo uma família de pulsos conhecida como Família de Pulsos Cosseno Levantado. Os pulsos cosseno levantado atendem ao critério de Nyquist e, como já dito, são largamente utilizados para experimentos reais.

A Figura 10 mostra a função de transferência do pulso de formação cosseno levantado. Os excessos laterais no espectro do sinal de resposta são representados por α (alfa). Esse fator variável, que pode crescer até o valor máximo de uma freqüência limite e decair a partir dessa até seu valor inicial, variando de maneira suave, é conhecido em PDS como *rollof*. Em sistemas MIR, o termo dual do *rollof* é denominado *cutoff*, ou simplesmente *cut*, e representa uma perda abrupta do nível da freqüência de resposta acima ou abaixo da freqüência limite (*i.e.*, quando o valor de α se aproxima do máximo e os excessos de largura de banda assintotam os limites laterais do pulso quadrado).

A família de pulsos do Cosseno Levantado é dada pela seguinte equação [29]:

$$x_{rc}(t) = \operatorname{sinc}\left(\frac{t}{T}\right)\left[\frac{\cos\left(\frac{\alpha\pi t}{T}\right)}{1 - \left(\frac{2\pi t}{T}\right)^2}\right],$$

onde α é o excesso da largura de faixa, citado no parágrafo anterior.

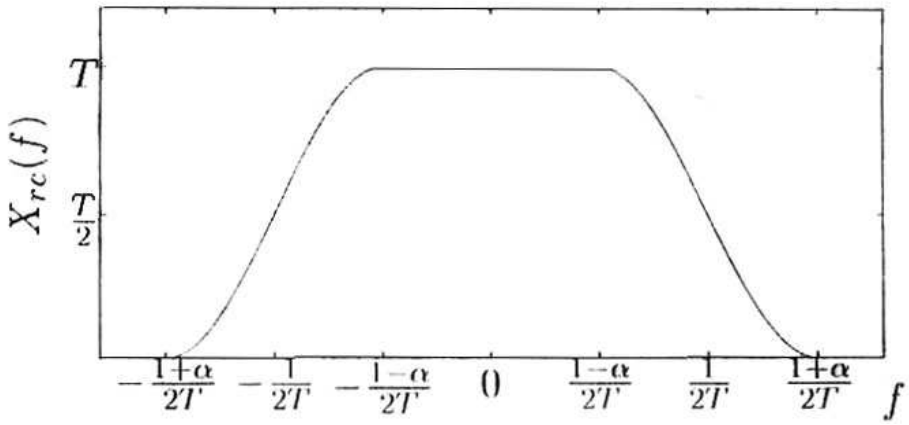


Figura 10. Função de Transferência do pulso de formação do Cosseno Levantado (Figura retirada de [29])

2.2.3 MFCC (*Mel Frequency Cepstral Coefficients*)

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 especialmente aplicações de compressão de dados [30]. Autores como [13] e [14], utilizam MFCC em conjunto com outras características que podem ser extraídas do sinal para auxiliar seus processos de classificação automática de áudio.

MFCC é uma escala perceptual de *pitches* (tons) baseada na escala Mel. A escala Mel foi proposta inicialmente por Stevens e Wolkman [31] e, assim como a escala de Decibéis, destina-se a realizar medidas do 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 freqüência da nota ouvida e sua freqüência fundamental, ou seja, o *pitch*. De maneira simples, o *pitch*, é a medida da resposta humana ao estímulo recebido por um som, representando a relação entre a faixa de freqüência em que esse som se encontra e a sua freqüência fundamental. A medida dessa distância é que define exatamente o *pitch* do sinal.

Qualitativamente falando, o *pitch* é a percepção do quão agudo ou grave um som é em relação ao acorde base do qual deriva (em teoria musical, o acorde base é também conhecido como tom médio da nota). Sendo assim, um acorde é mais alto (mais agudo), ou possui maior *pitch*, se possuir uma alta freqüência e, de modo análogo, um acorde é mais baixo (mais grave), ou possui menor *pitch*, se possuir uma baixa freqüência [23].

2.2.4 Energia

A Energia de um sinal é uma medida quantitativa que está diretamente ligada aos máximos e mínimos das suas freqüências. Alguns autores mostram que o uso combinado de medidas como energia e magnitude média auxilia de forma efetiva na descoberta de porções do sinal onde principia ou termina o silêncio (ruído de fundo), além capacidade de fazer distinção entre vozes masculinas e femininas. Dessa forma, aplicações para a área de reconhecimento de voz, fonemas e sílabas são alcançados sem maiores problemas ou custos (financeiros e computacionais) [5].

O valor da Energia de um sinal discreto pode ser calculado como [5]:

$$E = \sum_{m=-\infty}^{\infty} x^2[m]$$

onde *x[m]* é o valor da amplitude do sinal no ponto *m*.

2.2.5 Magnitude

A Magnitude de um sinal é um valor sempre positivo capaz de indicar alguns aspectos quantitativos como, por exemplo, o seu volume em decibéis. Usualmente, uma série de operações matemáticas como soma e subtração pode ser realizada entre magnitudes de sinais para diversos fins, entre eles a suavização de altas freqüências para auxiliar na verificação do início e fim das zonas de silêncio do sinal (*i.e.*, ruído de fundo). A comparação entre magnitudes é comumente realizada com escalas logarítmicas e em PDS, ela é definida pelo resultado da Transformada de Fourier aplicada ao sinal de entrada previamente amostrado. Esse procedimento é utilizado mais comumente em aplicações que lidam com processamento de voz [5], onde a suavização do sinal é necessária para tratamentos posteriores de reconhecimento dos segmentos de voz produzidos sem a vibração das cordas vocais (sons *unvoiced*) ou com a vibração de cordas vocais (sons *voiced*). A Magnitude de um sinal discreto é calculada como [5]:

$$M = \sum_{m=-\infty}^{\infty} |x[m]|$$

onde *x(m)* é o valor da amplitude do sinal no ponto *m*. Apesar de ser bem mais simples de calcular do que a energia, a Magnitude apresenta resultados menos precisos quando aplicada a discriminação de áreas de ruído ou silêncio.

2.2.6 Molduras de Análises

Molduras de Análises são porções consideravelmente menores de um sinal que ainda possuem características suficientes para permitir a extração de dados efetivamente usáveis para diversos tipos de processamento (como classificação do sinal em um ou mais gêneros musicais, por exemplo). A principal vantagem de particionar um sinal em diversos outros segmentos menores é que o ganho em tempo em seu processamento é considerável para o caso de bases de dados muito grandes. Essas porções precisam ser pequenas o suficiente para manter as características da freqüência e do espectro da magnitude relativamente estáveis e, ainda assim, ser suficientemente grandes para conter dados relevantes para os processamentos que se deseja realizar [14]. O método de divisão dos sinais em molduras de análises é bastante popular e largamente utilizado por diversos autores e projetos renomados na área como [5], [13], [14] e [17]. Em geral, esse sistema é amplamente utilizado nas áreas de tratamento e processamento de voz, classificação e mineração de dados em MIR. Tanto a Energia quanto a Magnitude podem ser calculadas para um sinal completo quanto apenas para uma moldura dele.

2.2.7 A Transformada de Fourier de Tempo Curto

A Transformada de Fourier de Tempo Curto (STFT - *Short Time Fourier Transform*) é uma transformada relacionada à transformada de Fourier que serve para determinar a freqüência e a fase de porções do sinal enquanto ele muda com o tempo. Há dois tipos de abordagem para o uso da STFT em PDS dependendo do tipo do sinal, se contínuo ou discreto. Para o primeiro caso, idealiza-se uma janela que vai ser multiplicada pelo sinal e as transformadas são obtidas à medida que a janela é deslizada pelo eixo. No segundo caso, o sinal é dividido em porções menores (Janelas de Análise ou *Analysis Frames*) e cada uma delas sofre uma Transformada. O resultado é adicionado a uma matriz, que guarda todos os dados sobre a magnitude e a fase do sinal.

O uso de STFT é bastante comum em abordagens que utilizam molduras de análises para auxiliar na extração de características de um conjunto de sinais que já passaram pelo processo de digitalização. Trabalhos como [14] e [32] são exemplos de alguns autores que utilizam a técnica.

2.2.8 Taxa de Passagem Pelo Zero

Em PDS, passagens pelo zero ocorrem todas as vezes que amostras sucessivas de um sinal possuem sinais algébricos diferentes (*i.e.*, ora positivos, ora negativos). A taxa de passagens pelo zero é uma medida simples do conteúdo da freqüência do sinal e pode representar, além de outros fatores, a quantidade de ruído que o sinal possui [5] [14]. Por exemplo, um período de um sinal senoidal tem duas passagens pelo zero, como pode ser visto na Figura 11.

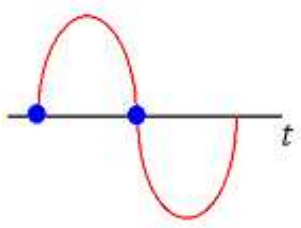


Figura 11. A função senoidal apresenta duas passagens pelo zero em um ciclo completo.

A Taxa de Passagem pelo Zero também é uma métrica que auxilia PDS na distinção entre sons audíveis, inaudíveis e ruídos de fundo. Comparam-se os três sinais analisando a quantidade de energia e a quantidade de passagens pelo zero que ele possui. Por exemplo, sons audíveis possuem altas quantidades energia e baixas taxas de passagem pelo zero. Sons inaudíveis ou fricativos possuem baixas quantidades energia, mas altas taxas de passagem pelo zero, enquanto o silêncio ou ruído de fundo, possui baixa energia e baixas taxas de passagem pelo zero. Dessa forma, pode-se isolar apenas as porções relevantes do sinal e eliminar o que não se deseja estudar.

Capítulo 3

Softwares de Apoio

Foram utilizados diversos *softwares* de apoio para realização das várias etapas envolvidas no pré-processamento de sinais, treinamento das redes, extração dos vetores de características e implementação de novas funcionalidades. São descritos a seguir, os principais programas auxiliares utilizados neste trabalho.

3.1 MARSYAS (*Music Analysis Retrieval and Synthesis for Audio Signals*)

O MARSYAS é uma suíte de aplicativos para análise, síntese e extração de características de arquivos de áudio, criada por George Tzanetakis e colocado em domínio público, regido pela GPL (*GNU Public License*) [33]. 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 MIR. Existe ainda a possibilidade de combinar esses módulos em redes colaborativas que podem ser controladas e modificadas dinamicamente sem interromper o processamento corrente dos dados. Também é fornecido um suporte completo de entrada e saída para diferentes formatos de arquivos de áudio, processamento de sinais e módulos de aprendizado de máquina (como por exemplo, geração de arquivos ARFF, para o WEKA, explicados na Seção subseqüente).

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. Usuários iniciantes são capazes de criar redes de objetos com os módulos pré-compilados e experimentar combinações de funcionalidades como filtros, extração de médias e variâncias dos MFCC (*Mel-Frequency Cepstral Coefficients*), ou extração de características baseadas na magnitude das STFT (*Short Time Fourier Transform*) (*i.e.*, médias e variâncias de Centróide Espectral, *Rollof*, etc). Usuários experientes podem criar redes complexas para atender os requerimentos de sistemas com propósitos mais específicos, escrever seus próprios códigos e interagir com os módulos pré-compilados para auxiliar nos trabalhos menores de extração de dados e análise de características.

Os executáveis pré-compilados que fazem parte do MARSYAS podem agir em arquivos de áudio separados ou em coleções de dados. As coleções consistem simplesmente de uma lista em texto puro, com os nomes dos arquivos separados por um caractere de nova linha. As listas

devem ser criadas com a extensão MF, para que sua leitura em processamentos futuros ocorra sem problemas.

Em sistemas GNU/Linux as coleções podem ser criadas através do utilitário *mkcollection*, que recebe dois parâmetros de entrada: o primeiro é o nome da coleção (seguido da extensão MF) e, o segundo, o diretório que contém os arquivos a serem adicionados à lista. Para criar uma lista das músicas que pertencem ao gênero Rock, o seguinte comando poderia ser usado: `# mkcollection rock.mf music/rock/`. Todos os arquivos contidos no diretório rock/, incluindo subdiretórios e seus respectivos conteúdos serão adicionados à coleção `rock.mf`. O ***mkcollection*** reconhece e adiciona automaticamente à coleção, arquivos com o formato AU e WAV, mas não verifica se eles estão com compressões válidas, suportados pelo MARSYAS.

Adicionalmente, se a biblioteca *libmad* [34] estiver devidamente instalada no sistema, o suporte para MP3 pode ser habilitado quando da compilação do MARSYAS, permitindo que todo o trabalho seja realizado com esse tipo de arquivo, ao invés dos formatos padrão (AU ou WAV). O MAD (*MPEG Audio Decoder*) é um decodificador de áudio de alta qualidade com suporte para as três camadas do MPEG (onde a camada 3 representa o formato MP3). Suas principais características são: capacidade de prover saídas de 24-bits PCM, computação da codificação dos dados usando aritmética de ponto fixo (ao invés de aritmética de ponto flutuante), uma implementação baseada inteiramente nos padrões ISO/IEC (*International Organization for Standardization*/*International Electrotechnical Commission*) e estar sob regimento da GPL.

Arquivos MP3 também são adicionados automaticamente à lista pelo *mkcolletion*, mas do mesmo modo que para os outros formatos, nenhuma verificação sobre a sua validade como arquivo de entrada para o MARSYAS será realizada. Alternativamente, uma coleção pode ser criada com o comando `ls`, usado para listar o conteúdo de diretórios em sistemas similares ou baseados no Unix. Para imitar o funcionamento do ***mkcollection*** e conseguir listar os subdiretórios e seus respectivos conteúdos, pode-se usar o seguinte comando: `# ls -R music/rock/*.mp3 > rock.mf`. O arquivo `rock.mf` é escrito como um arquivo de texto comum e pode posteriormente ser tratado de maneira apropriada para retirada de caracteres indesejados com comandos simples como *grep* ou a partir de editores comuns em sistemas similares ao Unix, como o VI ou Emacs.

