JOSÉ ANTONIO ALVES DE MENEZES

EXPLORAÇÃO DE CARACTERÍSTICAS ANALÍTICAS PARA CLASSIFICADORES AUTOMÁTICOS DE ÁUDIO ATRAVÉS DE OTIMIZAÇÃO MULTIOBJETIVO

Recife

2016

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA APLICADA

EXPLORAÇÃO DE CARACTERÍSTICAS ANALÍTICAS PARA CLASSIFICADORES AUTOMÁTICOS DE ÁUDIO ATRAVÉS DE OTIMIZAÇÃO MULTIOBJETIVO

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

Área de concentração: Computação Inteligente.

Orientador: Prof. Dr. Giordano Ribeiro Eulálio Cabral

Recife

2016

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas da UFRPE
Nome da Biblioteca, Cidade-PE, Brasil

M543e Menezes, José Antonio Alves de
Exploração de características analíticas para classificadores automáticos de áudio através de otimização multiobjetivo / José Antonio Alves de Menezes. – 2016.
82 f. : il.

Orientador(a): Giordano Ribeiro Eulálio Cabral .

Dissertação (Mestrado) – Universidade Federal Rural de Pernambuco, Programa de Pós-Graduação em Informática Aplicada, Recife, BR-PE, 2016.
Inclui apêndice(s) e referências.

1. Classificação automática de áudio 2. Feature learning 3. Espaço analítico 4. Algoritmos evolucionários 5. Otimização multiobjetivo I. Cabral, Giordano Ribeiro Eulálio, orient. II. Título

CDD 004

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA APLICADA

EXPLORAÇÃO DE CARACTERÍSTICAS ANALÍTICAS PARA CLASSIFICADORES AUTOMÁTICOS DE ÁUDIO ATRAVÉS DE OTIMIZAÇÃO MULTIOBJETIVO

JOSÉ ANTONIO ALVES DE MENEZES

Dissertação julgada adequada para obtenção do título de mestre em Informática Aplicada, defendida e aprovada por unanimidade, em 31/08/2016, pela Comissão Examinadora.

Orientador:

_____

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

Banca Examinadora:

_____

Prof. Dr. Cícero Garrozi
Universidade Federal Rural de Pernambuco
DEInfo-UFRPE

_____

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

Agradecimentos

Quero agradecer as pessoas e instituições que com o seu apoio tornaram possível a realização deste trabalho.

Em primeiro lugar, a minha família que me ajudou de maneira singular nesses últimos anos, me apoiando, incentivando e sobretudo compreendendo. Certamente, sem meus queridos pais e irmã toda trajetória seria mais difícil.

Ao meu orientador Giordano Cabral por todo dinamismo, confiança e disposição fundamentais para que realizássemos uma boa pesquisa. Da paciência a generosidade e com competência onipresente, são algumas das características que me fazem admirá-lo e toma-lo como modelo de pesquisador.

Agradeço a banca avaliadora que sem sombra de dúvida acrescenta e põe rigor ao trabalho.

Também ao Programa PPGIA e todos os que o compõe. Secretaria, coordenação, professores e demais. Especialmente ao professor Cícero que sugeriu a ideia que nortearia meu trabalho nos meses que se seguiram. Também não posso deixar de agradecer ao aluno do Centro de Informática da UFPE Bruno Tavares pela excelente ajuda técnica.

A Universidade Federal Rural de Pernambuco, a saber, desde que saí do colegial, tem sido a minha segunda casa.

Finalmente, agradeço a Deus por ter me permitido viver tamanha experiência de aprendizado e amadurecimento, ensinando-me a cada dia a *fides quaerens intellectum* que tanto me inspira.

No mais, obrigado a todos que me acompanharam em algum momento dessa trajetória, torcendo pelo meu sucesso.

“Mais vale o fim de uma coisa do que seu começo, mais vale a paciência do que a pretensão. ”

Ecle 7, 8.

Resumo

A escolha de características de áudio sempre foi um tema de bastante interesse entre os especialistas em classificação automática de sons, que veem nessa etapa do processo a parte talvez mais importante dos esforços em resolver problemas de classificação. É nesse sentido que surgem técnicas de *Feature Learning* com o intuito de conceber novas características que se adequem ao modelo de classificação. Entretanto essas técnicas em geral independem de domínio de conhecimento, podendo ser aplicadas nos mais diversos tipos de dados. Contudo abordagens dependentes de domínio inferem um tipo de conhecimento restrito ao campo que se estuda. Nesse sentido o áudio constitui um campo com possibilidade para *Feature Learning* que utilize conhecimento específico desse campo. Muitas são as técnicas que procuram melhorar o desempenho da geração de novas características acústicas, dentre elas se destaca aquela que utiliza algoritmos evolucionários para explorar um espaço analítico de funções. Entretanto, os esforços dispendidos até então deixam espaço para melhoras. O intuito desse trabalho é propor e avaliar uma alternativa multiobjetivo para a exploração de características analíticas de áudio. Além do método, que por si já contribui para o intuito, foram organizados experimentos para validação do mesmo através da provação de um protótipo computacional que implementasse a solução proposta. Ao fim foi constatado a efetividade do modelo e a garantia de que ainda há espaços para melhora no segmento escolhido.

Palavras-chave: Classificação automática de áudio, *feature learning*, espaço analítico, algoritmos evolucionários, otimização multiobjetivo.

Abstract

To choice audio features has been a very interesting theme for audio classification experts. They have seen that this process is probably the most important effort to solve the classification problem. In this sense, surge techniques of *Feature Learning* for generate new features more suitable for classification model than conventional features. However, these techniques generally do not depend on knowledge domain and they can apply in various types of raw data. However, less agnostic approaches learn a type of knowledge restricted to the area studded. The audio data requires a specific knowledge type. There are many techniques that seek to improve the performance of the new generation of acoustic characteristics, among which stands the technique that use evolutionary algorithms to explore analytical space of function. However, the efforts made leave opportunities for improvement. The purpose of this work is to propose and evaluate a multi-objective alternative to the exploitation of analytical characteristics of audio. In addition, experiments were arranged to be validated the method, with the help a computational prototype that implemented the proposed solution. After it was found the effectiveness of the model and ensuring that there is still opportunity for improvement in the chosen segment.

Key-words: Automatic audio classification, feature learning, analytical space, evolutionary algorithms, multi-objective optimization.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

ACE	*Autonomous Classification Engine*
AE	Algoritmo Evolucionário
ARFF	*Attribute-Relation File Format*
CAA	Classificação Automática de Áudio
CE	Computação Evolucionária
CUDA	*Compute Unified Device Architecture*
EDS	*Extractor Discovery System*
FFT	*Fast Fourier Transform*
ICA	*Independent Components Analysis*
K-NN	*K-Nearest Neighbor*
LDA	*Linear Discriminant Analysis*
MFCC	*Mel Frequency Cepstral Coefficient*
MIR	*Musical Information Retrieval*
NSGA	*Nondominated Sorting Genetic Algorithm*
PAES	*Pareto Archived Evolution Strategy*
PCA	*Principal Components Analysis*
RMS	*Root Main Square*
SMAC	*Sequencial Model-based Algorithm Selection*
SPEA	*Strength Pareto Evolutionary Algorithm*
SVM	*Support Vector Machine*

Sumário

1. Introdução

Classificação Automática de Áudio (CAA) é um tema de grande interesse para pesquisadores e profissionais do segmento de Recuperação de Informação Musical (*Musical Information Retrieval* – MIR). Devido a sua relevância muitos são os desafios que surgem decorrentes da exigência cada vez maior de melhores resultados. A tarefa envolve muitos conceitos no segmento de inteligência computacional, como aprendizado de máquina, reconhecimento de padrões, *feature learning*, otimização, além de outra gama de conhecimentos específicos: processamento de sinais acústicos e atributos de áudio. Nesse contexto, recebe uma grande importância a atividade de busca de boas características de áudio.

1.1. Motivação

Devido à complexidade que os problemas reais de CAA podem ter, muitas vezes o processo de concepção de características acústicas torna-se artesanal, demandando conhecimento especializado e tempo de projeto. Consequentemente abordagens analíticas, que automatizam a geração de novas características, vem se mostrando promissoras, pois dispensam conhecimento de especialista, são escaláveis, menos custosas, além de serem adaptáveis para o reuso em distintos problemas de classificação.

No geral, técnicas de *Feature Learning* (BENGIO, 2013) são primeiramente pensadas, justamente por não dependerem de informações específicas do domínio de conhecimento do problema, sendo úteis nos mais diversos campos como: processamento de imagens, vídeo, sensores, etc. Entretanto o áudio possui particularidades as quais abrem caminho para outras alternativas. Nisso infere-se que pesquisas envolvendo o domínio de conhecimento da CAA

podem obter ganhos relevantes no tocante ao desempenho da classificação, facilidade de uso, tempo de projeto ou execução.

