

DALTON FRANCISCO DE ARAÚJO

**BUSCA COMO SISTEMA DE APOIO À MELHORIA  
DE CLASSIFICADORES AUTOMÁTICOS DE  
ÁUDIO**

Recife

2014

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO  
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA APLICADA

**BUSCA COMO SISTEMA DE APOIO À MELHORIA  
DE CLASSIFICADORES AUTOMÁTICOS DE  
ÁUDIO**

Dissertação apresentada ao Programa de Pós-Graduação em Informática Aplicada como exigência parcial à obtenção do título de Mestre.

**Área de Concentração: Computação Inteligente e Modelagem**

Orientador: Prof. Dr. Giordano Ribeiro Eulálio Cabral

Recife  
2014

FICHA CATALOGRÁFICA PREPARADA PELA BIBLIOTECA CENTRAL DA UFRPE.

S586d      Araújo, Dalton Francisco de  
Busca como sistema de apoio à melhoria de classificadores automáticos de  
áudio / Dalton Francisco de Araújo. – Recife, 2014.

90f. : il.

Orientador: Giordano Ribeiro Eulálio Cabral.

Dissertação (Mestrado em Informática Aplicada) – Universidade Federal  
Rural de Pernambuco, Departamento de Estatística e Informática, Recife,  
2014.

Inclui referências e anexo(s).

1. Inteligência artificial 2. Aprendizado de máquina 3. Classificação  
automática de sons 4. Vigilância 5. Tiro

I. Cabral, Giordano Ribeiro Eulálio, orientador II. Título

CDD 001.642

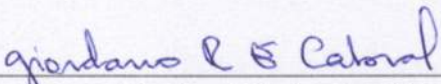
UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO  
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA APLICADA

BUSCA COMO SISTEMA DE APOIO À MELHORIA DE  
CLASSIFICADORES AUTOMÁTICOS DE ÁUDIO

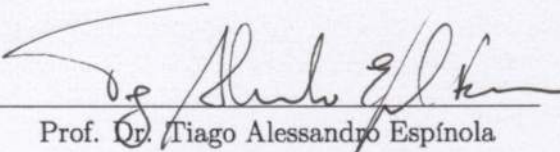
DALTON FRANCISCO DE ARAÚJO

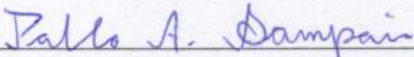
Dissertação julgada adequada para obtenção  
do título de mestre em Informática Aplicada,  
defendida e aprovada por unanimidade, em  
28/02/2014, pela Comissão Examinadora.

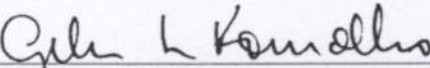
Orientador:

  
Prof. Dr. Giordano Ribeiro Eulálio Cabral  
Universidade Federal Rural de Pernambuco

Banca Examinadora:

  
Prof. Dr. Tiago Alessandro Espínola  
Ferreira  
Universidade Federal Rural de Pernambuco  
DEInfo-UFRPE

  
Prof. Dr. Pablo Azevedo Sampaio  
Universidade Federal Rural de Pernambuco  
DEInfo-UFRPE

  
Prof. Dr. Geber Lisboa Ramalho  
Universidade Federal de Pernambuco  
CIn-UFPE

Dedico à minha família.

# Agradecimentos

Agradeço, primeiramente, a Deus por tudo.

Agradeço também às pessoas e às instituições sem as quais dificilmente completaria o presente trabalho.

Aos meus pais, Ivoneide e Francisco, pela dedicação que tiveram a criar seus filhos e pelos investimentos em nossa educação mesmo em momentos difíceis.

Ao meu irmão, Faraday, por estar sempre disposto a ouvir e a ajudar

À minha noiva, Auristela, pela compreensão e apoio durante esse período.

Ao meu orientador, Giordano Cabral, pelo constante incentivo e pela compreensão que demonstrou nos momentos em que estive sob grande demanda. Além de me proporcionar, até em situações informais, verdadeiras aulas que foram além dos limites de uma orientação de mestrado.

Ao coordenador do Programa de Pós-Graduação em Informática Aplicada, Tiago Ferreira, por estar sempre presente quando nós estudantes precisamos, assim como pela cessão de servidores para realização dos experimentos deste trabalho.

A todos os docentes que viabilizaram a implantação deste programa de pós-graduação em 2011 e que em breve, certamente, será um dos mais bem avaliados do país.

À Universidade Federal Rural de Pernambuco por ter proporcionado todas as condições necessárias para realização deste mestrado.

À Daccord Music Software S.A pela cessão dos dados para experimentos, além da oportunidade de trocar experiências com seus desenvolvedores.

*"Muito aprendizado pode torná-lo realmente louco."*

**At 26, 24.**

# Resumo

A classificação automática de sons está presente de várias formas na sociedade. No entanto, ainda há espaços para melhorias desses sistemas. Enquanto alguns classificadores específicos obtêm índices satisfatórios de acertos, outros ainda carecem de técnicas e métodos que os façam atingir desempenhos desejáveis, principalmente quando se tratam de problemas que envolvam classes muito próximas umas das outras ou que envolvam grandes quantidades de classes. Existem ferramentas que são utilizadas pela comunidade que trabalha com esse tipo de problema para auxiliar em partes do processo de elaboração de um classificador automático de áudio. Porém, nota-se a ausência de ferramentas ou orientações que auxiliem os desenvolvedores na seleção do algoritmo de classificação a ser utilizado pelo classificador automático. Eles acabam realizando uma busca manual com ferramentas, como Weka, RapidMiner, Torch ou PyBrain, que permitem a seleção de algoritmos de classificação e seus respectivos parâmetros, para conceber um classificador dada uma base de dados. Nesse cenário, este trabalho propôs o SAC (Search for Audio Classification) que ao invés dessa busca manual, realiza uma busca automática. Em particular, foi adotado o método SMAC que já é utilizado na literatura para otimização de parâmetros. Em dois grupos de experimentos realizados: um com classificação de disparos de arma de fogo em ambientes públicos e outro com classificação de emoções nas músicas, os resultados mostraram uma diminuição média na taxa de erro em cerca de 11,26% em relação as taxas já registradas.

**Palavras-chave:** inteligência artificial; aprendizado de máquina; classificação automática de sons; vigilância; tiro.



# Abstract

Automatic classification of sounds is present in many forms in society. However, there is margin for improvement of these systems. While some specific classifiers obtained satisfactory rates, others still lack techniques and methods that make them achieve desirable performance, especially when dealing with problems involving classes very close to each other, or that involve large amounts of classes. There are tools that are used by the community working with this type of problem. However, there isn't a lack of tools or guidelines to aid developers in the selection of classification algorithm to be used by classifier. They end up doing a manual search tools like Weka, RapidMiner, Torch or PyBrain, which allow the selection of classification algorithms and their parameters to induce a classifier given a database. In this scenario, this paper proposes the SAC (Search for Audio Classification) that instead of this manual search, performs an automatic search. In particular, we adopted the SMAC method was already used in the literature for parameter optimization. In two groups of experiments, one with a rating of shooting a firearm in public places, and one with rating of emotions in the songs, the results showed an average decrease in error rate about 11.26% relative rates ever recorded.

**Key words:** artificial intelligence; machine learning; automatic classification of sounds; surveillance; shot.

# Lista de Figuras

1	Vídeomonitoramento de espaços públicos na cidade do Recife. . . . .	p. 2
2	Áudio Alerta: Equipamento de captura de som e imagem. . . . .	p. 4
3	Tarefa da aprendizagem de máquina (KUBAT et al., 1998). . . . .	p. 10
4	Hierarquia do aprendizado (MONARD; BARANAUSKAS, 2003). . . . .	p. 11
5	Resultados de análises não supervisionadas, grupos resultantes diferenciados por cores (WIKIPEDIA, 2014a). . . . .	p. 12
6	Abordagem supervisionada. . . . .	p. 13
7	Etapas básicas para concepção de um classificador. . . . .	p. 15
8	Exemplo de árvore de decisão para determinar o sucesso ou não na bilheteria de um filme (COPPIN, 2012). . . . .	p. 16
9	Funcionamento de um simples $k$ -NN, com $k = 3$ e $k = 9$ . . . . .	p. 19
10	Exemplo de Regressão Localmente Ponderada com regressão linear. . .	p. 20
11	Três funções de ativação (COPPIN, 2012). . . . .	p. 21
12	Modelo não-linear de um neurônio (SIMON, 2001). . . . .	p. 22
13	RNA em camadas. . . . .	p. 22
14	Visão geral de um detector e classificador de áudio (DUFAUX et al., 2000)- modificada. . . . .	p. 23
15	Extractor Discovery System (EDS), (PÉREZ, 2011) . . . . .	p. 27
16	Meta-aprendizado: aquisição de conhecimento (VILALTA et al., 2004). . .	p. 29
17	Meta-aprendizado: processo de recomendação (VILALTA et al., 2004). . .	p. 29
18	Weka. . . . .	p. 31
19	RapidMiner. . . . .	p. 32
20	Marsyas extração de MFCC. . . . .	p. 35

21	Sonic Visualiser. . . . .	p. 36
22	Dinâmica de uma busca em profundidade (COPPIN, 2012). . . . .	p. 38
23	Espaço com máximos locais (COPPIN, 2012). . . . .	p. 39
24	SMAC software. . . . .	p. 41
25	Proposta de solução. . . . .	p. 44
26	Validação cruzada. . . . .	p. 49
27	Experimento 1 – tipos de sons que compõem a base de dados. . . . .	p. 51
28	Experimento 1 – curva ROC. . . . .	p. 53
29	Modelo de Tellegen-Watson-Clark de humor tirado de Galinha e Pais-Ribeiro (2005) com modificações. . . . .	p. 57
30	Comparativo das taxas de erro de Tsoumakas et al. (2008) e experimento 2. . . . .	p. 59
31	Experimento 2 – redução percentual da taxa de erro alcançada o SAC. . . . .	p. 60

# Lista de Tabelas

1	Exemplo de conjunto de dados utilizado em processos de aprendizado de máquina para problemas de classificação. . . . .	p. 11
2	Principais métodos que compõem o espaço de busca dos experimentos.	p. 49
3	Experimento 1.1 – resultados obtidos com o SAC para o Áudio Alerta com 2 classes. . . . .	p. 52
4	Experimento 1.1 – matriz de confusão. . . . .	p. 53
5	Experimento 1.2: resultados obtidos com o SAC para o Áudio Alerta com 4 classes. . . . .	p. 54
6	Experimento 1.3: resultados obtidos com o SAC para o Áudio Alerta com 49 classes. . . . .	p. 54
7	Comparação dos erros: Áudio Alerta original vs SAC. . . . .	p. 55
8	Matriz de confusão algoritmo original do Áudio Alerta para o problema de duas classes. . . . .	p. 55
9	Experimento 2 – classes do experimento. . . . .	p. 57
10	Experimento 2 – SAC: resultados por classe. . . . .	p. 58
11	Taxa de erro (%) por classe para cada algoritmo testado por Tsoumakas et al. (2008). . . . .	p. 58
12	Experimento 2 – tabela comparativa das taxas de erro obtidas. . . . .	p. 58
13	Resultados finais: melhorias obtidas. . . . .	p. 59

# Sumário

<b>1</b>	<b>Introdução</b>	p. 1
1.1	Motivação . . . . .	p. 1
1.2	Objetivos . . . . .	p. 4
1.3	Principais Contribuições . . . . .	p. 5
1.4	Organização do Trabalho . . . . .	p. 5
<b>2</b>	<b>O Problema</b>	p. 7
2.1	Desafios . . . . .	p. 8
2.2	Requisitos . . . . .	p. 8
<b>3</b>	<b>Estado da Arte</b>	p. 10
3.1	Aprendizado de Máquina . . . . .	p. 10
3.1.1	Análise de dados, overfitting e compreensão dos modelos gerados	p. 12
3.1.2	Abordagens supervisionadas . . . . .	p. 13
3.2	Classificadores Automáticos . . . . .	p. 15
3.2.1	Principais Abordagens . . . . .	p. 15
3.2.1.1	Simbólico . . . . .	p. 16
3.2.1.2	Estatístico . . . . .	p. 17
3.2.1.3	Baseado em Instâncias . . . . .	p. 19
3.2.1.4	Conexionista . . . . .	p. 20
3.2.1.5	Genético . . . . .	p. 22
3.3	Classificadores de Áudio . . . . .	p. 23
3.3.1	Coefficientes Mel Frequency Cepstral (MFCCs) . . . . .	p. 25

3.3.2	Soluções Existentes . . . . .	p. 25
3.3.2.1	Abordagem Exploratória . . . . .	p. 26
3.3.2.2	Abordagem Sistemática . . . . .	p. 26
3.4	Meta-aprendizado . . . . .	p. 28
3.5	Ferramentas . . . . .	p. 30
3.5.1	Weka . . . . .	p. 30
3.5.2	RapidMiner . . . . .	p. 30
3.5.3	PyBrain . . . . .	p. 32
3.5.4	Torch . . . . .	p. 32
3.5.5	Auto-WEKA . . . . .	p. 33
3.5.6	Extração de Características de Áudio . . . . .	p. 34
3.5.6.1	Marsyas . . . . .	p. 35
3.5.6.2	Sonic Visualiser . . . . .	p. 36
3.6	Busca . . . . .	p. 37
3.6.1	Busca em Profundidade . . . . .	p. 37
3.6.2	Subida da colina . . . . .	p. 38
3.6.3	Problemas de Otimização Combinatória . . . . .	p. 39
3.6.3.1	Simulated Annealing . . . . .	p. 39
3.6.3.2	Sequential Model-based Algorithm Configuration (SMAC)	p. 41
<b>4</b>	<b>Proposta de Solução</b>	p. 42
4.1	Preparação da Busca . . . . .	p. 45
4.2	Requisitos da Proposta . . . . .	p. 45
<b>5</b>	<b>Avaliação</b>	p. 47
5.1	Metodologia . . . . .	p. 47
5.2	Experimento 1: Áudio Alerta . . . . .	p. 50
5.2.1	Descrição da base de dados . . . . .	p. 51

5.2.2	Resultados Obtidos . . . . .	p. 52
5.2.2.1	Problema com duas classes . . . . .	p. 52
5.2.2.2	Problema com quatro classes . . . . .	p. 53
5.2.2.3	Problema com 49 classes . . . . .	p. 54
5.2.3	Discussão dos Resultados . . . . .	p. 55
5.3	Experimento 2: Classificação de Emoções em Músicas . . . . .	p. 55
5.3.1	Resultados Obtidos . . . . .	p. 56
5.3.2	Discussão dos Resultados . . . . .	p. 57
5.4	Consolidação dos Resultados SAC . . . . .	p. 59
5.4.1	Economicidade . . . . .	p. 60
<b>6</b>	<b>Considerações Finais</b>	p. 62
6.1	Perspectivas Futuras . . . . .	p. 63
<b>Anexo A – Sequential Model-Based Algorithm Configuration (SMAC)</b>		
	- Transcrição parcial de Hutter et al. (2013b)	p. 65
	<b>Referências</b>	p. 67

# 1 Introdução

Problemas de classificação automática de áudio são clássicos na literatura. Existem diversos tipos de abordagens e aplicações. Um exemplo é o reconhecimento de disparos de arma de fogo que vem sendo utilizado em áreas urbanas com o objetivo de melhorar a segurança pública.

Métodos de aprendizado de máquina são empregados nessas soluções. Pesquisadores da área tentam melhorar cada dia mais a precisão dessas ferramentas.

Este trabalho lança uma proposta: alterar a forma de seleção do algoritmo de aprendizado de máquina a ser utilizado em um dado problema. A efetividade dessa solução é colocada à prova com dois problemas que envolvem dados reais. Toda essa investigação, reflexões e conclusões são detalhadas neste texto.

## 1.1 Motivação

Esta pesquisa foi fortemente influenciada por um projeto em andamento, no estado de Pernambuco, chamado Áudio Alerta: uma solução para o áudio monitoramento em espaços públicos.

A motivação original foi baseada em um problema específico dentro da problemática do Áudio Alerta. Contudo, no decorrer das investigações, as fronteiras iniciais foram sendo expandidas trazendo impactos a problemas, além daquele inicial.

Um dos principais desafios para os órgãos de segurança pública é conseguir maximizar eficientemente a cobertura de ações de vigilância em suas jurisdições. O que se torna muito difícil quando levadas em consideração as grandes dimensões das áreas urbanas e a quantidade limitada de agentes de segurança disponíveis.

Diante da inviabilidade de cobrir todas as áreas públicas com o efetivo humano disponível, nos últimos anos, um setor que ganhou destaque nessa área de atuação foi a de vigilância eletrônica.



O monitoramento eletrônico, antes aplicado principalmente para cobrir áreas privadas, vai ganhando espaço em áreas essencialmente públicas. Sendo visto como ferramenta que pode proporcionar vários benefícios quando aplicado à segurança pública, como a otimização de veículos e agentes em ações, uma maior agilidade do serviço, a prevenção de delitos, a detecção de atitudes suspeitas, a identificação de suspeitos, a elucidação de crimes, o controle do trânsito, entre outros (MELLO, 2011).

Geralmente, esses sistemas funcionam com agentes monitorando a distância por meio de telões. Assim, tendo os acontecimentos em tempo real no raio de alcance das câmeras. Esses operadores, ao detectarem visualmente alguma ação suspeita, entram em contato com as equipes mais próximas da região para averiguação *in loco*.



(a) Câmera instalada em via pública (PERNAMBUCO, 2014)



(b) Central de Monitoramento (RECIFE, 2014)

Figura 1: Videomonitoramento de espaços públicos na cidade do Recife.

Estima-se que a Inglaterra seja um dos países que mais investe nesse tipo de recurso. Segundo RT (2013), há no país cerca de 70000 câmeras geridas pela polícia e demais autoridades em território britânico. Ainda pouco quando comparado às mais de 4,1 milhões de propriedade particular.

No Brasil, apesar do aparelhamento com essa finalidade ter iniciado efetivamente nos últimos anos, algumas análises já começam a ser registradas na literatura. Como Jardwski (2012), que faz uma análise da eficácia do uso desses sistemas na cidade de Colombo no Paraná. Entre diversas observações, nesse estudo é apontada a redução em até 90% em alguns tipos de ocorrências policiais nas áreas em que as câmeras foram instaladas. Mesmo índice apresentado, também em casos específicos, por Guimarães (2007), que faz uma análise mais crítica com vistas aos impactos nas restrições das liberdades individuais e coletivas com a adoção desses recursos.

Contudo, como defendido por Ditton e Britain (apud RATCLIFFE, 2006), o vídeo-

monitoramento não pode ser tratado como a panaceia da segurança pública nos centros urbanos. Mesmo assim, sua utilidade tem se apresentado aceitável diante dos resultados apresentados. E a partir de pesquisas, como a desenvolvida neste trabalho, essa alternativa pode ser aprimorada para maiores ganhos da sociedade.

Apesar das vantagens, eventos suspeitos que ocorrem a poucos metros das câmeras podem passar despercebidos, uma vez que, em geral, o monitoramento é restrito à direção da câmera naquele momento particular. Ou seja, o equipamento pode estar dirigido para o lado oposto ao da ocorrência, o que compromete a sua eficácia nessas circunstâncias.

Além disso, mesmo que direcionada ao evento, o operador pode não ser capaz de identificar a irregularidade por estar distraído ou acometido por algum vício de ação (MELLO, 2011; JARDWSKI, 2012). E em alguns casos, necessitando contar com a sorte para perceber alguma anormalidade em momentos de baixa iluminação ou em rápidas ações (GUIMARÃES, 2007).

Uma maneira de mitigar essas situações e melhorar a aplicabilidade destes sistemas é a incorporação de novas direções ao processo, além da visão já explorada. Uma solução possível é abranger também o acompanhamento do áudio dos ambientes monitorados.

Ao contrário dos olhos, os ouvidos são sensíveis a sons chegando de todas as direções, como tal, constituem os sensores do sistema de alerta principal do ser humano, capazes de detectar sinais que indiquem situações de alerta mesmo quando estão dormindo (FAHY, 2000). O que mostra o potencial que pode ter o som também em sistemas de monitoramento eletrônico.

No entanto, o controle manual desse sistema seria impraticável, uma vez que o controle simultâneo de vários ambientes geraria uma miscelânea de áudios. O que não contribuiria para evitar eventuais cenários frágeis do processo já existente.

Ao invés disso, ao detectar a ocorrência de um evento de som suspeito, o equipamento poderia ser direcionado automaticamente para o emissor e enviar um alerta à central de monitoramento. E é nesse sentido que surge o Áudio Alerta.

O Áudio Alerta é um produto da Daccord Music Software S.A, com incentivos da Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco (FACEPE) e Financiadora de Estudos e Projetos (FINEP), que ao detectar um disparo de arma de fogo em uma área de atuação, envia um alerta a uma central de monitoramento e direciona a câmera instalada no local para o ponto de origem do evento.

Para isso, o sistema envolve a aplicação de uma série de conhecimentos: geolocalização,



Figura 2: Áudio Alerta: Equipamento de captura de som e imagem.

transmissão de dados, captura de som e de imagem, codificação, tratamento de imagem, compressão, armazenamento, etc., além da capacidade de perante um som de alto impacto, classificar essa ocorrência como sendo ou não um tiro de arma de fogo.

Inicialmente, este trabalho foi proposto com intuito de melhorar esse sistema de classificação do Áudio Alerta. No entanto, os caminhos trilhados para isso mostraram que o problema é mais abrangente.

O método aplicado ao Áudio Alerta não é exclusividade. Existem diversas aplicações que recorrem ao mesmo método para desenvolvimento de classificadores de sons, como classificação de música por gêneros (GOLUB, 2000), reconhecimento de voz (LIPPMANN, 1989), classificação do choro de crianças (SAHA et al., 2013), classificação de espécies de animais (LEE et al., 2008) etc.

As propostas de melhorias poderiam ter impactos nos problemas de classificação automática de áudio, de forma geral, o que se tornou a grande motivação deste trabalho.

## 1.2 Objetivos

Os objetivos principais deste trabalho são: melhorar o desempenho de classificadores automáticos de áudio, e; reduzir custos, como tempo e dinheiro, para concepção desses tipos de classificadores.

No decorrer deste trabalho, será mostrado que ainda existem espaços para melhorias em relação à corretude de muitos sistemas de classificação de áudio existentes. Assim como a importância de estudos nesse sentido, que podem trazer impactos positivos mais

abrangentes, como a outros tipos de classificadores, tendo em vista algumas similaridades no desenvolvimento de um classificador de áudio com outros sistemas de classificação.

O objetivo é reduzir a taxa de erro desses classificadores de áudio prezando por alguns requisitos básicos, como facilidade de adoção por outros pesquisadores e profissionais que tratem desse tipo de problema.

### 1.3 Principais Contribuições

Uma das principais contribuições deste trabalho é a convergência de conhecimentos já desenvolvidos, como métodos de aprendizado de máquina e de otimização de parâmetros aplicados à melhoria de classificadores automáticos de sons.

Assim como uma proposta de quebra de paradigma dentro da área de processamento de áudio em relação à seleção de um algoritmo de aprendizado de máquina e seus parâmetros para um problema dado, através da inclusão de um processo de busca automática.

### 1.4 Organização do Trabalho

Esta dissertação está organizada e escrita em seis capítulos de forma a facilitar o entendimento tanto de pessoas da área de inteligência artificial quanto àquelas que nunca tenham tido experiência anterior com o aprendizado de máquina, abrangendo também as que por algum motivo tenham interesse em ler este trabalho.

Logo no capítulo 2 é descrito o problema que foi objeto de investigação durante a execução deste trabalho. Como também os desafios que se imaginavam ter pela frente e os requisitos necessários a qualquer solução que pudesse surgir.

O capítulo 3 apresenta os principais conhecimentos teóricos relacionados com o problema tratado e que fundamentam as soluções desenvolvidas neste trabalho. Conhecimentos considerados necessários para compreensão das discussões levantadas, como noções de aprendizado de máquina, classificadores automáticos e classificadores de áudio.

O capítulo 4 descreve a proposta de solução, ou seja, o caminho escolhido para ser trilhado em busca de uma resposta para o problema levantado.