Além do ***mkcollection***, diversos outros binários foram úteis no desenvolvimento deste trabalho. Dentre eles destacam-se o ***extract*** e o ***bextract***, para a extração dos vetores de características dos arquivos de áudio, o ***sfinfo***, que fornece informações a respeito do arquivo, como taxa de amostragem, quantidade de canais, etc e o ***pitchextract***, utilizado para extração do contorno fundamental de freqüência (tons), ou *pitch*, para sinais de áudio monofônicos e polifônicos. A seguir são descritos com mais detalhes cada um dos binários de maior relevância para este estudo.

O ***extract*** é um executável usado para extração de vetores de características de um arquivo de áudio por vez. Com ele foram realizados alguns testes iniciais como a extração de médias e variâncias do Centróide Espectral e *Rollof*. Um exemplo do uso de *extract* para obtenção de médias e variâncias dos MFCC da música *Yellow Submarine* é dado a seguir: `# extract -e SVMFCC YellowSubmarine.wav`.

Outro utilitário bastante explorado foi o ***bextract***, que realiza basicamente as mesmas funções que o ***extract***, contudo, possui a capacidade adicional de receber e tratar mais de um arquivo de áudio por vez. O ***bextract*** pode extrair vetores de características de coleções inteiras, e é capaz de gerar arquivos ARFF, usados para alimentar o WEKA, onde os testes e treinamentos podem ser levados adiante. Apesar de o MARSYAS possuir um classificador Gaussiano simples que pode ser usado como algoritmo de aprendizado de máquina, o autor resolveu portar a maior parte dessa implementação para o WEKA, uma vez que é um aplicativo mais especializado e direcionado para esses propósitos. Ainda assim, é possível utilizar o *bextract* para realizar

classificação automática de arquivos de áudio em tempo real criando *plug-ins* com extensão MPL, que alimentam um outro binário da suíte, chamado *sfplugin*. A criação desses *plug-ins* se faz ao mesmo tempo em que se faz a criação dos arquivos ARFF utilizados no WEKA.

A seguir, um breve exemplo de como utilizar o *bextract* para criar um *plug-in* capaz de distinguir um conjunto de músicas (representado pelo arquivo **`musicas.mf`**) de um arquivo que contém apenas voz (representado pelo arquivo **`discurso.mf`**): **`# bextract -e STFT musicas.mf discurso.mf -w entradaParaWeka.arff -p plugin_classificacao.mpl`**. E aqui, o uso do *plug-in* com o utilitário *sfplugin*: **`# sfplugin -p plugin_classificacao.mpl new.wav`**. Nesse caso o arquivo **`new.wav`** vai ser classificado de acordo com os parâmetros contidos em **`plugin_classificacao.mpl`**.

No presente trabalho, os módulos *bextract* e *sfplugin* foram utilizados como descrito no parágrafo anterior apenas para realizar testes comparativos com os classificadores implementados no WEKA.

Outro dos módulos que acompanham o MARSYAS e que foi largamente utilizado é o ***sfinfo***. Esse binário representa um papel importante para o auxílio da extração de características dos arquivos de áudio. Com ele é possível coletar informações valiosas para o processamento dos dados, como quantidade de canais de áudio e taxa de amostragem dos sinais. Sua utilização em conjunto com os binários *bextract* e *sfplugin* ajudou nas pré-classificações e no pré-processamento das bases de dados de entrada. A Figura 12 ilustra a extração de um conjunto padrão de características de uma música em formato WAV.

```
root@moka:~/Desktop# sfinfo PinkFloyd-UsAndThem.wav
File Name       PinkFloyd-UsAndThem.wav
File Format     Microsoft RIFF WAVE Format (wave)
Data Format     16-bit integer (2's complement, little endian)
Audio Data      81308160 bytes begins at offset 44 (2c hex)
                2 channels, 20327040 frames
Sampling Rate   44100.00 Hz
Duration        460.931 seconds
```

Figura 12. Visualização de características básicas de um arquivo WAV.

Existem alguns projetos menores, internos ao MARSYAS, que constituem interfaces gráficas para realização das tarefas de extração de vetores de características e todas as outras funcionalidades que são acessíveis através de linhas de comando. Algumas dessas interfaces foram idealizadas em Java e outras em QT4 (uma biblioteca C++ para desenvolvimento de interfaces). O uso desses módulos favorece trabalhos cujas bases de dados são relativamente pequenas, no entanto, para o caso de bases de dados grandes e com muitos parâmetros, a interface de linha de comando é mais apropriada, especialmente por conta de seu maior desempenho. Como o presente trabalho utiliza uma base de dados relativamente extensa e com muitos parâmetros, nenhuma das implementações de GUI's (*Graphical User Interface*) foi utilizada.

3.2 WEKA (*Waikato Environment Knowledge for Analysis*)

O WEKA [35] é um *software* desenvolvido na Nova Zelândia pela Universidade de Waikato, utilizando a linguagem Java. Atualmente, o WEKA e seu código fonte estão em domínio público, regidos pela GPL. 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*. 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).

O WEKA utiliza um formato específico de arquivo, que recebe a extensão ARFF, que armazena os dados de uma maneira específica. Como apresentado na Seção anterior, o MARSYAS exporta o resultado das análises e extrações de características dos arquivos de áudio diretamente para arquivos ARFF, já corretamente formatados. A partir desses arquivos os treinamentos puderam ser concluídos de maneira bastante prática e direta. Um exemplo detalhado de um dos arquivos ARFF encontra-se no Apêndice 1.

A interface gráfica do WEKA é composta por quatro módulos diferentes. Cada módulo permite que o usuário realize praticamente as mesmas funções de maneiras levemente distintas. A Figura 13 mostra a tela inicial do WEKA e as quatro opções de trabalho disponíveis. A primeira opção, denominada *Simple CLI*, fornece um modo de trabalho simples em linha de comando, ideal para máquinas com pouco poder de processamento e que enfrentam problemas de desempenho para trabalhar com telas gráficas. Além disso, a interface *Simple CLI* supre a necessidade de sistemas que não possuem seus próprios aplicativos de linha de comando (ou ainda, auxilia o desenvolvedor que está trabalhando em sistemas cujas interfaces de linha de comando não são suficientemente poderosas). A segunda opção, *Explorer*, é uma tela no estilo “aponte e clique”, utilizada para pré-processamentos e aplicação de técnicas de aprendizado de máquina.

Os outros dois modos, *Experimenter* e *KnowledgeFlow*, também podem ser utilizados para os mesmos fins, embora possuam apresentações gráficas distintas. O módulo *Experimenter* é um ambiente para realização de experimentos que permite a condução de testes estatísticos entre diferentes esquemas de aprendizado. O *KnowledgeFlow*, por sua vez, provê praticamente as mesmas funcionalidades que se pode obter com o *Explorer*, com uma pequena diferença, a sua interface suporta o estilo “arraste-e-solte” (*drag-and-drop*) de interação. Ademais, o módulo *KnowledgeFlow* permite a realização de técnicas de aprendizado incremental.

Para o presente estudo, foi utilizada a versão 3.4.8a do WEKA e todos os testes e treinamentos foram realizados com apenas uma das quatro implementações de interação com o usuário, o módulo *Explorer*.

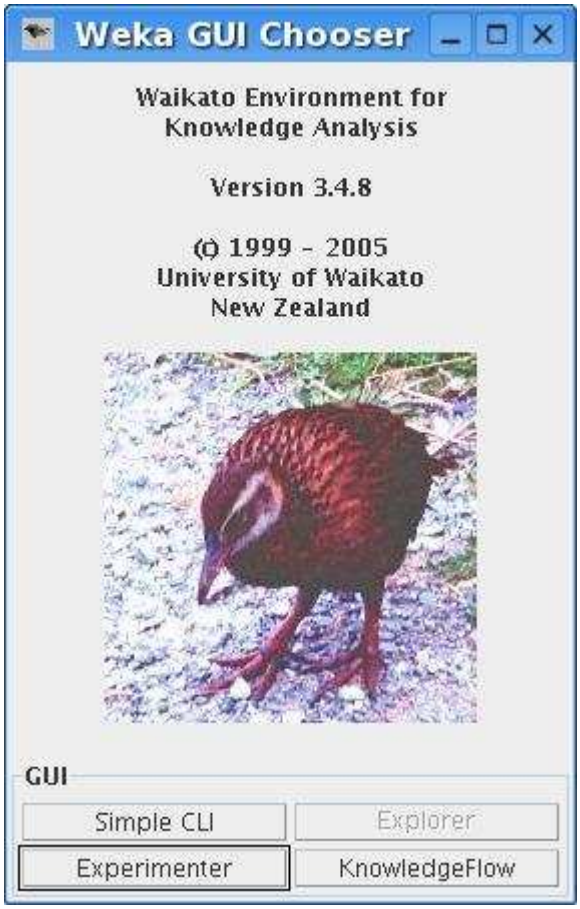


Figura 13. Tela inicial do WEKA.

A tela do módulo Explorer (Figura 14) permite que o pesquisador realize uma série de pré-processamentos com os dados de entrada. Na aba *Preprocess* pode-se escolher entre diversas aplicações de pré-processamento como discretização, normalização, transformação e combinação. As bases de dados poderão ser carregadas a partir de arquivos (ARFF no caso presente), a partir de uma URL (botão *Open URL*...) ou a partir de um Banco de Dados (botão *Open DB*...). Além disso, o WEKA oferece diversas opções de filtros para o pré-processamento, mas não foram utilizadas quaisquer implementações desses últimos no atual estudo. Ao abrir pela primeira vez o programa, todas as abas além da *Preprocess* estarão desabilitadas e só se tornarão ativas a partir do momento que um arquivo de dados válido for carregado (e possivelmente pré-processado). Esse tipo de comportamento impede que o pesquisador cometa certos erros quando estiver lidando com suas bases de dados. Ao forçar uma seqüência lógica de passos a seguir, os desenvolvedores do WEKA aproveitaram para simplificar a interação com seu *software*, traçando uma maneira correta para sua utilização, evitando, dessa forma, que a maioria dos potenciais erros relacionados ao tratamento dos dados ocorra.

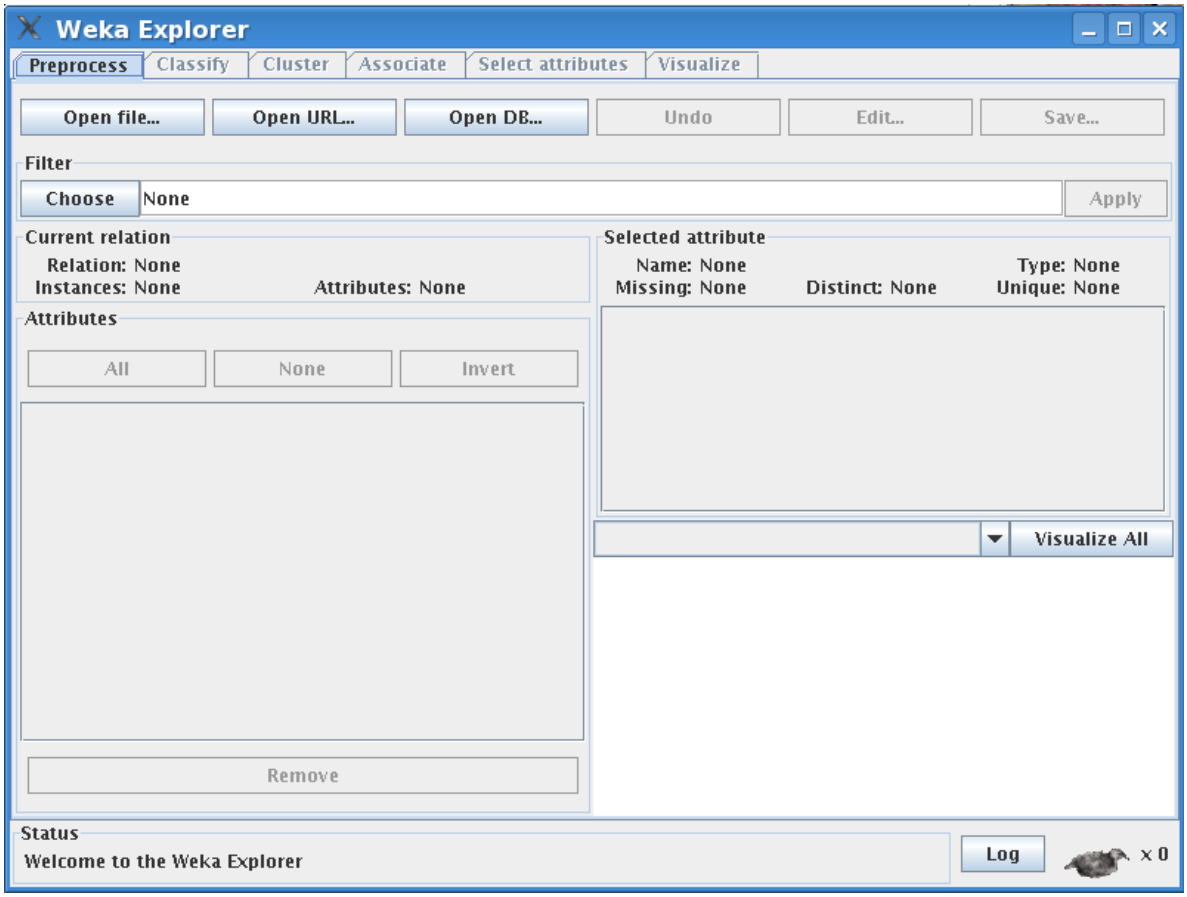


Figura 14. Interface *Explorer*.