Com o uso de computação evolucionária tem-se conseguido melhoras significativas em abordagens de domínio específico, como o *Extractor Discovery System* (EDS), proposto por Pachet (2003), que utiliza programação genética para explorar um espaço analítico de funções e encontrar novas características de áudio. A técnica se propõe a evoluir características individualmente e retornar o conjunto daquelas exploradas pela busca. Entretanto essa abordagem, apesar da evolução ao longo dos anos seguintes ao seu lançamento (PACHET; ROY, 2007, 2009), encontra-se estagnada. Possivelmente isso se deve as alternativas recentes que vem ganhando espaço, a saber, o *Deep Learning*, que tem sido cada vez mais aplicado em problemas de MIR (ZHOU; LERCH, 2015). O *Deep Learning* possui a qualidade de aprendizado de características, e pode ser entendida no contexto desse trabalho como uma solução de *Feature Learning*. Ele apresenta-se como uma opção caixa-preta e, portanto, nem sempre é possível extrair conclusões das características aprendidas pela técnica. Visando conceber características acústicas inteligíveis identificamos no EDS oportunidade de melhorias, uma das quais envolve o uso de algoritmos evolucionários multiobjetivo e que podem restaurar o caráter inovador da solução. Ocorre que alguns problemas de CAA requerem a minimização exclusiva dos falsos positivos ou negativos da matriz de confusão. Os métodos de algoritmos genéticos simples, como o EDS, não atendem a essa necessidade. Por esses métodos, o objetivo só pode ser alcançado por um efeito colateral, em que a solução obtém boa acurácia e consequentemente minimiza erros. Mas acreditamos que essa particularidade pode ser tratada de forma mais restrita. Além do mais, o processo do EDS envolve duas etapas no tocante a escolha de características, a saber, a otimização de características de áudio e a seleção daquelas mais relevantes.

A utilização de algoritmos multiobjetivo ocasiona não só uma melhor adequação ao tipo de problema acima citado, como também abre a possibilidade de simplificar o processo de escolha de características, fazendo simultaneamente o que o EDS faz em duas etapas.

A falta de soluções abertas desse tipo constitui um outro motivador para este trabalho. O EDS é um sistema proprietário.

1.2. Objetivos

Visando uma abordagem menos agnóstica pensamos em uma melhoria na busca analítica de novas características de áudio que utilizasse algoritmos evolucionários multiobjetivo. Disso foi originada a seguinte questão de pesquisa: “É possível melhorar o desempenho de um conjunto de características de áudio utilizado em atividades de CAA através da otimização multiobjetivo das características desse conjunto? ”. Afim de responder essa questão, o direcionamento da pesquisa se resumiu aos objetivos apresentados a seguir.

1.2.1. Geral

Analisar o poder de algoritmos evolucionários multiobjetivo na concepção de características analíticas de áudio.

1.2.2. Específicos

- Analisar a técnica de classificação de áudio e o processo de aprendizado de características, no que tange a utilização de algoritmos genéticos, identificando limitações e oportunidade de melhorias.
- Contribuir com o entendimento do processo de composição analítica de características de áudio e da complexidade do espaço analítico que as contêm.
- Propor uma estratégia de otimização de características num processo de classificação automática de áudio, que utilize algoritmos evolucionários multiobjetivo aplicados num espaço analítico.

1.3. Contribuições

Uma vez que na literatura não há proposta que envolva a construção de características analíticas pelo método adotado por este trabalho, e que ao mesmo tempo utilizasse algoritmos multiobjetivo, foi necessário desenvolver um protótipo que incorporasse a solução proposta. Através das comparações com a técnica mono-objetivo e com uma técnica de *feature learning* independente de domínio, pudemos confirmar nossa hipótese inicial de que é possível obter melhores resultados com a abordagem proposta por este trabalho.

Realizou-se experimentos com duas bases de dados para problemas reais distintos: reconhecimento de disparo de arma de fogo e identificação de nasalidade. Foi constatado que o uso de otimização multiobjetivo de características de áudio pode melhorar a acurácia do modelo de classificação, além de se adequar melhor às necessidades específicas de alguns problemas.

1.4. Organização do Documento

Após o presente capítulo segue-se mais cinco capítulos, com o intuito de apresentar sistematicamente o resultado da pesquisa realizada, onde são enfatizados o problema, a solução proposta e os métodos de validação da mesma.

No Capítulo 2 o contexto e a complexidade do problema são abordados, assim como são detalhados os requisitos esperados de uma possível solução.

O Capítulo 3 compreende o referencial teórico sobre o qual este trabalho se sustentou e também o estado da arte das propostas de solução. Aqui são apresentadas algumas ferramentas, como a já citada EDS, seus métodos empregados, tecnologias e processos.

No Capítulo 4 a solução proposta é apresentada, frisando o seu diferencial em relação ao estado da arte, seu algoritmo global, bem como a arquitetura e os detalhes de execução do protótipo construído.

A solução logo é validada no Capítulo 5 através da realização de dois experimentos envolvendo problemas de classificação de áudio reais. É exposta a análise quantitativa dos dados obtidos e busca-se consolidar os resultados verificando a adequação com o objetivo inicial e a adequação aos requisitos esperados.

Por fim o trabalho é concluído no Capítulo 6, onde as principais contribuições são sintetizadas e as lacunas deixadas e oportunidades para trabalhos futuros são expostos.

2. Descrição do Problema

Com a popularização cada vez maior das tecnologias de áudio digital, a quantidade de possibilidades de aplicações que demandam CAA (Classificação Automática de Áudio), na indústria e na vida cotidiana das pessoas, acaba aumentando significativamente e, portanto, melhorias nas soluções existentes sempre são esperadas.

2.1. O Contexto

Sabe-se que a escolha de boas características é determinante para a eficiência de atividades de classificação, uma vez que delimitam de modo razoável cada classe do problema (MCKAY, 2005), (YASLAN, 2006). Entretanto nem sempre é fácil determinar quais características são as mais adequadas para resolver determinado problema, e por isso estratégias de concepção de novas características são importantes. Diante disso surge o interesse nas técnicas de aprendizado de características (*Feature Learning*), como: PCA (*Principal Components Analysis*) (BURKA, 2010), ICA (*Independent Components Analysis*) (ERONEN, 2003) e *Deep Learning* (HUMPHREY et al. 2012), dentre outras. *Feature Learning* também é importante na redução do custo de medição e do risco de sobreajuste (*overfitting*), uma vez que pode reduzir a dimensão do espaço de características, capacitando o mecanismo de classificação generalizar observações. É, portanto, um procedimento que melhora não somente os resultados como também os custos da classificação.

Entretanto é de nosso interesse que, além de prover bons resultados, a busca de boas características seja inteligível, de tal maneira que as características resultantes do processo tenham significado para o profissional da área. Apesar das alternativas de *Feature Learning* acima citadas serem efetivas, também são genéricas quanto ao domínio de conhecimento em que podem ser aplicadas. Além de algumas serem caixa-preta, sendo contrárias ao nosso

interesse. Visamos uma solução menos agnóstica e inteligível que possibilite, dessa forma, uma interpretação para os estudiosos.

É nesse contexto que surgem técnicas de aprendizado de características específicas para o áudio, como a exploração automática do espaço analítico de características acústicas (Seção 3.3.1), nesta linha se destaca o EDS (*Extractor Discovery System*) (PACHET; ROY, 2009) com seu método de geração de novos atributos através do uso de computação evolucionária.

2.2. O Problema

São três os problemas que envolvem CAA que desejamos abordar nesse trabalho, no tocante a seleção de características. São elas: a necessidade de encontrar novas características; a dificuldade na concepção das mesmas e a possibilidade de melhora da técnica de Computação Evolucionária como método de solução desses problemas.

A necessidade de encontrar novas características de áudio se deve a situação de que aquelas conhecidas (por ex. *Fast Fourier Transform* – FFT, Mel *Frequency Cepstral Coefficient* – MFCC, *Zero Crossing*, *Root Main Square* – RMS, etc.) muitas vezes não resolvem satisfatoriamente problemas de classificação restritos, sendo necessário, porém custoso, conceber manualmente características acústicas que se adequem melhor a natureza do problema. Nessa concepção artesanal não há garantia de se chegar a soluções satisfatórias. E ainda que se chegue a boas características, dificilmente essas poderão ser utilizadas em outro problema de classificação, uma vez que cada problema possui suas classes particulares. Daí a necessidade de um método automático e rápido que possa encontrar essas características.

A principal dificuldade envolve a infinitude do espaço de busca, que faz da solução ótima ilimitada, sendo assim, a concepção de características é inviável por busca exaustiva. No final a tarefa pode se resumir em um processo de tentativa e erro, ainda que se utilize métodos exatos computacionais. É nesse sentido que algoritmos evolucionários se apresentam como uma boa opção para aproximação da solução ótima, graças a sua natureza heurística.

Além das problemáticas anteriores parte-se da hipótese de que também é possível melhorar a eficiência na resolução de problemas de classificação quaisquer, através da busca heurística de novas características. Isso porque, dado um problema de classificação de áudio,

sendo o espaço de busca infinito, é hipoteticamente possível, até certo ponto, encontrar melhores características do que as atuais (PACHET; ROY, 2007). Essa hipótese, por sua vez foi sendo validada nos trabalhos referentes ao EDS (CABRAL et al, 2005), (PACHET; ZILS, 2003), (PACHET; ROY, 2007).