Essa proposta é avaliada através de dois grupos de experimentos com problemas reais. Os procedimentos, resultados e discussões estão no capítulo 5.

Por fim, há as conclusões gerais do trabalho, as reflexões e sugestões de trabalhos

futuros, tudo relatado no sexto e último capítulo desta dissertação de mestrado.

## 2 O Problema

Classificação de áudio é um problema científico consagrado, abordado de diferentes formas por engenheiros e cientistas de diversos campos do conhecimento, que encontraram na classificação automática de áudio, além de um grande desafio, uma enorme fonte de possibilidades para realização dos mais diversos experimentos e testes.

Herrera-Boyer et al. (2003) listam algumas situações as quais a classificação automática de sons é um recurso relevante: na biologia (por exemplo, para a identificação de animais pertencentes a uma determinada espécie ou para catalogar recursos comunicativos) (FRISTRUP; WATKINS, 1995; MILLS, 1995; POTTER et al., 1994), nos diagnósticos médicos (por exemplo, para detecção de condições anormais de órgãos vitais) (SHIYONG et al., 1998; BULLER; LUTMAN, 1998; SCHÖN et al., 2001), na segurança (MCLAUGHLIN et al., 1997), em operações militares (por exemplo, para detecção de um motor inimigo se aproximando ou para a identificação de armas) (GORMAN; SEJNOWSKI, 1988; ANTONIC; ZAGAR, 2000; DUBNOV; TISHBY, 1997) e na descrição de conteúdo multimídia (por exemplo, para ajudar na classificação de cenas de vídeos ou detecção de objetos) (LIU et al., 1998; LIENBART et al., 1999).

Vários registros de modelos para classificação automática de classes específicas de sons já existem na literatura, como Malheiro et al. (2004), que utiliza redes neurais artificiais (RNA) para classificação musical, Tzanetakis e Cook (2002a) que utilizam Modelos de Misturas Gaussianas (GMM) e  $k$ -Vizinhos mais Próximos ( $k$ -NN) (AHA et al., 1991) para classificar músicas entre gêneros, Golub (2000), que utiliza Modelo Linear Generalizado (GLM), redes neurais do tipo Multilayer Perceptron (MLP) e  $k$ -NN para classificação de estilos de músicas e Herrera-Boyer et al. (2003), que faz uma ampla revisão na literatura com a finalidade de apresentar um modelo de classificação automática de sons de instrumentos musicais.

Contudo, constata-se na literatura que ele continua um problema aberto. Para determinadas classificações os métodos chegam a baixas taxas de erro, como alguns exemplos citados por Herrera-Boyer et al. (2003) e Dufaux et al. (2000), mas para boa parte ainda

há muito o que evoluir. Tornando-o um importante desafio para a ciências diante de suas potencialidades.

## 2.1 Desafios

Vários fatores trazem complexidade à classificação automática de sons. Primeiramente, é uma habilidade rara mesmo em seres humanos, especialmente quando as classes são próximas, como acordes, espécies de pássaros e tipos de armas de fogo. Além disso, o conjunto domínio utilizado geralmente é formado por uma grande quantidade de exemplos, o que reflete no custo computacional de processamento.

Esses fatores, se não levados em consideração, podem facilmente tornar inviável qualquer proposta de solução para o problema.

Por outro lado, uma proposta de melhoria de classificadores de áudio pode se converter, a depender de pesquisas nesse sentido, também em melhorias em diversos outros processos que demandem sistemas de classificação automática, extrapolando os limites da classificação de sons. Trazendo consequências positivas à área de Inteligência Artificial, à ciência e à sociedade de forma geral.

## 2.2 Requisitos

É necessário estabelecer critérios que viabilizem a avaliação das soluções que possam surgir, de forma que a qualidade esteja intimamente ligada à conformidade dos requisitos, como definido por Crosby (1979): "qualidade é conformidade aos requisitos".

Existem diversos fatores de qualidade que podem ser utilizados para esse tipo de análise (MEYER, 1988). No entanto, para este trabalho em particular, serão considerados, prioritariamente, os seguintes:

- **corretude** — quantitativo, leva em consideração o quanto do objetivo é satisfeito. É a capacidade de ser isento de erros ou faltas. É um fator essencial quando se trata de processos de classificação automática. Raramente é útil um classificador que apresente uma corretude desprezível;
- **usabilidade** — qualitativo, detalha a facilidade de utilização de um sistema. Diante da variedade de profissionais de diferentes áreas que trabalham na classificação de

sons, é desejável que uma solução possa ser de fácil utilização para uma maior quantidade de pessoas e de problemas;

- disponibilidade de código — qualitativo, descreve se o código da solução está disponível para que outros interessados possam manipulá-lo e desenvolvê-lo caso tenham interesse. Sendo uma área amplamente abordada, é importante que contribuições sejam abertas para novas contribuições. Assim, pode-se ter um número maior de pessoas contribuindo, o que reduz os custos e a duplicação de esforços.
- economicidade — quantitativo, descreve o quanto de, neste caso, recursos financeiros são consumidos ao se adotar um sistema.

Para o problema posto, foi feito um questionamento junto à equipe do Áudio Alerta, que identificou outros fatores importantes. Mas sua adoção depende do tipo de solução e a viabilidade de verificação dos mesmos:

- escalabilidade — quantitativo, descreve se a solução está preparada para crescer uniformemente, se é capaz de manter o desempenho com diferentes volumes de informações;
- tempo de processamento — quantitativo, o tempo que a solução consome para apresentar uma resposta;
- tempo de implementação — quantitativo, tempo estimado para modelar um problema através da solução apresentada.

Dado o tipo de problema, a qualidade de uma solução é validada principalmente pela sua correção. Uma vez que classificadores automáticos têm sua qualidade estimada pelo custo associado às suas previsões (FACELI et al., 2011). Para essa mensuração, este trabalho adota a *taxa de erro*, métrica detalhada na seção 3.1.2.

Diante dos resultados dessas análises poder-se-á identificar a importância e o grau de satisfatibilidade da solução, assim como os grupos de problemas e usuários que estarão mais bem atendidos pela proposta.



## 3 Estado da Arte

O problema de classificação de áudio envolve uma série de conceitos, métodos e técnicas existentes. Este capítulo apresenta os principais conhecimentos teóricos que fundamentam as soluções desenvolvidas neste trabalho. Conhecimentos considerados necessários para compreensão das discussões levantadas.

A organização deste capítulo pode ser descrita em três seguimentos: primeiramente são tratados os fundamentos teóricos relacionados à classificação automática com ênfase especial à classificação automática de áudio, em segundo, na seção 3.5, são apresentadas ferramentas que auxiliam na aplicação desses conhecimentos, e por fim, na seção 3.6 são abordadas algumas técnicas de buscas, que estão envolvidas com a solução proposta neste trabalho.

### 3.1 Aprendizado de Máquina

Aprendizado de Máquina (AM) é um segmento extremamente importante da Inteligência Artificial (IA) cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado, bem como a construção de sistemas capazes de adquirir conhecimento automaticamente (MONARD; BARANAUSKAS, 2003; COPPIN, 2012, p. 233).

Entre várias definições na literatura, Mitchell (apud FACELI et al., 2011, p. 3), define aprendizado de máquina como: "A capacidade de melhorar o desempenho na realização de alguma tarefa por meio da experiência."

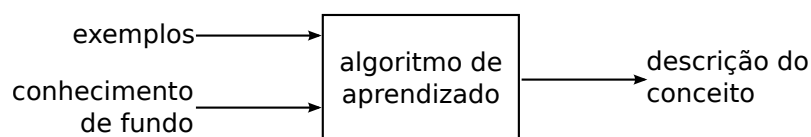


Figura 3: Tarefa da aprendizagem de máquina (KUBAT et al., 1998).

Uma forma de obtenção de conhecimento é através da indução, uma forma de inferência lógica, considerada a mais utilizada pelo cérebro humano, que permite obter

conclusões genéricas a partir de um conjunto particular de exemplos (MONARD; BARANAUSKAS, 2003). É o chamado método de aprendizado indutivo, baseado no princípio de que se for encontrada uma função capaz de mapear corretamente um grande conjunto de dados de treinamento, então ela também mapeará corretamente dados não observados anteriormente, generalizando através da experiência adquirida (COPPIN, 2012, p. 236).

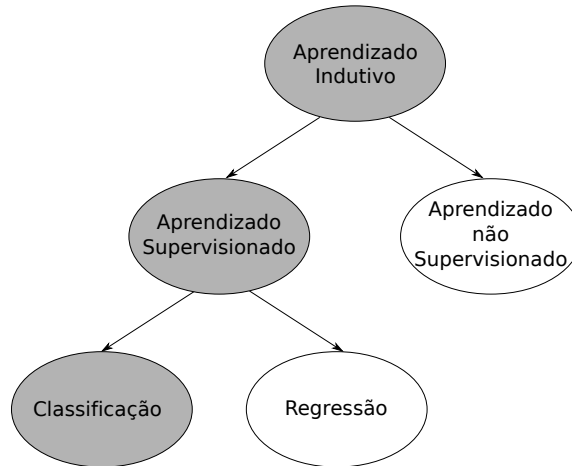


Figura 4: Hierarquia do aprendizado (MONARD; BARANAUSKAS, 2003).

O aprendizado indutivo pode ser dividido em supervisionado, também chamado de preditivo, e não - supervisionado, também conhecido como método descritivo por alguns autores. No aprendizado supervisionado, é fornecido ao algoritmo de aprendizado ou indutor, um conjunto de exemplos denominado conjunto de treinamento com os quais o rótulo da classe associada é conhecido. Isto é, fornece-se instâncias em que o atributo "classe" já está informado (MONARD; BARANAUSKAS, 2003).

Nesse caso, o objetivo do algoritmo de indução é construir um modelo que possa determinar corretamente a classe de novos exemplos ainda não rotulados (MONARD; BARANAUSKAS, 2003). Na seção 3.1.2 será mostrado que esse processo produz ou um **classificador** ou um **regressor** a depender do conjunto de classes trabalhada.

Tabela 1: Exemplo de conjunto de dados utilizado em processos de aprendizado de máquina para problemas de classificação.

Instância	Atributo 1	Atributo 2	...	Atributo $n$	Classe
1	54	0,0	...	4,9	Tiro
2	34	0,1	...	5,7	Buzina
⋮	⋮	⋮	⋮	⋮	⋮
$m$	53	0,0	...	6,2	Frenagem

Já no aprendizado não-supervisionado, o indutor analisa os exemplos fornecidos sem conhecer o atributo de saída da classe e tenta determinar se alguns deles podem ser

agrupados, de alguma maneira, formando grupos ou clusters. Após a determinação dos agrupamentos, normalmente, é necessária uma análise para determinar o que cada cluster significa no contexto do problema que está sendo analisado (MONARD; BARANAUSKAS, 2003).

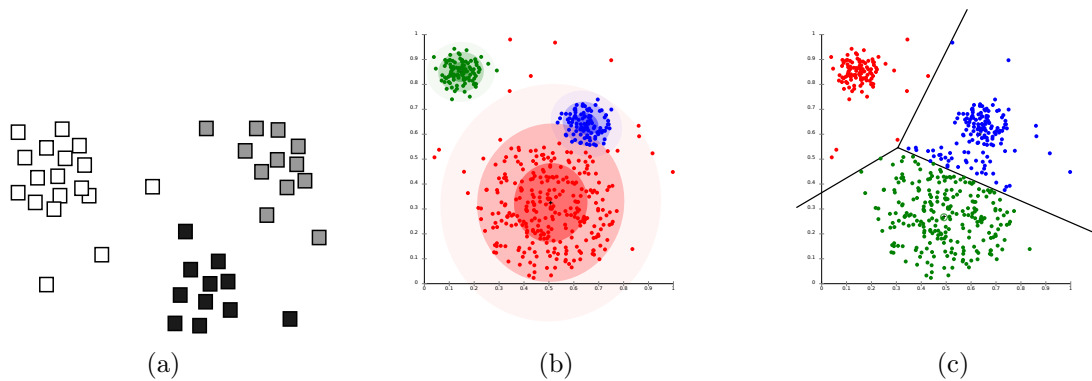


Figura 5: Resultados de análises não supervisionadas, grupos resultantes diferenciados por cores (WIKIPEDIA, 2014a).

### 3.1.1 Análise de dados, overfitting e compreensão dos modelos gerados

A qualidade das informações contidas na base de dados utilizada em problemas de aprendizado de máquina tem forte influência no desempenho das soluções geradas. Uma base composta por instâncias pouco representativas ou com ruídos, dados inconsistentes ou imperfeições podem provocar uma baixa taxa de acerto da ferramenta. Isto também ocorre quando um algoritmo indutor não captura os padrões existentes nos dados (FACELI et al., 2011).

Em relação aos problemas relacionados ao conjunto de dados, existem técnicas de pré-processamento que são frequentemente utilizadas para a melhoria dessas bases com o objetivo de eliminar ou minimizar essas situações. Há ainda a possibilidade de adaptação das informações ao algoritmo utilizado, o que também pode levar à construção de ferramentas mais fiéis à distribuição dos dados, reduzindo sua complexidade computacional e tornar mais fáceis, bem como rápidos os ajuste de parâmetros e o posterior uso dessas ferramentas (FACELI et al., 2011).

Algoritmos que realizam processo de treinamento ainda podem cair em um problema de superadaptação, ou *overfitting*, identificado quando o modelo gerado consegue classificar corretamente todos os dados de treinamento, mas um desempenho não tão bom ao classificar dados não observados, isto é, há a perda de generalidade (COPPIN, 2012,

p. 245). É o caso em que o modelo praticamente memoriza as saídas do conjunto de treinamento, restringindo sua especialidade a apenas ela.

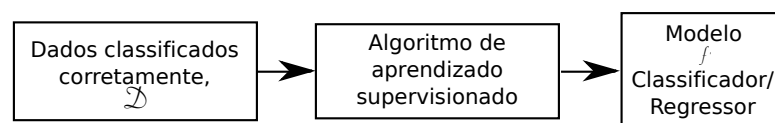
Sobre a interpretação do modelo gerado independente do tipo de abordagem, alguns algoritmos desenvolvem sua própria representação do conhecimento nem sempre de fácil compreensão de leitura, como as redes neurais artificiais, os métodos matemáticos, e os estatísticos, que tipicamente envolvem cálculos numéricos de coeficientes e pesos. Enquanto outros objetivam a criação de estruturas de conhecimento simbólico que atende ao princípio da compreensibilidade. Características que podem influenciar na escolha de um algoritmo indutor (KUBAT et al., 1998).

### 3.1.2 Abordagens supervisionadas

Sistemas de aprendizado supervisionado aprendem ao serem apresentados a dados de treinamento pré-classificados. Nesse caso, os algoritmos trabalham para gerar, a partir desses dados de entrada, um modelo que se aproxime ao máximo possível das saídas desejadas dos exemplos que compõem essa base.

Se o domínio for um conjunto de valores nominais, tem-se um problema de classificação e o estimador gerado é um classificador. Se o domínio for um conjunto infinito e ordenado de valores, tem-se um problema de regressão que gera um regressor (FACELI et al., 2011, p. 54).

Matematicamente falando, podemos considerar o conjunto  $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$ , como sendo o conjunto treinamento, onde o algoritmo indutor terá como objetivo gerar uma função  $f'$  que se aproxime ao máximo de  $y_i$  para cada  $f'(x_i)$ .



Assim, para cada nova instância:  $x$   
 $f(x)=y$ , espera-se que  $y$ =classe de  $x$

Figura 6: Abordagem supervisionada.

Sendo possível fazer a seguinte distinção de problemas:

- Classificação:  $y_i = f(x_i) \in \{c_1, \dots, c_m\}$ , ou seja,  $f(x_i)$  assume valores em um conjunto discreto, não ordenado;
- Regressão:  $y_i = f(x_i) \in \mathbb{R}$ , ou seja,  $f(x_i)$  assume valores em um conjunto infinito e

ordenado de valores.

A apresentação de uma base de treinamento a diferentes algoritmos de aprendizado de máquina pode resultar em diferentes modelos de classificação ou regressão com saídas diferentes. Além disso, diferenças no conjunto de treinamento, variações na ordem de apresentação dos exemplos durante o treinamento e processos internos estocásticos podem fazer com que até o mesmo algoritmo de aprendizado de máquina encontre fronteiras diferentes a cada novo treinamento (FACELI et al., 2011, p. 54).

A estimativa de qualidade de uma ferramenta de aprendizado supervisionado é dada pelo custo associado às saídas do modelo  $f'$ . Usualmente, o cálculo desse custo é dado da seguinte forma para um conjunto com  $n$  instâncias (FACELI et al., 2011, p. 56):

- Classificação: o custo de uma classificação incorreta  $I(f'(x_i) \neq y_i)$  é 1 e de uma correta  $I(f'(x_i) = y_i)$  é 0. Assim a taxa total de erro é:

$$erro(f') = \frac{1}{n} \sum_{i=1}^n I(y_i \neq f'(x_i));$$

- Regressão: é dado pelo erro quadrático médio,

$$erro(f') = \frac{1}{n} \sum_{i=1}^n (f'(x_i) - y_i)^2.$$

Dessa forma, quanto menor valor de  $erro(f')$  melhor é o desempenho do classificador ou regressor  $f'$ , respectivamente.

Ainda segundo Faceli et al. (2011), o complemento da taxa de erro corresponde à taxa de acerto ou **acurácia** do modelo:

$$acuracia(f') = 1 - erro(f').$$

A abordagem supervisionada é amplamente utilizada para os problemas que envolvem classificação de áudio. A justificativa é que, em se tratando de sons, normalmente é conhecido o conjunto de classes envolvido e exemplos desses grupos, características de aplicações de aprendizado supervisionado.

Portanto, neste texto não serão aprofundadas as demais abordagens, por considerar que não é conhecimento essencial ao que está proposto.

## 3.2 Classificadores Automáticos

Segundo o dicionário Silva e Michaelis (1998), um classificador é aquele que classifica ou se emprega em classificar algo. O ser humano é um inerente classificador. Um exemplo é quando, por meio de seus cinco sentidos perceptivos, ele realiza vários tipos de classificação e em diferentes níveis com relativa facilidade como o reconhecimento do rosto de uma pessoa, que se dá por meio de reconhecimento de padrões; quando aprende, por exemplo, o que deve ser observado em uma fisionomia para poder identificar um rosto conhecido após terem tido vários exemplos de rostos com identificação clara (FACELI et al., 2011).

Diante de grandes quantidades de elementos e informações a serem distribuídas em classes e em grupos, aliados à complexidade de algumas situações, até para os seres humanos, torna-se inviável ou mesmo impossível uma classificação manual.

Nesse sentido, recursos de Inteligência Artificial, em especial de Aprendizado de Máquina, se tornaram grandes aliados. Sistemas mais independentes e autônomos são projetados com o objetivo de realizar atividades de classificação em grande escala automaticamente e sem a necessidade de supervisão de um especialista.

Em aprendizado de máquina, conforme formalizado na seção 3.1.2, um classificador é uma função  $f'$  que para cada elemento de entrada  $x_i$  atribui uma classe, ou rótulo,  $f'(x_i) \in C$  como saída, onde  $C$  é um conjunto discreto e não ordenado.

O desenvolvimento de um classificador segue três etapas básicas: a extração de características dos objetos a serem classificados, a seleção de características que melhor representem as classes e com isso, essas informações são fornecidas ao algoritmo de classificação que induz um classificador (HERRERA-BOYER et al., 2003). Alguns tipos desses algoritmos de classificação serão abordados a seguir.

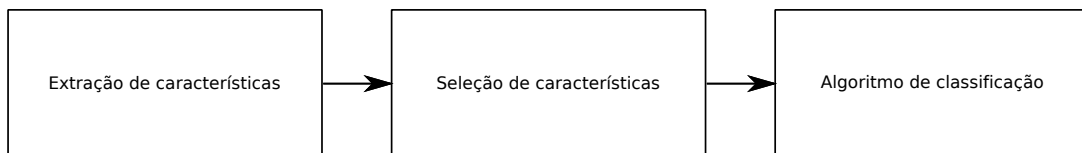


Figura 7: Etapas básicas para concepção de um classificador.

### 3.2.1 Principais Abordagens

Os algoritmos utilizados para conceber classificadores podem ser agrupados em paradigmas que estão ligados à forma de adquirir, tratar e representar o conhecimento. As principais abordagens em um universo de aprendizado de máquina são: simbólica,

estatística, baseada em instâncias, conexionista e evolucionista (ou genética) (MONARD; BARANAUSKAS, 2003).

Todas essas abordagens, umas com mais frequência que outras, são utilizadas na classificação automática de áudio. Essa conexão será tratada na seção 3.3.

Individualmente, elas formam grandes ramos da ciência da computação com ramificações evoluídas e complexas que sua apresentação extrapolam os objetivos deste trabalho. Portanto, serão tratadas a seguir de forma breve, assim como os principais algoritmos de aprendizado de máquina que a seguem, que são apresentados para trazer uma melhor compreensão das particularidades de cada abordagem.

### 3.2.1.1 Simbólico

Os sistemas de aprendizado simbólico buscam construir representações simbólicas de um conceito através da análise de exemplos e contra-exemplos desse conceito. As representações simbólicas estão tipicamente na forma de alguma expressão lógica, árvore de decisão, regras ou rede semântica (MONARD; BARANAUSKAS, 2003).

#### Árvores de Decisão:

A indução de árvores de decisão é uma das formas mais simples, ainda assim mais bem-sucedidas, de algoritmos de aprendizagem. Ela alcança uma decisão executando uma sequência de testes. Cada nó interno corresponde a um teste do valor de uma das propriedades e suas ramificações a partir do nó são identificadas com os valores possíveis do teste (RUSSELL; NORVIG, 2004, p. 633).

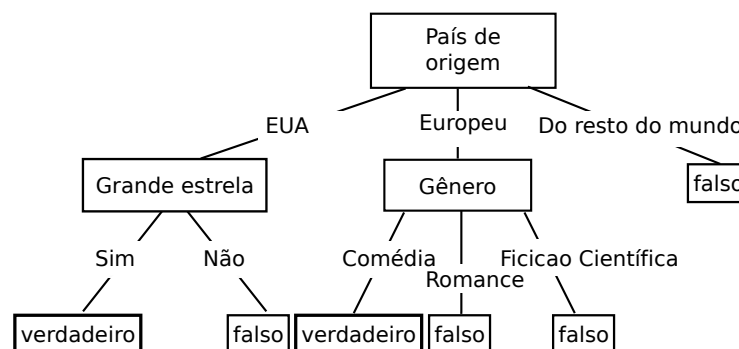


Figura 8: Exemplo de árvore de decisão para determinar o sucesso ou não na bilheteria de um filme (COPPIN, 2012).

O algoritmo de árvore de decisão mais conhecido é o ID3, que constrói uma árvore a partir do topo. Os nós são selecionados pela escolha de atributos do conjunto de dados

de treinamento que forneçam o máximo de informações (ganho de informação) sobre os dados, transformando, então, essas características em perguntas. Ganho de informação é definido como redução de entropia. Isso é o que garante ao ID3 encontrar a menor árvore de decisão possível de classificar corretamente os dados de treinamento (COPPIN, 2012, p. 242).

A entropia de um conjunto de dados,  $S$ , formado por exemplos positivos e negativos de um valor objetivo, é dado pela fórmula (COPPIN, 2012, p. 243):

$$H(S) = -p_1 \log_2 p_1 - p_0 \log_2 p_0,$$

onde  $p_1$  é definido como a proporção dos dados de treinamento que inclui exemplos positivos e  $p_0$  relativo aos exemplos negativos. A entropia será zero quando todos os exemplos estiverem concentrados em apenas um dos valores (positivo ou negativo). E assumirá o valor de 1 quando houver 50% positivos e 50% negativos.

O cálculo da entropia total do atributo é dado pela soma das entropias de cada subconjunto de valores possíveis para esse atributo.

Após se obter o valor total de cada atributo, o que representar a menor entropia se torna a pergunta do nó atual da árvore. Processo que é repetido até esgotados todos os atributos. Formando ao final uma árvore que, cada nó interno testa um atributo, cada ramo corresponde a um valor do atributo e cada folha representa uma classe.