A aba *Classify* contém as implementações dos classificadores fornecidos pelo WEKA. Um amplo conjunto dos mais populares classificadores encontra-se idealizado no programa. Entre eles destacam-se as Redes Bayesianas, Máquinas de Vetor de Suporte (*Support Vector Machines* - SVM), kNN (*k-nearest Neighbor*), MLP (*Multi Layer Perceptron*), Listas de Decisão e diversos outros. Também é possível realizar regressões a partir da aba *Classify*.

A aba *Cluster* permite ao pesquisador criar experimentos que sejam capazes de realizar associações com os dados usando *clusters*. Essa técnica consiste em separar os dados em porções menores que o total, formando conjuntos ou classes que possuem características similares e certo grau de similaridade entre si. Se o grau de similaridade entre elementos de conjuntos diferentes se cruza, esses elementos podem ser promovidos para outros conjuntos ou mesmo passar a não fazer parte de conjunto algum (nesses casos diz-se que foi detectada uma novidade). A aba *Associate* auxilia a rede a aprender regras de associação a partir do conjunto de dados. A aba *Select Attributes* permite a escolha dos atributos mais relevantes para o processamento de dados e, por fim, a aba *Visualize* permite a visualização interativa da plotagem em gráficos bidimensionais.

No trabalho atual, foram utilizadas as técnicas kNN [36] e MLP [37] para classificação dos arquivos de áudio, em conformidade com a proposta inicial de estudar técnicas diferentes para obter bases de comparação com a nova implementação proposta. Não faz parte do escopo desse trabalho explanar detalhadamente o funcionamento das técnicas de classificação utilizadas, uma vez que ambas são largamente conhecidas e utilizadas para diversos fins, em inúmeros projetos de maturidade sólida no meio científico e comercial. Existem ótimas referências para pesquisas mais profundas a respeito do assunto nos seguintes trabalhos: [14], [35], [36] e [37].

Na Figura 15, pode-se ver o carregamento da base de dados utilizada nos processos de classificação. Todas as bases de dados possuem 16 atributos, que comportam os vetores de características, mais 1 atributo extra, que é a saída de dados (*output*)

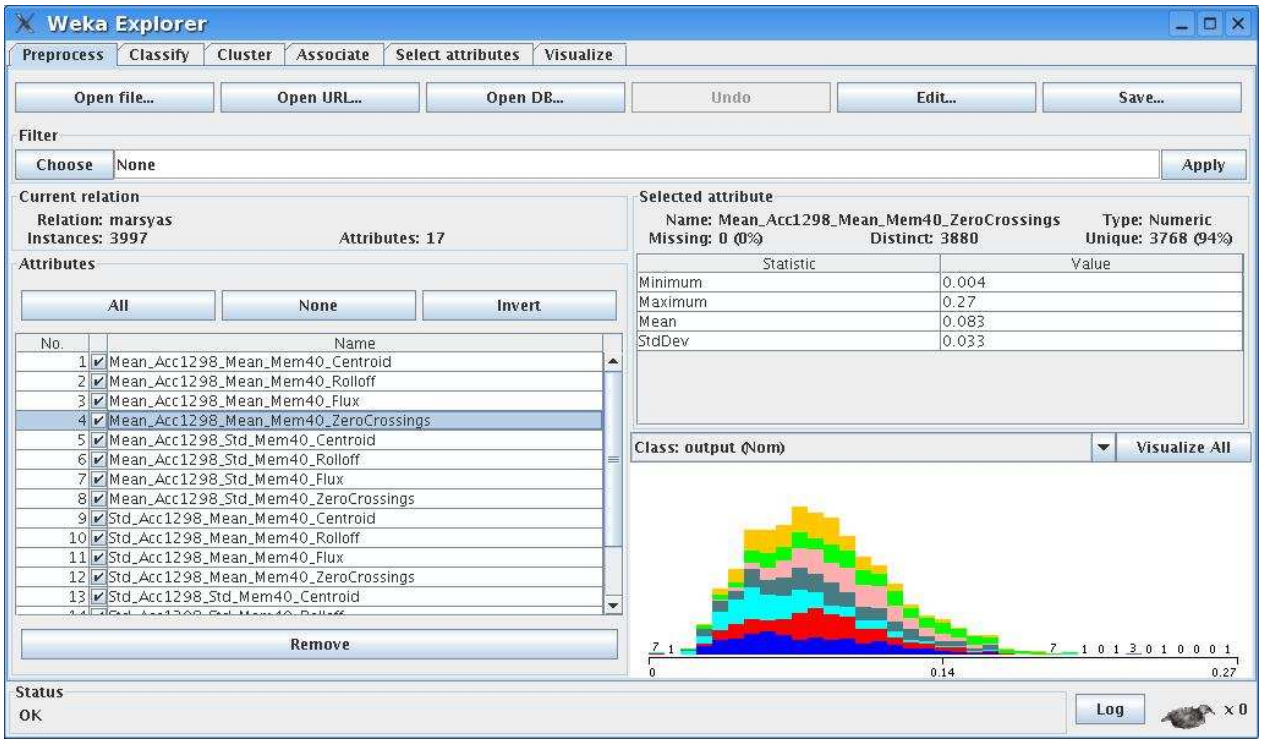


Figura 15. Carregamento das bases de dados no WEKA.

Como dito anteriormente, foram utilizadas duas técnicas para a realização das classificações, uma implementação do método MLP e do método kNN. A Figura 16 apresenta a visualização das técnicas de aprendizado do WEKA, mais especificamente do grupo *lazy*, onde se encontra o kNN (IBK), enquanto a Figura 17 mostra o grupo *functions*, onde está localizada a técnica MPL.

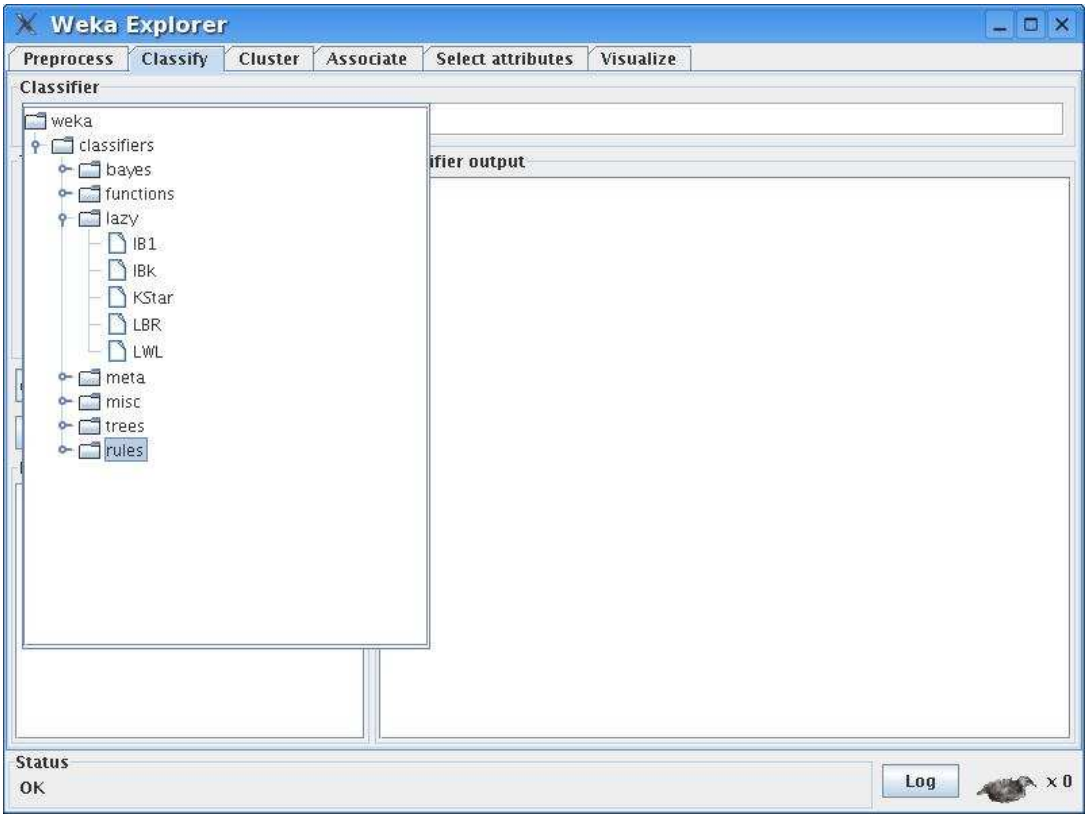


Figura 16. Visualização da técnica de aprendizado kNN, do grupo *lazy*.

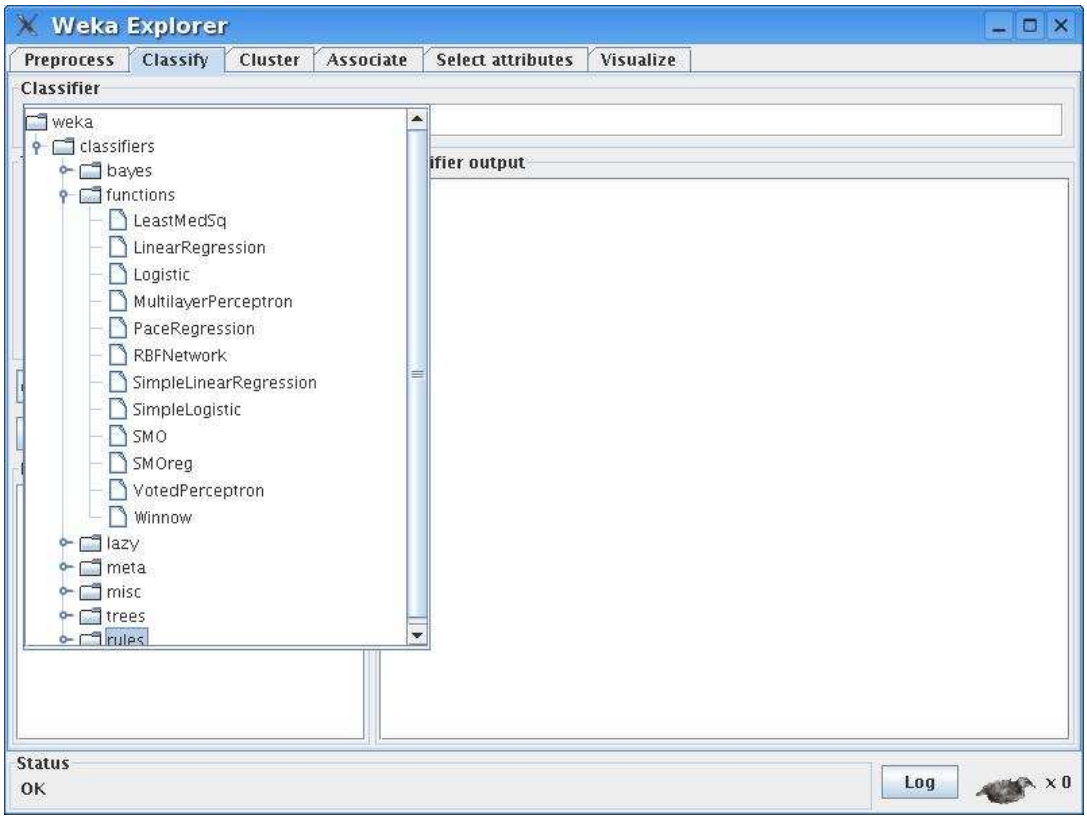


Figura 17. Visualização da técnica de aprendizado MLP, do grupo *functions*.

Capítulo 4

Classificação de Áudio: Experimentos e Resultados

As bases de dados utilizadas no presente estudo fazem parte, em sua grande maioria, da coleção particular do autor. Processos como a normalização da taxa de *bits* por segundo (*bit rate*), *downsampling*, freqüências máximas utilizadas em cada sinal e quantidade de dados pertencente a cada gênero são explanadas pormenorizadamente nesse Capítulo. Ademais, são revisitados alguns dos conceitos e pilares importantes da classificação de áudio no cenário do entretenimento mundial (visão mercadológica), bem como a importância e o crescimento dos estudos da área para o ambiente científico (visão acadêmica).

4.1 Separação de Gêneros e Bases de Dados

A separação dos arquivos de áudio em classes pode ser realizada através do uso de propriedades estatísticas ou mecanismos de aprendizagem de máquina. Em ambos os casos, a extração de vetores de características é uma das principais e mais importantes etapas. O processo consiste na obtenção de um conjunto de dados que seja suficiente para caracterizar de maneira correta um segmento de áudio.

Em geral, as características extraídas do sinal são as chamadas texturas de timbre [15] e algumas delas exercem influências mais profundas que outras quanto à qualidade da classificação. Alguns aspectos de um sinal de áudio auxiliam pobremente certos classificadores na tarefa de separação (como por exemplo, o *pitch*, em [14]), causando, desse modo, um custo computacional efetivamente desnecessário se forem incluídos no conjunto analisado. Distinguir e utilizar os aspectos do sinal que exercem maior influência na diferenciação entre os gêneros propostos (em sua maioria essas características pertencem ao domínio espectral e temporal do sinal) é uma das necessidades principais deste estudo, uma vez que essa resposta será de grande importância para apontar o algoritmo com melhor eficiência na classificação. Essa distinção foi feita com base nos trabalhos de outros autores, como [17], [14] e [32], que detalham suas experiências, indicando quais os melhores resultados obtidos a partir das diversas técnicas utilizadas.