Contudo implementações como o EDS são passíveis de melhora, uma vez que apenas utilizam otimização mono-objetivo e, pelo que consta na literatura, carecem de uma forma objetiva de tratar certas especificidades de problemas de classificação.

Muitas vezes em uma atividade de CAA não se está apenas interessado em melhorar a acurácia do método utilizado, mas resolver o problema de tal modo que determinadas condições sejam satisfeitas. A não satisfação dessas condições pode até desencadear consequências graves, dependendo da aplicação (ARAÚJO, 2014). Tomemos como exemplo um sistema de monitoramento de saúde: o sistema pode até errar em emitir um alarme falso sobre a situação de um paciente, mas, dependendo do que se está monitorando, não pode deixar de emitir o alarme verdadeiro na ocorrência de algum evento anormal. Sendo assim, a taxa de acerto do sistema, em si mesma, não é o principal aspecto a ser melhorado. Nesse caso a otimização deve ser direcionada a diminuir os erros relacionados aos alarmes verdadeiros. Portanto as características que devem emergir do processo de aprendizado devem ser capazes de definir com precisão os eventos críticos do problema, satisfazendo os requisitos demandados.

Além do mais, a abordagem do EDS necessita de um processo complementar. Ao final da busca se tem um número considerável de características. Mas ainda é necessário selecionar aquelas que são de fato relevantes para a solução de um problema. Dessa forma, métodos de seleção ainda são exigidos. Numa estratégia onde se busca evoluir o conjunto e não somente uma característica em particular, ao final do processo já se tem o melhor conjunto de características, isentando-se da necessidade de métodos de seleção pós-otimização.

Algoritmos evolucionários multiobjetivo apresentam-se como boa alternativa para essas questões.

2.3. Solução Esperada

Esperamos que uma solução encontre, em tempo viável, um conjunto de características que possam representar os dados de uma tarefa de classificação automática, através do emprego de técnicas de computação evolucionária multiobjetivo.

Dentre outros critérios de satisfação a serem perseguidos para a solução desenvolvida, seguem os seguintes:

- Corretude. Atividades de classificação devem ser o tanto quanto possível isentas de erros. A estratégia de otimizar características deve influenciar na melhora da acurácia do classificador;
- Adequação. A solução esperada deve levar em conta as possíveis especificidades de determinados problemas, facilitando os meios de satisfazer as restrições de cada um deles.
- Disponibilidade de código. Sendo um tema bastante relevante é importante que as soluções sejam abertas para novas contribuições diminuindo os custos e esforços;
- Reusabilidade e escalabilidade da técnica em distintos problemas de CAA. Embora não seja comum reutilizar as características de áudio de um problema de classificação particular, a técnica para encontrá-las deve ser adaptável a qualquer tipo e tamanho de problema;
- Economia de conhecimento especializado. Diferentemente da fase de implementação, em que é necessário conhecer a natureza das características de áudio, na fase de execução seria importante haver independência de conhecimento especializado para tornar a solução mais acessível a desenvolvedores de tecnologias inteligentes de áudio.

Uma vez conhecido o problema e elencados os critérios da solução, realizou-se pesquisa a fim de encontrar soluções e saber quais critérios atendiam. Constatamos a ineficácia do estado da arte e propusemos uma alternativa que contemplasse esses critérios em sua totalidade. Nos capítulos a seguir são apresentados o estado da arte, a solução proposta e a validação da mesma.

3. Estado da arte

Este capítulo apresenta os fundamentos teóricos envolvidos no problema de geração automática de características analíticas de áudio. Também apresenta o que existe de solução dentro do escopo da pesquisa: métodos de computação evolucionária para exploração do espaço analítico de funções acústicas.

3.1. Aprendizado de Máquina

Aprendizado de máquina é o nome que se dá às técnicas computacionais utilizadas para reconhecer padrões, corrigir erros, fazer inferência e generalizações através do aprendizado automático. Mitchell (apud FACELI et al., 2011, p. 3) define o aprendizado de máquina como "a capacidade de melhorar o desempenho na realização de alguma tarefa por meio da experiência". Russell e Norvig (2013, p. 605) dizem ainda que "um agente estará aprendendo se melhorar o seu desempenho nas tarefas futuras de aprendizagem após fazer observações sobre o mundo".

O Aprendizado de Máquina tem sido um tema de grande interesse nas pesquisas de Inteligência Artificial, devido à vasta gama de aplicações que o subcampo possui. Robótica, mercado financeiro, tratamento de imagens, classificação automática e jogos são algumas das aplicações conhecidas do aprendizado de máquina. Ao supor que o propósito da computação está em auxiliar atividades humanas, é natural pensar em um computador que aprenda, pois, o aprendizado é uma atividade humana das mais notáveis. A medida que se avança no campo, cada vez mais a máquina se torna útil ao homem.

A técnica sugere que a partir de um conjunto de dados a máquina possa aprender os padrões existentes acerca desses dados, de modo a descrever algum conceito de sua realidade (Figura 1).

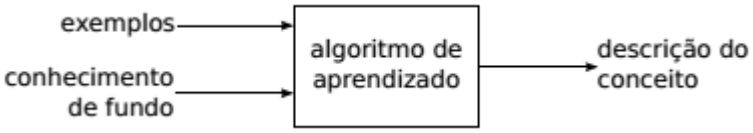

Figura 1: Esquema de aprendizado de máquina (KUBAT et al., 1998)

Uma forma de aprendizado de ampla aplicabilidade consiste em, partindo de um conjunto de pares de entrada e saída, aprender uma nova função que prevê a saída para novas entradas (RUSSELL; NORVIG, 2013).

> 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 apud ARAÚJO, 2014, p. 11)

Partindo de uma perspectiva de classificação automática de música, a técnica é utilizada para aprender mapeamentos de dados musicais que podem ser utilizados para classificar o áudio (MCKAY, 2010). A literatura normalmente refere-se às unidades classificadas como exemplos, amostras, instâncias ou vetores de características, já as categorias nas quais essas unidades são classificadas são em geral chamadas de classes, e as relações entre essas podem ser descritas através de ontologias de classes (MCKAY, 2010).

O processo de classificação requer primeiramente extração de características das instâncias a serem classificadas. As características são informações descritivas do tipo numérico ou nominal que podem ser extraídas de cada instância de áudio e, em seguida, utilizadas num algoritmo de classificação para realização do mapeamento. Algumas vezes essas características são chamadas de atributos, variáveis ou entradas.

3.1.1. Paradigmas de Aprendizado de Máquina

O aprendizado indutivo do qual nos referimos nessa seção pode ser dividido em supervisionado ou não-supervisionado (Figura 2). É no segmento do primeiro que tarefas de classificação automática estão situadas.

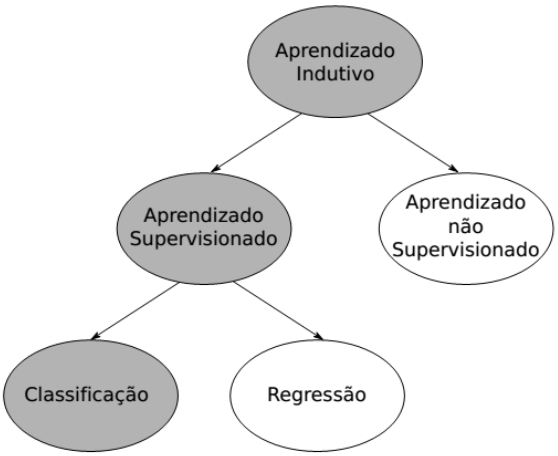

Figura 2: Hierarquia do Aprendizado de Máquina (MONARD; BARANAUSKAS, 2003)

No aprendizado supervisionado, a máquina aprende através de uma base de exemplos anotadas e, já sabendo a classe de cada um, infere o padrão correspondente as suas classes e é capaz de realizar tarefas de classificação ou regressão sobre novos exemplos (MONARD; BARANAUSKAS, 2003). Há diversos classificadores que atuam sob este paradigma, alguns deles, como as redes neurais *feedforward*, aprendem com a retropropagação do erro, ou seja, os pesos dos nós são ajustados de acordo com a diferença entre o resultado obtido na classificação e o rótulo previamente conhecido de cada instância, até haver uma convergência, tornando o ajuste irrisório, uma vez que o erro para de variar significativamente (MCKAY, 2010).

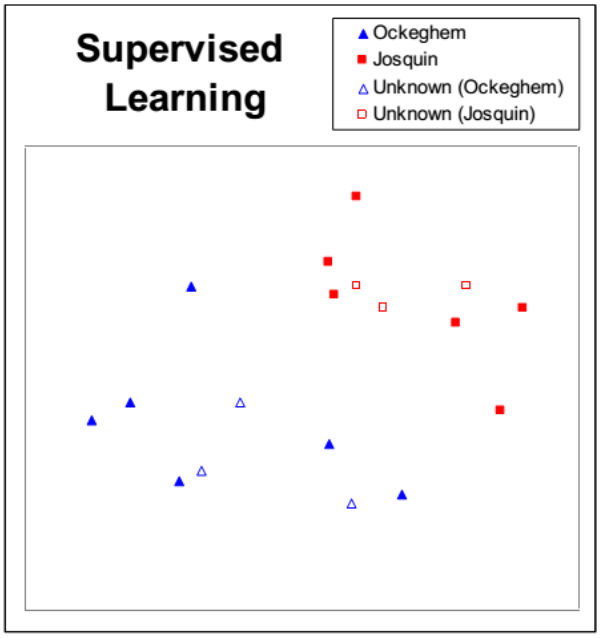

Figura 3: Exemplo de aprendizado supervisionado (MCKAY, 2010, p. 287).