### **Logistic Model Trees (LMT)**

Logistic Model Tree (LANDWEHR et al., 2005a) é um modelo que combina conhecimentos de regressão logística com aprendizado via árvores de decisão.

Elas adaptam a ideia das árvores de modelo para problemas de classificação com a diferença de usar regressão logística em vez de regressão linear. É usado um processo de adaptação por etapas para construir os modelos logísticos de regressão nas folhas, redefinindo incrementalmente aqueles construídos em níveis superiores da árvore.

#### **3.2.1.2 Estatístico**

A ideia geral da abordagem estatística consiste em utilizar métodos estatísticos para encontrar uma boa aproximação do conceito induzido. Um dos métodos que se destacam nesse tipo de abordagem é o aprendizado Bayesiano, que utilizam um método probabilístico baseado no conhecimento prévio do problema, o qual é combinado com os exemplos de treinamento para determinar a probabilidade final de uma hipótese (MONARD; BARA-



NAUSKAS, 2003).

### Redes Bayesianas

Uma forma de trabalhar com tarefas supervisionadas em aprendizado de máquina, principalmente quando as informações disponíveis são incompletas ou imprecisas, é por meio do uso de algoritmos baseados no método Bayesiano. Esse método assume que a probabilidade de um evento  $A$ , que pode ser uma classe (por exemplo, um doente apresentar um determinada doença), dado um evento  $B$ , que pode ser o conjunto de valores dos atributos de entrada (por exemplo, ter um resultado positivo em um exame de raios X), não depende apenas da relação entre  $A$  e  $B$ , mas também da probabilidade de observar  $A$  independente de observar  $B$  (MITCHELL, 1997 apud MONARD; BARANAUSKAS, 2003).

O desmembramento se origina de Jain et al.; Kotsiantis et al. (apud RODRIGUES, 2012):

A probabilidade de um evento discreto  $A$  ocorrer é  $P(A)$ , e dada uma variável aleatória contínua ( $x$ ) a função de densidade de probabilidade é definida como  $p(x)$ . Para um vetor de variáveis aleatórias  $\mathbf{x}$ , a função de densidade de probabilidade é definida como  $p(\mathbf{x})$ , e a probabilidade condicional de  $A$  dado  $B$  é  $P(A|B)$ . Para o vetor  $\mathbf{x}$ ,  $p(\mathbf{x}|A)$  é definida como a probabilidade condicional de  $\mathbf{x}$  dado  $A$ .

A abordagem Bayesiana supõe que as probabilidade de cada classe  $P(\omega_i|\mathbf{x})$ , definida como:

$$P(\omega_i|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_i)P(\omega_i)}{p(\mathbf{x})},$$

onde:

- $p(\mathbf{x}|\omega_i)$  é a probabilidade condicional de classe;
- $P(\omega_i)$  é a probabilidade a priori;
- $p(\mathbf{x}) = \sum_{j=1}^c P(\omega_j)p(\mathbf{x}|\omega_j)$  que é a evidência;
- $P(\omega_i|\mathbf{x})$  é a probabilidade a posteriori.

A regra de decisão de Bayes que resulta em uma mínima taxa de erro é dada como "Decidir  $\omega_i$  se  $P(\omega_i|\mathbf{x}) \geq P(\omega_j|\mathbf{x}), \forall j \neq i$ ". Conhecendo-se as distribuições de probabilidades, classifica-se sempre com a classe que tem maior probabilidade de ocorrer para um dado  $\mathbf{x}$ , conforme:

$$f_B(\mathbf{x}) = \arg_i \max_{\omega_i \in C} P(\omega_i|\mathbf{x}).$$

### 3.2.1.3 Baseado em Instâncias

Os métodos baseados em instâncias partem da filosofia básica de que se conhecendo um exemplo em memória, ao surgir um outro similar, a classe atribuída será a mesma do primeiro.

Esse tipo de aprendizado é conhecido como *lazy learning* (aprendizado preguiçoso). Esses sistemas necessitam manter os exemplos na memória para classificar novos exemplos, o que pode ser uma desvantagem, dependendo da quantidade de instâncias a serem memorizadas.

Assim, saber quais instâncias de treinamento devem ser mantidos na memória do sistema para comparação é muito importante. O ideal é manter os mais representativos do problema (MONARD; BARANAUSKAS, 2003).

#### ***k*-Nearest Neighbors (*k*-NN)**

O *k*-NN, *k*-vizinhos mais próximos, é um exemplo de aprendizado baseado em instâncias. Ele opera em situações as quais cada instância pode ser definida por um vetor de  $n$  dimensões, onde  $n$  é o número de atributos usados para descrever cada instância e as classificações sejam valores discretos. Os dados de treinamento são armazenados e, quando uma nova instância for encontrada, ela será então comparada aos dados de treinamento para encontrar seus vizinhos mais próximos (COPPIN, 2012, p. 247).

Os vizinhos mais próximos são aqueles que estão mais próximos seguindo a distância Euclidiana. O cálculo da distância entre dois elementos  $A = \langle a_1, \dots, a_n \rangle$  e  $B = \langle b_1, \dots, b_n \rangle$  é feito da seguinte forma:

$$d = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}.$$

Ordenados os  $k$  vizinhos mais próximos da nova instância, a classificação atribuída a ela será a classe com maior ocorrência entre eles.

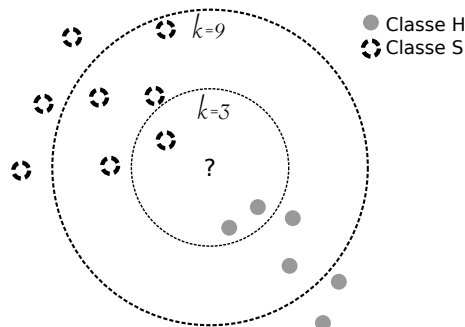


Figura 9: Funcionamento de um simples *k*-NN, com  $k = 3$  e  $k = 9$ .

Note que na Figura 9 se  $k = 3$  a nova instância "?" será rotulada como sendo da classe  $H$ , já se  $k = 9$ , "?" será classificado como pertencente à classe  $S$ . Isso mostra a importância do  $k$  para um classificador desse tipo.

Ao contrário da árvore de decisão, o algoritmo  $k$ -NN tem bom desempenho com dados de entrada com ruído (COPPIN, 2012, p. 247).

### Regressão Localmente Ponderada

A Regressão Localmente Ponderada usa exemplos de treinamento ponderado por sua distância para formar uma aproximação a uma função  $f'$ . Essa função de aproximação pode ser uma função linear, quadrática, rede neural ou alguma outra.

Dada uma instância a classificar  $x_q$ , a abordagem constrói uma aproximação  $f'$  usando os vizinhos de  $x_q$ . Essa aproximação é utilizada para calcular  $f'(x_q)$ .

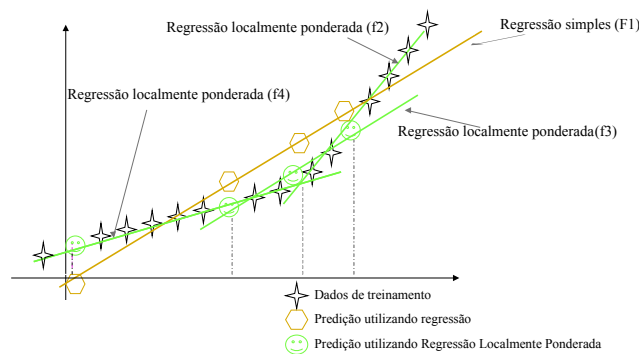


Figura 10: Exemplo de Regressão Localmente Ponderada com regressão linear.

Uma das formas de atribuição de pesos se dá pela forma denominada tri-cube:

$$W(x_i) = \left(1 - \left|\frac{x - x_i}{d(x)}\right|^3\right)^3,$$

onde  $x$  consiste na posição da abscissa associada ao valor a ser suavizado,  $x_i$  representa um ponto qualquer entre os vizinhos considerados;  $d(x)$  corresponde à distância ao longo da abscissa entre o ponto avaliado  $x$  e o vizinho mais distante (PIRES, 2012).

#### 3.2.1.4 Conexionista

A abordagem conexionista leva em consideração também aspectos biológicos do aprendizado. Tenta-se reproduzir a dinâmica do cérebro humano em sistemas, por meio de redes neurais.

### Redes Neurais Artificiais

Assim como as do cérebro humano, as redes neurais artificiais (RNA) são compostas por neurônios que são unidades de processamento simples. Mas, quando interligado a outros em uma rede, podem gerar resultados para problemas complexos.

Nas redes artificiais, cada neurônio recebe uma série de entradas às quais é aplicada uma função de ativação o que resulta no valor de saída do neurônio. Há uma série de funções que podem ser utilizadas nos neurônios. Algumas das funções de ativação mais comumente usadas estão ilustradas na Figura 11 (COPPIN, 2012).

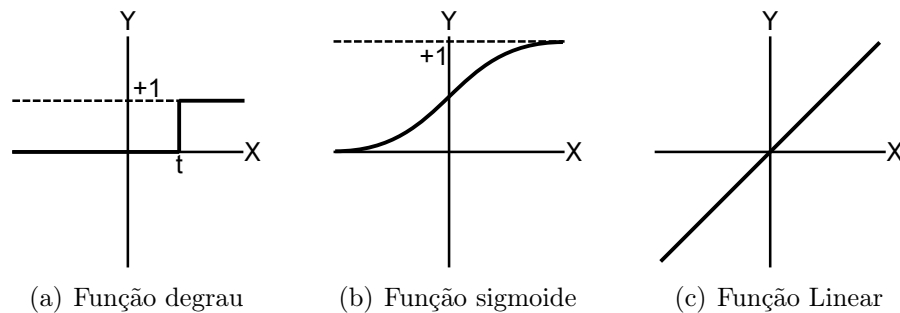


Figura 11: Três funções de ativação (COPPIN, 2012).

Ao usar a função de ativação degrau, as entradas para o neurônio são somadas após cada uma ter sido multiplicada por seu respectivo peso e esta soma é comparada com um limiar  $t$ . Se a soma for maior do que o limiar, o neurônio ativará e terá um nível de ativação de  $+1$ . Caso contrário, ficará inativo com nível de ativação  $0$ . Em algumas redes a inativação assume o valor de  $-1$  em vez de zero. (COPPIN, 2012)

Dessa forma, matematicamente, o comportamento do neurônio será expresso como:

$$X = \sum_{i=1}^n \omega_i x_i,$$

onde  $X$  é a soma ponderada das  $n$  entradas do neurônio, de  $x_1$  a  $x_n$ , na qual cada entrada  $x_i$  é multiplicada pelo seu peso corresponde  $\omega_i$ .

Já um neurônio que use a função de ativação linear usa apenas a soma ponderada de suas entradas como nível de ativação, enquanto os que utilizam uma função sigmoide converte entradas na faixa de  $-\infty$  a  $+\infty$  em um nível de ativação no intervalo aberto  $(0, 1)$ .

Uma rede neural é formada por vários neurônios conectados, podendo haver mais de uma camada de neurônios (camadas ocultas). Durante a fase de treinamento das saídas obtidas são calculados os erros em comparação às saídas desejadas. Esses valores são

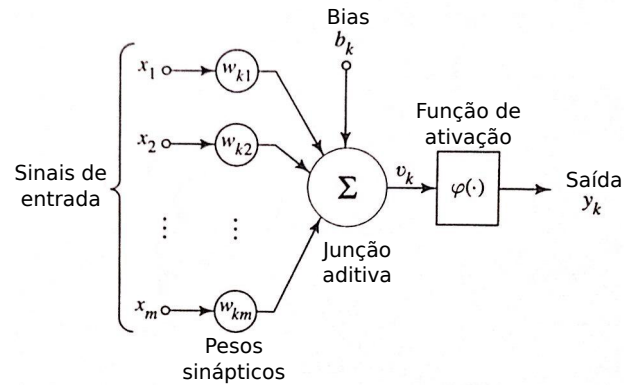


Figura 12: Modelo não-linear de um neurônio (SIMON, 2001).

então utilizados para ajuste de todos os pesos da rede. Até que esse erro chegue a um valor nulo ou abaixo de um limite pré-determinado pelo usuário ou por limite de iterações de ajuste.

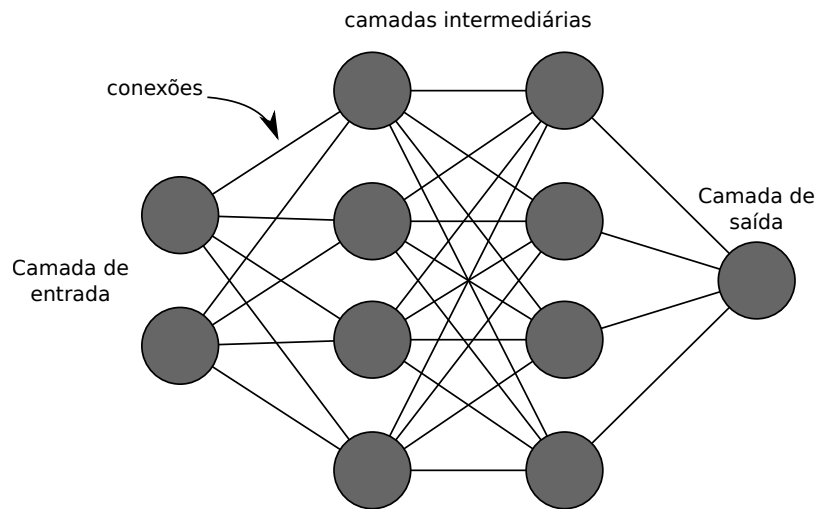


Figura 13: RNA em camadas.

Assim, se obtém um classificador pronto para classificar todos os novos exemplos que são dados como entrada.

### 3.2.1.5 Genético

Esta abordagem é derivada do modelo evolucionário de aprendizado e possui uma analogia direta com a Teoria da Evolução das Espécies de Charles Darwin a qual afirma que a maior probabilidade de sobrevivência e reprodução em um meio ambiente de competição de recursos pertence aos indivíduos mais fortes e mais bem adaptados. Esse é o princípio da *Seleção Natural*.

Um classificador evolucionário consiste de uma população de elementos de classificação que competem para fazer a predição. Elementos que possuem uma performance fraca são descartados, enquanto os elementos mais fortes proliferam, produzindo variações de si mesmos (MONARD; BARANAUSKAS, 2003). A dinâmica destes modelos herdam propriedades genéticas como aptidão e operações como cruzamento e mutação.

Uma representação para Algoritmos Genéticos (AG) é a que foi usada por John Holland: uma cadeia de *bits* é conhecida como um **cromossomo** e cada *bit* é conhecido como **gene**. Termos ligados diretamente à genética, o que mostra a proximidade dos AG com processos biológicos. Há ainda as **populações** que consistem em conjuntos de cromossomos que muitas vezes é considerado um indivíduo completo dentro de uma população (COPPIN, 2012, p. 335).

### 3.3 Classificadores de Áudio

A classificação automática de áudio é um subconjunto dos problemas de classificação. É uma ramificação heterogênea que envolve aplicações em vários domínios em uma área de pesquisa própria.

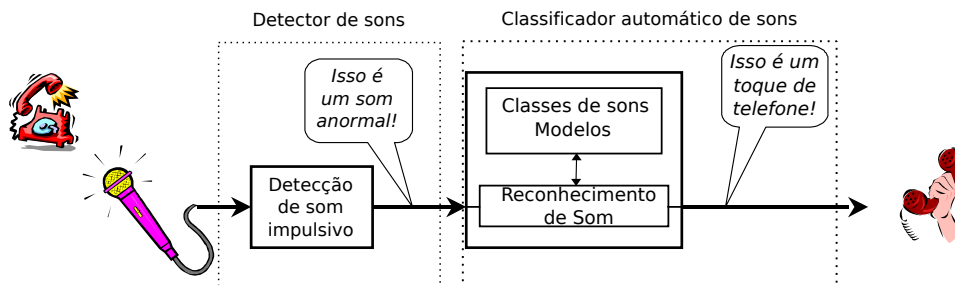


Figura 14: Visão geral de um detector e classificador de áudio (DUFAUX et al., 2000)-modificada.

Primeiramente, é importante destacar as abordagens de classificação de áudio existentes. Geralmente, os classificadores de áudio podem seguir dois caminhos: classificar de forma **perceptiva** ou classificar de forma **taxonômica** um som. A primeira forma considera experimentos utilizando seres humanos ou algum sistema artificial que simule alguns de seus processos perceptivos (HERRERA-BOYER et al., 2003).

Por outro lado, existe também a forma taxonômica que tenta encontrar entre características extraídas de um elemento semelhanças comuns a grupos previamente estabelecidos. Nesse caso, é realizada uma classificação pela similaridade das características do elemento com aquelas apresentadas pelos grupos.

Assim como outros problemas de classificação, o desenvolvimento de uma ferramenta de classificação automática de áudio é uma tarefa trabalhosa e complexa. Herrera-Boyer et al. (2003) descreve o processo geral para obtenção de um classificador de áudio em quatro etapas:

- Listas de características são selecionadas para descrever as amostras;
- Os valores para estas características são computados;
- Um algoritmo de aprendizagem que utiliza as características selecionadas para discriminar entre as famílias de instrumentos ou classes é aplicado;
- O desempenho do processo de aprendizagem é avaliado através da classificação de novas amostras de som (validação cruzada).

O áudio tem várias especificidades em comparação a outros objetos classificáveis. A começar pela sua forma. O sinal de áudio é um elemento físico, ondulatório e analógico. É classificado na Acústica, ciência que o estuda, como um fenômeno ondulatório causado pelos mais diversos objetos e que é capaz de se propagar através dos diferentes estados físicos das matérias.

Antes de alimentar um algoritmo de classificação com uma base de treinamento, técnicas são necessárias para transformar sons em sinais digitais, para transpor de um sinal contínuo para uma representação discreta. Dessa nova representação são extraídas os atributos de cada exemplo, também conhecidos como características de áudio.

Um número considerável de características de áudio tem sido usado para as tarefas de análise e recuperação de áudio baseado em conteúdo. Tradicionalmente, essas características foram classificadas em características temporais, como *Zero Crossing Rate (ZCR)*; características de energia, como *Short Time Energy (STE)*; características espectrais, por exemplo, momentos espectrais e nivelamento espectral; características perceptivas, como *loudness*, *sharpness* ou *Coefficientes Mel Frequency Cepstral (MFCCs)* (VALENZISE et al., 2007).

Todas essas técnicas de extração de informações têm vários desmembramentos e possibilidades de combinações, o que é muito explorado por pesquisadores pela sua importância dentro do processo e influência sobre o desempenho de um classificador gerado.

### 3.3.1 Coeficientes Mel Frequency Cepstral (MFCCs)

Os coeficientes Mel Frequency Cepstral (MFCCs) são exemplos de características acústicas que são utilizadas para reconhecimento de voz e modelagem de música (TSOUMAKAS et al., 2008 apud LOGAN et al., 2000). Autores como Tzanetakis e Cook (2002a) utilizam MFCC em conjunto com outras características que podem ser extraídas do sinal para auxiliar seus processos de classificação automática de áudio (FILHO, 2006).

MFCC é uma escala perceptual de pitches (tons) baseada na escala Mel. A escala Mel foi proposta inicialmente por Stevens e Volkman (STEVENS; VOLKMANN, 1940 apud FILHO, 2006) 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 frequência da nota ouvida e sua frequência fundamental, ou seja, o pitch. De maneira simples, o pitch é a medida da resposta humana ao estímulo recebido por um som a qual representa a relação entre a faixa de frequência em que esse som se encontra e a sua frequência fundamental. A medida dessa distância é que define exatamente o pitch do sinal (FILHO, 2006).

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 frequência e, de modo análogo, um acorde é mais baixo (mais grave) ou possui menor pitch se possuir uma baixa frequência (RABINER; JUANG, 1993 apud FILHO, 2006).

Para derivar características MFCCs, o sinal de som é dividido em quadros e o espectro de amplitude é calculado para cada quadro. Em seguida, é calculado o seu logaritmo e convertido em escala Mel. (LOGAN et al., 2000)

### 3.3.2 Soluções Existentes

Os classificadores automáticos de áudio existentes podem também ser categorizados em dois grupos: um grupo **dominante** formado pelos classificadores concebidos através de uma **abordagem exploratória** e outro ainda **recente** no qual os classificadores são desenvolvidos a partir de uma **abordagem sistemática**.



### 3.3.2.1 Abordagem Exploratória

Esse grupo é composto em maior parte por soluções criadas para problemas específicos. Por exemplo, quando está se desenvolvendo um problema de recomendação no qual o objetivo é conhecer, dentre um elenco de estilos de músicas, a qual estilo um novo exemplo pertence. Isto é, o método já é diretamente associado aos dados. As pessoas desenvolvem métodos e algoritmos diferentes.

Nesse sentido, é possível citar alguns exemplos de classificadores, como Saha et al. (2013), que tenta classificar o choro de crianças; Tzanetakis e Cook (2002b) que realiza a classificação automática de músicas por gênero; Navratil et al. (2010) que classifica explosões e tiros para a indústria militar; Lee et al. (2008) que classifica pássaros pelo sons emitido; etc. Além de diversos produtos comerciais que, geralmente, não são registrados por meio de trabalhos acadêmicos.

Essa abordagem é um método que acaba por ser mais exploratório, no qual cada desenvolvedor recorre muito a conhecimentos intuitivos no sentido de querer um classificador para o seu problema específico.

Esse tipo de abordagem ainda é bem mais utilizada do que a abordagem sistêmica que será tratada a seguir.

### 3.3.2.2 Abordagem Sistemática

Por outro lado, o outro grupo é uma exceção **recente** que pode ser classificado como detentor de uma abordagem sistemática, no qual o usuário pode encaixar sua base de dados a ferramentas existentes que auxiliam de alguma forma durante o processo, tentando assim sistematizar a criação de um classificador de áudio.

Um exemplo é o **Extractor Discovery System (EDS)**(PACHET; ZILS, 2004) cujo objetivo principal é, dada uma base de dados de sons, encontrar as melhores características, os atributos mais representativos.

O EDS foi desenvolvido na Sony CSL e é uma abordagem baseada em heurísticas para extrair automaticamente descritores de músicas a partir de sinais acústicos.

O sistema utiliza em alguns processos programação genética, como para construir funções de extração com composições de operadores de processamento de sinal, como log, variância e transformada rápida de Fourier (FFT).