As texturas de timbre são calculadas utilizando-se técnica de moldura de análises [32], onde uma janela de tempo colhe amostras de curta duração do sinal e processa esses dados, gerando um conjunto de informações que funciona como um indicador para os classificadores

como o WEKA. Algumas técnicas como *clustering* e kNN são capazes de confrontar esses dados e extrair informações úteis a respeito de cada arquivo em particular, acomodando-os em classes de acordo com as similaridades encontradas nos processamentos. O resultado da análise pode ser apenas um valor numérico ou um vetor de valores. A representação das texturas de timbre pode ser feita através da avaliação de certo conjunto de características [15] como Centróide Espectral [5] [14], média do sinal [5], entropia e taxa de passagem pelo zero [32] [15]. A Taxa de Passagem pelo Zero, por exemplo, é capaz de indicar alguns comportamentos específicos, como ruídos contidos no sinal.

Dependendo da quantidade de características que se deseja utilizar para fazer a análise, o custo computacional da classificação aumenta grandemente sem retornar maiores vantagens para a precisão do classificador, tornando o trabalho do algoritmo desnecessariamente longo. Dessa forma, evitou-se coletar mais características do que as suficientemente necessárias para classificar corretamente um segmento de áudio.

Os pré-processamentos, as separações e manipulação dos dados das bases são explicitados a seguir.

4.1.1 Pré-processamentos das Bases de Dados

Foram realizados dois pré-processamentos nos arquivos de áudio do conjunto de testes. O primeiro tem a finalidade de encontrar as porções do sinal que possuam as características mais relevantes para a classificação, evitando o processamento de espaços que contenham dados com pouca ou nenhuma relevância para a distinção entre as classes. A extração das porções de dados nesse primeiro passo foi realizada com o auxílio do *software* de apoio MARSYAS, através do comando `bextract`, cuja utilização já foi pormenorizadamente explicada no Capítulo 3. O segundo pré-processamento consistiu em realizar um *downsampling* dos sinais com fins de diminuir o custo computacional do classificador sem afetar o desempenho do correlacionamento das classes. Depois de finalizados esses dois passos iniciais, o algoritmo ganha em desempenho, pois necessita processar porções de dados significativamente menores sem necessariamente perder os detalhes essenciais que indicarão como separar os gêneros de cada classe.

Duas bases principais de áudio digital foram utilizadas para realizar os testes com o algoritmo de classificação. O primeiro conjunto é uma base já conhecida dos trabalhos [15] e [14], contento 1000 títulos no total. Há 10 gêneros pré-existentes para essa base: *Blues*, *Classical*, *Country*, *Disco*, *Hip-hop*, *Jazz*, *Metal*, *Pop*, *Reggae*, e *Rock*. Cada gênero possui 100 músicas compactadas em formato AU, com duração de 30 segundos cada. AU é um formato de áudio bastante simples, que possui um cabeçalho identificador de seis palavras de 32 *bits*, seguido dos dados que compõem efetivamente o áudio, organizados no estilo *big-endian* (*bytes* de maior ordem vêm primeiro). O formato foi desenvolvido pela Sun Microsystems e por um tempo, foi o padrão para arquivos de áudio em sistemas Unix [38].

Uma vez que essa base de dados faz parte de trabalhos de reconhecimento e credibilidade sólidos no meio acadêmico, ela foi utilizada no treinamento dos algoritmos de classificação. Com exceção dos gêneros *Hip-Hop*, *Metal*, *Disco* e *Country*, que não se encontram presentes no segundo grupo de dados, todos os outros arquivos foram utilizados nos treinamentos.

A segunda base de dados consiste de 3.997 títulos de arquivos de áudio, comprimidos inicialmente como MP3 à taxa de 128Kbps e freqüência de 44.100Hz, que fazem parte da coleção particular do autor desta monografia. Os gêneros principais que constituem o segundo conjunto são os seguintes: *Blues*, *Classical*, *Jazz*, *Pop*, *Reggae*, *Rock* e *Samba*. Observa-se claramente, pela comparação com a lista de gêneros mostrada no parágrafo anterior, que todos os gêneros do segundo grupo encontram-se no primeiro, com exceção de *Samba*, que constitui a única novidade no conjunto a classificar. Como essa classe não existe na base de dados de testes, parte do

conjunto de músicas pré-rotulado como *Samba* foi transportada para o grupo de testes e, o restante dos arquivos, passou a constituir os dados novos a classificar.

Uma vez que o segundo conjunto possui 7 gêneros, a divisão acomodou equitativamente em cada classe, 571 arquivos. Todas as músicas passaram por um processo de *downsampling*, para diminuir pela metade a freqüência original na qual se encontravam. A necessidade de realização desse processo encontra explicação no fato de que as características das texturas de timbre do sinal são preservadas, enquanto o arquivo com menor freqüência torna-se efetivamente menor e, sendo assim, pode ser mais rapidamente processado pelo classificador. Trabalhou-se, dessa forma, com arquivos MP3, comprimidos a uma taxa de 128Kbps, freqüência de 22.050Hz e 30 segundos de duração.

Para alguns dos gêneros existem subclassificações e as Tabelas 1 a 7 ilustram algumas delas. No entanto, a proposta deste trabalho não leva em conta a distinção entre as subclasses existentes, apenas entre os gêneros principais. Uma vez que as diferenças significativas entre as subclassificações de um gênero principal estão raramente dentro de um padrão bem definido, por conta da própria característica livre e "despojada" que a música possui, é praticamente impossível delinear as nuances determinantes de uma subclassificação ou outra. Dessa forma, todas as subclasses de *Rock* deverão ser consideradas corretas se forem classificadas como *Rock* apenas, todas as subclasses de *Pop* estarão classificadas corretamente se forem rotuladas como *Pop* apenas e assim por diante.

É necessário deixar claro que as Tabelas ilustram apenas uma pequena porção da grande diversidade e pluralidade de estilos que pode ser encontrada hoje no meio musical, mas não é, de maneira alguma, uma classificação definitiva, tampouco exaustiva dos gêneros que vêm se formando a partir dos existentes.

Tabela 1. Subclassificações para o gênero *Pop.*

Pop
Pop Rock
Jpop
BritPop
Synth Pop
Dream Pop
Twee Pop

Tabela 2. Subclassificações para o gênero *Rock.*

Rock
Rock & Roll
Gothic Rock
Indie Rock
Punk Rock
Post Punk
Glam Rock
Psychedelic Rock
Alternative Rock
Folk Rock
Folk

Tabela 3. Subclassificações para o gênero *Jazz.*

Jazz
Free Jazz
Avant-Garde
Fusion
Bebop
Latin Jazz
Dixieland
Acid Jazz

Tabela 4. Subclassificações para o gênero *Classical.*

Classical
Antique Music
Medieval
Renaissance
Barroque
Classical
Romantic
20th Century (Post-Romatic)
Contemporary Classical Music

Tabela 5. Subclassificações para o gênero *Reggae.*

Reggae
Dancehall
Ragga
Toasting
Roots Reggae

Tabela 6. Subclassificações para o gênero *Samba.*

Samba
Samba de breque
Samba-Canção
Samba-Enredo
Samba-Reggae
Samba de Roda
Bossa Nova

Tabela 7. Subclassificações para o gênero *Blues.*

Blues
Country Blues
Jump Blues
Jazz Blues
Piano-Blues
Boogie-Woogie
Delta Blues
Soul
Soul Blues

4.2 Validação Cruzada

Para realizar uma classificação mais apurada, também foi utilizada a técnica de Validação Cruzada que, por sua vez, permite estimar a capacidade de generalização de um classificador, *i.e.*, propicia saber se os resultados não foram apenas decorados pela rede neural artificial. Formalmente, pode-se dizer que a Validação Cruzada é uma técnica estatística, que consiste em particionar o conjunto de treinamento em partes aproximadamente iguais e fazer com que o classificador utilize uma porção inicial para treinamento e as subseqüentes para validação ou teste.

Uma das abordagens possíveis para fazer validação Cruzada é separar manualmente os dados. Por exemplo, no presente trabalho, o conjunto de testes consiste de 3997 arquivos, que poderiam ser particionados em três subconjuntos, dois com 1332 arquivos e um com 1333. Realizar-se-iam, então, três execuções para cada classificador e em cada rodada um terço do conjunto de testes seria utilizado para treinamento, enquanto os dois terços restantes ficaram para realização dos testes. Para a primeira rodada, o primeiro terço seria usado como treinamento e a soma do segundo e do terceiro para testes. Para a segunda rodada, o segundo terço seria utilizado como conjunto de treinamento e a concatenação do primeiro e terceiro terços para testes. E, por fim, na terceira rodada o terceiro terço seria utilizado para treinamento e o primeiro e segundo terços para testes.

Essa abordagem permite que todo o conjunto de testes participe da classificação, gerando um resultado usualmente mais confiável. No entanto, o WEKA já realiza essa separação automaticamente quando a opção *Cross Validation* na aba *Classify* é escolhida. A Figura 18 mostra a escolha da opção *Cross Validation* com 10 blocos ou *folders* (padrão utilizado nesse trabalho).

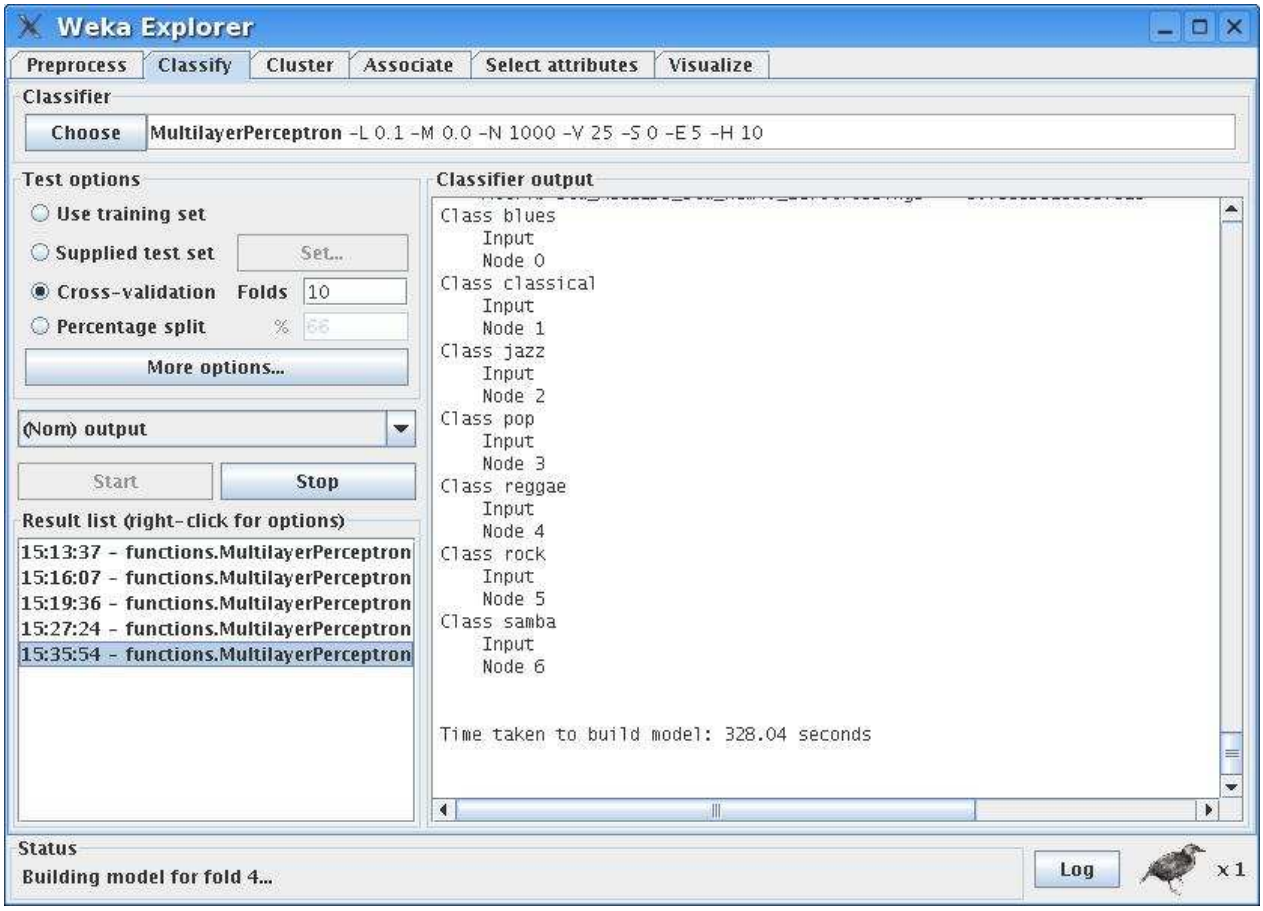


Figura 18. Configurando o WEKA para fazer Validação Cruzada.

4.3 Resultados das Classificações

4.3.1 Resultados das Classificações usando a técnica MLP

Foram utilizadas diversas configurações diferentes para os experimentos realizados com MLP, a fim de encontrar uma que melhor se adequasse ao problema. Dado que a quantidade de instâncias utilizadas nesse trabalho é significativamente maior que as usadas em todos os outros nos quais ele se baseou, os parâmetros utilizados naqueles, não puderam ser aproveitados nesse.

Fixou-se o uso de apenas uma camada escondida e uma taxa de aprendizado de 0,3. Para as épocas, foram escolhidos três valores: 1000, 10000 e 50000. A taxa de inicialização dos pesos foi tomada como 0 (sem aleatoriedade na inicialização) e 10. Por fim, foram usados 10, 20 e 40 neurônios nas camadas escondidas, totalizando uma soma de 18 (3x3x2) testes ao todo. Novamente, a decisão quanto ao número de camadas e neurônios escondidos, diferentemente de outros parâmetros utilizados nesse projeto, não foi baseada nos trabalhos de outros autores, mas sim, a partir de uma série de experimentos, onde os que produziram resultados menos significativos foram descartados. Por exemplo, todas as taxas de aprendizado abaixo de 0,3 não chegavam, sequer, a fazer o algoritmo convergir.