Na Figura 3 várias características são projetadas em duas dimensões para facilitar a visualização. Neste exemplo o modelo é treinado com instâncias rotuladas de composições dos músicos Johannes Ockeghem e Josquin Desprez, em seguida lhes são oferecidas seis novas instâncias não rotuladas, as quais o modelo já treinado infere que três delas são composições de Ockeghem e as outras três de Josquin.

No aprendizado não-supervisionado a máquina aprende conceitos a partir de exemplos sem que esses estejam anotados. Ela procura inferir se esse conjunto de exemplos pode ser agrupado de algum modo, formando *clusters* (MONARD; BARANAUSKAS, 2003). O treinamento de um modelo não-supervisionado consiste na tarefa de inferir para quais valores de características fazem uma instância pertencente a um determinado *cluster*. Cada *cluster* pode ser interpretado como uma classe, contudo não há garantia de que esse auto aprendizado de classes seja útil ou possua significado em domínios específicos (MCKAY, 2010, p.285).

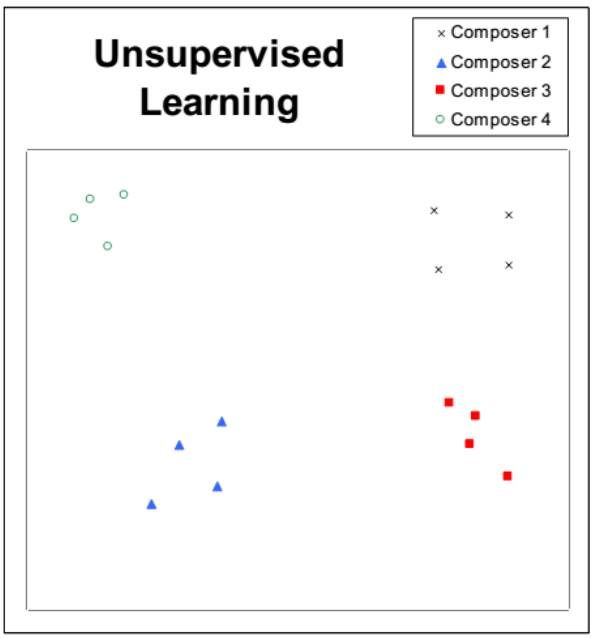

Figura 4: Exemplo de aprendizado não-supervisionado (MCKAY, 2010, p. 288).

A Figura 4 ilustra um modelo que é treinado com dezesseis peças musicais sem rótulo e as separa em quatro grupos baseado nas similaridades e diferenças entre elas. Espera-se que cada grupo deva corresponder a um compositor.

3.1.2. Classificadores

Um componente fundamental na classificação são os algoritmos classificadores. Esses algoritmos são a base para a criação de modelos de classificação treinados.

Existe uma variedade de algoritmos de classificação, que utilizam de diferentes abordagens, seja simbólica, estatística, baseada em instâncias, conexionista ou evolucionista (MONARD; BARANAUSKAS, 2003 apud ARAÚJO, 2014, p. 16). Árvores de decisão, redes neurais artificiais, SVM, k-NN (RUSSELL; NORVIG, 2013) são alguns dos classificadores mais disseminados. Cada qual, procura resolver o problema do aprendizado de um modo diferenciado. O k-NN (*k nearest neighbor*), por exemplo, é usado para quando os valores de classificação são discretos (ARAÚJO, 2014). Classifica um novo exemplo através do cálculo da distância entre seus k ($k \geq 1$) vizinhos mais próximos. Dependendo do vetor de características essa distância pode ser a Euclidiana, de Hamming, etc. (RUSSELL; NORVIG, 2013). De acordo com o valor de k, um limite de decisão é estabelecido e determinará em que classe o novo exemplo estará contido.

É perceptível aqui que quanto maior a quantidade de características, mais complexo e custoso torna-se o problema, ao mesmo tempo, uma quantidade pequena de características pode não carregar a quantidade de informação necessária para a classificação. Seja como for, em geral os algoritmos classificadores apresentam esse mesmo comportamento e tem na seleção de características um gargalo que pode comprometer ou resolver todo o processo.

Após o treinamento de um modelo é importante que se busque um meio de validar a qualidade, pois o baixo erro não é um indicador seguro a respeito do quão bem o modelo treinado faz generalizações de novos exemplos. Por isso é comum reservar entre 10% a 20% das amostras de treinamento para fazer validação (MCKAY, 2010).

A validação cruzada (*cross-validation*) é um método conhecido para avaliar a performance do classificador. Consiste em dividir o conjunto das amostras conhecidas, comumente chamadas de *ground-truth*, em x partes iguais. A validação consiste em fazer x iterações de treinamento e validação, onde cada parte dividida constitui a amostra de validação em alguma iteração (*fold*) e as demais servirão para o treinamento. É sempre relativo determinar qual o melhor valor de x, depende muito da quantidade de *ground-truth* e do poder computacional que se dispõe. Em geral o número de *folds* varia entre cinco e trinta (MCKAY, 2010, p. 293).

Ainda existem algumas variantes do *cross-validation*, podemos citar dentre elas a *leave-one-out* que faz $k = n$ para *ground-truth* pequenos (tamanho n), *bootstrapping*, *5 x 2* (MCKAY, 2010). Também existem outros métodos de validação de modelos de classificação que podem ser encontrados na literatura (ALPAYDIN, 2014).

Ao se executar uma classificação, tem-se como resultado a matriz de confusão, que é uma tabela $n \times n$ (n = número de classes) que dispõe de todos os resultados da execução confrontados com o seu valor real.

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \\ c_{n1} & \cdots & & c_{mn} \end{bmatrix}$$

Figura 5: Definição da matriz de confusão (CORRÊA, 2012)

Cada valor c_{ij} da matriz corresponde ao número de amostras da classe i classificadas como sendo da classe j. Isso implica dizer que a diagonal é constituída dos elementos que representam uma classificação correta.

Para ilustrar melhor esse conceito, tomemos um problema de classificação binária com classes *A* e *B*. A matriz de confusão é constituída dos exemplos classificados como *A* e que de fato são *A* (verdadeiros positivos – VP), dos exemplos classificados como *B* e que de fato são *B* (verdadeiros negativos – VN), além dos exemplos que são *A* mas que foram classificados como *B* (falsos negativos – FN) e dos exemplos que são *B* e que foram classificados como *A* (falsos positivos– FP). A Figura 6 ilustra uma matriz de confusão, onde as linhas significam os exemplos em sua classe exata e as colunas, o resultado da classificação.

```
CONFUSION MATRIX:
   a   b   <-- classified as
726 224 |   a = PISTOLA
214 736 |   b = FOGOS
```

Figura 6: Matriz de confusão para 950 instâncias de arma de fogo e 950 instâncias de fogos de artifício. Fonte: O Autor.

O conceito de matriz de confusão permite-nos conceber algumas métricas para avaliar o poder discriminatório de um dado modelo de classificação, dentre as quais três podem ser destacadas: acurácia, sensibilidade e especificidade.

A Acurácia é a medida obtida pela seguinte equação, onde VP, VN, FP e FN são os elementos da matriz de confusão, e descreve a probabilidade de se acertar na classificação:

$$Acurácia = \frac{(VP+VN)}{(VP+FN+FP+VN)} \quad \text{(Eq. 1)}$$

A Sensibilidade descreve a probabilidade de um legítimo verdadeiro positivo ser classificado como verdadeiro positivo e é a obtida pela seguinte equação:

$$Sensibilidade = \frac{(VP)}{(VP+FN)} \quad \text{(Eq. 2)}$$

A especificidade descreve a probabilidade de um legítimo verdadeiro negativo ser classificado como verdadeiro negativo, sendo obtida pela seguinte equação:

$$Especificidade = \frac{(VN)}{(VN+FP)} \quad \text{(Eq. 3)}$$

3.1.3. Seleção de Características

É o segmento do Aprendizado de Máquina que desperta bastante interesse da comunidade de MIR por ser "a parte mais importante da classificação automática" (MCKAY, FUJINAGA, 2005, p.2).

A motivação para a seleção de características está no fato de que a quantidade de características afeta diretamente o desempenho do classificador. Quanto maior o número de características, maiores o custo computacional e a dificuldade de entender o problema. Somado a isso está também o fato de algumas características não agregarem valor discriminatório (capacidade de distinguir entre classes) ou até mesmo prejudicarem o treinamento de um novo modelo. Baseado nisso, é esperado que se encontre o menor conjunto possível de características sem que se perca qualidade da informação discriminatória. É chamada de entropia o ganho de informação.