Dado um banco de dados de sinais de áudio com seus valores perceptivos associados,

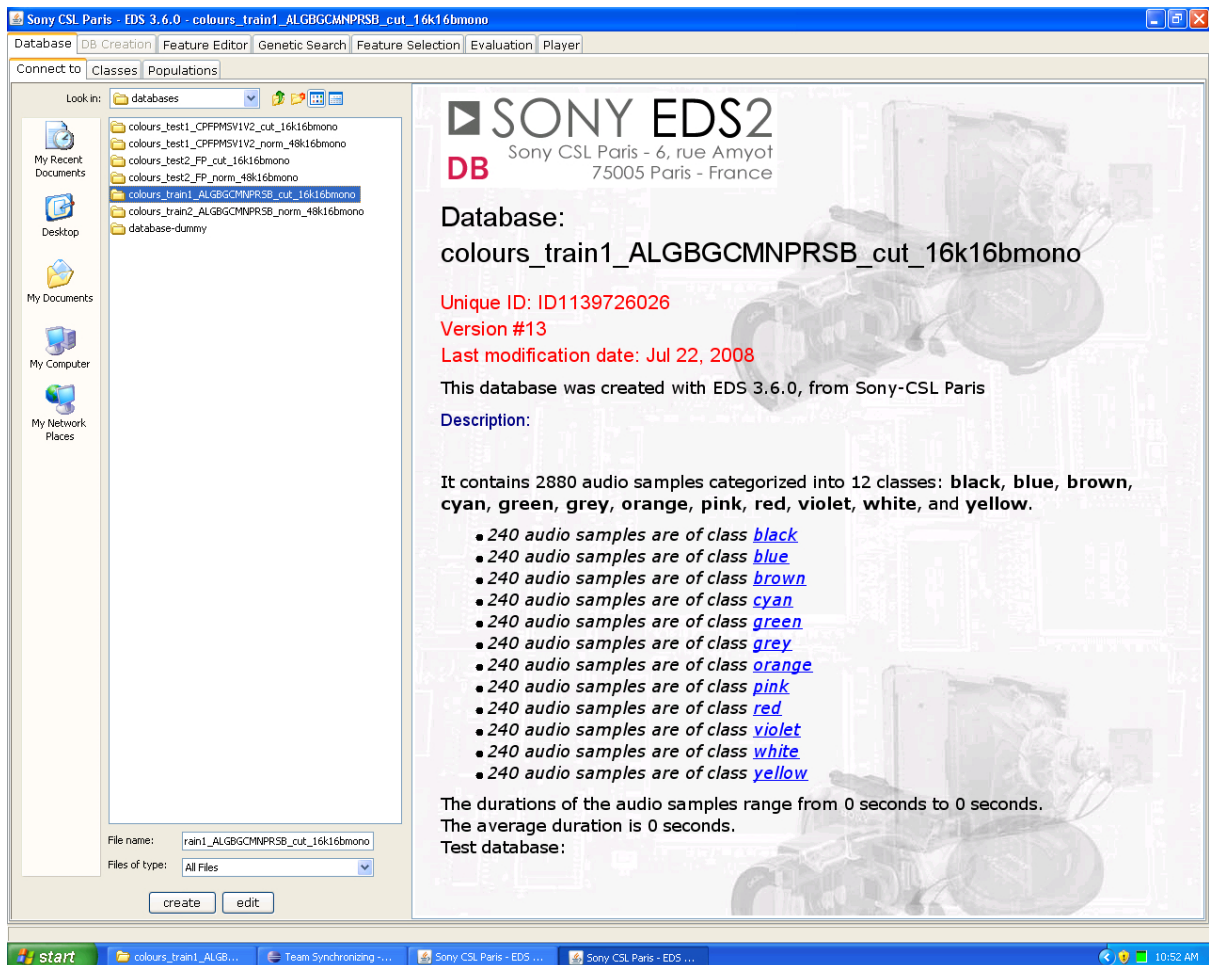


Figura 15: Extractor Discovery System (EDS), (PÉREZ, 2011)

o EDS (Figura 15) é capaz de generalizar um descritor. Esse descritor é construído através da combinação de uma pesquisa genética que objetiva encontrar características de processamento de sinal relevantes e algoritmos de aprendizado de máquina (CABRAL et al., 2005; CABRAL et al., 2006).

O passo final no processo de extração é escolher e induzir um modelo. O EDS dispõe de alguns algoritmos de aprendizado de máquina (regressão linear, árvores de decisão,  $k$ -NN, regressão ponderada localmente, redes neurais artificiais, etc.) os quais são **selecionados pelo usuário**, esse podendo realizar testes e ajustes de parâmetros. Como saída, o EDS cria um modelo que classifica um arquivo de áudio passado como argumento (CABRAL et al., 2006).

A **principal desvantagem** do EDS é que não está disponível para livre utilização da comunidade que trabalha com esse tipo de problema. É utilizada restritamente pela Sony e suas subsidiárias. Além disso, ferramentas semelhantes não são facilmente encontradas,

o que torna o grupo de classificadores desenvolvidos através dessa abordagem sistêmica quase insignificante diante da dimensão do grupo formado por aqueles concebidos de forma mais exploratória.

### 3.4 Meta-aprendizado

Conforme Faceli et al. (2011, p. 270), um dos principais desafios enfrentados no momento da utilização de algoritmos de aprendizado de máquina em novos conjuntos de dados é a escolha do algoritmo mais apropriado dentro do universo de alternativas existentes. Faltam regras ou dicas que auxiliem na escolha do algoritmo mais apropriado para um dado problema. Essa escolha geralmente ou ocorre por tentativa e erro ou é influenciada pela disponibilidade do algoritmo em uma ferramenta computacional utilizada, ou por experiências passadas do usuário ou por sugestões de especialistas em aprendizado de máquina. Essas abordagens, além de serem subjetivas, podem ter um custo computacional elevado.

No cenário atual, em que não existe um único algoritmo de aprendizado de máquina que seja sempre melhor que os demais para todos os problemas (WOLPERT, 1996 apud FACELI et al., 2011, p. 270) (MONARD; BARANAUSKAS, 2003), o meta-aprendizado surge como uma área que, dado um problema, estuda formas de seleção do(s) algoritmo(s) de aprendizado mais promissor(es) por meio de acúmulo de conhecimento. Ou como definido por Giraud-Carrier et al. (apud PRUDÊNCIO; LUDERMIR, 2007), Meta-Aprendizado é um arcabouço desenvolvido no campo de Aprendizado de Máquina com o objetivo de selecionar algoritmos de aprendizado.

O aprendizado no meta-aprendizado se dá por meio de experiências obtidas através de observações do desempenho de diversos algoritmos de aprendizado quando aplicados a vários conjuntos de dados (BRAZDIL et al., 2009 apud FACELI et al., 2011, p. 271).

Esse aprendizado é transformado em um meta-exemplo que é armazenado, gerando um acúmulo de experiências, mais especificamente, cada meta-exemplo armazena as seguintes informações, conforme Prudêncio e Ludermir (2007):

- as características que descrevem um dado problema (como o número de exemplos de treinamento, número de atributos, entropia do atributo classe, ...) que possam influenciar no desempenho de um algoritmo de aprendizado de máquina;
- informações de desempenho dos algoritmos quando aplicados ao problema (como o

melhor algoritmo, erros obtidos, tempos de execuções, ...).

A Figura 16 representa a dinâmica de aprendizado de um sistema de meta-aprendizado.

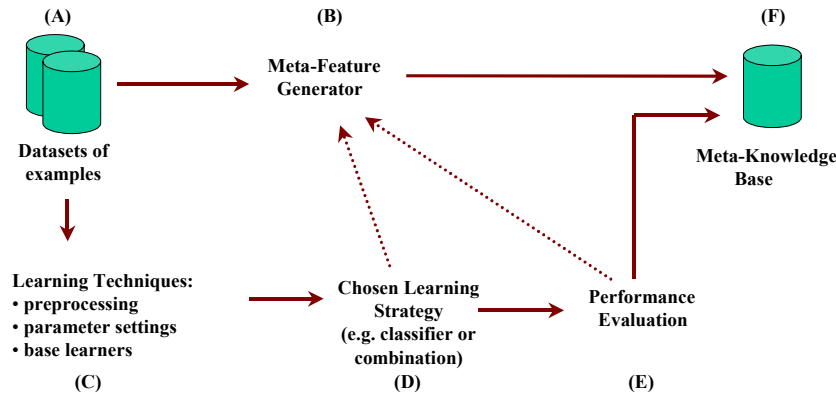


Figura 16: Meta-aprendizado: aquisição de conhecimento (VILALTA et al., 2004).

A decisão do algoritmo ou dos algoritmos mais apropriados para um conjunto de dados pressupõe a utilização de medidas que avaliem o desempenho dos mesmos. A princípio, qualquer medida de desempenho empregada na avaliação dos algoritmos candidatos pode ser adotada. Assim como combinações entre diferentes medidas. (FACELI et al., 2011).

Essas informações são utilizadas para dar origem a um sistema de recomendação (Figura 17), que também pode ser baseado em aprendizado de máquina, em um meta-nível, capaz de sugerir os algoritmos mais promissores para o problema de entrada. Caso o resultado seja mais de um algoritmo, o sistema pode ainda utilizar um *ranking* para ordená-los de acordo com seu desempenho perante alguma medida de avaliação (BRAZDIL et al., 2003).

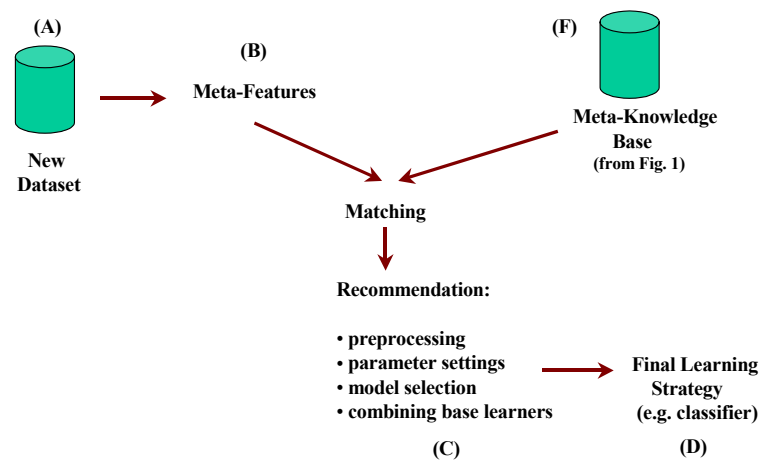


Figura 17: Meta-aprendizado: processo de recomendação (VILALTA et al., 2004).

Apesar das contribuições que o meta-aprendizado pode proporcionar à área de aprendizado de máquina, há dificuldades de se encontrar trabalhos na literatura que os relaci-

onem com problemas de classificação automática de áudio. Assim como ferramentas disponíveis que auxiliem na utilização desse conhecimento. O que torna o meta-aprendizado um conhecimento ainda restrito à pesquisas científicas.

## 3.5 Ferramentas

Como visto na seção 3.2, existem etapas consideradas básicas para a elaboração de um classificador independente do problema de classificação tratado. E como acabam sendo tarefas recorrentes na área de Aprendizado de Máquina, é natural que existam ferramentas para diminuir o esforço de pesquisadores da área durante esses processos.

A seguir, serão descritas algumas dessas ferramentas.

### 3.5.1 Weka

O WEKA (Waikato Environment for Knowledge Analysis) é uma coleção de algoritmos de aprendizado de máquina para tarefas de mineração de dados escritos na linguagem de programação Java.

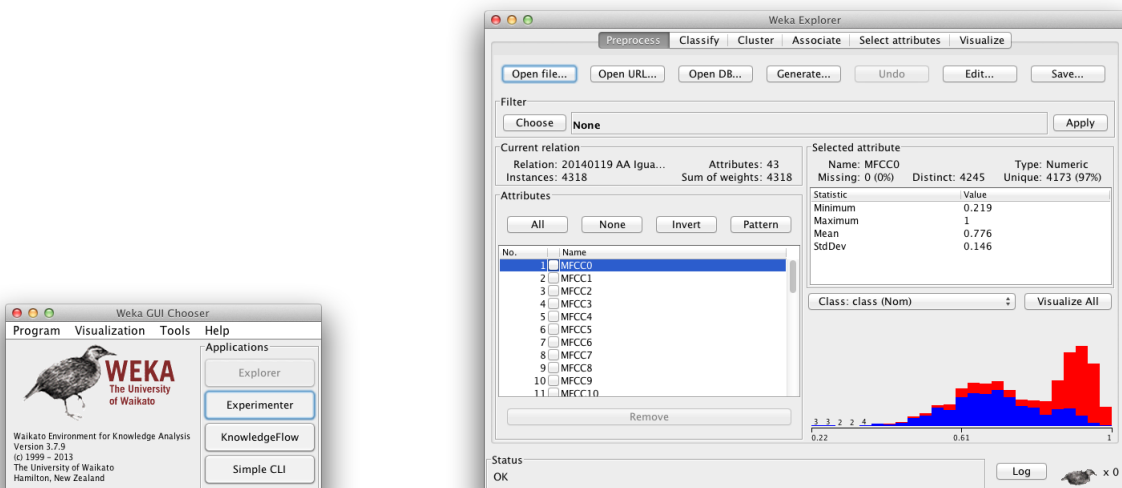
Ele contém ferramentas para pré-processamento de dados, classificação, regressão, agrupamento, regras de associação e visualização. Pode ser utilizado diretamente ou chamado dentro de um código em Java.

Foi desenvolvido por pesquisadores na Universidade de Waikato na Nova Zelândia e utiliza licença GNU General Public License (GPL) (HALL et al., 2009), disponibilizando seus códigos fontes em repositório público (<http://www.cs.waikato.ac.nz/ml/weka/>).

O WEKA (Figura 18) utiliza um formato de arquivo com extensão *.arff*, que armazena os dados de uma maneira específica. Alguns softwares, como o MARSYAS (seção 3.5.6.1), já exportam dados diretamente para esse formato. Assim como o Weka é capaz de importar dados em outros formatos, como *.csv* e *.m*, esse último originário do Matlab.

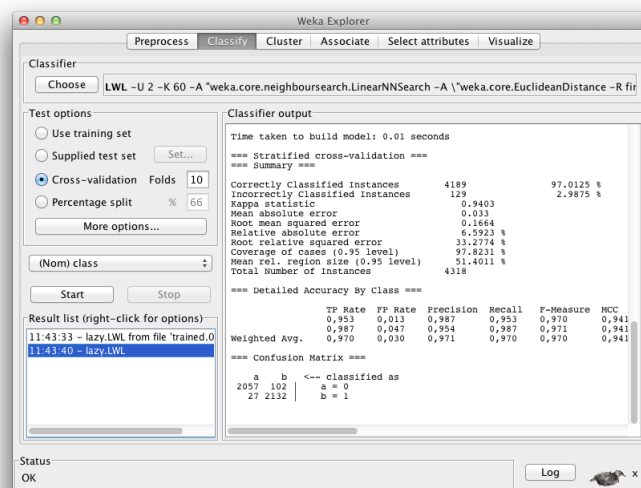
### 3.5.2 RapidMiner

O RapidMiner (Figura 19) é uma plataforma de software desenvolvido pela empresa de mesmo nome, que fornece um ambiente integrado para a aprendizagem de máquina, mineração de dados, mineração de texto, análise preditiva e análise de negócios. Ele é usado para aplicações comerciais e industriais, bem como para pesquisas, educação,



(a) Weka.

(b) Weka: Pré-processamento.



(c) Weka: Seleção de algoritmo e resultados.

Figura 18: Weka.

formação, prototipagem rápida e desenvolvimento de aplicações dando suporte a todas as etapas de processos de mineração de dados, incluindo resultados de visualização, validação e otimização.

O núcleo da ferramenta, assim como as versões anteriores estão disponíveis através da licença OSI-certified open source em <http://rapidminer.com/>. (HOFMANN; KLINKENBERG, 2013 apud WIKIPEDIA, 2014b).

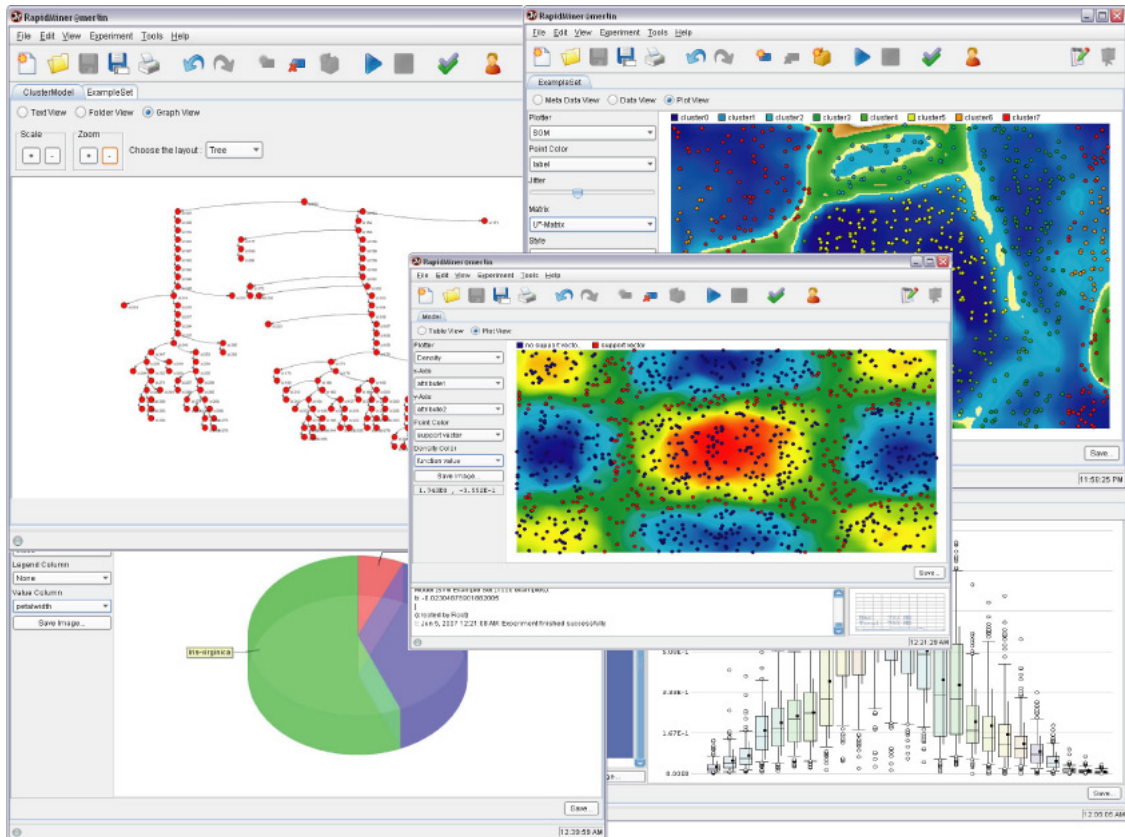


Figura 19: RapidMiner.

### 3.5.3 PyBrain

O Pybrain (Python-Based Reinforcement Learning, Artificial Intelligence and Neural Network Library) é uma biblioteca de aprendizado de máquina versátil para Python. Seu objetivo é oferecer, de forma simples e fácil de usar, poderosos algoritmos para tarefas de aprendizado de máquina, incluindo uma variedade de ambientes pré-definidos para testar e comparar os algoritmos dos usuários.

A ferramenta é código aberto e está licenciada sob a Licença de Software BSD (SCHAUL et al., 2010) e pode ser encontrada em <http://pybrain.org/>.

### 3.5.4 Torch

O Torch é uma estrutura de computação científica com amplo apoio para os algoritmos de aprendizado de máquina. Ele pode ser usado para treinar MLPs, RBFs, HMMs, Gauss Misturas Kmeans, Misturas de especialistas, Parzen do Windows,  $k$ -NN, e pode ser facilmente estendido para que você possa adicionar seus próprios algoritmos de aprendizado de máquina. Torch é atualmente desenvolvido na Idiap Research Institute, New York

University e NEC Laboratories America, distribuído sob a licença BSD e está disponível em [torch.ch/torch3/](http://torch.ch/torch3/).

É executado em Linux, FreeBSD e MacOS X. Mas pacotes Core também funcionam no Windows (COLLOBERT et al., 2011; MUSIC-IR, 2014).

### 3.5.5 Auto-WEKA

Com as ferramentas citadas nas seções anteriores é possível realizar vários testes com diferentes algoritmos. Basicamente o usuário entra com uma base e seleciona um algoritmo classificador e a parametrização. A partir desses dados, a ferramenta cria um classificador com um subconjunto da base de dados (caso seja escolhido um método supervisionado), executa o teste com a outra parte da base e retorna ao usuário informações importantes, como acurácia, erro médio e outras medidas de desempenho. Assim como a possibilidade de exportar o classificador gerado, caso desejado.

Apesar da chance de praticamente dispensar o usuário da tarefa de implementar um algoritmo de classificação para conhecer o comportamento com a base de dados que detém, ainda fica ao arbítrio do usuário a seleção do algoritmo de classificação e a respectiva parametrização que deseja utilizar diante de um universo de inúmeras possibilidades.

O Auto-WEKA (THORNTON et al., 2012) é um exemplo de ferramenta recente que tenta automatizar esse processo.

O Auto-WEKA visa auxiliar usuários não especializados em técnicas de aprendizado de máquina na seleção de algoritmos de classificação. A abordagem mais simples para a seleção de um classificador amplamente adotado entre os não-especialistas é a adoção de um algoritmo de classificação apenas pela sua intuição ou popularidade do método, sem qualquer consideração técnica (THORNTON et al., 2012).

Ao ser apresentado a um conjunto de dados, o sistema realiza uma otimização para recomendar ao usuário um algoritmo de classificação com respectiva parametrização relacionado ao problema, além de fornecer a estimativa da taxa de erro para o modelo sugerido.

A base de algoritmos de aprendizado de máquina, considerada por esse sistema, é composta por algoritmos que compõe o Weka (tratado na seção 3.5.1). No entanto, a depender da base de treinamento, essa quantidade pode ser reduzida pelo próprio sistema, uma vez que nem todos os algoritmos são aplicáveis a todos os conjuntos de dados.



Para o Auto-Weka, a escolha de um algoritmo em si é considerado também a escolha de um parâmetro. Assim o problema é reduzido à uma otimização de hiper-parâmetros. E para tal, são utilizados dois procedimentos de otimização bayesiana: o TPE (BERGSTRA et al., 2011 apud THORNTON et al., 2012) e o SMAC (HUTTER et al., 2011 apud THORNTON et al., 2012).

### Seleção do modelo:

Dado um conjunto  $\mathcal{A}$  de algoritmos de aprendizado e uma quantidade limitada de dados de treinamento  $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ , o objetivo do Auto-WEKA é determinar o algoritmo  $A \in \mathcal{A}$  que otimiza a performance em  $\mathcal{D}$ .

Para isso, o conjunto  $\mathcal{D}$  é dividido nos subconjuntos disjuntos de treinamento  $\mathcal{D}_{\text{train}}^{(i)}$  e validação (ou teste)  $\mathcal{D}_{\text{valid}}^{(i)}$ .

A performance é estimada pela aplicação da função de aprendizado  $f_i$  do modelo  $A$  sob  $\mathcal{D}_{\text{train}}^{(i)}$  e avalia a performance dessa função em  $\mathcal{D}_{\text{valid}}^{(i)}$ . Isso permite escrever o problema de seleção de modelo como:

$$A \in \arg \min_{A \in \mathcal{A}} \frac{1}{k} \sum_{i=1}^k \mathcal{L}(A, \mathcal{D}_{\text{train}}^{(i)}, \mathcal{D}_{\text{valid}}^{(i)}),$$

onde,  $\mathcal{L}(A, \mathcal{D}_{\text{train}}^{(i)}, \mathcal{D}_{\text{valid}}^{(i)})$  é a taxa de erro alcançada por um algoritmo  $A$  quando treinado com  $\mathcal{D}_{\text{train}}^{(i)}$  e avaliado com  $\mathcal{D}_{\text{valid}}^{(i)}$ . Usando classificação  $k$ -fold cross-validation, que divide o conjunto de treinamento em  $k$  partes iguais e atribui  $\mathcal{D}_{\text{train}}^{(i)} = \mathcal{D} \setminus \mathcal{D}_{\text{valid}}^{(i)}$  para  $i = 1, \dots, k$ .

Ao final de seu trabalho, Thornton et al. (2012) não só realiza algumas análises e comparações, mas também conclui que o Auto-WEKA frequentemente supera outros métodos de seleção de algoritmos, principalmente empíricos e especialmente em grandes conjuntos de dados.

A diferença entre o Auto-Weka e os sistemas de meta-aprendizado citados na seção 3.4 é que apesar de serem todos sistemas de recomendação, o Auto-WEKA não salva na memória experiências acumuladas para serem levadas em consideração em futuras recomendações. Ele apenas faz uma busca naquele instante e retorna o resultado.

### 3.5.6 Extração de Características de Áudio

Além das ferramentas já citadas neste trabalho, como Weka, RapidMiner, PyBrain, Torch (seção 3.5), Auto-Weka (seção 3.5.5), EDS (seção 3.3.2.2) existem outras ferramentas utilizadas pela comunidade científica e profissionais da área de aprendizado de máquina

para auxiliar tarefas em projetos de classificação automática de áudio. O Marsyas (TZANETAKIS; COOK, 2000) e o Sonic Visualiser (CANNAM et al., 2006; CANNAM et al., 2010) são exemplos dessas ferramentas.

### 3.5.6.1 Marsyas

O Marsyas (Music Analysis, Retrieval and Synthesis for Audio Signals) é um framework de software não só para prototipagem rápida e experimentação com a análise de áudio, mas também para síntese com ênfase específica em sinais de música e Music Information Retrieval (MIR).