A função de ativação utilizada pelas duas camadas escondidas do treinamento MLP foi a sigmóide logística e o algoritmo para cada treinamento foi o *back-propagation*. Ambas as escolhas foram empíricas, apesar de constituírem o padrão utilizado pelo WEKA e de serem

citadas em trabalhos como [14] e [39]. A conexão entre os neurônios de cada camada foi total. Por fim, o conjunto de validação foi fixado em 25% e, por conta da validação cruzada, utilizando a opção 10 *folders*, cada base foi treinada e testada 10 vezes.

Uma das grandes dificuldades encontradas para realização dos treinamentos com MLP é o seu custo computacional efetivamente muito alto. Os treinamentos longos, que resultam em diferenças pouco expressivas nos primeiros momentos do ajuste das características, fazem com que a técnica sofra com tempos muitos extensos de teste até que se chegue a uma configuração ótima.

Um resumo contendo os melhores resultados das configurações e treinamentos realizados com o MLP pode ser visto na Tabela 8.

Tabela 8. Resumo dos parâmetros de configuração dos treinamentos MLP.

Quantidade de camadas escondidas	1
Neurônios nas camadas escondidas	10, 20 e 40
Taxa de aprendizagem	0,3
Conjunto de validação	25%
Épocas de Treinamento	1000, 10000 e 50000
Algoritmo de treinamento	*back-propagation*

Na Tabela 9, estão delineados os resultados mais expressivos dos testes realizados com MLP. Note-se que a configuração 5 obteve um êxito maior na classificação dos gêneros musicais, atingindo mais de 50% de acerto, apesar de ser apenas levemente mais rápida na classificação que a configuração 6.

Tabela 9. Melhores resultados MLP.

Configuração	Taxa de aleatoriedade	Número de Neurônios escondidos	Taxa de Aprendizado	Número de épocas	Tempo de treinamento	Taxa de acerto (%)
1	0	10	0,3	1000	00:25:15	47,8359
2	0	20	0,3	10000	00:34:38	48,5364
3	0	40	0,3	1000	00:50:22	48,9867
4	10	10	0,3	1000	00:20:45	48,9367
5	10	20	0,3	1000	00:30:15	50,5629
6	10	40	0,3	10000	00:38:12	50,5378

Na Tabela 10 está a matriz de confusão gerada pela configuração 5. Pode-se verificar que algumas confusões já esperadas [14] [17] como as que ocorrem entre *Blues* e *Jazz* e *Blues Classical*, demonstram a dificuldade de se delinear diferenças entre gêneros que utilizam, geralmente os mesmo instrumentos musicais, possuem a mesma batida, a mesma velocidade de tempo e vocalizações similares. Ao mesmo tempo, nota-se facilmente que o estilo Reggae foi o que conseguiu ser classificado com maior precisão, das as suas peculiaridades de batida, tempo, e taxa de ruído no sinal (Taxa de Passagem pelo Zero). É interessante perceber que a menor taxa de confusão entre gêneros ocorre exatamente entre o *Reggae* e a música Clássica (gênero *Classical*), uma vez que as similaridades instrumentais, vocais e outros detalhes como a velocidade rítmica de ambos, são completamente distintos.

Tabela 10. Matriz de confusão para o melhor resultado da MLP.

Classificação	a	b	c	d	e	f	g
a = blues	271	71	118	21	9	49	42
b = classical	64	360	93	3	0	18	33
c = jazz	95	89	243	33	20	34	57
d = pop	36	25	65	150	47	116	132
e = reggae	11	2	21	15	468	15	39
f = rock	54	44	68	96	26	205	78
g = samba	28	15	50	54	61	29	334

Adicionalmente, foram extraídas as curvas ROC (*Receiver Operating Characteristics*) de todos os gêneros para a melhor configuração, criadas automaticamente pelo WEKA. A análise dessas curvas mostra que, ainda que a classificação de instâncias corretas tenha sido baixa (pouco acima dos 50%), a probabilidade de acerto é alta, permanecendo, para a maioria dos casos, acima de 70%. As curvas ROC para os gêneros *Reggae*, *Classical* e *Samba*, são mostradas nas Figuras 19, 20 e 21. A área abaixo da curva (*Área Under ROC*) retrata a qualidade da classificação. Quanto mais próxima de 1, melhor o desempenho do classificador. Para todos os casos, inclusive nos testes com as técnicas kNN, a área sob a curva é obtida plotando as taxa de Falsos Positivos *versus* a de Verdadeiros Positivos, ou seja, cruzando a taxa de acerto *versus* a taxa de erro na classificação de cada gênero.

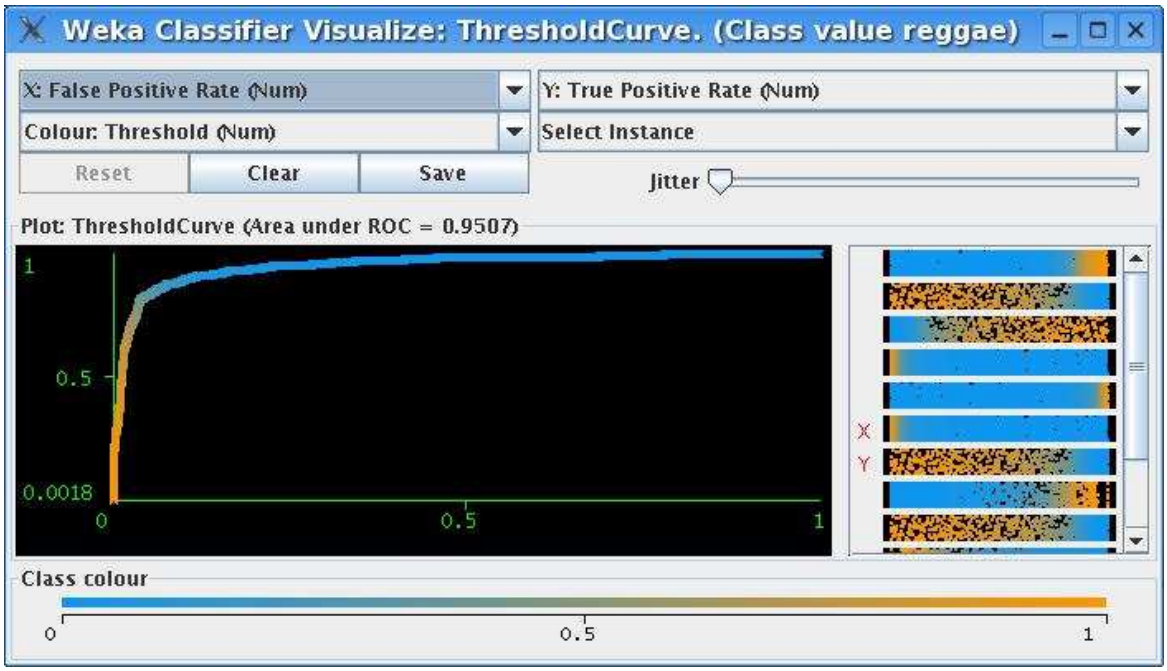


Figura 19. Curva ROC do gênero *Reggae*, para o melhor resultado da técnica MLP.

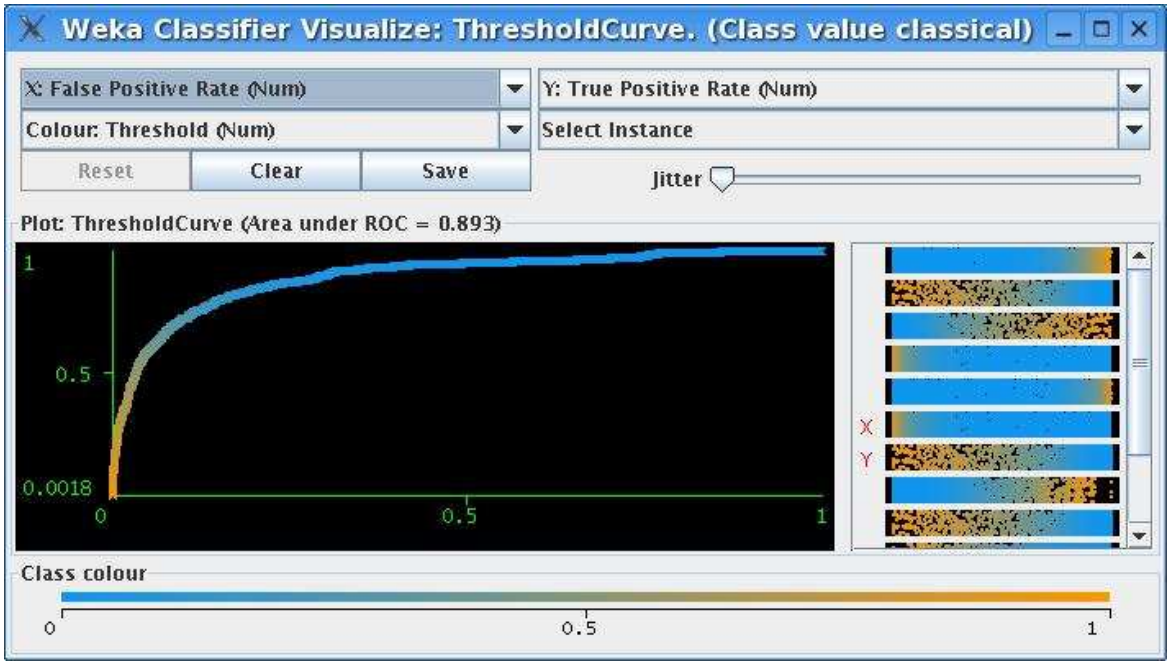


Figura 20. Curva ROC do gênero *Classical*, para o melhor resultado da técnica MLP.

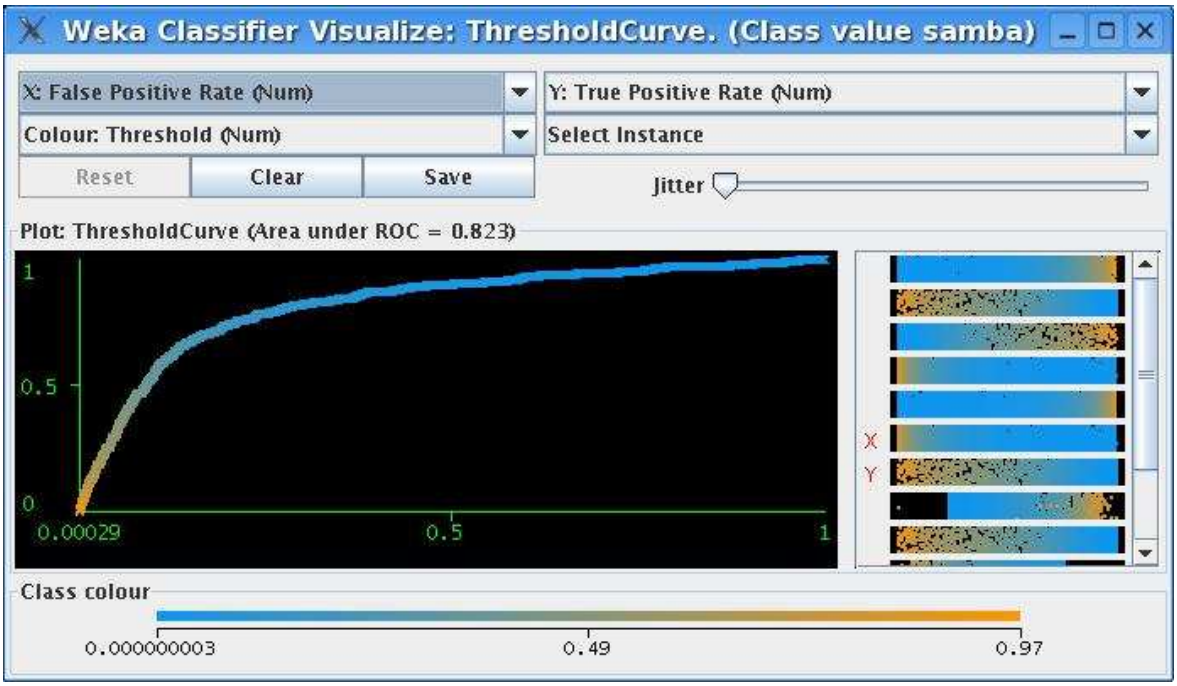


Figura 21. Curva ROC do gênero *Samba*, para o melhor resultado da técnica MLP.

Por fim, a Figura 22 mostra os erros do classificador para o melhor resultado dos testes com a técnica MLP. Os quadrados representam instâncias classificadas erroneamente e, as cruzes, mostram os gêneros classificados corretamente. Podem-se notar algumas peculiaridades acontecidas no experimento de maneira mais simples, apenas olhando para o gráfico. Por exemplo, é fácil notar que a classificação do gênero *Pop* foi relativamente pobre em relação à da classe *Reggae*.