Apesar da importância da questão, a seleção de características é muitas vezes realizada de forma empírica ou subjetiva, daí a importância de técnicas que assegurem uma boa escolha.

Existem dois modos de realizar a seleção: através de filtros, onde não se usa a informação do classificador e através de *wrappers* (BLUM; LANGLEY, 1997), onde se usa a resposta do classificador para selecionar o atributo.

No que se refere ao algoritmo de seleção, é comum encontrar duas alternativas, *forward* ou *backward*. A primeira consiste em iniciar sem nenhum atributo o conjunto final e adicionar um a um cada atributo, à medida que se avalia o conjunto, baseado em algum critério. A segunda alternativa consiste em iniciar o conjunto final com todos os atributos e eliminar atributos até que se chegue a uma condição de parada. No entanto, a solução que este trabalho propõe é diferenciada, ao fazer uma busca bidirecional, mesclando as duas estratégias.

A seguir são suscintamente apresentados dois típicos métodos de seleção de características, suas vantagens e desvantagens:

3.1.3.1. Busca Exaustiva

A clássica busca exaustiva consiste em computar todas as possibilidades de combinação entre as características de um conjunto, essa técnica tanto pode ser na direção *forward* quanto *backward*, desde que o critério de parada seja a análise de todas as possibilidades, garantindo a seleção do subconjunto ótimo aplicado àquele conjunto de características analisado. No entanto essa estratégia vem acompanhada de uma enorme desvantagem. O tempo de busca cresce exponencialmente a medida que o conjunto de características aumenta. Isto implica dizer que, para alguns tamanhos de conjunto, a busca exaustiva torna-se inviável.

Na prática a busca exaustiva é utilizada para grupos de atributos muito pequenos, o que restringe muito sua aplicação. O jMIR, software que também realiza classificação no contexto de informação musical, realiza busca exaustiva para até seis características, por exemplo (MCKAY, 2010).

3.1.3.2. Busca Genética

Visando sanar as desvantagens de métodos de busca exaustivos, busca-se heurísticas que possibilitem aproximar-se de uma solução ótima em tempo viável. Uma dessas alternativas é a busca genética (SIEDLECKI; SKLANSKY, 1989). Ela inspira-se na teoria da evolução biológica para gerar indivíduos (subconjunto de características) bons, que podem servir de solução para o problema.

Um candidato a solução na busca genética é representado como um vetor booleano com tamanho igual ao do conjunto de atributos, onde cada espaço corresponde a uma característica desse conjunto. Os espaços do vetor podem conter ‘1’ – indicando que a característica deve ser selecionada, ou ‘0’ – indicando que a característica deve ser descartada. Assim, partindo de um conjunto de soluções candidatas, aplica-se um algoritmo genético (EINBEN, 2003), a fim de melhorar esse conjunto de soluções candidatas.

A busca genética tem a vantagem de ser viável em termos de tempo, mesmo não encontrando a solução ótima, consegue aproximar-se bastante dela, sendo satisfatória. No entanto, está limitada ao conjunto inicial de características, e não consegue conceber nenhuma nova, como faz os métodos de *feature learning*, muitas vezes necessários para uma gama de problemas.

3.2. Feature Learning

Feature Learning ou *Representation Learning* é o nome dado para designar o conjunto de técnicas computacionais que concebem novas características para representação computacional de dados brutos (BENGIO, 2013). A motivação para o *feature learning* se dá porque dados reais, tais como imagem, vídeo, música ou sensores, muitas vezes são redundantes, complexos e altamente variáveis. Como dito anteriormente, é comum projeto *ad-hoc* de novas características para representar bem esses dados, contudo a dificuldade de reuso e os custos envolvidos motivam a busca pelas alternativas computacionais.

Existem dois tipos de *feature learning*, o supervisionado e o não-supervisionado, e podem ser entendidos de maneira puramente similar aos conceitos de aprendizado de máquina. O aprendizado de características supervisionado gera novos atributos a partir de uma base de dados anotada, enquanto o aprendizado não-supervisionado ignora que a base esteja anotada. Dentre os métodos do primeiro tipo podemos citar LDA (*Linear Discriminant Analysis*) (CORRÊA, 2012) e as redes neurais. Dos métodos não-supervisionados encontra-se o PCA (*Principal Component Analysis*) (CORRÊA, 2012) e ICA (*Independent Component Analysis*) (HYVÄRINEN, 2000). Há também aquelas técnicas inspiradas no *Deep Learning* como as máquinas de Boltzmann restritas (COATES, 2010), que podem ser vistas no contexto do aprendizado de características, entretanto o *Deep Learning* é um conceito abrangente e que está mais relacionado a arquitetura de soluções de *Feature Learning*, utilizando redes neurais artificiais com múltiplas camadas (daí o *deep*) na representação dos dados.

3.2.1. Análise de Componentes Principais

Devido à ampla utilização em MIR, a Análise de Componentes Principais (PCA) merece uma atenção particular (MCKAY, 2010). A técnica aprende novas características, através da aplicação de transformações geométricas no espaço de características, que resultam combinações lineares das características originais (CORRÊA, 2012).

PCA também é interessante por reduzir a dimensionalidade do problema. Consiste em fazer um mapeamento entre o espaço de original de características de dimensão *d* e um novo espaço de dimensão *k*, onde $k < d$. Essas dimensões do novo espaço são chamadas de componentes principais. Uma vez ranqueados de acordo com a sua variância, os componentes com menor variância (representam poucos dados na projeção ao novo espaço), podem ser descartados sem muita perda de informação, tornando a dimensão do problema menor. A redução de dimensionalidade acontece devido a possibilidade de as variáveis do espaço original estarem linearmente correlacionadas, já que os componentes principais do novo espaço não estão.

Informações complementares sobre a análise de componentes principais podem ser encontradas em Costa (2001) e Duda (2012).

3.3. Classificação de áudio

Em aplicações que demandam inteligência na manipulação de áudio digital, CAA tem sido amplamente utilizada. A Figura 7 apresenta uma ontologia dos subproblemas e respectivas aplicações na área.

As aplicações são abrangentes e causam impacto em uma variedade de áreas como telecomunicações, segurança, saúde, biologia, música, dentre outras. Essas aplicações abrangem a detecção de atividade vocal (SOHN et al., 1999), reconhecimento de fala (VARILE; ZAMPOLLI, 1997), detecção de idioma (MUTHUSAMY et al., 1994), identificação de usuário por voz (LIPPMANN, 1989), reconhecimento de emoções (KWON, 2003), monitoramento de saúde (cordas vocais, respiração, etc.), localização de fontes sonoras (GUO, 2005), detecção de pestes (POTAMITIS, 2006), taxonomia biológica (LEE et al., 2008), comunicação com animais, áudio monitoramento, reconhecimento de eventos sonoros, como choro de criança (SAHA et al., 2013) e uma gama de aplicações musicais. Estas últimas são promissoras por causa da quantidade cada vez maior de conteúdo musical

produzido e compartilhado. Enquadram-se nessas aplicações o reconhecimento de estilos musicais (GOLUB, 2000), reconhecimento de canções, reconhecimento e modelagem de estilos de composição e de compositores, detecção de instrumentos (EGGINK; BROWN, 2003), separação de fontes sonoras, transcrição musical (acordes, notas, ritmos, ataques, andamento, tonalidade) (KLAPURI, 2007), entre outras (POTAMITIS, 2008).

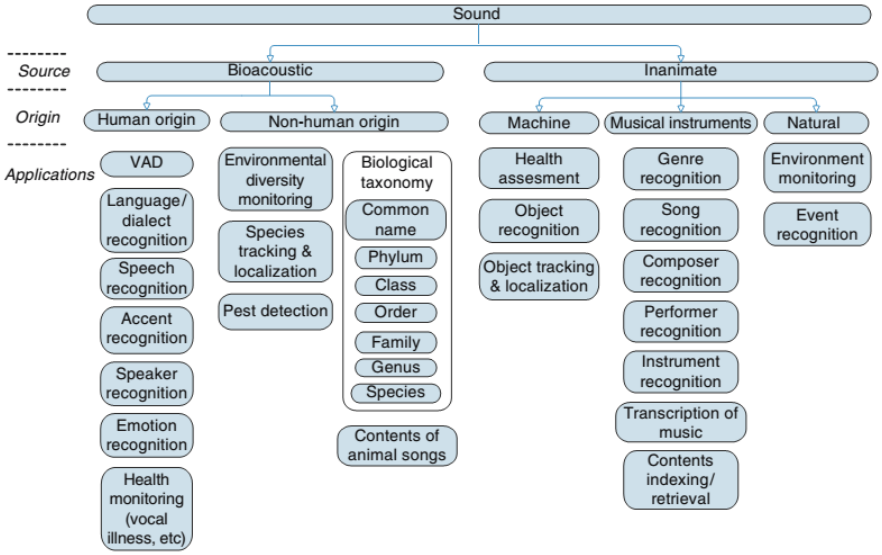

Figura 7: Taxonomia dos sons na perspectiva de aplicações humanas (POTAMITIS, 2008)

No que diz respeito a implementação de um novo classificador, três etapas estão envolvidas: seleção/extração de características de áudio, treinamento e validação. Na seleção de características, o som, que pode ser ruído ou harmônico (música), precisa ser representado computacionalmente por um conjunto de atributos (frequência de onda, histograma de frequências, espectrais, etc.) (PEETERS, 2004). Em métodos tradicionais essa etapa trabalha com características de áudio pré-definidas. Após o pré-processamento das características, são escolhidas aquelas que melhor representam as classes do problema a fim de treinar um novo classificador utilizando alguma base de áudio. Por fim o classificador obtido é validado utilizando uma base distinta da base de treinamento.