Ele fornece uma arquitetura geral, extensível e flexível, que permite a experimentação com algoritmos e oferece um desempenho rápido que é útil no desenvolvimento de ferramentas de análise em tempo real de áudio. Uma variedade de blocos de construção existentes, que formam a base da maioria dos algoritmos publicados em computação auditiva, já estão disponíveis como parte do pacote, além de possibilitar pacotes adicionais.

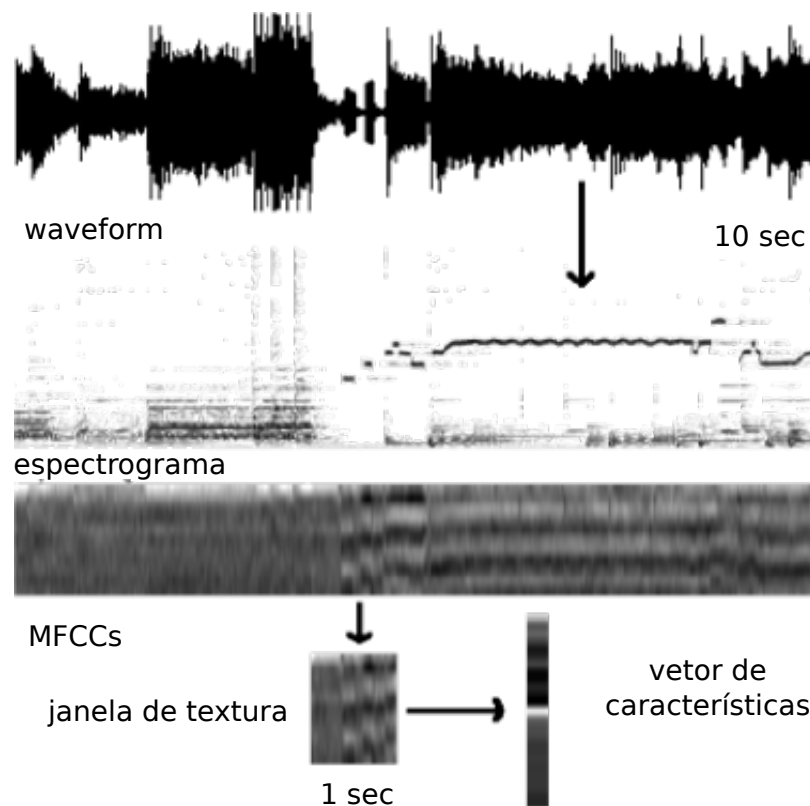


Figura 20: Marsyas extração de MFCC.

A ferramenta é desenvolvida por George Tzanetakis e é distribuída sob licença GNU

General Public License (GPL) havendo também licenças comerciais (TZANETAKIS; COOK, 2000).

A Figura 20 representa o processo de extração de MFCCs pelo Marsyas.

### 3.5.6.2 Sonic Visualiser

O Sonic Visualiser é definido por Cannam et al. (2010) como um aplicativo para a extração automática, navegação e edição de descritores de áudio projetado para auxiliar o estudo e compreensão do conteúdo de dados de áudio com particular ênfase para as gravações musicais.

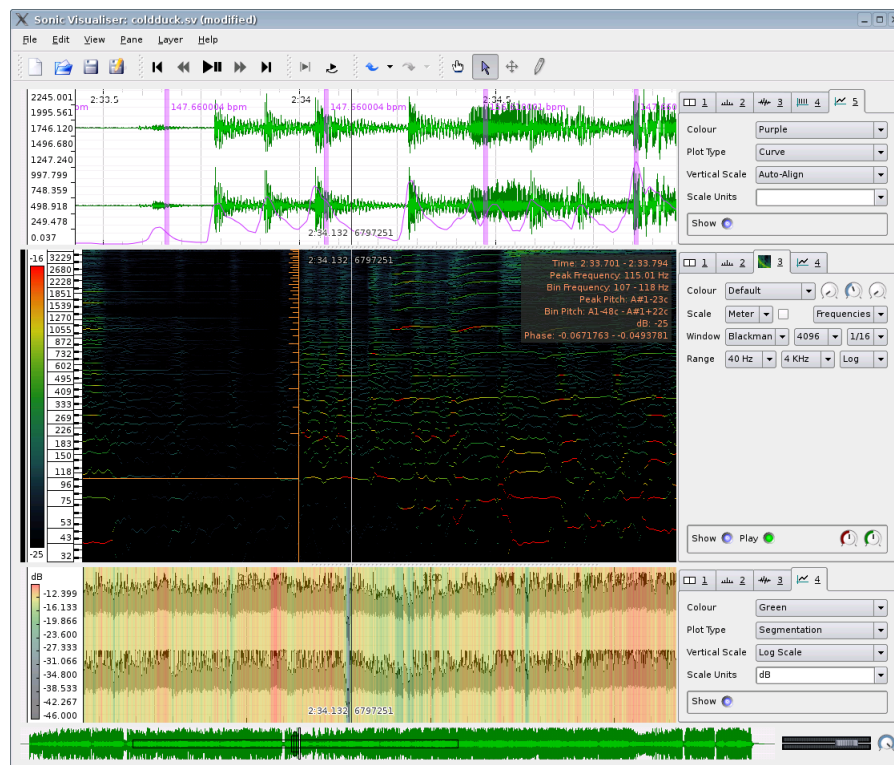


Figura 21: Sonic Visualiser.

Seu objetivo declarado é de ser "o primeiro programa que você recorre quando quer estudar uma gravação musical em vez de simplesmente ouvi-la". Para isso, ele tem uma interface que se assemelha a aplicativos familiares de edição de áudio, um conjunto de instalações de visualização úteis padrão e suporte para um formato de plug-in para os métodos de análise automatizados adicionais.

O Sonic Visualiser (Figura 21) é um Software Livre distribuído sob a GNU General Public License (v2 ou posterior) e está disponível para Linux, Mac OS / X e Windows.

A ferramenta foi desenvolvida no Centro Música Digital no Queen Mary, Universidade

de Londres.

## 3.6 Busca

Busca é definida por (LINDEN, 2008, p. 51) como o problema básico da computação. Todo problema pode ser descrito como a tentativa de alcançar um determinado objetivo, isto é, de chegar a um determinado estado em que uma certa condição é satisfeita. Essa condição é chamada de estado objetivo do problema.

Estados são representações de situações possíveis para o problema assumir. Existe um estado inicial (como começa), estados intermediários e estado final. Frequentemente, é desejável que a sequência de estados assumidos por um problema até seu objetivo seja o menor possível e que não consuma muitos recursos.

Uma busca pode ser formalmente representada pelo conjunto  $\{E, A, I, O\}$ , onde (LINDEN, 2008, p. 53):

- $E$  representa os estados que o problema pode assumir (conjunto de estados);
- $A$  representa as ações que fazem mudar de estado (função sucessor);
- $I$  representa o estado inicial do problema (como começa);
- $O$  representa o(s) estado(s) objetivo(s) do problema.

Busca utiliza uma série de técnicas para solucionar grandes problemas de otimização combinatória, que é simplesmente um problema que pode ser expresso em termos de encontrar o melhor conjunto de valores para um grupo de variáveis (COPPIN, 2012).

Existem algoritmos para solução de problemas de busca que diante de um problema fornecido, retornam uma solução em forma de sequência de ações para se alcançar ao estado objetivo do problema (RUSSELL; NORVIG, 2004, p. 62). Exemplos desses algoritmos são: busca em profundidade, busca em largura, subida da colina, custo uniforme e busca gulosa.

### 3.6.1 Busca em Profundidade

Busca em profundidade é assim chamada por seguir cada caminho até a sua maior profundidade antes de seguir para um outro caminho. Esse princípio é ilustrado na Figura 22.

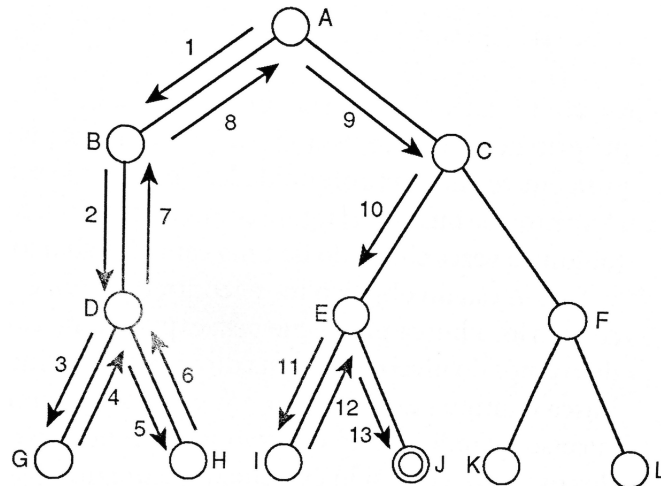


Figura 22: Dinâmica de uma busca em profundidade (COPPIN, 2012).

Tomando como base a árvore de busca disposta na Figura 22, supondo o início do lado esquerdo para o direito, a busca em profundidade envolve descer pelo caminho mais à esquerda na árvore até achar uma folha. Se este for um estado objetivo, a busca foi concluída e será relatado sucesso (COPPIN, 2012, p. 66).

Se a folha não representar um estado objetivo, a busca retrocederá ao primeiro nó anterior e que tenha um caminho ainda não explorado. No caso da Figura 22, após explorar G, voltaria ao ponto D, que ainda há o ponto H não explorado.

Essa dinâmica é repetida até localizar o estado objetivo. É um tipo de busca **exaustiva**, que explora todos os estados ordenados em sequência.

### 3.6.2 Subida da colina

Segundo Coppin (2012, p. 87) subida da colina é um exemplo de um método de busca **informado**, pois ele utiliza informação sobre o espaço de busca de uma maneira razoavelmente eficiente. Se tentar escalar uma montanha, em dia de neblina, com um altímetro, mas sem mapa, o escalador utilizaria uma abordagem de subida da colina:

- Verifique a altura a alguns centímetros da sua posição corrente em cada direção: norte, sul, oeste e leste;
- Tão logo encontre uma posição que tenha altura maior que a altura corrente, vá para lá e recomece o algoritmo;
- Se todas as direções levam para lugares mais baixos que a corrente, então pare e assuma que chegou ao topo.

Nem sempre o teste de parada desse método é válido. Ele pode ser induzido ao erro em espaços que tenham máximos locais. Para espaços assim, existem outros tipos de busca melhor indicados, como *simulated annealing* (seção 3.6.3.1).

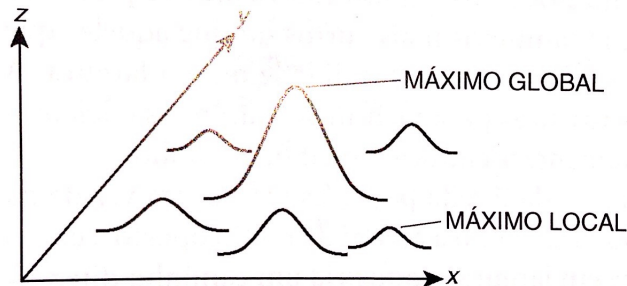


Figura 23: Espaço com máximos locais (COPPIN, 2012).

### 3.6.3 Problemas de Otimização Combinatória

Um problema de otimização combinatória é simplesmente um problema que pode ser expresso em termos de encontrar o melhor conjunto de valores para um grupo de variáveis discretas (COPPIN, 2012).

A tarefa de encontrar a melhor combinação de parâmetros para um algoritmo de aprendizado de máquina dado um problema, por exemplo, é uma tarefa de otimização. A alocação de professores a turmas, escalonar máquinas e funcionários em fábricas, selecionar melhores rotas para ônibus, e outros veículos também são problemas de otimização combinatória (COPPIN, 2012).

Alguns métodos de busca são utilizados frequentemente para solução desses problemas. São métodos também conhecidos como **otimização local**, pois eles tentam otimizar um conjunto de valores, mas encontram frequentemente máximos locais em vez dos máximos globais. Eles partem de alguma configuração inicial (geralmente aleatória), fazendo pequenas alterações até atingir um estado "otimizado" (COPPIN, 2012).

O simulated annealing e o subida da colina fazem parte desse grupo de métodos.

#### 3.6.3.1 Simulated Annealing

Em metalurgia, annealing é o processo usado para temperar ou endurecer metais e vidros aquecendo-os a alta temperatura e depois esfriando-os gradualmente, permitindo assim que o material seja misturado em um estado cristalino de baixa energia (RUSSELL; NORVIG, 2004, p. 114).

Simulated Annealing é uma meta-heurística que é aplicada a problemas combinatoriais multi-valorado, nos quais é necessário escolher valores para muitas variáveis a fim de produzir um valor específico para alguma função global, dependente de todas as variáveis do sistema (COPPIN, 2012, p. 112).

Dado um estado de busca, o Simulated Annealing realiza uma pequena alteração nesse espaço. Caso obtenha uma melhor energia no sistema como um todo, esse novo estado é aceito. Caso contrário é aplicado o critério de aceitação de Boltzman, calculado da seguinte forma:

$$e^{(-dE/T)},$$

onde  $T$  é a temperatura corrente do sistema e  $dE$  é o ganho de energia do estado anterior para o novo.

A partir desse valor é gerado um número aleatório  $x$  entre 0 e 1. Caso  $x < e^{(-dE/T)}$ , então o novo estado é aceito, caso contrário um novo estado é selecionado e novo teste realizado.

Note que quanto maior  $T$ , maior a possibilidade de aceitação do novo estado avaliado. E quanto mais  $T$  for próximo de 0 a probabilidade de aceitar um estado "pior" é menor. Esse movimento estocástico permite a esse método de busca evitar ficar preso em máximos e mínimos locais, a depender do problema.

Com as iterações, um plano de resfriamento é aplicado, o que determina o modo pelo qual a temperatura será reduzida. Dois populares planos de resfriamento são (COPPIN, 2012, p. 113):

- $T_{\text{nova}} = T_{\text{antiga}} - dT$ ;
- $T_{\text{nova}} = C \times T_{\text{antiga}}$ , com  $C < 1$ .

Assim como o plano de resfriamento, a escolha da temperatura inicial do sistema é muito importante, e deve ser cuidadosamente escolhidos de acordo com a natureza do problema a ser solucionado (COPPIN, 2012, p. 114).

Quando a temperatura,  $T$  atinge zero, o sistema congela e, dependendo do sucesso do sistema, será identificado o estado de melhor energia para o problema levantado.

### 3.6.3.2 Sequential Model-based Algorithm Configuration (SMAC)

SMACs são modelos desenvolvidos por Hutter et al. (2011) baseados em florestas aleatórias (BREIMAN, 2001a) que recentemente vêm sendo aplicados à seleção de parâmetros para algoritmos de classificação.

Esses modelos geram árvores aleatórias nas quais cada nó corresponde a uma combinação de parâmetros aceitável pelo problema. Uma das vantagens desse sistema é que ele dá suporte a variados tamanhos de vetores de parâmetros, verificando em cada análise se os parâmetros estão "ativos" para cada caso analisado, concentrando esforços apenas nos que estiverem ativos.

O modelo funciona em busca de combinações de parâmetros que apresente progressiva melhoria de desempenho. Para um estado substituir o atual, ele deve ter desempenho melhor do que o estado atual.

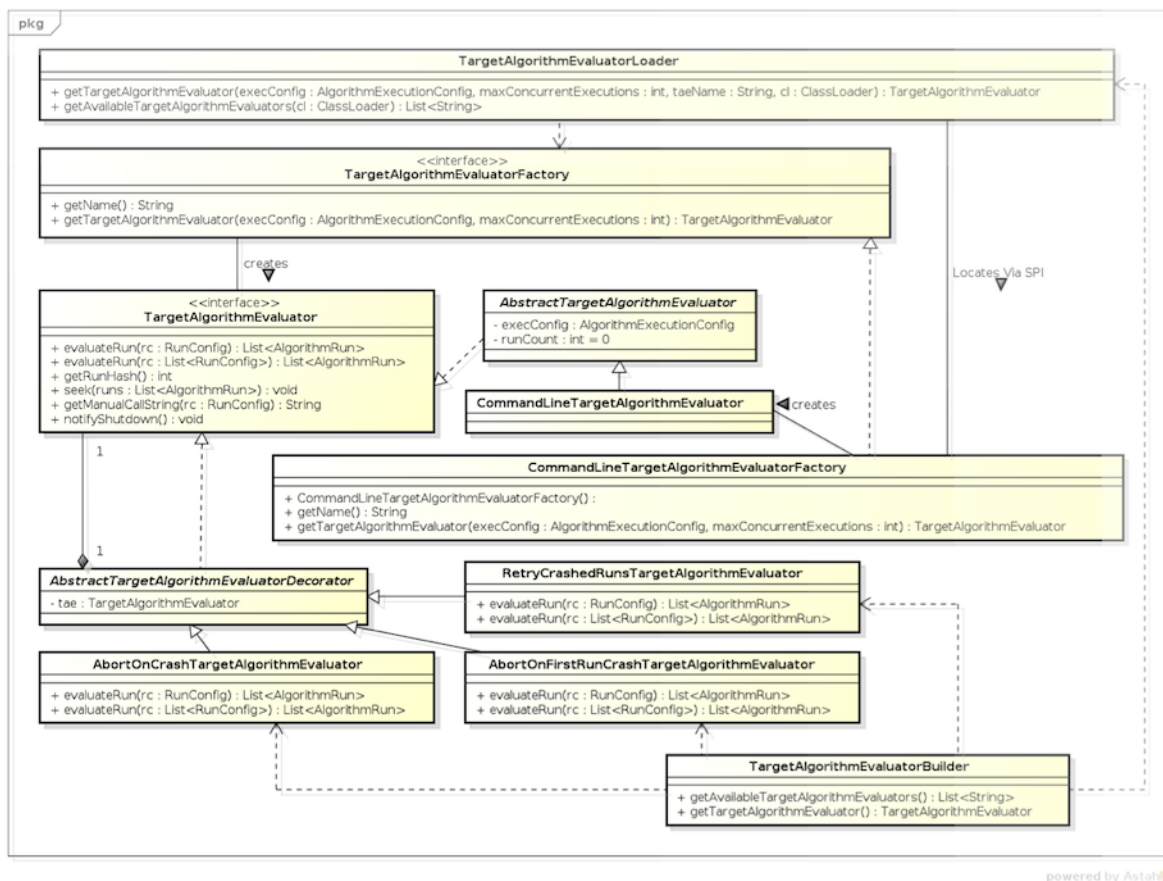


Figura 24: SMAC software.

Uma descrição mais detalhada do SMAC pode ser encontrada no Apêndice A, no qual consta uma transcrição parcial do autor do método.



## 4 Proposta de Solução

A proposta deste trabalho é reunir alguns dos argumentos e conhecimentos citados no capítulo 3 com o propósito de melhorar o desempenho dos classificadores automáticos de áudio.

Propostas de melhorias para esses classificadores poderiam surgir de diferentes formas, como: a criação de um novo algoritmo de aprendizado de máquina, o refino de soluções existentes, a composição de modelos para gerar sistemas híbridos, etc. No entanto, este trabalho prezou por uma solução que fosse facilmente integrada aos procedimentos já realizados pelos desenvolvedores.

Como visto e ilustrado na seção 3.2, o processo de criação de um classificador automático tem basicamente três fases: a extração de características, a seleção de características representativas e a execução de um algoritmo de aprendizado de máquina que induz um modelo classificador.

Em especial, quando se tratam de classificadores para problemas de identificação de áudio, existem ferramentas que podem tanto auxiliar na primeira etapa, como o Marsyas e o Sonic Visualiser (seção 3.5.6) quanto na segunda fase como o Extractor Discovery System (seção 3.3.2.2) que também dá suporte à primeira fase.

No entanto, quando chega a terceira e última fase, momento em que é selecionado um algoritmo de aprendizado de máquina e seus parâmetros para gerar o classificador, faltam regras ou dicas que auxiliem nessa tarefa (FACELI et al., 2011).

Mesmo diante de inúmeras possibilidades de combinações de algoritmos de aprendizado existentes e seus parâmetros, bem como da inexistência de um algoritmo que apresente o melhor desempenho para todos os problemas, a seleção desses algoritmos basicamente se baseiam na popularidade ou disponibilidade do método (quando se tem um usuário inexperiente), ou experiências passadas do usuário. (MONARD; BARANAUSKAS, 2003; FACELI et al., 2011; THORNTON et al., 2012)

Ferramentas, como Weka, RapidMiner, PyBrain e Torch, todos descritos na seção

3.5, são conhecidas pela comunidade que trabalha problemas de classificação de áudio e são indicadas pela International Music Information Retrieval Systems Evaluation Laboratory (IMIRSEL) (IMIRSEL, 2014a). Inclusive foram utilizadas para desenvolvimento do Áudio Alerta. Nessas ferramentas geralmente ocorre que os desenvolvedores inserem uma base de treinamento e alternam **manualmente** vários tipos de algoritmos de aprendizado e parâmetros até encontrarem uma combinação que retorne um classificador o qual apresente um resultado satisfatório, adotando-o como solução para suas necessidades.

A escolha do algoritmo de classificação tem impacto direto no desempenho do classificador gerado. E é ainda maior quando o problema envolve classes que tenham características semelhantes, como o disparo de arma de fogo e fogos de artifício, ou quando envolvem grandes quantidades de classes, situações em que é difícil obter classificadores com altas taxas de acertos.

É importante destacar que, como frequentemente os algoritmos de aprendizado de máquina existentes aceitam valores pertencentes à  $\mathbb{R}$  em algum de seus parâmetros, o espaço formado por todas as combinações possíveis por um só algoritmo desse tipo já é ilimitado. O que inviabiliza a tentativa de testar **todas** essas combinações em um problema dado para se encontrar a combinação que apresente o melhor desempenho.

Dessa forma, o que está proposto neste trabalho é que seja inserido, ao processo básico de desenvolvimento de um classificador, meios que auxiliem na escolha do algoritmo de aprendizado de máquina e de seus respectivos parâmetros.

Descartada a possibilidade de análise uma a uma de todas as soluções possíveis, este trabalho sugere a inclusão de uma busca inteligente ao processo seletivo de algoritmos de classificação e seus respectivos parâmetros. Esses sistemas não necessariamente analisam todas as soluções de um espaço de busca, e são adotados para a solução de problemas semelhantes na literatura. Seria a **automatização** de um processo que atualmente, quando acontece, geralmente é realizado de forma **manual**.

Além disso, ao final de seus processos, buscas retornam sequências de soluções que foram visitadas. Isso possibilita que, em vez de recomendar apenas uma solução (algoritmo, parametrização), seja possível ordenar uma lista das soluções mais promissoras entre as que foram analisadas. Com isso, evita-se o risco de indicar apenas uma sugestão que eventualmente não seja de possível implementação.

No decorrer do texto, essa proposta será referenciada como **SAC (Search for Audio Classification)** (Figura 25).

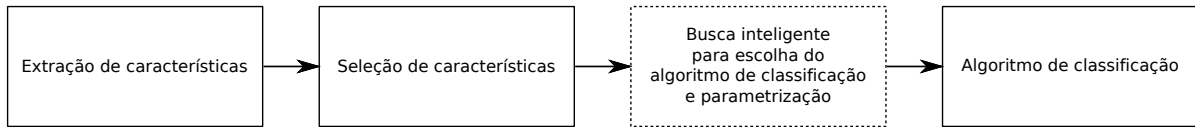


Figura 25: Proposta de solução.

É possível encontrar na literatura registros de tentativas de automatizar a definição dos parâmetros de um algoritmo específico de aprendizado, como Hutter et al. (2011) e outros relacionados à meta-aprendizado (seção 3.4) para seleção de modelos, como Brazdil et al. (2003), Giraud-Carrier et al. (2004), Brazdil et al. (2009), assim como outros também já citados neste trabalho.

Entretanto, o que se percebe é que, excluindo-se Thornton et al. (2013) que em paralelo vem tentando trilhar caminho parecido, mas para problemas mais gerais e em forma de uma ferramenta, não foram encontrados durante os esforços de realização deste trabalho, estudos que combinassem essas necessidades em forma de busca para a solução de problemas, em especial para aqueles que envolvessem sinais de áudio.