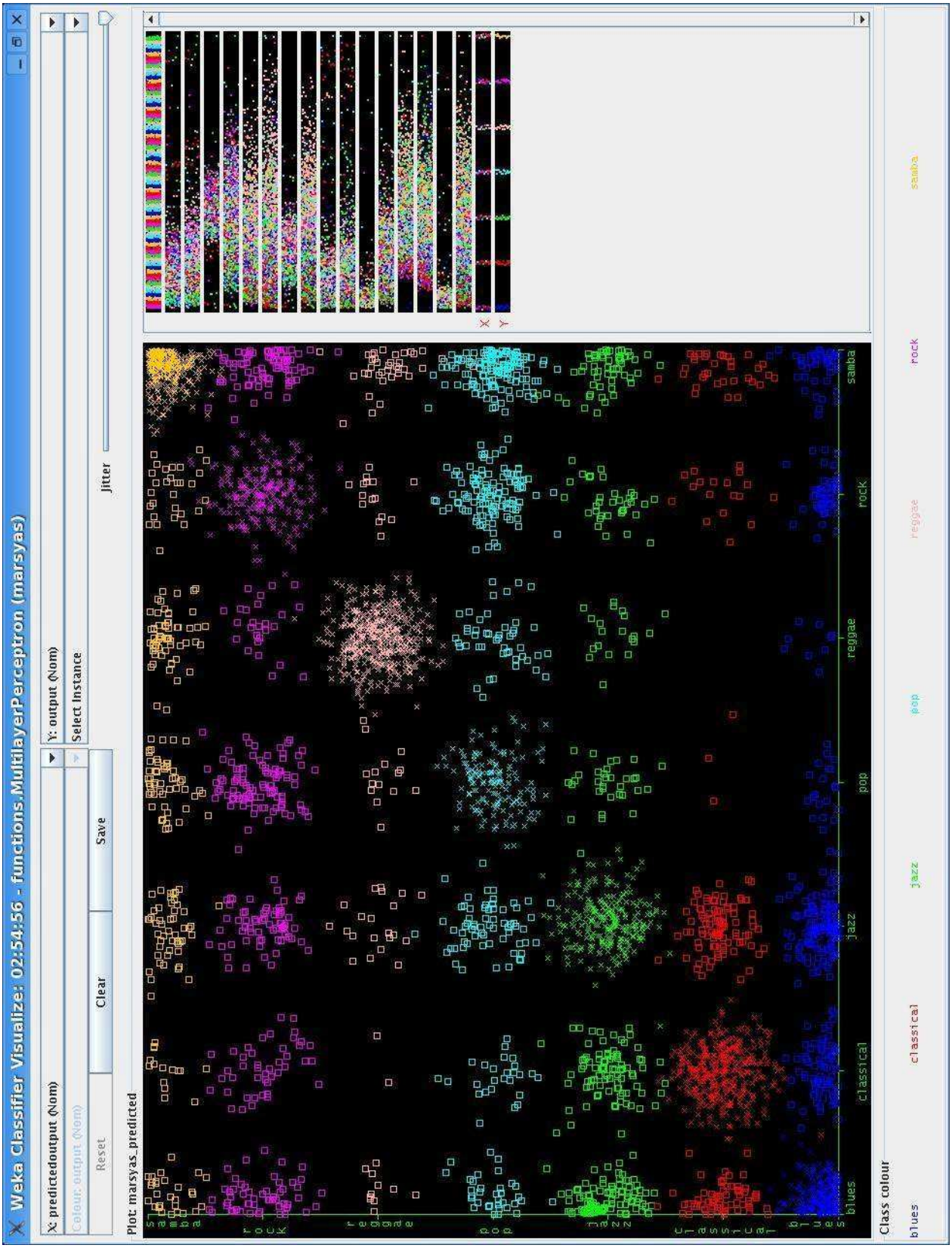


Figura 22. Gráfico dos erros de classificação para a técnica MLP. As classes aparecem no rodapé da figura, na seguinte ordem: *Blues*, *Classical*, *Jazz*, *Pop*, *Reggae*, *Rock* e *Samba*.

4.3.2 Resultados das Classificações usando a técnica kNN

Ao contrário do modo de operação dos experimentos com MLP, os parâmetros para as classificações com a técnica kNN (k-*Nearest Neighbors* – k-Vizinhos Mais Próximos) não foram obtidos empiricamente. Uma vez que a técnica possui poucos parâmetros, escolheu-se seguir as indicações da literatura e dos outros trabalhos nos quais esta monografia se baseou.

O kNN é uma técnica não-paramétrica, relativamente eficiente e bastante rápida, se comparada ao MLP. Desse modo, apenas um valor da configuração é modificado de um treinamento para o outro, o valor de k. Adotou-se, a exemplo de [14], e sob conselho da literatura [16], três valores ímpares para k: 1, 3 e 5. Novamente, e pelos mesmos motivos explicitados anteriormente na Seção 4.2, foi utilizada validação cruzada em 10 blocos. A Tabela 11 mostra os resultados dos treinamentos usando kNN.

Tabela 11. Resultados dos treinamentos com kNN.

Número de k no Treinamento	Tempo de processamento	Instâncias Classificadas corretamente	Erro médio quadrático
K = 1	00:04:30	1963 (49,118%)	0,3809
K = 3	00:03:23	1826 (45,6843%)	0,3169
K = 5	00:04:31	1834 (45,8844%)	0,3288

O melhor resultado obtido foi para o valor de k = 1, apesar de que o tempo de processamento para k = 3 foi o menor de todos. Ainda assim, nota-se que o tempo de processamento para cada experimento é significantemente menor que os tempos envolvidos nos treinamentos MLP, pois, uma vez que o método possui poucos parâmetros ajustáveis para modificar, o treinamento e a obtenção de resultados expressivos é bem mais direto se comparado à outra técnica.

Na Tabela 12, podemos ver a matriz de confusão para o melhor caso (K = 1). Como esperado, uma das mais altas taxas de confusão ocorreu com o gênero Pop, que teve uma grande porcentagem de classificações erradas como Samba. Algumas características do sinal, como Taxa de Passagem pelo Zero, e velocidade da batida, explicam o alto grau de confusão entre um gênero e outro.

Também como esperado, o gênero *Jazz* foi muito confundido com *Blues* e *Classical*, especialmente as instâncias de *Jazz Be-bop*, que possuem orquestras de metais e muita incidência do som de piano, largamente presente nas outras duas classes.

Por fim, os dois gêneros que sofreram menos influência um do outro e, por conseqüência, menos erros de classificação entre si, foram *Reggae* e *Classical*. As diferenças de batida, Taxa de Passagem pelo Zero e *Rollof*, entre cada gênero são fatores que podem explicar a corretude da classificação ente as duas instâncias.

Tabela 12. Matriz de confusão para o treinamento kNN, com k = 1.

Classificação	a	b	c	d	e	f	g
a = blues	292	55	60	50	12	58	44
b = classical	44	421	73	7	1	14	11
c = jazz	84	82	229	62	25	58	31
d = pop	50	38	66	155	42	95	125
e = reggae	15	0	17	41	415	20	63
f = rock	74	46	65	106	37	159	84
g = samba	25	21	29	106	42	56	292

Novamente, algumas curvas ROC geradas a partir do conjunto de treinamento ainda indicam que a probabilidade de acerto é alta (acima de 70%), apesar de as taxa de classificações corretas ser relativamente baixa (abaixo de 50%). As Figuras 23, 24 e 25 mostram as curvas ROC geradas para os gêneros *Classical*, *Blues* e *Reggae*.

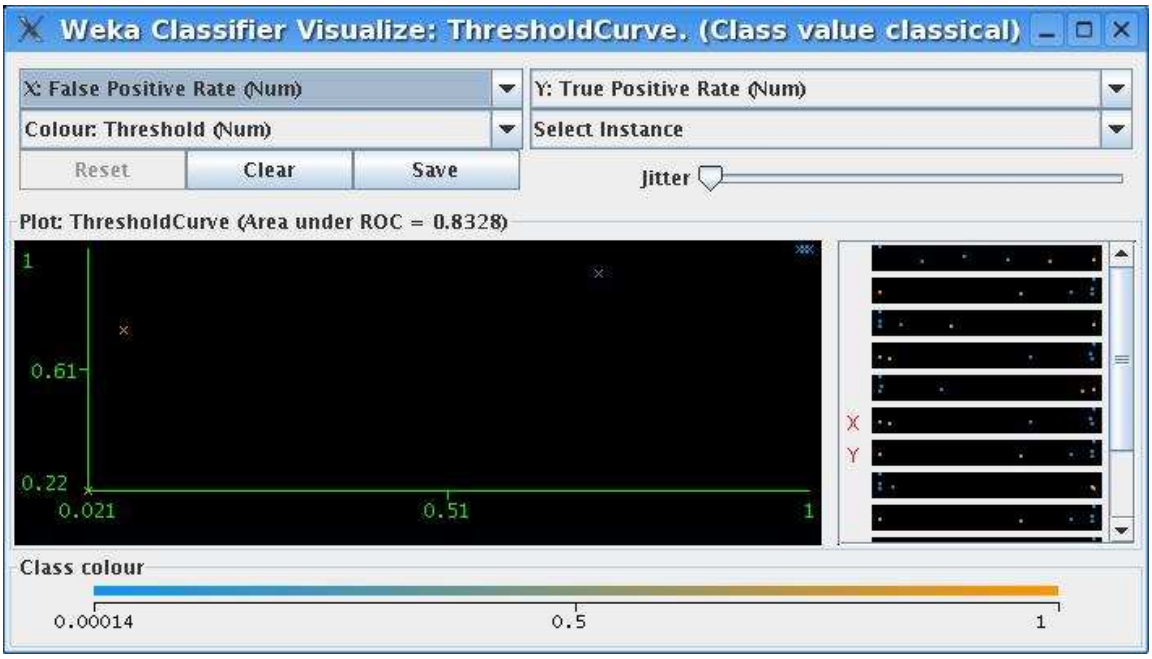


Figura 23. Curva ROC para o gênero *Classical*, usando a técnica kNN, com k = 1.

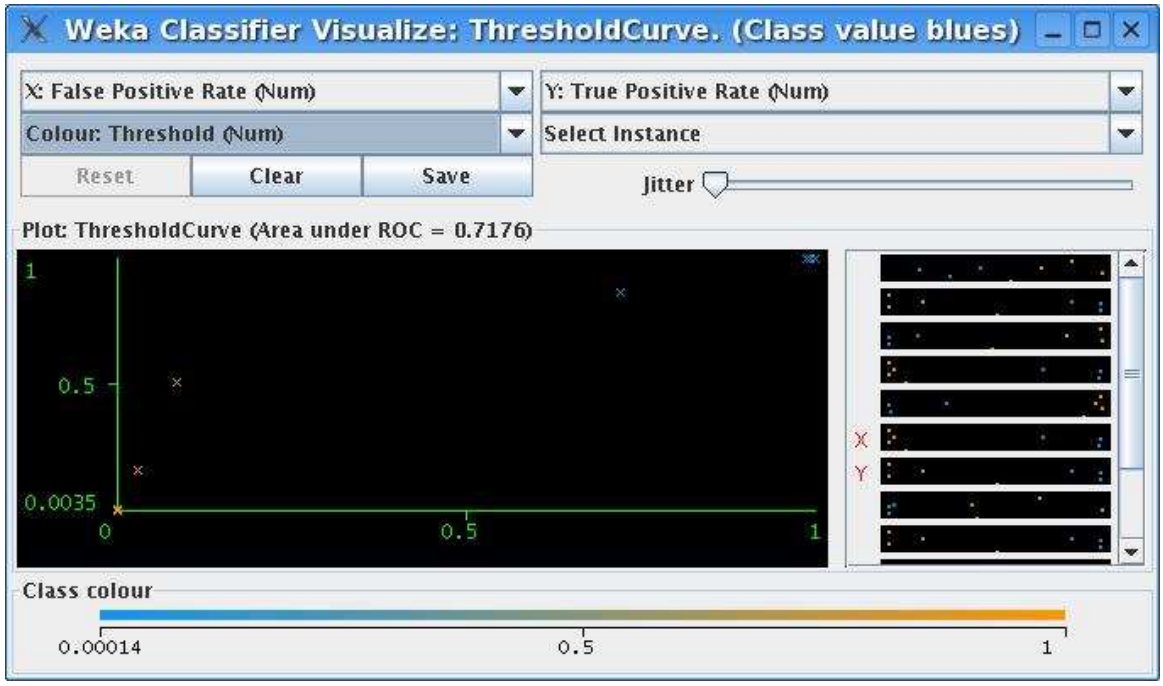


Figura 24. Curva ROC para o gênero *Blues*, usando a técnica kNN, com k = 1.

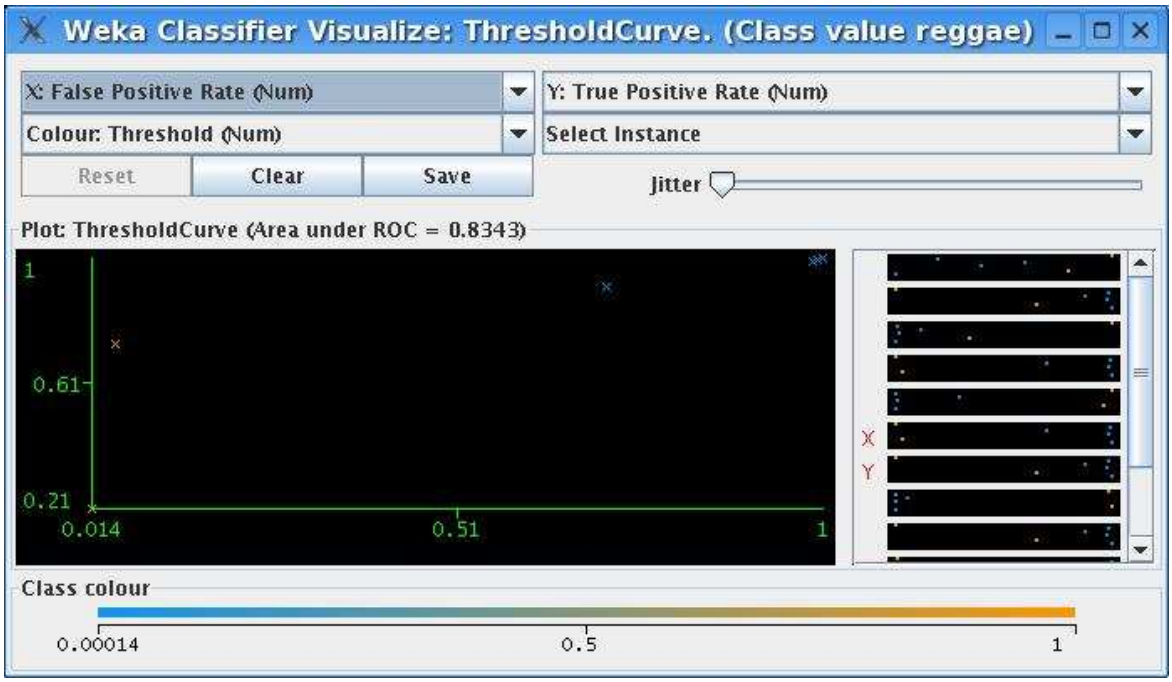


Figura 25. Curva ROC para o gênero *Reggae*, usando a técnica kNN, com k = 1.