A escolha de boas características é determinante para um bom mapeamento entre as instâncias de áudio e as classes do problema. CAA incorpora muitas das técnicas convencionais do segmento de classificação automática. Entretanto algumas estratégias de *Feature Learning* são particulares, uma vez que o objeto tratado, o áudio, possui atributos específicos, que não se aplicam a imagens, textos, ou outra forma de dados.

O classificador por sua vez é escolhido de acordo com a sua eficiência. Porém a determinação de qual é o melhor não é exata, depende intrinsecamente da natureza do problema, bem como dos atributos sonoros envolvidos. Quando se quer buscar o melhor classificador pode-se fazer através da otimização de hiper-parâmetros para algoritmos de classificação, solução investigada por Araújo (2014), que emprega método de SMAC (*Sequencial Model-based Algorithm Selection*) (HUTTER et al., 2011) para definir o melhor classificador e ajuste de seus parâmetros.

Os esforços demandados na seleção de características e classificadores acabam por resultar em duas abordagens distintas:

- *Bag-of-Frame* (WEST; COX, 2014) (AUCOUTURIER et al, 2007): Aqui as amostras de sons são divididas em quadros (*frames*), possivelmente sobrepostos, e para cada quadro é computado um vetor de características. Esses vetores são agregados (daí o *bag*) e utilizados no restante do processo: seleção do subconjunto de características, treinamento e validação do classificador. Atualmente essa abordagem serve para uma variada gama de problemas como: classificação de gênero musical, instrumento, nasalidade, identificação de voz, humor; etc.
- *Ad-hoc* (PACHET; ROY, 2007): Apesar de *Bag-of-Frame* ser eficiente em muitos casos, há uma classe de problemas de menor abstração e, portanto, mais difíceis de resolver. Por exemplo, é fácil distinguir entre Rock e Jazz, no entanto é difícil distinguir, até mesmo para um ser humano, entre subgêneros como Be-bop e Hard-bop (ambos do Jazz) isso porque existe muita semelhança e diferenças sutis entre os estilos. Uma abordagem *Ad-hoc* visa conceber um conjunto novo de características que possibilitem essa distinção. Isso pode ser feito com a aplicação de diversas funções sobre o sinal sonoro (ex. aplicar um filtro no sinal e depois uma FFT no resultado obtido), com a desvantagem que esse processo exige conhecimento especialista e é custoso uma vez que se dá por tentativa e erro. Além do mais o reuso é raro e, portanto, as características analíticas obtidas por esse processo dificilmente servirão para outro problema.

É nesse contexto que as técnicas de geração automática de características ganham importância, principalmente aquelas que utilizam algoritmos evolucionários (RITTHOF, 2002), (MIERSWA, 2005), (PACHET; ZILS, 2003) aproximando-se de soluções ótimas com menor esforço computacional.

3.3.1. Espaço genérico x Espaço analítico

É importante para a compreensão do enfoque tomado por este trabalho ter a clareza do que são características de áudio genéricas (diferente dos Operadores Genéricos da Seção 3.5.2.2) e características de áudio analíticas. Esse conceito é encontrado e explicado com profundidade em Pachet e Roy (2007, 2009).

Numa abordagem de classificação de áudio que utiliza características genéricas (atributos de alto nível pré-concebidos, ex.: *Zero Crossing*, *Root Main Square* – RMS, *Mel Frequency Cepstral Coefficient* – MFCC, etc.) (PEETERS, 2004) não há preocupação com a melhoria dessas características, limitando-se a apenas selecionar aquelas que são mais relevantes através de técnicas de redução de dimensionalidade. O conjunto de características no qual essas técnicas operam pode ser chamado de espaço genérico.

Uma diferença crucial na utilização de características analíticas, em contraste com a abordagem anterior, é que elas consistem em atributos gerados a partir de mudanças nas características genéricas, havendo um interesse em melhorar o desempenho particular das mesmas através de análises. Sendo assim, entre as abordagens, esta última acrescenta somente uma etapa a mais no processo de seleção: a busca heurística no espaço analítico, que visa encontrar novas características de áudio partindo de um conjunto genérico. É comumente denominado Aprendizado de Características o esforço que se faz nesse sentido.

Sendo o espaço analítico complexo, nem todo método exato é viável para efetuar a busca das características ótimas. No aprendizado de características de áudio, a utilização da Computação Evolucionária ainda é muito pouco explorada, contudo o campo é abrangente e seu potencial em CAA vem sendo negligenciado. Talvez isso se deva aos avanços recentes do *Deep Learning* (HUMPHREY et al. 2013), (ZHOU; LERCH, 2015), (BOULANGER et al., 2013). Entretanto resultados que surgem através da exploração do campo são particularmente interessantes por serem específicos ao domínio de conhecimento de MIR, como a abordagem utilizada pelo EDS (*Extractor Discovery System*) (CABRAL et al, 2005), (PACHET; ZILS, 2003), (PACHET; ROY, 2007).

3.4. Computação Evolucionária

Algoritmos evolucionários (AE) são algoritmos estocásticos baseados na metáfora da biologia de como as espécies evoluem (EIBEN; SMITH, 2003) A computação evolutiva ou

evolucionária (CE), por sua vez, é uma subárea da ciência da computação e independe das ciências biológicas, tendo essas últimas apenas servido de inspiração e cedido a terminologia. No entanto ainda há a possibilidade de aplicar CE em pesquisas das ciências biológicas visto que a área originalmente surgiu desse esforço.

A metáfora consiste em modelar um problema (ambiente), representar as soluções candidatas para o problema (indivíduos) e definir um método que mensure a qualidade dessas soluções (aptidão). O algoritmo comporta-se de modo similar ao processo de evolução natural: ao longo de ciclos (gerações) os indivíduos se cruzam (trocam informações), sofrem mutações, passam por seleção natural e ao final do processo, estabelecido por uma condição de parada, sobrevivem os mais aptos.

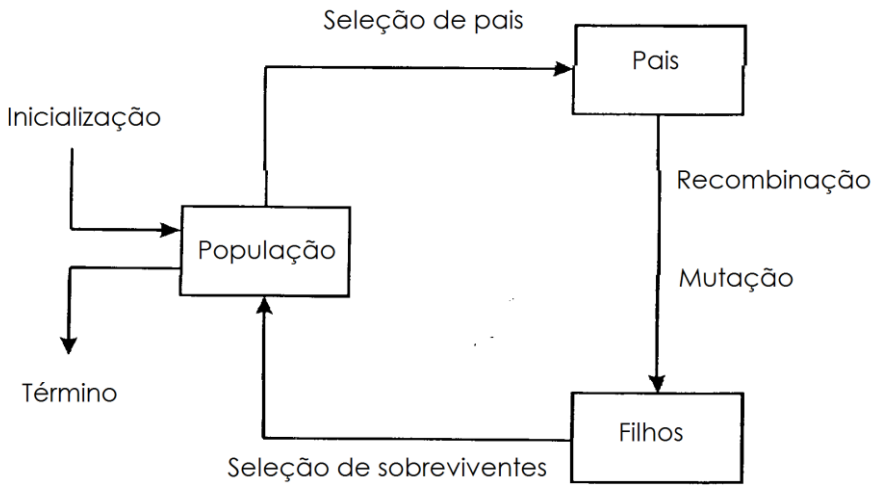

Figura 8: Esquema geral de AE. Eiben e Smith (2003, p.17)

Em geral CE demanda a implementação de três mecanismos para um algoritmo ser considerado evolucionário: a geração da população inicial, a função de avaliação dos indivíduos e os operadores. A seguir são descritos cada um desses mecanismos.

Geração da população inicial: A criação da população inicial geralmente se dá de forma aleatória. Os indivíduos gerados nesse mecanismo serão o ponto de partida para o surgimento de novos indivíduos, portanto a população inicial é possivelmente composta em média pelas soluções de menor qualidade envolvidas no processo.

Função de aptidão: O cálculo da aptidão consiste em um método de determinar quão boa uma solução é. Isso estará diretamente envolvido com a probabilidade do indivíduo

sobreviver e gerar novas soluções. Tipicamente, busca-se maximizar as aptidões presentes nas populações.