Para adotar um sistema de busca para essa aplicação, um dos fatores que se tem que definir é a forma de combinar em uma só busca a definição do algoritmo de classificação e seus parâmetros. E para isso, neste trabalho, o algoritmo de classificação também será considerado um parâmetro de uma solução possível. Dessa forma, o problema é reduzido a apenas otimização de parâmetros, possibilitando o aproveitamento de trabalhos já existentes.

Neste trabalho foi adotada a técnica Sequential Model-based Algorithm Configuration (SMAC) (HUTTER et al., 2011), descrita na seção 3.6.3.2 e Apêndice A, por avaliar que já é disponibilizado o código para otimização de parâmetros com essa técnica. Mas poderia ter sido adotado qualquer tipo de busca. Não sendo recomendadas buscas que tentem explorar todos os estados possíveis ou buscas cegas, tendo em vista a alta dimensionalidade do espaço de soluções.

Alguns experimentos foram realizados para averiguar o comportamento do proposto **SAC** diante de problemas de classificação de áudio. Os procedimentos, resultados e análises são detalhados no próximo capítulo.

## 4.1 Preparação da Busca

O sistema de execução do algoritmo SAC aceita dois tipos de parâmetros: numérico e categórico. Para um parâmetro numérico  $x$  são indicados um valor mínimo  $a$ , um máximo  $b$ , um valor padrão (opcional) e o conjunto do qual aceita valores, se  $\forall x \in \mathbb{R} : a \leq x \leq b$  ou  $\forall x \in \mathbb{Z} : a \leq x \leq b$ . No caso de um parâmetro categórico  $y$  são aceitos o conjunto de valores do qual  $y \in \{\langle \text{valor1} \rangle, \dots, \langle \text{valorn} \rangle\}$  e um valor padrão (opcional).

Além disso, é configurado um arquivo chamado *Conditional Parameter Clauses* que informa em quais situações cada parâmetro é ativado. Para os experimentos, nesse arquivo foram informadas as relações que indicavam quais parâmetros estariam ativos para cada algoritmo de aprendizado de máquina. Assim como as conexões com os algoritmos de aprendizado de máquina propriamente dito para estimativa de desempenho.

Os resultados são obtidos através de um arquivo *log* de saída constando a trajetória percorrida até o término do processo de busca e respectivas taxas de erro médio estimadas.

## 4.2 Requisitos da Proposta

É possível fazer uma análise dos principais requisitos citados na seção 2.2 em relação ao SAC. Alguns desses fatores estão implícitos na própria estrutura da solução, como a disponibilidade de código e a usabilidade.

Em relação à disponibilidade de código, o desejo é que o método SAC esteja disponível à maior quantidade de pessoas possíveis. Para isso, todos os recursos utilizados ficarão disponíveis em repositórios públicos para livre utilização e aperfeiçoamento.

Como dito antes, essa é uma área que abrange profissionais de várias áreas do conhecimento, como artes, medicina, computação, engenharia, estatística, biologia, etc. Baseado nisso, uma preocupação durante o desenvolvimento da proposta é que ela fosse algo que pessoas de todas essas áreas conseguissem usá-la sem necessidade de conhecimentos aprofundados em aprendizado de máquina e que fosse facilmente reutilizada em outros problemas de classificação de sons. Assim, a proposta foi elaborada de forma a ser integrada a ferramentas já utilizadas por essas pessoas como o aproveitamento da coletânea de algoritmos de aprendizado de máquina disponível em ferramentas como as citada na seção 3.5.

Já a corretude da solução será aferida através dos experimentos e serão destacadas nos seus respectivos resultados, levando em consideração a estimativa de erro de cada

classificador obtido.

Da mesma forma a economicidade da proposta será debatida em uma seção específica (5.4.1) deste trabalho.

## 5 Avaliação

A proposta de solução para o problema levantada no capítulo anterior foi testada em experimentos que envolvessem problemas reais de classificação automática de áudio. A descrição, análise e discussão desses experimentos estão detalhados neste capítulo.

### 5.1 Metodologia

A hipótese a ser avaliada afirma que é possível melhorar o desempenho de classificadores automáticos de áudio por meio da realização de uma busca automática para seleção do algoritmo de classificação e dos respectivos parâmetros a serem utilizados para desenvolver esses classificadores.

Para isso, foram realizados experimentos controlados com dois problemas reais distintos: o reconhecimento automático de disparos de armas de fogo e a classificação de emoções contidas nas músicas. No decorrer do texto, esses ensaios serão tratados como experimento 1 e experimento 2, respectivamente.

A escolha desses problemas foi influenciada por três fatores: ter envolvimento com uma situação real; ter disponíveis as bases de dados relacionadas; e já existir classificadores para o problema, de forma que a definição do algoritmo adotado tenha sido por meio de seleção manual (exploratória), sem um procedimento automático (sistemático) como proposto na hipótese deste trabalho.

Não se espera que com apenas esses dois experimentos se possa provar que haverá melhoria para todo problema de classificação de áudio. Espera-se ter apenas um **indício** da efetividade da solução SAC.

Além do Áudio Alerta, foram buscadas outras bases de dados de domínios diferentes, com dificuldades de classificação diferente, de forma a realizar testes mais abrangentes da hipótese, como reconhecimento de voz, e desafios lançados no Music Information Retrieval Evaluation eXchange (MIREX) (IMIRSEL, 2014b), evento anual da comunidade internaci-

onal de avaliação de sistemas de recuperação de informação musical. No entanto, questões de tempo limitaram a apenas mais um experimento. Mas as ferramentas e métodos ficam disponíveis para que pesquisadores possam realizar estudos e inferências complementares em <https://code.google.com/p/sac-thesis-2014/>.

Para avaliação da capacidade de generalização dos classificadores gerados foi utilizada a técnica Validação Cruzada (*r-fold cross-validation*) com  $r=10$ . Segundo Kohavi (1995b), esse é o melhor método de amostragem de dados a ser usado para a seleção de um modelo.

Kohavi (1995b) chega a essa conclusão após revisar métodos de estimativa de precisão e realizar comparações entre dois mais comuns na literatura: cross-validation e bootstrap. Para isso foram realizados mais de meio milhão de experimentos comparativos gerando classificadores com árvores de decisão e redes bayesianas. Mesmo em situações em que o poder computacional viabilizava a utilização em mais folds (dobras) amostragens em 10-folds apresentaram melhores resultados.

O método de validação cruzada (Figura 26) funciona da seguinte forma (FACELI et al., 2011):

1. um conjunto de exemplos é dividido em  $r$  subconjuntos de tamanho aproximadamente igual;
2. os objetos de  $r - 1$  partições são utilizados no treinamento de um algoritmo de classificação;
3. o classificador gerado é testado sobre os elementos da partição que não foi utilizada no treinamento;
4. os processos 2 e 3 são repetidos  $r$  vezes, utilizando em cada ciclo uma partição diferente para teste no passo 3;
5. O desempenho final do classificador é dado pela média do desempenho observado para cada subconjunto de teste.

Esse método foi o mesmo utilizado para desenvolvimento dos classificadores que já existiam para os problema de classificação abordados nos experimentos 1 e 2. Como a intenção nos experimentos é criar novos classificadores por meio dos mesmos procedimentos da elaboração dos anteriores, alterando-se apenas a etapa de seleção do algoritmo classificador e seus parâmetros para uma análise comparativa dos desempenhos, foi decidido manter também o tipo de amostragem.

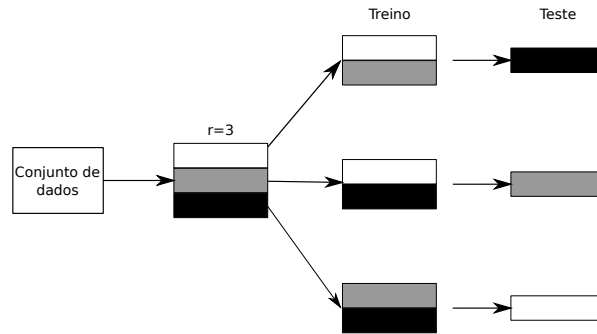


Figura 26: Validação cruzada.

As buscas foram realizadas em um espaço formado por 32 subespaços. Para cada algoritmo de aprendizado (os principais estão listados na Tabela 2) foi formado um subespaço com todas combinações possíveis dele com seus parâmetros. Como muitos desses algoritmos adotam em ao menos um parâmetro valores pertencente à  $\mathbb{R}$ , **esses espaços formaram um espaço ilimitado de soluções.**

Tabela 2: Principais métodos que compõem o espaço de busca dos experimentos.

Algoritmo
Árvore de Decisão - C.45 (QUINLAN, 1993)
PART Decision List (FRANK; WITTEN, 1998)
Simple Decision Table Majority (KOHAVI, 1995a)
$k$ Vizinhos mais próximos ( $k$ NN) (AHA et al., 1991)
Regressão Localmente Ponderada (LWL) (ATKESON et al., 1996)
Multinomial Logistic Regression (CESSIE; HOUWELINGEN, 1992)
Redes Neurais Artificiais - Multilayer Perceptron
Voted Perceptron (FREUND; SCHAPIRE, 1998)
Redes Baysianas/Naive Bayes (JOHN; LANGLEY, 1995)
Redes Baysianas/Naive Bayes Multinomial (MCCALLUM; NIGAM, 1998)
Floresta de Decisão - Random Subspace Method (HO, 1998)
K* (CLEARY; TRIGG, 1995)
Florestas de Decisão construídas com Random Subspace Method (HO, 1998)
Florestas Aleatórias (BREIMAN, 2001b)
Máquina de vetores de suporte (SVM) com Sequential Minimal Optimization (PLATT, 1998)
Logistic Model Trees (LMT) (LANDWEHR et al., 2005b)
Repeated Incremental Pruning to Produce Error Reduction (RIPPER) (COHEN, 1995)
1R (HOLTE, 1993)
Adaboost M1 (FREUND; SCHAPIRE, 1996)
Bagging (BREIMAN, 1996)
Regression Methods (FRANK et al., 1998)
Stacking (WOLPERT, 1992)
Vote (KITTLER et al., 1998)

A corretude de cada solução é mensurada por meio da taxa de classificações incorretas apresentada pelo classificador gerado  $f_S$  em uma base de dados com  $n$  instâncias, métrica



usualmente adotada para esse tipo de avaliação, como visto na seção 3.1.2:

$$erro(f_S) = \frac{1}{n} \sum_{i=1}^n I(y_i \neq f_S(\mathbf{x}_i)).$$

Quanto menor o erro, melhor será considerada a solução formada pelo vetor ⟨algoritmo classificador, sequência de parâmetros⟩.

Após iniciado, os sistemas de buscas têm critérios que indicam o momento de interromper sua execução. Em métodos definidos como *ótimos*, aqueles que são capazes de sempre retornar a melhor solução de todas possíveis, o critério de parada é apenas um: quando identificada essa solução ótima.

Diante das inúmeras possibilidades de soluções para o espaço de busca formado por algoritmos de classificação e seus parâmetros, seria inviável aguardar o sistema de busca analisar todas as soluções possíveis e indicar a solução ótima para qualquer problema de classificação de áudio.

No entanto um estado ideal, caracterizado por uma solução que apresente uma estimativa de  $erro(f_S) = 0$ , eventualmente pode ser localizado pelo processo de busca para um problema específico, sem a necessidade de análise de todas as soluções no espaço de busca. "Eventualmente" porque esse estado ideal ainda não deve existir para muitos problemas de classificação, considerando o atual estado da arte. Mas por ser possível, compõe o primeiro critério de parada do processo de busca proposto neste trabalho.

Como o primeiro critério de busca nem sempre é atendido, foi necessário estabelecer um segundo critério que leva em consideração o tempo de processamento. Esse valor é informado pelo usuário. Em particular, para este trabalho foi estabelecido inicialmente 12 horas de processamento contínuo, em alguns casos, esse período foi ampliado. Pela ausência de relatos na literatura, esse tempo máximo foi estimado levando em consideração apenas a disponibilidade de tempo para realização dos experimentos pelo autor, carecendo de uma investigação maior.

## 5.2 Experimento 1: Áudio Alerta

O Áudio Alerta é uma solução desenvolvida pela Daccord Music Software S.A. que integra monitoramento de áudio aos sistemas de vídeo segurança com o objetivo de monitorar áreas urbanas. Uma das suas principais funções é, ao detectar um disparo de arma de fogo dentro de seu raio de cobertura, enviar um aviso à central de monitoramento ao

mesmo tempo que rotaciona a câmera local para a direção do ocorrido.

Esse sistema envolve a aplicação de uma série de conhecimentos: geolocalização, transmissão de dados, captura de som e de imagem, codificação, tratamento de imagem, compressão, armazenamento, etc. Mas uma dessas funções se destaca quando levado em consideração a abordagem deste trabalho: a capacidade de, diante de um som de alto impacto, classificá-lo como sendo som ou não de tiro de arma de fogo.

É sobre esse típico problema de classificação automática de áudio que foi realizado o primeiro experimento deste trabalho. Mais que isso, o Áudio Alerta foi um grande motivador durante toda essa investigação.

### 5.2.1 Descrição da base de dados

A classificação automática de disparos de armas de fogo é abordada na literatura por Valenzise et al. (2007) e Gerosa et al. (2007), que utilizam bases de dados compostas por coeficientes MFCC para gerar classificadores capazes de diferenciar um tiro de outro som qualquer.

O Áudio Alerta também assume valores de coeficientes MFCC como atributos. Uma base nesse formato foi disponibilizada para a realização dos experimentos deste trabalho.

Essa base é composta por exemplos de áudios comuns aos ambientes urbanos, como fogos, buzinas, frenagens, alarmes, latidos, chuva, palmas, sirenes, aviões, helicópteros, espirros, tiros de diferentes calibres e munições, etc. Cada instância é descrita em 42 atributos e uma classe.



Figura 27: Experimento 1 – tipos de sons que compõem a base de dados.

## 5.2.2 Resultados Obtidos

Este experimento foi subdividido em três problemas com diferentes números de classes: 2, 4 e 49. Considerada a mesma base de dados, quanto maior o número de classes, mais específica e mais complexa é a classificação. Por exemplo, no de duas classes só foi considerado se o som analisado era tiro de arma de fogo ou não. Já no experimento com 49 classes tinha que ser identificado também o calibre e a munição utilizada para provocar aquele som.

### 5.2.2.1 Problema com duas classes

Para a primeira parte deste experimento, foram selecionadas 4318 instâncias de um total de 17066 de forma que a amostra ficasse balanceada com mesma quantidade de ocorrências negativas (0) e positivas (1) de disparos de arma de fogo.

Neste primeiro experimento, a busca realizada expandiu quinze possíveis soluções entre três algoritmos de aprendizado de máquina diferentes. Dessas, as cinco que apresentaram menor taxa de erro estão mostradas na Tabela 3.

Tabela 3: Experimento 1.1 – resultados obtidos com o SAC para o Áudio Alerta com 2 classes.

Quantidade de Classes	Algoritmo Sugerido	Taxa de Erro (%)
2	Regressão Localmente Ponderada (LWL)	2.98
2	Regressão Localmente Ponderada (LWL)	3.06
2	Regressão Localmente Ponderada (LWL)	3.17
2	Logistic Model Trees (LMT)	4.31
2	Logistic Model Trees (LMT)	4.32

Verifica-se que no ranking final houve a repetição de algoritmos de aprendizado de máquina. Mas com taxas de erro diferentes.

Essa diferença entre resultados para os mesmos algoritmos se dá pela variação dos parâmetros desses métodos, ocasionando a criação de classificadores diferentes e que, neste caso, apresentam rendimentos distintos.

Como existem algoritmos que utilizam algum tipo de operação estocástica em seus processos internos, ainda que em situações as quais se repita um mesmo algoritmo de classificação, com os mesmos parâmetros e mesmos exemplos de dados, é possível que se obtenham classificadores diferentes (FACELI et al., 2011).

Considerando o tipo de problema investigado neste experimento, é importante fazer

uma análise sobre a taxa de falso-negativos apresentada nos resultados. Para o Áudio Alerta, uma ocorrência de falso-negativo é registrada quando um som verdadeiramente de tiro (1) é classificada como negativa (0) pelo classificador.

Para um sistema de vigilância não é interessante a ocorrência de falsos-negativos. Nesse caso específico, é menos custoso a ocorrência de um falso-positivo, um alarme falso. Por exemplo, um tiro não detectado pode deixar passar a oportunidade de identificar o autor de um crime. Por outro lado, a detecção de um som falso-positivo pode ser rapidamente verificado por um agente na central de monitoramento e ignorado.

Neste experimento, o resultado mais promissor apresentou uma taxa de falso-negativo de 1,25%.

Tabela 4: Experimento 1.1 – matriz de confusão.

	Positivo	Negativo
Positivo	2132	102
Negativo	27	2057

A parametrização da Regressão Localmente Ponderada que apresentou menor erro é formada por uma função logística aplicada localmente aos 60 vizinhos mais próximos, calculados pela distância euclidiana, e com atribuição de pesos na forma tri-cube.

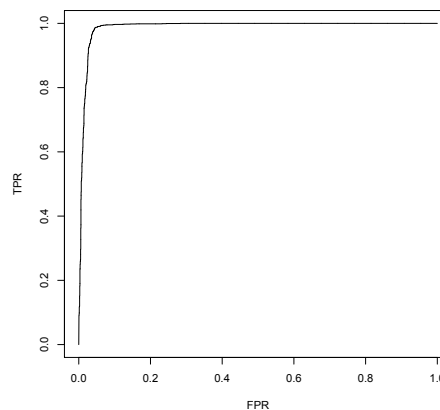


Figura 28: Experimento 1 – curva ROC.

### 5.2.2.2 Problema com quatro classes

Em uma segunda parte foram utilizadas quatro classes considerando as 17066 instâncias da base de dados:

1. positivo: pequeno calibre (revólveres);

2. positivo: calibre intermediário (pistolas);
3. positivo: grosso calibre (espingardas, fuzis, metralhadoras e submetralhadoras);
4. negativo (frenagens, fogos, etc).

Com essa distribuição de classes o SAC indicou como mais promissor o algoritmo de Regressão Localmente Ponderada (LWL) novamente. Porém, com uma sequência de parâmetros diferente da proposta no experimento com apenas duas classes (seção 5.2.2.1).

Dessa vez, a parametrização é composta por uma rede neural artificial do tipo multilayer perceptron, com uma camada oculta, e taxa de aprendizado 0.98 aplicada aos 30 vizinhos mais próximos.

Com essa configuração a taxa registrada de classificações erradas foi de 3.95%.

Tabela 5: Experimento 1.2: resultados obtidos com o SAC para o Áudio Alerta com 4 classes.

Quantidade de Classes	Algoritmo Sugerido	Taxa de Erro (%)
4	Regressão Localmente Ponderada (LWL)	3.95

### 5.2.2.3 Problema com 49 classes

Em um terceiro momento foram utilizadas classes ainda mais específicas, 49 classes ilustradas na Figura 27, também utilizando a totalidade de exemplos da base de dados.

Nesse cenário, o SAC apresentou taxa de erro de 34,19% com o algoritmo de aprendizado do tipo Florestas Aleatórias com seis árvores após 12 horas de processamento.

Nesse caso, foi dado continuidade no experimento por mais 140 horas consecutivos, totalizando 152 horas de processamento. Após esse período o SAC apresentou novo resultado com 26,81% de taxa de erro com Regressão Localmente Ponderada (LWL) aplicada um árvore de decisão aos 10 vizinhos mais próximos.

Tabela 6: Experimento 1.3: resultados obtidos com o SAC para o Áudio Alerta com 49 classes.

Processamento (h)	n. de Classes	Algoritmo Sugerido	Taxa de Erro (%)
12	49	Floresta Aleatória	34.19
152	49	Regressão Localmente Ponderada (LWL)	26.81

### 5.2.3 Discussão dos Resultados

O Áudio Alerta é um produto já em comercialização pela Daccord Music S.A. Portanto, existe um classificador rodando nos aparelhos que estão em funcionamento. Para efeito de análise do desempenho do SAC no experimento 1, foi aplicado a mesma base de dados ao algoritmo 3-NN em utilização no sistema comercializado. A Tabela 7 apresenta esses dados comparativos.

Tabela 7: Comparação dos erros: Áudio Alerta original vs SAC.

Quantidade de Classes	Áudio Alerta original(%)	SAC (%)	Redução (%)
2	3.57	2.98	16,53
4	5.10	3.95	22,55
49	25.91	26.81	-3,47

Nota-se que o SAC em sua busca automática limitada por tempo de processamento apresentou resultados superiores em 2/3 dos problemas trabalhados no experimento 1 quando comparado ao sistema original do Áudio Alerta desenvolvido por especialistas da área de classificação automática de sons.

Uma análise da taxa de falso-negativo do problema de duas classes também foi realizada. Nesse aspecto o SAC apresentou uma redução de 25,15% da ocorrência de falso-negativos.

Tabela 8: Matriz de confusão algoritmo original do Áudio Alerta para o problema de duas classes.

	Positivo	Negativo
Positivo	2123	118
Negativo	36	2041

## 5.3 Experimento 2: Classificação de Emoções em Músicas

Os seres humanos, por natureza, são emocionalmente afetados pela música. Como fala a famosa citação do filósofo-alemão Friedrich Nietzsche: "*Sem música, a vida seria um erro*". Como os bancos de dados de música crescem em tamanho e número, a recuperação de música pela emoção está se tornando uma tarefa importante para diversas aplicações, tais como a seleção em dispositivos móveis (TOLOS et al., 2005), sistemas de recomendação de música (CAI et al., 2007), rádio e TV, musicoterapia, etc (TSOUMAKAS et al., 2008).

Nesse sentido, existem na literatura trabalhos que buscam realizar a classificação automática das emoções contidas nas músicas, como Lu et al. (2006), Yang et al. (2006), Yang et al. (2008) que trabalham com uma classe de emoções e Li e Ogihara (2003), Wieczorkowska et al. (2006), Li e Ogihara (2006) e Lu et al. (2006) que fazem uma abordagem por meio de classificação multi-rótulo.

Há também Tsoumakas et al. (2008), que apesar de trabalhar uma classificação simultânea das músicas em várias classes de emoções, também apresenta dados de desempenho para cada classe individual (Tabela 11) e, igualmente, disponibiliza a base de dados utilizada. A mesma empregada neste segundo experimento.

Essa base é composta por 593 instâncias, cada uma com 72 atributos ritmos e de timbre que representam janelas de músicas de 30 segundos após os 30 segundos iniciais da canção. Todos extraídos com auxílio do Marsyas (seção 3.5.6.1).

Os 72 atributos são compostos da seguinte forma:

- 64 equivalem às médias, desvio padrão, média do desvio padrão e o desvio padrão do desvio padrão: de 13 coeficientes MFCCs, 1 centroide espectral, 1 atenuação espectral e 1 fluxo espectral ( $((13 \times MFCC + 3 \times FFT) \times 4 = 64)$ );
- 8 equivalem às seguintes características retiradas dos dois maiores picos selecionados por histograma: amplitude, beats por minuto (BPM), relação high-to-low de suas BPM, além de três recursos calculados pela soma dos bins do histograma entre 40-90, 90-140 e 140-250 BPMs respectivamente.

As classes envolvidas, em livre tradução<sup>1</sup>, baseadas no modelo de Tellegen-Watson-Clark de humor (Figura 29), são: surpresa, felicidade, calma, quietude, tristeza, raiva. Essas classes serão tratadas como: L1, L2, L3, L4, L5 e L6 respectivamente. E seus exemplos na base de dados não formam necessariamente conjuntos disjuntos em relação às classes, pois um exemplo pode pertencer a mais de uma classe.

### 5.3.1 Resultados Obtidos

Neste segundo experimento, os algoritmos de aprendizado mais promissores obtidos por meio do SAC estão registrados na Tabela 10 com suas respectivas taxas de erro.

<sup>1</sup>Termos originais em Tsoumakas et al. (2008): amazed-surprised, happy-pleased, relaxing-calm, quiet-still, sad-lonely, angry-fearful.

Tabela 9: Experimento 2 – classes do experimento.