Por fim, a Figura 26 mostra os erros do classificador para o melhor resultado dos testes com a técnica kNN. Os quadrados representam instâncias classificadas erroneamente e, as cruzes, mostram os gêneros classificados corretamente. Alguns dos fatos que ocorreram no experimento são mais facilmente percebidos a partir desse gráfico. Por exemplo, a confusão na classificação entre os gêneros *Classical*, *Blues* e *Jazz* e a facilidade na distinção entre a classe *Classical* e *Reggae* ficam visualmente explícitos. Da mesma forma, observa-se a grande maioria das instâncias da classe *Reggae* foi classificada corretamente, existindo maior confusão apenas, com o gênero *Samba*.

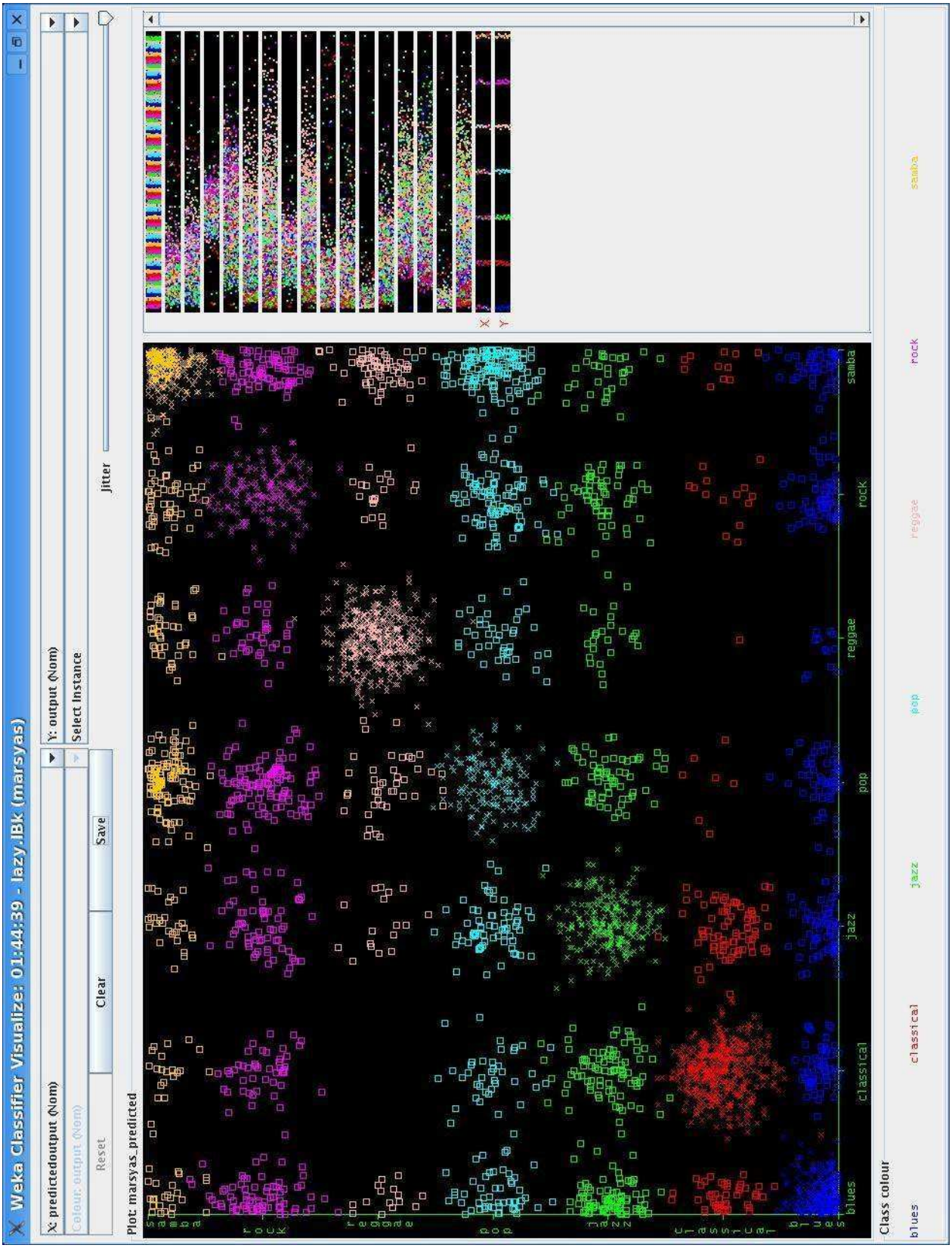


Figura 26. Gráfico dos erros de classificação para a técnica kNN. As classes aparecem no rodapé da figura, na seguinte ordem: *Blues*, *Classical*, *Jazz*, *Pop*, *Reggae*, *Rock* e *Samba.*

4.4 Comparação entre as Classificações

Ao contrário da expectativa inicial, também baseada no sucesso de trabalhos de outros autores, as redes MLP demonstraram um desempenho levemente superior ao da técnica kNN. O tratamento dos dados e a manipulação dos parâmetros iniciais para os experimentos MLP, mostraram que com apenas uma camada escondida, uma quantidade média de tempo de treinamento de aproximadamente 50000 ciclos e um fator de aleatoriedade de, pelo menos 10, as classificações entre os sete gêneros tornam-se mais aceitáveis, pois a taxa de acerto sobre para mais de 50%. Enquanto isso, o melhor desempenho alcançado pelo kNN não conseguiu atingir esse mínimo. Contudo, sua matriz de confusão aponta erros de classificação bastante plausíveis, como por exemplo a confusão entre os gêneros *Blues*, *Jazz* e *Classical*, também presentes em [17] e [14]. Além disso, a técnica gera curvas de previsão de características cujas áreas sob a curva ROC são maiores que 60%, indicando que a técnica ainda possui uma boa probabilidade de acerto.

A Tabela 13, mostra uma comparação entre os melhores resultados dos experimentos com kNN e MLP. Novamente, pode-se chegar a uma série de conclusões a respeito do desempenho de cada método empregado, como por exemplo, o fato de MLP ter conseguido um melhor desempenho, apesar de levar um tempo muito grande para finalizar os experimentos.

Em ambas as técnicas, as matrizes de confusão demonstram que classes como *Reggae* e *Classical* são facilmente distinguíveis, enquanto que as diferenças ente *Classical*, *Blues* e *Jazz*, são mais difíceis de verificar. A explicação para esses acontecimentos reside no fato de que as similaridades de velocidade da batida, tipos de instrumentos, utilizados e vocalização são características extremamente distintas entre o Reggae e o Classical, fazendo com que o primeiro possua uma taxa de ruído (Taxa de Passagem Pelo Zero) e um *Rollof* (a taxa de decaimento do da freqüência do sinal) maiores que o segundo, enquanto que as mesmas características são bastante similares entre os gêneros *Classical*, *Blues* e *Jazz*. Nota-se, especialmente, que a distinção entre Blues e Jazz não é trivial, uma vez que, até cognitivamente, os dois gêneros se interceptam em diversos aspectos, especialmente na instrumentação e vocalização de grande parte de suas músicas.

No resultado de melhor desempenho da kNN, muitas instâncias do gênero Pop foram confundidas com Samba, fato que não acontece para as classificações que utilizaram MLP. Também se deve notar que, no melhor resultado da kNN cerca de 19% das instâncias de *Rock* foram erroneamente classificadas como Pop; enquanto que na MLP, essa porcentagem baixa para aproximadamente 17%.

Tabela 13. Comparação entre os melhores resultados dos testes.

Comparação dos melhores Resultados dos testes (MLP x kNN)					
Técnica	Quantidade de instâncias usadas nos treinamentos	Tempo de processamento dos dados (hh:mm:ss)	Erro de Validação Cruzada	Taxa de acerto (%)	Número de Ciclos
kNN	Todos os padrões	00:04:30	0,3809	49,1118	-
MLP	Todos os padrões	00:30:15	0,3041	50,5629	1000

Capítulo 5

Conclusões e Trabalhos Futuros

O principal e mais ambicioso objetivo da atenção desse trabalho foi abordar o problema da classificação automática de áudio digital aproveitando estudos, ensaios e outras propostas já existentes no meio acadêmico, a fim de encontrar uma maneira mais genérica e completa de solucionar algumas das necessidades pendentes, deixadas pelos seus antecessores. Algumas das propostas encontravam dificuldade em classificar algum gênero específico, dada a quantidade reduzida de características extraídas dos sinais de áudio, outros se detinham apenas em classificar segmentos que continham voz, diferenciar ou separar o ruído de fundo dos sinais de áudio efetivos e alguns, ainda, encontravam severas dificuldades e respostas pouco expressivas por conta da escolha de um conjunto pobre das características envolvidas no processo de classificação e de seu inter-relacionamento.

Foram utilizados duas técnicas para classificação dos segmentos de áudio, o método kNN e o MLP. Ambas as escolhas basearam-se em um estudo abrangente de uma série de trabalhos da área, onde foi identificado que as duas técnicas em questão apresentaram, na grande maioria das vezes, as respostas mais significativas e satisfatórias. O uso dos classificadores ficou por conta do simulador WEKA, que já possui uma série de implementações de redes neurais artificiais, cujos parâmetros podem ser modificados de acordo com os treinamentos que se deseja realizar. Em todos os experimentos, foram realizadas validações cruzadas em 10 blocos, a fim de aprimorar o resultado da classificação e utilizar, em sua totalidade, as bases de dados.

Para criação dos arquivos utilizados com o WEKA e para a extração dos vetores de características dos sinais de áudio, utilizou-se a suíte de aplicativos MARSYAS. Esse *software* permite a extração das texturas de timbre, dos coeficientes cepstrais de freqüência, da além de outros coeficientes dos sinais de áudio, de maneira prática e automática, do mesmo modo que a geração dos arquivos ARFF, formato básico do simulador WEKA.

As classificações usando redes MLP apresentaram um desempenho levemente superior àquele da técnica kNN, apesar de seu tempo de processamento, manipulação de dados e configuração de parâmetros ser bastante superior.

Apesar de os resultados obtidos encontram-se próximos daqueles de bom desempenho na área de MIR e PDS, a taxa de acerto para a classificação dos gêneros ainda foi menor que a esperada (levando em conta a área de Redes Neurais e Aprendizado de Máquinas, onde as taxas de acerto são, comumente, próximas a 70%). Contudo, pode-se considerar que o desempenho do experimento foi satisfatório, dada a existência de pouquíssimos estudos que utilizam arquivos de MP3 para classificação e, ainda mais, por se tratar de uma abordagem nova, utilizando uma combinação de duas técnicas antes empregadas em arquivos cujos formatos (AU, MIDI) são menos complexos que o do MPEG Layer III.

Ademais, fica claro que um ataque ainda mais incisivo ao problema da padronização dos gêneros, para aprimorar a separação técnica e cognitiva das bases; o aumento do poder dos extratores de características, para que sejam capazes de obter ainda mais dados relevantes para a distinção das classes; e, por fim, um estudo mais abrangente e mais aprofundado a respeito de outras técnicas de classificação como Máquinas de Vetor de Suporte ou SVM (*Support Vector Machine*), Transformadas Wavelet, Redes RBF (*Radial Basis Function*), Redes PNN-DDA (*Probabilistic Neural Network – Dynamic Decay Adjustment*) e regras de separação Taxonômica de Dados, fazem parte dos estudos futuros que podem ser levados a cabo a partir desse.

A implementação de técnicas mais eficientes de extração de vetores de características de arquivos MP3 passa por, pelo menos, três projetos que se encontram atualmente regidos pela GPL. Dessa forma, a contribuição para seu aprimoramento constitui um trabalho futuro relativamente simples, contudo, muito importante de se realizar.

Da mesma forma, o estudo e subseqüente uso de outras técnicas ainda não experimentadas para conjuntos de dados como os que foram utilizados nesse trabalho, é tarefa relativamente simples, embora extremamente custosa de fazer. Nesse âmbito, uma proposta mais específica para trabalhos futuros, seria a utilização de Transformadas Wavelet ao invés de técnicas de Redes Neurais Artificiais, para a classificação dos vetores de características extraídos dos segmentos de áudio. Muito embora a complexidade matemática das técnicas de que usam Transformadas Wavelet seja bastante elevada, outros autores já reportaram um bom desempenho em suas experiências.

Um dos únicos e grandes problemas que se pode enfrentar em relação às propostas iniciais de trabalhos subseqüentes é justamente a busca por uma padronização dos gêneros musicais. Uma vez que a mutabilidade da área musical cresce livre de controle, documentação ou regulamentação, apenas alguns gêneros já presentes há muito tempo no mundo do entretenimento podem, ou devem, ser considerados suficientemente “seguros”, a ponto de fazer parte de uma e apenas uma categoria, ao invés de pertencer (cognitiva e tecnicamente falando) a uma miríade de gêneros distintos.

Além desses, uma outra proposta para melhorar os resultados obtidos, é realizar uma estratificação manual dos dados, normalizando ao máximo a porcentagem de instâncias iguais em diferentes classes, bem como levar em conta algumas teorias mais profundas da matemática e da estatística como a Teoria dos Grandes Números (*Law Of Large Numbers*) [40], para melhorar o pré-processamento das bases de dados, diminuindo, dessa forma, os erros médios e quadrados e aumentando a taxa de acertos na classificação.