Operadores: São os mecanismos que atuam no algoritmo evolucionário para obtenção de uma nova geração. São três os tipos de operadores básicos:

- Seleção – Responsável por selecionar os indivíduos mais aptos para a geração de descendentes e a definição dos sobreviventes;
- Recombinação ou cruzamento – É o operador que implementa o método de reprodução entre indivíduos na geração de novos descendentes. Em geral os indivíduos envolvidos trocam aleatoriamente parte de sua estrutura e originam outros indivíduos, ocasionalmente melhores do que os pais;
- Mutação – É outro operador responsável pela geração de um novo indivíduo, porém similarmente a mutação biológica, provoca mudanças aleatórias na composição do indivíduo original.

Além dos mecanismos acima descritos, alguns parâmetros do algoritmo evolucionário precisam ser ajustados. O tamanho da população, a condição de parada que costuma ser um limite na quantidade de gerações ou na quantidade de avaliações. Também é necessário definir as taxas de recombinação e de mutação, além de outros parâmetros típicos para um método evolucionário específico.

Existem vários tipos de algoritmos evolucionários, dentre eles o mais conhecido é o *algoritmo genético*, que pode ser simples ou multiobjetivo, entretanto outras abordagens existem, como a *estratégia evolutiva* e a *programação genética*, cada qual difere na forma de representação das soluções e nos operadores que satisfazem essas representações (EIBEN; SMITH, 2003). A soluções podem ser representadas como *string* binária, vetores de valores reais, árvores ou máquinas de estado finitos. Os operadores de recombinação e mutação dependem exclusivamente do tipo de representação escolhido, a seleção é o único operador que independe da representação pois baseia-se somente na aptidão.

Dos operadores de seleção podemos destacar o torneio, roleta e *rank*, cada qual tem a característica de atribuir de algum modo probabilidade de seleção de indivíduos baseado em suas aptidões. Isso possibilita as soluções piores serem selecionadas ocasionalmente, implicando na chance de se fugir de ótimos locais.

3.4.1. Comportamento dos Métodos Evolucionários

Golberb (1989) esquematiza a performance de CE na solução de problemas, situando-os como uma opção intermediária entre a busca aleatória por soluções e métodos específicos. Essa visão foi gradualmente mudando por causa das novas teorias e aplicações da técnica e atualmente se busca combinar várias estratégias em um algoritmo híbrido (EIBEN, SMITH, 2003). Na Figura 9 é possível constatar a flexibilidade do método evolutivo para solucionar problemas e sua pouca variação de performance, só é ultrapassado por métodos específicos de solução de problema, o que nem sempre é viável em termos de custo de projeto. AEs portanto ganham importância no cenário.

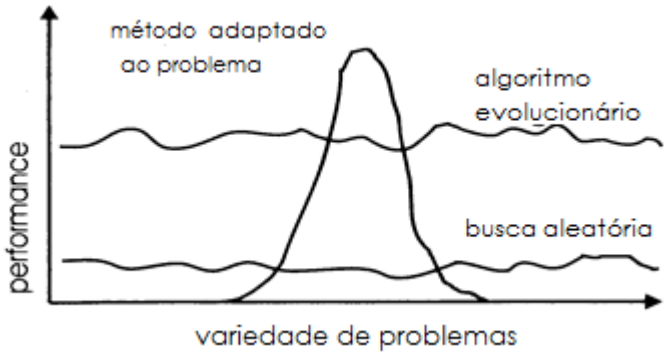

Figura 9: Visão de Golberg sobre a performance dos métodos de solução de problemas (GOLBERG, 1989, apud, EIBEN; SMITH, 2003, p. 32).

Durante as iterações do algoritmo evolutivo é possível perceber a tendência em se atingir os ótimos locais ou globais, isto é, os indivíduos das populações mais jovens tendem a distribuir-se em torno dos ótimos, como ilustra a Figura 10.

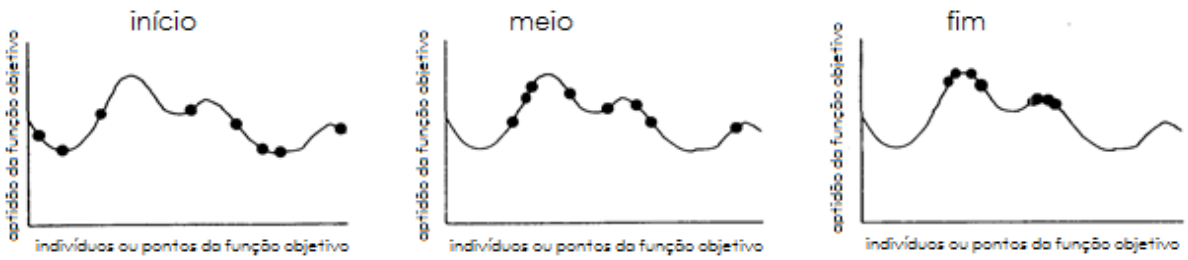

Figura 10: Progresso típico de EA em termos de distribuição populacional (EIBEN; SMITH, 2003, p. 30).

Esse comportamento pode sugerir inicialmente que se aumente a quantidade de gerações na execução do AE, já que quanto mais recente for uma população mais próxima da

solução ótima ela estará. No entanto a vantagem de execuções longas vai até certo ponto. Como mostra a Figura 11, depois da metade das iterações a aptidão dos indivíduos pouco cresce. Dependo do quanto se precise alcançar em termos de aptidão pode ser mais vantajoso não insistir em execuções prolongadas.

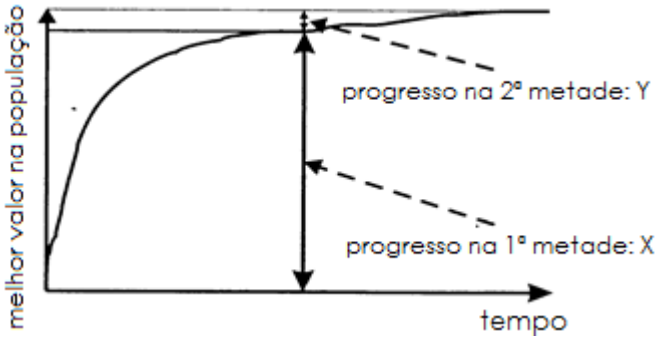

Figura 11: Ilustração de quão longa deve ser ou não ser a execução de um AE (EIBEN; SMITH, 2003, p. 31).

Um outro comportamento relevante diz respeito a vantagem de usar alguma inteligência na criação da população inicial, a fim de agilizar o progresso. Contudo Eiben e Smith (2003) afirmam parecer questionável esse esforço extra, uma vez que o estado da população inicial criada heuristicamente poderia ser atingido em pouco tempo de execução (Figura 12). Evidentemente, para alguma modelagem de problema esse tempo pode não ser tão curto, ou a eficiência da evolução depender disso. É o caso dos *padrões* e *heurísticas* utilizados no sistema EDS apresentado na seção 3.5.2, que são aplicados logo na população inicial, influenciando o sentido da busca.

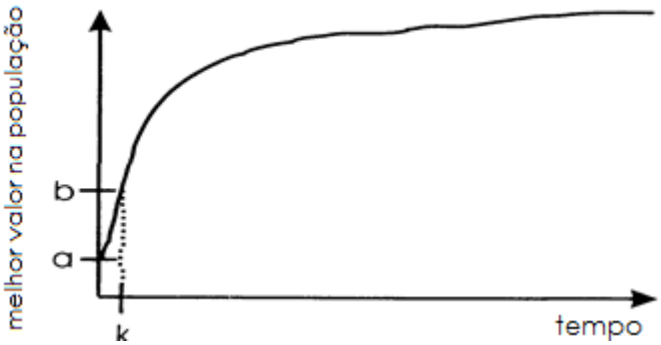

Figura 12: Ilustração do porquê heurísticas de inicialização devem ou não ser usadas como esforço adicional (EIBEN; SMITH, 2003, p. 31).

3.4.2. Paradigmas de Computação Evolucionária

Existem dois tipos de abordagem quanto ao objetivo de um AE. A otimização pode ser mono-objetivo ou multiobjetivo (DEB, 2011).

Algoritmos mono-objetivo geralmente são mais simples de serem implementados. Consiste na ideia de que uma solução candidata deve satisfazer um único objetivo apenas, seja ele minimizar, maximizar ou aproximar algum valor. A avaliação do indivíduo, portanto será melhor quando o mesmo estiver próximo do objetivo, e ele terá boa aptidão. Ao longo das operações genéticas os descendentes tenderão a ter aptidões mais próximas do objetivo. Os Algoritmos Genéticos Simples são exemplos de métodos que implementam esse paradigma.

No paradigma multiobjetivo ocorre que uma solução candidata deve satisfazer mais de um objetivo. É de se esperar que a melhor solução tenha simultaneamente os melhores valores para seus múltiplos objetivos, contudo há situações em que esses objetivos estão em conflito, isto é, à medida que um deles é satisfeito outro deles pode deixar de ser. Assim não é possível designar uma solução única para o problema, mas ainda é necessário definir quando um indivíduo é mais apto que outro. O conceito de dominância (DEB, 2011) nos ajuda a resolver essa questão da seguinte forma (MIETTINEN, 2012):

Definição 1. Uma solução *a* domina uma solução *b*, se ambas satisfazem as seguintes condições:

- A solução *a* não é pior que *b* em nenhum dos objetivos.
- A solução *a* é melhor que *b* em pelo menos um objetivo.