Classe	Algoritmo Classificador	Exemplos
L1	surpresa	173
L2	felicidade	166
L3	calma	264
L4	quietude	148
L5	tristeza	168
L6	raiva	189

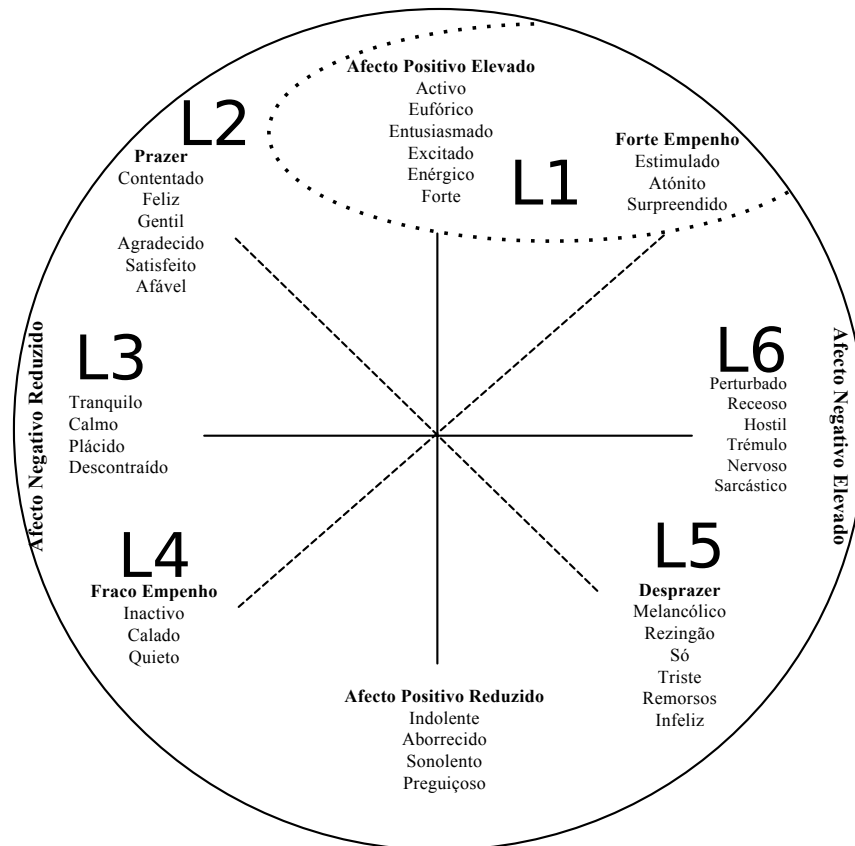


Figura 29: Modelo de Tellegen-Watson-Clark de humor tirado de Galinha e Pais-Ribeiro (2005) com modificações.

### 5.3.2 Discussão dos Resultados

Para avaliar o desempenho do Experimento 2 e fundamentar essa discussão, mais uma vez foram resgatados trabalhos científicos anteriores que utilizaram a mesma base de dados para o problema de classificação dos sentimentos em músicas.

Para isso, tentou-se reproduzir as mesmas condições do trabalho de Tsoumakas et al. (2008) para se possibilitar uma posterior análise comparativa com seus resultados. Como nos resultados desse trabalho original o desempenho dos classificadores foram mensurados por meio da acurácia, foram adaptados também os resultados finais do experimento 2,



Tabela 10: Experimento 2 – SAC: resultados por classe.

Classe	Algoritmo Classificador	Horas de Processamento	Taxa de Erro (%)
L1	Logistic Model Trees (LMT)	12	18.16
L2	14-NN	12	23.76
L3	Florestas Aleatórias	12	17.15
L4	Random Subspace Method	12	7.92
L5	Random Subspace Method	12	14.58
L6	Simple Logistic	209	16.62

sendo registrados na Tabela 12.

Para uma fidelidade maior da comparação de resultados, deveria-se reproduzir também os algoritmos do trabalho original como foi feito no caso do Áudio Alerta (Experimento 1 - seção 5.2). Mas isso não foi possível por ausência de informações necessárias para essa reprodução, como por exemplo, os parâmetros para cada algoritmo de aprendizado de máquina utilizado. Portanto, foram assumidos os resultados registrados pelo autor em seu trabalho para quatro algoritmos de classificação multirótulos diferentes: relevância binária (BR), label powerset (LP), random k-labelsets (RAKEL) e multi-classe  $k$ NN.

Tabela 11: Taxa de erro (%) por classe para cada algoritmo testado por Tsoumakas et al. (2008).

Classe	BR	LP	RAKEL	ML $k$ NN	Média
L1	21.00	20.94	<b>20.18</b>	25.54	21.92
L2	28,85	26.20	<b>24.13</b>	28.05	26.81
L3	22.80	22.95	<b>21.46</b>	27.79	23.75
L4	10.03	10.08	<b>9.69</b>	20.31	12.53
L5	<b>17.13</b>	19.07	17.64	29.49	20.83
L6	<b>16.78</b>	18.58	17.62	25.78	19.69

Tabela 12: Experimento 2 – tabela comparativa das taxas de erro obtidas.

Classe	Resultados na Literatura	SAC	Redução em (%)
L1	20.18	18.16	<b>10.01</b>
L2	24.13	23.76	<b>1.53</b>
L3	21.46	17.15	<b>20.08</b>
L4	9.69	7.92	<b>18.27</b>
L5	17.13	14.58	<b>14.89</b>
L6	16.78	16.62	<b>0.95</b>

O subproblema classe L6 consumiu 209 horas para obter um resultado superior do que o já registrado na literatura. Enquanto os demais em 12 horas já apresentavam resultados significativos.

Verifica-se na Tabela 12 que no cenário definido neste experimento, o desempenho foi registrado superior com os algoritmos recomendados ao ser utilizado o SAC, em comparação com os melhores desempenhos em negrito na Tabela 11. Apresentando uma **redução média de 10,96%** sobre a taxa de erro do problema.

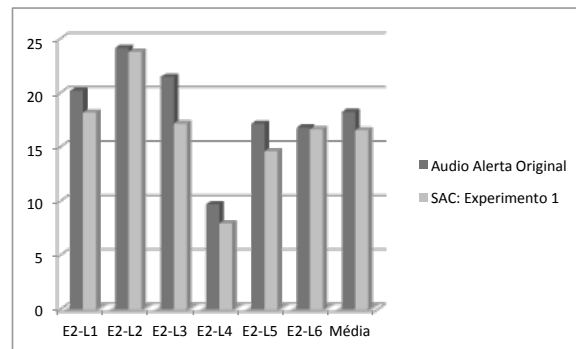


Figura 30: Comparativo das taxas de erro de Tsoumakas et al. (2008) e experimento 2.

## 5.4 Consolidação dos Resultados SAC

A proposta do SAC é melhorar a qualidade dos classificadores automáticos de áudio. Para isso ele foi posto a prova com problemas reais que já haviam soluções desenvolvidas por especialistas, o que aumentou o desafio.

Em seus resultados, o SAC apresentou uma melhoria em 8 de 9 experimentos realizados neste trabalho, mesmo desconhecendo as soluções que já existiam. O que mostra o potencial de sistemas de buscas automáticas quando aplicadas a esses processos.

A Tabela 13 e a Figura 31 mostram as melhorias alcançadas durante os experimentos.

Tabela 13: Resultados finais: melhorias obtidas.

Problema	Taxa de Erro SAC (%)	Redução da Taxa de Erro (%)
Experimento 1 - 2 classes	2.98	16.53
Experimento 1 - 4 classes	3.95	22.55
Experimento 1 - 49 classes	26.81	-3.47
Experimento 2 - emoção: surpresa	18.16	10.01
Experimento 2 - emoção: felicidade	23.76	1.53
Experimento 2 - emoção: calma	17.15	20.08
Experimento 2 - emoção: quietude	7.92	18.27
Experimento 2 - emoção: tristeza	14.58	14.89
Experimento 2 - emoção: raiva	16.62	0.95
<b>Média</b>		<b>11.26</b>

Eram esperados resultados que eventualmente não trouxessem melhorias em comparação a soluções desenvolvidas por **especialistas**, como os obtidos no experimento 2 para a classe L6, sistemas de busca estão passíveis a essas situações. Isso só seria evitado em 100% dos casos se fossem utilizados métodos capazes de sempre encontrar a melhor solução de todas, o que geralmente não ocorre com problemas com a complexidade e dimensionalidade dos investigados neste trabalho.

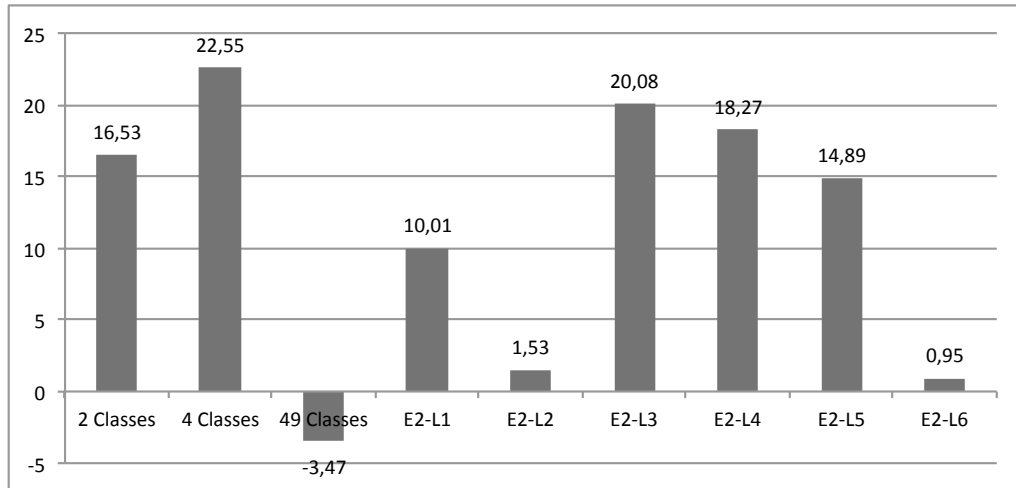


Figura 31: Experimento 2 – redução percentual da taxa de erro alcançada o SAC.

### 5.4.1 Economicidade

Uma discussão importante é sobre a redução de custo financeiro na elaboração de um classificador automático de som ao se aplicar a busca automática do SAC.

A Daccord Music conta em sua equipe com profissionais com alto nível de capacitação, incluindo mestres e doutores na área de Ciências da Computação. A manutenção de uma equipe com essas características geralmente requer uma folha de pagamento elevada.

Segundo informações da equipe de desenvolvimento do Áudio Alerta, na seleção do algoritmo de aprendizado, treinamento e testes com o sistema que hoje é utilizado e que os resultados foram trabalhados neste trabalho, foi necessário o empenho de dois cientistas durante cerca de duas semanas e meia.

Se for contabilizadas seis horas diárias de trabalho, conclui-se que foram necessárias aproximadamente 75 horas de trabalho contínuo. Se ainda for levado em consideração que foram dois analistas, obtém-se 150 horas de especialistas que não são facilmente encontrados no mercado.

Enquanto isso, os resultados dos experimentos deste trabalho levaram, em geral, 12

horas de processamento, e 152 horas no caso que consumiu mais tempo, onde o pesquisador só precisou estar presente nas duas horas iniciais, para configuração do SAC, e duas horas finais para interpretação dos resultados.

## 6 Considerações Finais

A classificação automática de sons está presente de várias formas na sociedade, seja na classificação de músicas (GOLUB, 2000), no reconhecimento de voz (LIPPMANN, 1989), na classificação do choro de crianças (SAHA et al., 2013) ou na classificação de animais (LEE et al., 2008), entre outros.

No entanto, foi mostrado neste trabalho que ainda há espaços para melhorias desses sistemas. Enquanto alguns classificadores específicos obtêm índices satisfatórios de acertos, outros ainda carecem de técnicas e métodos que os façam atingir desempenhos desejáveis, principalmente, quando se tratam de problemas que envolvam classes muito próximas umas das outras ou que envolvam grandes quantidades de classes.

As principais técnicas de desenvolvimento de classificadores estão concentradas na área de Inteligência Artificial, mais precisamente, em Aprendizado de Máquina. Uma revisão dos principais conteúdos relacionados foi realizada neste trabalho como as principais definições e abordagens de algoritmos de classificação.

Os classificadores de áudio são elaborados, geralmente, em três etapas básicas: a extração de características dos sons a serem classificados; a seleção de características que melhor representem as classes envolvidas; a definição de algoritmo de classificação para induzir um classificador.

Existem ferramentas que são utilizadas pela comunidade que trabalha com esse tipo de problema. Porém, nota-se a ausência de ferramentas ou orientações que auxiliem os desenvolvedores na seleção do algoritmo de classificação a ser utilizado pelo classificador. Eles acabam realizando uma busca manual com ferramentas, como Weka, RapidMiner, Torch ou PyBrain, que permitem a seleção de algoritmos de classificação e seus respectivos parâmetros para induzir um classificador dado uma base de dados.

Nesse cenário, este trabalho propôs o **SAC: Search for Audio Classification** que em vez dessa busca manual, realizada pelos desenvolvedores de classificadores de som, realiza uma busca automática. Em particular, foi adotado como busca o método SMAC

que já era utilizado na literatura para otimização de parâmetros.

Essa proposta foi testada em sete ensaios divididos em dois grupos de experimentos com dados reais: a classificação de disparos de arma de fogo em ambientes públicos e a classificação de emoções nas músicas.

O SAC apresentou melhorias em 8 dos 9 ensaios, mesmo concorrendo com soluções desenvolvidas por especialistas. Ao final, os resultados mostraram uma redução média de cerca de 11,26% da taxa de erro.

Em particular, em comparação com os resultados do Áudio Alerta já existentes, o SAC apresentou resultados em menos tempo e conseqüentemente com um custo financeiro inferior ao investido pela Daccord Music nesse processo específico. Principalmente se destacou as características da equipe de especialistas envolvidos, e seu custo de manutenção.

O desejo era de poder afirmar que a proposta de solução seria capaz de melhorar todo tipo de classificador de áudio. Contudo, essa solução carece de experimentos mais aprofundados que não foram possíveis de realizar no momento por limitações de tempo. Mas as ferramentas e técnicas utilizadas ficam para que pesquisadores interessados possam dar continuidade a essa linha de investigação.

A conclusão que se pode tirar é que o SAC, uma alternativa sistêmica, aberta e econômica, auxilia no processo de seleção do algoritmo de classificação e seus respectivos parâmetros para serem aplicados a problemas de classificação automática de áudio. Principalmente, em situações em que não se pode contar com a ajuda de um especialista da área, ou quando se tem pouco tempo para definir uma solução, ou ainda quando se quer validar uma solução existente.

## 6.1 Perspectivas Futuras

Os resultados obtidos neste trabalho são indícios que o emprego de uma busca automática para seleção de algoritmos de classificação e seus parâmetros pode trazer benefícios para os classificadores automáticos de áudio e talvez também para os classificadores automáticos de forma geral.

No entanto, essas afirmações ainda carecem de estudos mais aprofundados, que analisem o comportamento da proposta em diferentes cenários, por exemplo:

- variações no tipo de busca utilizada além do SMAC, como heurísticas, algoritmos

genéticos, simulated annealing, etc;

- diversificação nos tipos de dados, classes e tamanho da base;
- aplicações a problemas de regressão;

É possível ainda que sistemas, como Weka, RapidMiner, Torch e PyBrain absorvam conceitos apresentados neste trabalho, aumentando a eficiência e a utilidade dessas ferramentas.

Para auxiliar nessa continuidade, todas os recursos utilizados neste trabalho ficarão disponíveis para novas pesquisas de interesse da comunidade científica.

# ANEXO A – Sequential Model-Based Algorithm Configuration (SMAC) - Transcrição parcial de Hutter et al. (2013b)

SMAC foi projetado para agir sobre funções *blackbox* que surgem na otimização de parâmetros em algoritmos. Formalmente, o problema de configuração de um algoritmo (CA) resolvido por SMAC pode ser declarado como: um algoritmo parametrizado  $A$  com espaço de configuração  $\Theta$ , uma distribuição de  $D$  casos problemas, e um desempenho  $m(\theta, \pi)$  que captura a performance de  $A$  com os parâmetros  $\theta \in \Theta$  para a instância  $\pi \sim D$ . Seja  $f(\theta) = \mathbb{E}_{\pi \sim D}[m(\theta, \pi)]$  observa-se o desempenho estimado de  $A$  dado os parâmetros  $\theta \in \Theta$  (onde a expectativa é de mais instâncias  $\pi$  tiradas de  $D$ ; no caso dos algoritmos randomizados, seria também sobre sequências aleatórias). O problema é então de encontrar uma parametrização  $\theta$  de  $A$  que minimiza  $f(\theta)$ .

SMAC está baseado no método estatístico sequential model-based optimization (SMBO) (JONES et al., 1998; MOCKUS et al., 1978) de funções custo. Para minimizar funções  $f : \Theta \rightarrow \mathbb{R}$  que não têm uma única forma de representação, são caras de avaliar, e não permitem a computação de gradientes, SMBO primeiro reúne dados iniciais e então realiza os seguintes passos:

1. com base nos dados coletados até então, construir um modelo que prevê uma distribuição de probabilidade para o valor de  $f$  em pontos arbitrários  $\theta \in \Theta$ ;
2. usar o modelo para quantificar a conveniência  $d(\theta)$  de aprender  $f(\theta)$  para cada  $\theta \in \Theta$  e selecionar  $\theta \in \arg \max_{\theta \in \Theta} d(\theta)$ ; e
3. avaliar  $f(\theta)$ , resultando em um novo ponto  $\langle \theta, f(\theta) \rangle$ .

A função de conveniência  $d$  serve para abordar a exploração/tradeoff entre a aprendizagem sobre peças novas, desconhecidas do espaço de parâmetros e intensificar a busca



localmente na região mais conhecida. SMAC usa uma função de conveniência clássica que avalia cada configuração  $\theta$  pela sua esperada melhoria positiva  $\mathbb{E}[I(\theta)] = \mathbb{E}[\max\{0, f_{\min} - f(\theta)\}]$  sobre o menor valor da função  $f_{\min}$  visto até o momento, onde a expectativa é tomada com relação ao modelo onde a expectativa é feita em relação ao modelo predições e pode ser calculado de forma fechada dado um modelo preditivo com distribuição Gaussiana  $\mathcal{N}(\mu_{\theta}, \sigma_{\theta}^2)$  (ver fórmula exata em Jones et al. (1998)).

No desenvolvimento de SMAC, estendeu-se abordagens SMBO existentes de vários modos a fim de obter um processo aplicável ao problema prático da configuração de algoritmos. Especificamente, desenvolveu-se mecanismos para manusear:

- espaços de parâmetros discretos e condicionais (HUTTER et al., 2011) (permitindo a otimização dos parâmetros do algoritmo categóricos e os parâmetros entre os quais existem dependências);
- substancial, o ruído não Gaussiano (HUTTER et al., 2009) (devido à variação na distribuição do tempo de execução do algoritmo sobre casos de problemas e de vários ensaios independentes sobre a mesma instância);
- avaliações da função parcialmente censuradas (HUTTER et al., 2013a) (devido ao algoritmo encerrado prematuramente é executado);
- um custo no tempo total disponível para a configuração do algoritmo, e não no número de avaliações da função (HUTTER et al., 2010), e
- otimização distribuídos em clusters de computadores (HUTTER et al., 2012).

Por padrão, SMAC usa modelos de florestas aleatórias (FA) em vez do processo de Gauss mais comumente utilizados, porque os modelos FA melhoraram o desempenho em otimização discreta e são relativamente fáceis de serem adaptados para acomodar dados censurados e parâmetros condicionais.

Na sua variante padrão, SMAC usa múltiplos inícios, um procedimento de busca local relativamente barato, a fim de otimizar  $\mathbb{E}[I(\theta)]$  sobre  $\theta \in \Theta$ .

# Referências

- AHA, D.; KIBLER, D.; ALBERT, M. Instance-based learning algorithms. **Machine Learning**, Kluwer Academic Publishers, v. 6, n. 1, p. 37–66, 1991. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1007/BF00153759>.
- ANTONIC, D.; ZAGAR, M. Method for determining classification significant features from acoustic signature of mine-like buried objects. In: CITESEER. **15th World Conference on Non-Destructive Testing. Rome**. [S.l.], 2000.
- ATKESON, C.; MOORE, A.; SCHAAL, S. Locally weighted learning. **AI Review**, 1996.
- BERGSTRA, J.; BARDENET, R.; BENGIO, Y.; KÉGL, B. et al. Algorithms for hyperparameter optimization. In: **25th Annual Conference on Neural Information Processing Systems (NIPS 2011)**. [S.l.: s.n.], 2011.
- BRAZDIL, P.; CARRIER, C. G.; SOARES, C.; VILALTA, R. **Metalearning: Applications to data mining**. [S.l.]: Springer, 2009.
- BRAZDIL, P.; SOARES, C.; COSTA, J. da. Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results. **Machine Learning**, Kluwer Academic Publishers, v. 50, n. 3, p. 251–277, 2003. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1023/A%3A1021713901879>.
- BREIMAN, L. Bagging predictors. **Machine Learning**, v. 24, n. 2, p. 123–140, 1996.
- BREIMAN, L. Random forests. **Machine Learning**, Kluwer Academic Publishers, v. 45, n. 1, p. 5–32, 2001. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1023/A%3A1010933404324>.
- BREIMAN, L. Random forests. **Machine Learning**, v. 45, n. 1, p. 5–32, 2001.
- BULLER, G.; LUTMAN, M. E. Automatic classification of transiently evoked otoacoustic emissions using an artificial neural network. **British Journal of Audiology**, v. 32, n. 4, p. 235–247, 1998. Disponível em: <http://informahealthcare.com/doi/abs/10.3109/03005364000000071>.
- CABRAL, G.; PACHET, F.; BRIOT, J.-P. Recognizing chords with eds: Part one. In: KRONLAND-MARTINET, R.; VOINIER, T.; YSTAD, S. (Ed.). **Computer Music Modeling and Retrieval**. Springer Berlin Heidelberg, 2006, (Lecture Notes in Computer Science, v. 3902). p. 185–195. ISBN 978-3-540-34027-0. Disponível em: [http://dx.doi.org/10.1007/11751069\\_17](http://dx.doi.org/10.1007/11751069_17).
- CABRAL, G.; PACHET, F.; BRIOT, J.-P.; PARIS, S. C. Automatic x traditional descriptor extraction: The case of chord recognition. In: **Proceedings of the 6th international conference on music information retrieval (ISMIR'2005), London**. [S.l.: s.n.], 2005.

CAI, R.; ZHANG, C.; WANG, C.; ZHANG, L.; MA, W.-Y. MusicSense: Contextual music recommendation using emotional allocation modeling. In: **Proceedings of the 15th International Conference on Multimedia**. New York, NY, USA: ACM, 2007. (MULTIMEDIA '07), p. 553–556. ISBN 978-1-59593-702-5. Disponível em: <http://doi.acm.org/10.1145/1291233.1291369>.

CANNAM, C.; LANDONE, C.; SANDLER, M. Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files. In: **Proceedings of the International Conference on Multimedia**. New York, NY, USA: ACM, 2010. (MM '10), p. 1467–1468. ISBN 978-1-60558-933-6. Disponível em: <http://doi.acm.org/10.1145/1873951.1874248>.

CANNAM, C.; LANDONE, C.; SANDLER, M. B.; BELLO, J. P. The sonic visualiser: A visualisation platform for semantic descriptors from musical signals. In: **ISMIR**. [S.l.: s.n.], 2006. p. 324–327.

CESSIE, S. le; HOUWELINGEN, J. van. Ridge estimators in logistic regression. **Applied Statistics**, v. 41, n. 1, p. 191–201, 1992.

CLEARY, J. G.; TRIGG, L. E. K\*. An instance-based learner using an entropic distance measure. In: **12th International Conference on Machine Learning**. [S.l.: s.n.], 1995. p. 108–114.

COHEN, W. W. Fast effective rule induction. In: **Twelfth International Conference on Machine Learning**. [S.l.]: Morgan Kaufmann, 1995. p. 115–123.

COLLOBERT, R.; KAVUKCUOGLU, K.; FARABET, C. Torch7: A matlab-like environment for machine learning. In: **BigLearn, NIPS Workshop**. [S.l.: s.n.], 2011.

COPPIN, B. **Inteligência artificial**. Reimpr. Rio de Janeiro: LTC, 2012.