Bibliografia

[1] VAN DER MERWE, P., “Origins of the Popular Style: The Antecedents of Twentieth-Century Popular Music”, Oxford: Clarendon Press, 1989.
[2] Disponível em: http://opihi.cs.uvic.ca/MARSYAS/, Último acesso: 11 de Março de 2006.
[3] Disponível em: http://www.cs.waikato.ac.nz/ml/weka/, Último acesso: 11 de Março de 2006.
[4] SHANNON, C. E., “A Mathematical Theory Of Communication”, Bell Systems Tech. J., Vol. 27, pp. 623-656, 1968
[5] RABINER, L. R. e SCHAFER, R. W., “Digital Signal Processing of Speech Signals”, Prentice-Hall, Inc., Englewood Cliff, Nova Jersey, 1978.
[6] Disponível em: http://www.amazon.com/exec/obidos/tg/browse/-/301668/ref=br_lr_/002-1163926-5498408, Último acesso: 12 de Março de 2006.
[7] Disponível em: http://music.barnesandnoble.com/styles/, Último acesso: 12 de Março de 2006.
[8] Disponível em: http://www.winamp.com/music/, Último acesso: 12 de Março de 2006.
[9] Disponível em: http://www.musicmatch.com/download/music_discovery_intro.htm, Último acesso: 12 de Março de 2006.
[10] Disponível em: http://www.xmms.org/about.php, Último acesso: 12 de Março de 2006.
[11] Disponível em: http://music.yahoo.com/musicvideos/default.asp, Último acesso: 12 de Março de 2006.
[12] Disponível em: http://music.aol.com/musicstyles/styles.adp, Último acesso: 23 de Março de 2006.
[13] DESHPANDE, H.; SINGH, R. e NAM, U., “Classification of music signals in the visual domain”, COST-G6 Conference on Digital Audio Effects, Irlanda, 2001.
[14] TZANETAKIS, G. e COOK P., “Musical Genre Classification of Audio Signals”, IEEE Transactions On Speech And Audio Processing, Vol. 10, nº 5, Julho, 2002.
[15] LAMBROU, T.; KUDUMAKIS, P.; SPELLER, R.; SANDLER, M. e LINNEY, A., “Classification of audio signals using statistical features on time and wavelet transform domains”, International Conference on Acoustic, Speech and Signal Processing, Seattle, E.U.A., Vol. 6, pp. 3621–3624, 1998.
[16] HAYKIN, S., “Redes Neurais”, Ed. Bookman, 2002.
[17] LI, T. e OGIHARA M., “Music Genre Classification with Taxonomy”, International Conference on Acoustic, Speech and Signal Processing, Filadélfia, EUA, 2005.
[18] MACMILLAN, N. e CREELMAN, C. D., “Detection Theory A User´s Guide”, Lawrence Erlbaum Associates Pub, 2a edição, 2005.
[19] Disponível em: http://www.apple.com, Último acesso: 02 de Outubro de 2006.
[20] Disponível em: http://www.microsoft.com/presspass/presskits/zune/default.mspx, Último acesso: 02 de Outubro de 2006.
[21] Disponível em: http://www.last.fm, Último acesso: 29 de Setembro de 2006.
[22] Disponível em: http://www.myspace.com, Último acesso: 02 de Outubro de 2006.
[23] RABINER, L. R. e HUANG, B. H., “Fundamentals of Speech Recognition”, Englewood Cliffs, 1993.

[24] SAUNDERS, J., “Real time discrimination of broadcast speech/music”, Proc. International Conference of Acoustics, Speech, Signal Processing (ICASSP), pp. 993-996, 1996.
[25] BERENZWEIG, A. L., e ELLIS, D. P., “Locating singing voice segments within musical signals,” Proc. International Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA) Mohonk, Nova Iorque, pp. 119–123, 2001.
[26] MACMILLAN, N. e CREELMAN, C. D., "Detection Theory", Lawrence Erlbaum Associates, 2005.
[27] OPPENHEIM, WILLSKY e NAWAB, “Signals and Systems”, 2nd Edition, Prentice Hall.
[28] Haykin, S. e Veen, B. V., “Sinais e Sistemas”, Ed. Bookman, 2001.
[29] PIMENTEL, C., “Notas de Aula do Professor Cecílio Pimentel”, Departamento de Eletrônica e Sistemas da Universidade Federal de Pernambuco – UFPE.
[30] ANAGNOSTOPOULOU, C.; FERRAND, M. e SMAILL, A., “Music and Artificial Inteligence”, Springer, Outubro 2002.
[31] STEVENS, S. e VOLKMAN, J., “The Relation of pitch to frequency”, American Journal of Psychology Vol. 53, Pág. 329, 1940.
[32] XU, C.; MADDAGE, C. N.; XU, C.; CAO, F. e TIAN, Q., “Musical genre classification using support vector machines”, International Conference on Acoustic, Speech and Signal Processing, Hong Kong, Japão, 2003.
[33] Disponível em: http://www.gnu.org/copyleft/gpl.html#TOC1, Último acesso: 15 de Março de 2006.
[34] Disponível em: http://www.underbit.com/products/mad/, Último acesso: 29 de Julho de 2006.
[35] WITTEN, H. I. e FRANK E., "Data Mining: Practical Machine Learning Tools and Techniques", 2a Edição, Morgan Kaufmann, São Francisco, 2005.
[36] MARKOU, M. e SINGH, S., “Novelty Detection: A review – Part 1: Statistical Approach”, Elsevier, Exter, Reino Unido, Julho, 2003.
[37] MARKOU, M. e SINGH, S., “Novelty Detection: A review – Part 2: Neural Netwroked Based Approaches”, Elsevier, Exter, Reino Unido, Julho, 2003
[38] Disponível em: http://www.opengroup.org/public/pubs/external/auformat.html, Último acesso: 02 de Outubro de 2006.
[39] LIU, Z.; WANG, Y. e CHEN, T.,“Áudio Feature Extraction And Analysis For Scene Segmentation And Classification”, Journal of VLSI Signal Processing, pp. 20, 61-79, 1998.
[40] BAUM, L. E. e KATZ, M., “Convergence Rates in the Law of Large Numbers”, Transactions of the American Mathematical Society, Vol. 120, No. 1, pp. 108-123, Outubro de 1965.

Apêndice 1

Exemplo de base de dados do WEKA

O conjunto a seguir exemplifica a estrutura interna dos arquivos ARFF utilizados pelo WEKA durante esse estudo. É válido ressaltar que todos os arquivos similares, que foram utilizados para realizar as classificações, foram gerados automaticamente pelo comando `bextract` e seus respectivos parâmetros, fornecido com a suíte de aplicativos para extração de vetores de características de sinais de áudio, o MARSYAS.

@relation marsyas
@attribute Mean_Acc1298_Mean_Mem40_Centroid real
@attribute Mean_Acc1298_Mean_Mem40_Rolloff real
@attribute Mean_Acc1298_Mean_Mem40_Flux real
@attribute Mean_Acc1298_Mean_Mem40_ZeroCrossings real
@attribute Mean_Acc1298_Std_Mem40_Centroid real
@attribute Mean_Acc1298_Std_Mem40_Rolloff real
@attribute Mean_Acc1298_Std_Mem40_Flux real
@attribute Mean_Acc1298_Std_Mem40_ZeroCrossings real
@attribute Std_Acc1298_Mean_Mem40_Centroid real
@attribute Std_Acc1298_Mean_Mem40_Rolloff real
@attribute Std_Acc1298_Mean_Mem40_Flux real
@attribute Std_Acc1298_Mean_Mem40_ZeroCrossings real
@attribute Std_Acc1298_Std_Mem40_Centroid real
@attribute Std_Acc1298_Std_Mem40_Rolloff real
@attribute Std_Acc1298_Std_Mem40_Flux real
@attribute Std_Acc1298_Std_Mem40_ZeroCrossings real
@attribute output {blues,classical,jazz,pop,reggae,rock,samba}

@data
0.034322,0.058450,0.006579,0.037674,0.013783,0.030541,0.002090,0.011517,0.013780,0.030257,0.001337,0.012240,0.030787,0.062026,0.000517,0.003726,blues
0.065193,0.136192,0.009005,0.085986,0.035844,0.087836,0.003621,0.038015,0.019186,0.041222,0.001052,0.022918,0.019154,0.050505,0.001170,0.016039,blues
0.027791,0.073252,0.007783,0.037472,0.017049,0.094149,0.002777,0.019616,0.018547,0.063433,0.001288,0.016353,0.024324,0.112692,0.000800,0.013866,blues
0.027072,0.059100,0.007658,0.038481,0.018705,0.070498,0.002634,0.022141,0.015158,0.040650,0.000725,0.008264,0.028012,0.075909,0.001743,0.010135,blues...

0.098470,0.200158,0.008456,0.115979,0.039027,0.076587,0.003270,0.038650,0.023409,0.044500,0.001193,0.025063,0.017658,0.036579,0.000689,0.014403,classical
0.087505,0.163083,0.009218,0.105635,0.030087,0.063184,0.003568,0.030677,0.029589,0.047275,0.001423,0.027798,0.016109,0.028586,0.000836,0.009096,classical
0.054928,0.110173,0.006639,0.063778,0.028387,0.087739,0.002192,0.030797,0.027901,0.078720,0.000918,0.029926,0.044568,0.110228,0.000882,0.041207,classical
0.033119,0.048601,0.006849,0.039768,0.010710,0.027155,0.002218,0.013119,0.015223,0.027575,0.001091,0.015496,0.017090,0.037188,0.000785,0.008245,classical...

0.091979,0.184808,0.008631,0.106889,0.045533,0.099009,0.003539,0.041340,0.020307,0.040750,0.000935,0.016411,0.016812,0.027861,0.000676,0.010456,jazz
0.065976,0.134036,0.008846,0.082438,0.020830,0.062767,0.003221,0.022341,0.023348,0.058658,0.001920,0.028406,0.016343,0.058850,0.001065,0.010232,jazz
0.035626,0.064825,0.007680,0.052416,0.026789,0.080728,0.002659,0.032459,0.021076,0.043023,0.001083,0.024587,0.040018,0.086324,0.000605,0.036255,jazz
0.083880,0.157964,0.008146,0.099164,0.032699,0.077555,0.003275,0.033597,0.017761,0.034403,0.001989,0.022201,0.017149,0.039324,0.001137,0.007316,jazz...

0.089895,0.192660,0.009094,0.118916,0.033831,0.087748,0.003652,0.037248,0.030660,0.064539,0.001364,0.033764,0.018249,0.041309,0.001095,0.017855,pop
0.055386,0.138705,0.008491,0.092791,0.037805,0.151320,0.002900,0.047925,0.021660,0.064613,0.000858,0.026051,0.024725,0.073371,0.000592,0.023030,pop
0.056758,0.116961,0.008814,0.081467,0.024408,0.087950,0.003140,0.030294,0.024010,0.058254,0.001962,0.028648,0.027792,0.075123,0.001011,0.024689,pop
0.118042,0.284548,0.009147,0.160231,0.045231,0.099824,0.003974,0.047116,0.022451,0.049402,0.001435,0.026603,0.029405,0.055381,0.001099,0.027320,pop..

0.053938,0.170828,0.007363,0.078257,0.040464,0.214159,0.002869,0.046124,0.014945,0.066872,0.000773,0.017468,0.029255,0.117012,0.000748,0.023938,reggae
0.052332,0.115077,0.007115,0.070912,0.036148,0.137352,0.002639,0.043868,0.020015,0.058289,0.000720,0.024578,0.024168,0.089271,0.000669,0.017264,reggae
0.063751,0.160221,0.007587,0.096485,0.049324,0.173700,0.002735,0.055701,0.018971,0.055469,0.000796,0.024040,0.020596,0.067704,0.000666,0.017510,reggae
0.090347,0.214780,0.008525,0.124533,0.066338,0.193323,0.003509,0.068498,0.033090,0.062669,0.001309,0.033647,0.051121,0.084983,0.001242,0.044893,reggae...

0.117415,0.244630,0.008890,0.150796,0.076882,0.149890,0.004513,0.072436,0.029204,0.057709,0.001296,0.030802,0.039717,0.068562,0.002169,0.037514,rock
0.110345,0.217848,0.009361,0.137765,0.056058,0.112031,0.004119,0.058306,0.033322,0.067831,0.001293,0.038014,0.025741,0.051819,0.000968,0.023617,rock
0.059257,0.116251,0.008354,0.083384,0.034458,0.094164,0.003138,0.040333,0.020762,0.054360,0.000993,0.023050,0.031413,0.065876,0.001059,0.027810,rock
0.108072,0.248744,0.009867,0.148631,0.057638,0.147318,0.004395,0.061241,0.025186,0.051338,0.001346,0.025871,0.025160,0.043361,0.001265,0.020650,rock…

0.063750,0.145206,0.007417,0.092493,0.063462,0.205753,0.002545,0.076117,0.030630,0.0801
82,0.000943,0.037691,0.045046,0.112985,0.000754,0.043613,samba
0.043785,0.088163,0.008134,0.064824,0.029649,0.090369,0.002688,0.036078,0.016418,0.0395
79,0.001124,0.020575,0.029023,0.079123,0.000702,0.027433,samba
0.065174,0.133874,0.008517,0.087681,0.032966,0.098102,0.003456,0.038558,0.014260,0.0393
74,0.000962,0.018022,0.022228,0.059787,0.001073,0.021769,samba
0.052314,0.105362,0.007597,0.065566,0.018648,0.053988,0.002425,0.022077,0.016603,0.0474
77,0.001265,0.021023,0.017782,0.042737,0.000704,0.014170,samba…