CROSBY, P. B. **Quality is free: The art of making quality certain**. [S.l.: s.n.], 1979.

DITTON, J.; BRITAIN, G. **The effect of closed circuit television on recorded crime rates and public concern about crime in Glasgow**. [S.l.]: CRU, 1999.

DUBNOV, S.; TISHBY, N. Analysis of sound textures in musical and machine sounds by means of higher order statistical features. In: **Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on**. [S.l.: s.n.], 1997. v. 5, p. 3845–3848 vol.5. ISSN 1520-6149.

DUFAUX, A.; BESACIER, L.; ANSORGE, M.; PELLANDINI, F. Automatic sound detection and recognition for noisy environment. In: **CITeseer. Proc. of the X European Signal Processing Conference**. [S.l.], 2000.

FACELI, K.; LORENA, A. C.; GAMA, J.; CARVALHO, A. C. P. L. F. d. **Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina**. Rio de Janeiro: Livros Técnicos e Científicos, 2011.

FAHY, F. J. **Foundations of engineering acoustics**. [S.l.]: Academic Press, 2000.

FILHO, M. d. C. S. **Classificação Automática de Gêneros de Áudio Digital**. Trabalho de Conclusão de Curso — Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, 2006.

FRANK, E.; WANG, Y.; INGLIS, S.; HOLMES, G.; WITTEN, I. Using model trees for classification. **Machine Learning**, v. 32, n. 1, p. 63–76, 1998.

FRANK, E.; WITTEN, I. H. Generating accurate rule sets without global optimization. In: SHAVLIK, J. (Ed.). **Fifteenth International Conference on Machine Learning**. [S.l.]: Morgan Kaufmann, 1998. p. 144–151.

FREUND, Y.; SCHAPIRE, R. E. Experiments with a new boosting algorithm. In: **Thirteenth International Conference on Machine Learning**. San Francisco: Morgan Kaufmann, 1996. p. 148–156.

FREUND, Y.; SCHAPIRE, R. E. Large margin classification using the perceptron algorithm. In: **11th Annual Conference on Computational Learning Theory**. New York, NY: ACM Press, 1998. p. 209–217.

FRISTRUP, K. M.; WATKINS, W. A. Marine animal sound classification. **The Journal of the Acoustical Society of America**, v. 97, n. 5, p. 3369–3369, 1995. Disponível em: <http://scitation.aip.org/content/asa/journal/jasa/97/5/10.1121/1.412646>.

GALINHA, I. C.; PAIS-RIBEIRO, L. Contribuição para o estudo da versão portuguesa da positive and negative affect schedule (panas): I - abordagem teórica ao conceito de afecto. **Análise Psicológica**, scielopt, v. 23, p. 209 – 218, 04 2005. ISSN 0870-8231.

GEROSA, L.; VALENZISE, G.; TAGLIASACCHI, M.; ANTONACCI, F.; SARTI, A. Scream and gunshot detection in noisy environments. In: **15th European Signal Processing Conference (EUSIPCO-07), Sep. 3-7, Poznan, Poland**. [S.l.: s.n.], 2007.

GIRAUD-CARRIER, C.; VILALTA, R.; BRAZDIL, P. Introduction to the special issue on meta-learning. **Machine Learning**, Kluwer Academic Publishers-Plenum Publishers, v. 54, n. 3, p. 187–193, 2004. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1023/B%3AMACH.0000015878.60765.42>.

GOLUB, S. Classifying recorded music. **MSc in Artificial Intelligence. Division of Informatics. University of Edinburgh**, 2000.

GORMAN, R.; SEJNOWSKI, T. Learned classification of sonar targets using a massively parallel network. **Acoustics, Speech and Signal Processing, IEEE Transactions on**, v. 36, n. 7, p. 1135–1140, 1988. ISSN 0096-3518.

GUIMARÃES, L. A. B. **Os limites da vigilância eletrônica em via pública: A experiência gaúcha**. 2007. Disponível em: <http://guayi3.hospedagemdesites.ws/?p=753>.

HALL, M.; FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. H. The weka data mining software: An update. **SIGKDD Explor. Newsl.**, ACM, New York, NY, USA, v. 11, n. 1, p. 10–18, nov. 2009. ISSN 1931-0145. Disponível em: <http://doi.acm.org/10.1145/1656274.1656278>.

HERRERA-BOYER, P.; PEETERS, G.; DUBNOV, S. Automatic classification of musical instrument sounds. **Journal of New Music Research**, v. 32, n. 1, p. 3–21, 2003. Disponível em: <http://www.tandfonline.com/doi/abs/10.1076/jnmr.32.1.3.16798>.

HO, T. K. The random subspace method for constructing decision forests. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 20, n. 8, p. 832–844, 1998. ISSN 0162-8828. Disponível em: <http://citeseer.ist.psu.edu/ho98random.html>.

HOFMANN, M.; KLINKENBERG, R. **RapidMiner: Data Mining Use Cases and Business Analytics Applications**. [S.l.]: CRC Press, 2013.

HOLTE, R. Very simple classification rules perform well on most commonly used datasets. **Machine Learning**, v. 11, p. 63–91, 1993.

HUTTER, F.; HOOS, H.; LEYTON-BROWN, K. Sequential model-based optimization for general algorithm configuration. In: COELLO, C. (Ed.). **Learning and Intelligent Optimization**. Springer Berlin Heidelberg, 2011, (Lecture Notes in Computer Science, v. 6683). p. 507–523. ISBN 978-3-642-25565-6. Disponível em: [http://dx.doi.org/10.1007/978-3-642-25566-3\\_40](http://dx.doi.org/10.1007/978-3-642-25566-3_40).

HUTTER, F.; HOOS, H.; LEYTON-BROWN, K. Parallel algorithm configuration. In: HAMADI, Y.; SCHOENAUER, M. (Ed.). **Learning and Intelligent Optimization**. Springer Berlin Heidelberg, 2012, (Lecture Notes in Computer Science). p. 55–70. ISBN 978-3-642-34412-1. Disponível em: [http://dx.doi.org/10.1007/978-3-642-34413-8\\_5](http://dx.doi.org/10.1007/978-3-642-34413-8_5).

HUTTER, F.; HOOS, H.; LEYTON-BROWN, K. Bayesian optimization with censored response data. **CoRR**, abs/1310.1947, 2013.

HUTTER, F.; HOOS, H.; LEYTON-BROWN, K. An evaluation of sequential model-based optimization for expensive blackbox functions. In: **Proceeding of the Fifteenth Annual Conference Companion on Genetic and Evolutionary Computation Conference Companion**. New York, NY, USA: ACM, 2013. (GECCO '13 Companion), p. 1209–1216. ISBN 978-1-4503-1964-5. Disponível em: <http://doi.acm.org/10.1145/2464576.2501592>.

HUTTER, F.; HOOS, H.; LEYTON-BROWN, K.; MURPHY, K. Time-bounded sequential parameter optimization. In: BLUM, C.; BATTITI, R. (Ed.). **Learning and Intelligent Optimization**. Springer Berlin Heidelberg, 2010, (Lecture Notes in Computer Science, v. 6073). p. 281–298. ISBN 978-3-642-13799-0. Disponível em: [http://dx.doi.org/10.1007/978-3-642-13800-3\\_30](http://dx.doi.org/10.1007/978-3-642-13800-3_30).

HUTTER, F.; HOOS, H. H.; LEYTON-BROWN, K.; MURPHY, K. P. An experimental investigation of model-based parameter optimisation: Spo and beyond. In: **Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation**. New York, NY, USA: ACM, 2009. (GECCO '09), p. 271–278. ISBN 978-1-60558-325-9. Disponível em: <http://doi.acm.org/10.1145/1569901.1569940>.

IMIRSEL. **MIR Community — The Tools We Use**. 2014. [Online; accessed 29-January-2014]. Disponível em: <http://www.music-ir.org/evaluation/tools.html>.

IMIRSEL, I. M. I. R. S. E. L. **MIREX**. 2014. [Online; accessed 08-May-2014]. Disponível em: [http://www.music-ir.org/mirex/wiki/MIREX\\_HOME](http://www.music-ir.org/mirex/wiki/MIREX_HOME).

JAIN, A.; DUIN, R. P. W.; MAO, J. Statistical pattern recognition: a review. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, v. 22, n. 1, p. 4–37, 2000. ISSN 0162-8828.

JARDWSKI, É. R. W. **Análise da eficácia do sistema de vídeo monitoramento da cidade de Colombo**. Projeto Técnico (Especialização em Gestão Pública Municipal) — Universidade Federal do Paraná, Curitiba, 2012.

JOHN, G. H.; LANGLEY, P. Estimating continuous distributions in bayesian classifiers. In: **Eleventh Conference on Uncertainty in Artificial Intelligence**. San Mateo: Morgan Kaufmann, 1995. p. 338–345.

JONES, D.; SCHONLAU, M.; WELCH, W. Efficient global optimization of expensive black-box functions. **Journal of Global Optimization**, Kluwer Academic Publishers, v. 13, n. 4, p. 455–492, 1998. ISSN 0925-5001. Disponível em: <http://dx.doi.org/10.1023/A%3A1008306431147>.

KITTLER, J.; HATEF, M.; DUIN, R. P. W.; MATAS, J. On combining classifiers. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, v. 20, n. 3, p. 226–239, Mar 1998. ISSN 0162-8828.

KOHAVI, R. The power of decision tables. In: **8th European Conference on Machine Learning**. [S.l.]: Springer, 1995. p. 174–189.

KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: **Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995. (IJCAI'95), p. 1137–1143. ISBN 1-55860-363-8. Disponível em: <http://dl.acm.org/citation.cfm?id=1643031.1643047>.

KOTSIANTIS, S. B.; ZAHARAKIS, I.; PINTELAS, P. **Supervised machine learning: A review of classification techniques**. 2007.

KUBAT, M.; BRATKO, I.; MICHALSKI, R. A review of machine learning methods. 1998.

LANDWEHR, N.; HALL, M.; FRANK, E. Logistic model trees. **Machine Learning**, Springer, v. 59, n. 1-2, p. 161–205, 2005.

LANDWEHR, N.; HALL, M.; FRANK, E. Logistic model trees. **Machine Learning**, v. 95, n. 1-2, p. 161–205, 2005.

LEE, C.-H.; HAN, C.-C.; CHUANG, C.-C. Automatic classification of bird species from their sounds using two-dimensional cepstral coefficients. **Audio, Speech, and Language Processing, IEEE Transactions on**, v. 16, n. 8, p. 1541–1550, 2008. ISSN 1558-7916.

LI, T.; OGIHARA, M. Detecting emotion in music. In: **ISMIR**. [S.l.: s.n.], 2003. v. 3, p. 239–240.

LI, T.; OGIHARA, M. Toward intelligent music information retrieval. **Multimedia, IEEE Transactions on**, v. 8, n. 3, p. 564–574, 2006. ISSN 1520-9210.

- LIENBART, R.; PFEIFFER, S.; EFFELSBERG, W. Scene determination based on video and audio features. In: **Multimedia Computing and Systems, 1999. IEEE International Conference on**. [S.l.: s.n.], 1999. v. 1, p. 685–690 vol.1.
- LINDEN, R. **Algoritmos Genéticos**. 2. ed. [S.l.]: Brasport, 2008.
- LIPPMANN, R. P. Review of neural networks for speech recognition. **Neural Computation**, MIT Press, v. 1, n. 1, p. 1–38, 2014/01/29 1989. Disponível em: <http://dx.doi.org/10.1162/neco.1989.1.1.1>.
- LIU, Z.; WANG, Y.; CHEN, T. Audio feature extraction and analysis for scene segmentation and classification. **J. VLSI Signal Process. Syst.**, Kluwer Academic Publishers, Hingham, MA, USA, v. 20, n. 1/2, p. 61–79, out. 1998. ISSN 0922-5773. Disponível em: <http://dx.doi.org/10.1023/A:1008066223044>.
- LOGAN, B. et al. Mel frequency cepstral coefficients for music modeling. In: **ISMIR**. [S.l.: s.n.], 2000.
- LU, L.; LIU, D.; ZHANG, H.-J. Automatic mood detection and tracking of music audio signals. **Audio, Speech, and Language Processing, IEEE Transactions on**, v. 14, n. 1, p. 5–18, 2006. ISSN 1558-7916.
- MALHEIRO, R.; PAIVA, R. P.; MENDES, A. J.; MENDES, T.; CARDOSO, A. Sistemas de classificação musical com redes neuronais. Universidade Católica Portuguesa. Instituto Universitário de Desenvolvimento e Promoção Social, 2004.
- MCCALLUM, A.; NIGAM, K. A comparison of event models for naive bayes text classification. In: **AAAI-98 Workshop on 'Learning for Text Categorization'**. [S.l.: s.n.], 1998.
- MCLAUGHLIN, J.; OWSLEY, L. M.; ATLAS, L. E.; BERNARD, G. D. Advances in real-time monitoring of acoustic emissions. In: **SOC AUTOMATIVE ENGINEERS INC. SAE CONFERENCE PROCEEDINGS P**. [S.l.], 1997. p. 291–297.
- MELLO, R. L. M. d. Câmeras de vigilância nas ruas e qualidade dos espaços públicos urbanos. **Revista LEVS**, n. 3, 2011.
- MEYER, B. **Object-oriented software construction**. [S.l.]: Prentice hall New York, 1988.
- MILLS, H. Automatic detection and classification of nocturnal migrant bird calls. **The Journal of the Acoustical Society of America**, v. 97, n. 5, p. 3370–3370, 1995. Disponível em: <http://scitation.aip.org/content/asa/journal/jasa/97/5/10.1121/1.412649>.
- MITCHELL, T. M. **Machine learning**. [S.l.]: McGraw Hill, 1997.
- MOCKUS, J.; TIESIS, V.; ZILINSKAS, A. The application of bayesian methods for seeking the extremum. **Towards Global Optimization**, Amsterdam: Elsevier, v. 2, n. 117-129, p. 2, 1978.
- MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. **Sistemas Inteligentes-Fundamentos e Aplicações**, Manole, p. 89–114, 2003.

MUSIC-IR. **The Tools We Use, version 1.5**. 2014. [Online; accessed 10-January-2014]. Disponível em: <http://music-ir.org/evaluation/tools.html>.

NAVRATIL, M.; DOSTALEK, P.; KRESALEK, V. Classification of audio sources using neural network applicable in security or military industry. In: **Security Technology (ICCST), 2010 IEEE International Carnahan Conference on**. [S.l.: s.n.], 2010. p. 369–374. ISSN 1071-6572.

PACHET, F.; ZILS, A. Evolving automatically high-level music descriptors from acoustic signals. In: WIIL, U. (Ed.). **Computer Music Modeling and Retrieval**. Springer Berlin Heidelberg, 2004, (Lecture Notes in Computer Science, v. 2771). p. 42–53. ISBN 978-3-540-20922-5. Disponível em: [http://dx.doi.org/10.1007/978-3-540-39900-1\\_5](http://dx.doi.org/10.1007/978-3-540-39900-1_5).

PÉREZ, G. C. **Automatic robust classification of speech using analytical feature techniques**. Projecte Final de Carrera — Universitat Politècnica de Catalunya, Barcelona, 2011.

PERNAMBUCO, G. d. E. d. **Sistema de som nas ruas para evitar crimes**. 2014. [Online; accessed 29-January-2014]. Disponível em: <http://www.pactopelavida.pe.gov.br/sistema-de-som-nas-ruas-para-evitar-crimes/>.

PIRES, P. R. d. M. **Processamento Inteligente de Sinais de Pressão e Temperatura Adquiridos Através de Sensores Permanentes em Poços de Petróleo**. Dissertação (Mestrado em Ciência e Engenharia do Petróleo — Universidade Federal do Rio Grande do Norte, Natal, RN, Brasil, 2012.

PLATT, J. Fast training of support vector machines using sequential minimal optimization. In: SHOELKOPF, B.; BURGESS, C.; SMOLA, A. (Ed.). **Advances in Kernel Methods - Support Vector Learning**. MIT Press, 1998. Disponível em: <http://research.microsoft.com/~jplatt/smo.html>.

POTTER, J. R.; MELLINGER, D. K.; CLARK, C. W. Marine mammal call discrimination using artificial neural networks. **The Journal of the Acoustical Society of America**, v. 96, n. 3, p. 1255–1262, 1994. Disponível em: <http://scitation.aip.org/content/asa/journal/jasa/96/3/10.1121/1.410274>.

PRUDÊNCIO, R. B.; LUDERMIR, T. B. Aprendizagem ativa para seleção de exemplos em meta-aprendizado. **Anais do XXVII Congresso da SBC - VI ENIA**, 2007.

QUINLAN, R. **C4.5: Programs for Machine Learning**. San Mateo, CA: Morgan Kaufmann Publishers, 1993.

RABINER, L.; JUANG, B.-H. **Fundamentals of Speech Recognition**. United States ed. Prentice Hall, 1993. Paperback. ISBN 0130151572. Disponível em: <http://www.worldcat.org/isbn/0130151572>.

RATCLIFFE, J. **Video surveillance of public places**. [S.l.]: US Department of Justice, Office of Community Oriented Policing Services, 2006.



RECIFE, P. d. C. **Novo Centro de Operações ampliará resultados na Segurança, Mobilidade e Ordenamento.** 2014. [Online; accessed 29-January-2014]. Disponível em: <http://www2.recife.pe.gov.br/novo-centro-de-operacoes-ampliara-resultados-na-seguranca-mobilidade-e-ordenamento>.

RODRIGUES, R. A. M. **Algoritmos Evolutivos aplicados ao Classificador baseado em Segmentos de Reta.** Dissertação (Mestrado em Ciência da Computação) — Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2012.

RT. **Big Brother next door? Most of UK's 6 milion CCTV cameras are privately owned.** 2013. [Online; acessado em 15-Outubro-2013]. Disponível em: <http://rt.com/news/cctv-uk-private-surveillance-918/>.

RUSSELL, S.; NORVIG, P. **Inteligência Artificial: uma abordagem moderna.** 5 reimpr.. ed. Rio de Janeiro: Elsevier, 2004.

SAHA, B.; PURKAIT, P.; MUKHERJEE, J.; MAJUMDAR, A.; MAJUMDAR, B.; SINGH, A. An embedded system for automatic classification of neonatal cry. In: **Point-of-Care Healthcare Technologies (PHT), 2013 IEEE.** [S.l.: s.n.], 2013. p. 248–251.

SCHAUL, T.; BAYER, J.; WIERSTRA, D.; SUN, Y.; FELDER, M.; SEHNKE, F.; RÜCKSTIESS, T.; SCHMIDHUBER, J. PyBrain. **Journal of Machine Learning Research**, 2010.

SCHÖN, P.-C.; PUPPE, B.; MANTEUFFEL, G. Linear prediction coding analysis and self-organizing feature map as tools to classify stress calls of domestic pigs (*sus scrofa*). **The Journal of the Acoustical Society of America**, v. 110, n. 3, p. 1425–1431, 2001. Disponível em: <http://scitation.aip.org/content/asa/journal/jasa/110/3/10.1121/1.1388003>.

SHIYONG, Z.; ZEHAN, C.; FEI, G.; LI, F.; SHOUZHONG, X. The knowledge-based signal analysis for a heart sound information system. In: **Engineering in Medicine and Biology Society, 1998. Proceedings of the 20th Annual International Conference of the IEEE.** [S.l.: s.n.], 1998. v. 3, p. 1622–1624 vol.3. ISSN 1094-687X.

SILVA, A. P. e; MICHAELIS, H. **Michaelis: moderno dicionário da língua portuguesa.** [S.l.]: Melhoramentos, 1998.

SIMON, H. **Redes Neurais: princípios e prática.** 2. ed. Porto Alegre: Bookman, 2001.

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

THORNTON, C.; HUTTER, F.; HOOS, H. H.; LEYTON-BROWN, K. Auto-weka: Automated selection and hyper-parameter optimization of classification algorithms. **CoRR**, abs/1208.3719, 2012.

THORNTON, C.; HUTTER, F.; HOOS, H. H.; LEYTON-BROWN, K. Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In: **Proceedings of the 19th ACM SIGKDD International Conference on Knowledge**

**Discovery and Data Mining**. New York, NY, USA: ACM, 2013. (KDD '13), p. 847–855. ISBN 978-1-4503-2174-7. Disponível em: <http://doi.acm.org/10.1145/2487575.2487629>.

TOLOS, M.; TATO, R.; KEMP, T. Mood-based navigation through large collections of musical data. In: **Consumer Communications and Networking Conference, 2005. CCNC. 2005 Second IEEE**. [S.l.: s.n.], 2005. p. 71–75.

TSOUMAKAS, K. T. G.; KALLIRIS, G.; VLAHAVAS, I. Multi-label classification of music into emotions. In: LULU. COM. **ISMIR 2008: Proceedings of the 9th International Conference of Music Information Retrieval**. [S.l.], 2008. p. 325.

TZANETAKIS, G.; COOK, P. Marsyas: a framework for audio analysis. **Organised Sound**, v. 4, p. 169–175, 12 2000. ISSN 1469-8153. Disponível em: [http://journals.cambridge.org/article\\_{\\_}S1355771800003071](http://journals.cambridge.org/article_{_}S1355771800003071).

TZANETAKIS, G.; COOK, P. Musical genre classification of audio signals. **Speech and Audio Processing, IEEE Transactions on**, v. 10, n. 5, p. 293–302, 2002. ISSN 1063-6676.

TZANETAKIS, G.; COOK, P. Musical genre classification of audio signals. **Speech and Audio Processing, IEEE Transactions on**, v. 10, n. 5, p. 293–302, 2002. ISSN 1063-6676.

VALENZISE, G.; GEROSA, L.; TAGLIASACCHI, M.; ANTONACCI, F.; SARTI, A. Scream and gunshot detection and localization for audio-surveillance systems. In: **IEEE. Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on**. [S.l.], 2007. p. 21–26.

VILALTA, R.; GIRAUD-CARRIER, C. G.; BRAZDIL, P.; SOARES, C. Using meta-learning to support data mining. **IJCSA**, v. 1, n. 1, p. 31–45, 2004.

WIECZORKOWSKA, A.; SYNAK, P.; RAŚ, Z. Multi-label classification of emotions in music. In: KTOPOTEK, M.; WIERZCHOŃ, S.; TROJANOWSKI, K. (Ed.). **Intelligent Information Processing and Web Mining**. Springer Berlin Heidelberg, 2006, (Advances in Soft Computing, v. 35). p. 307–315. ISBN 978-3-540-33520-7. Disponível em: [http://dx.doi.org/10.1007/3-540-33521-8\\_30](http://dx.doi.org/10.1007/3-540-33521-8_30).

WIKIPEDIA. **Cluster analysis** — **Wikipedia, The Free Encyclopedia**. 2014. [Online; accessed 24-January-2014]. Disponível em: [http://en.wikipedia.org/wiki/Data\\_clustering](http://en.wikipedia.org/wiki/Data_clustering).

WIKIPEDIA. **RapidMiner** — **Wikipedia, The Free Encyclopedia**. 2014. [Online; acessado em 10-Janeiro-2014]. Disponível em: <http://en.wikipedia.org/wiki/RapidMiner>.

WOLPERT, D. H. Stacked generalization. **Neural Networks**, Pergamon Press, v. 5, p. 241–259, 1992.

WOLPERT, D. H. The lack of a priori distinctions between learning algorithms. **Neural Computation**, MIT Press, v. 8, n. 7, p. 1341–1390, 2014/01/07 1996. Disponível em: <http://dx.doi.org/10.1162/neco.1996.8.7.1341>.

YANG, Y.-H.; LIN, Y.-C.; SU, Y.-F.; CHEN, H. A regression approach to music emotion recognition. **Audio, Speech, and Language Processing, IEEE Transactions on**, v. 16, n. 2, p. 448–457, 2008. ISSN 1558-7916.

YANG, Y.-H.; LIU, C.-C.; CHEN, H. H. Music emotion classification: A fuzzy approach. In: **Proceedings of the 14th Annual ACM International Conference on Multimedia**. New York, NY, USA: ACM, 2006. (MULTIMEDIA '06), p. 81–84. ISBN 1-59593-447-2. Disponível em: <http://doi.acm.org/10.1145/1180639.1180665>.