No caso de um indivíduo ser dominado, isso significa que os indivíduos que o dominam terão maior aptidão e, portanto, probabilidade de sobreviver e produzir descendentes. Ao final do processo aproveita-se o conjunto das soluções não-dominadas, o qual designa-se *Fronteira de Pareto* (VAN VELDHUIZEN; LAMONT, 1998) (Figura 13).

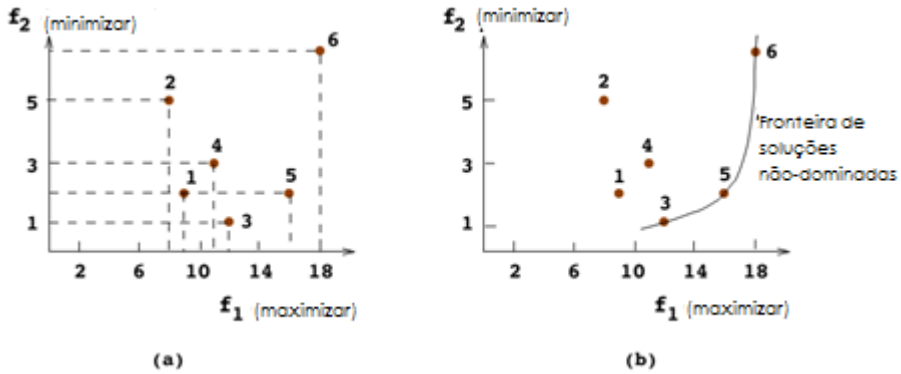

Figura 13: Conjunto de soluções candidatas (a) e a respectiva fronteira de Pareto destacada (b) (DEB, 2011).

São exemplos de algoritmos que implementam o paradigma multiobjetivo: NSGA-II (DEB, 2002) e SPEA2 (ZITZLER et al., 2001), do tipo algoritmo genético e PAES (KNOWLES; CORNE, 1999) do tipo estratégia evolucionária.

3.4.3. Programação Genética

Um tipo de AE particularmente interessante para este trabalho é a programação genética (KOZA, 1992) por causa de sua relação com o tema abordado. É um método evolucionário muito aplicado em atividades de aprendizado de máquina como predição e classificação.

A programação genética difere dos demais tipos pela representação não sequencial das soluções. Utiliza uma estrutura em árvore para constituir um indivíduo e, decorrente disso, implementações específicas para operadores de recombinação e mutação. A recombinação ocorre pela troca válida de sub-árvores entre indivíduos e a mutação consiste em mudança aleatória na árvore. Já a seleção de pais é proporcional à aptidão dos indivíduos, e essas são relacionadas a taxa de sucesso nas atividades de classificação ou predição.

O espaço de busca explorado pelo método é um conjunto de fórmulas (aritméticas, lógicas ou programas), assim com a aplicação das operações genéticas dessas fórmulas tendem a construir outras fórmulas que produzem resultados melhores na resolução do problema. Informações adicionais sobre programação genética são encontradas em Koza (1994) e Kinnear (1994).

3.5. Ferramentas

Visto que já foram apresentados os principais conceitos que envolvem o problema, nesta seção apresentamos algumas das ferramentas que aplicam tais conceitos.

Para o estudo das ferramentas buscou-se levar em conta o quanto elas implementam as teorias, métodos e técnicas da área pesquisada, isto é, ambas possuem mecanismos de treinamento, validação de modelos de classificação, aprendizado de características e métodos evolucionários no processo. Essas ferramentas são: o jMIR (McKAY, 2010), que é uma tecnologia desenvolvida para auxiliar atividades de recuperação de informação musical em geral e, o EDS (PACHET, 2003), que tem o foco mais específico na geração de novas características de áudio.

3.5.1. jMIR

O jMIR é uma suíte de aplicativos em código-aberto Java para uso em pesquisas de MIR. Foi proposta por Cory McKay (2010) a fim de oferecer vasto suporte as mais variadas aplicações de MIR. O jMIR possibilita a manipulação do sinal acústico tanto em formato de sinal digital (Wave, MP3 e etc.) quanto em formato simbólico (MIDI) e serve para as mais diversas aplicações como: mineração de dados de áudio na Web, classificação de áudio, dentre outras. É uma ferramenta completa em termos de MIR e referência na área.

Precisamente a característica de classificação automática de áudio do jMIR nos torna interessados nessa ferramenta. Um problema de CAA é resolvido por ela utilizando dois de seus módulos: jAudio e o ACE. Cada um pode ser entendido pelas seguintes finalidades:

- *jAudio Feature Extractor* (MCENNIS, 2005): Módulo responsável por extrair características das amostras. Traz consigo um variado conjunto de características de áudio e sinal (FFT, MFCC, normalização, compactude, histograma, etc.) e a possibilidade de salvar seus valores, para cada amostra do problema, em formato padrão do jMIR, ACE XML, ou até mesmo no formato ARFF próprio do Weka (HOLMES, 1994). Dentre as demais funcionalidades da ferramenta estão, por exemplo, a possibilidade de gravação do áudio, execução, ajuste da taxa de amostragem, fragmentação e normalização do sinal.

- ACE (*Autonomous Classification Engine*) (MCKAY et al., 2005): É o módulo de Aprendizado de Máquina do jMIR. Responsável por aplicar algoritmos de classificação aos valores extraídos pelo jAudio e associá-los a categorias. Ao todo implementa sete tipos de classificadores (k-NN; Naive Bayesiano; Árvore de Decisão C4.5; Multilayer Perceptron; *Support Vector Machine*; Adaboost e *Bagging* com C4.5 e também três técnicas de redução de dimensionalidade: PCA, busca exaustiva e busca genética.

Este trabalho está interessado em estratégias evolucionárias para o aprendizado de novas características, no entanto a busca genética realizada pelo ACE não corresponde aos nossos requisitos, pois como explicado na subseção 3.1.3.2, só reduz a dimensão do problema, não concebe novas características, logo a técnica a qual estamos interessados é a PCA a qual não se limita em redução de dimensionalidade, mas desenvolve novas características, os denominados componentes principais. Apesar dos variados métodos para o *Feature Learning* nos limitamos ao PCA por dois motivos: ser bem conhecido e ter sido aplicado por uma ferramenta aberta e bastante disseminada na área de computação musical. O jMIR portanto torna-se uma de nossas bases comparativas por implementar métodos de *Feature Learning* convencionais. Sua técnica é reutilizável e escalável para distintos problemas de classificação, dispensa conhecimento especializado na implementação e realiza a busca em um tempo razoável. No entanto, pôde-se verificar empiricamente que, nem sempre a utilização do PCA melhora a eficiência do algoritmo, além disso a técnica não oferece possibilidade de satisfação de restrições.

O jMIR por si mesmo faz uso da abordagem de classificação de áudio aqui denominada *Bag-of-frame*, isto é, sem o uso de qualquer método de aprendizado de característica, a ferramenta divide as amostras de áudio em janelas e para cada uma são computados os vetores de características. Já com o auxílio do PCA, a ferramenta ganha novos aspectos, preservando muito do processo *Bag-of-frame*, porém os atributos acústicos resultantes proveem de um processo analítico.

Foi uma ferramenta importante para este trabalho pois, sendo código-aberto, dispôs de muitas características de áudio codificadas em seu módulo extrator e também algoritmos de classificação, os quais foram aproveitados nas implementações realizadas.

3.5.2. Extractor Discovery System (EDS)

O EDS (*Extractor Discovery System*) foi desenvolvido no laboratório da Sony CSL[1] em Paris e apresenta uma boa proposta no aprendizado de novas características de áudio. O sistema da Sony utiliza técnicas de programação genética (KINNEAR, 1994), porém é uma tecnologia proprietária e pouco se pode saber sobre os detalhes de implementação da mesma além do encontrado na literatura acadêmica.

Nesta subseção é apresentado a técnica usada para exploração do espaço analítico de características acústicas tomando como base a estratégia adotada pelo EDS.

O sistema é um esforço *Ad-hoc* de geração de características. Cada atributo de áudio utilizado nas classificações realizadas pela ferramenta provém de um processo de tentativa e erro ao longo da execução de seu algoritmo genético.

Partindo de um conjunto finito de operadores elementares, ex.: Matemáticos (adição, multiplicação por escalar, média, etc.); de processamento de sinais (Transformada de Fourier, filtros, *spectral centroid*, etc.); específicos para música (*Pitch or Ltas*), busca-se combinar esses operadores de forma válida a fim de se obter expressões, como apresenta a Figura 14, onde (A) pode ser entendida como a média dos cinco primeiros coeficientes ceptrais (MFCC) da derivada do sinal 'x'. (B) Valor médio da energia (RMS) de uma sucessão de quadros (*split*) de 32 amostras do sinal normalizado 'x'. A aplicação desses operadores define uma nova característica de áudio, também chamada de função.

```
(A) Mean(Mfcc(Diferentiation(x),5))
(B) Median(Rms(Split(Normalize(x),32)))
```

Figura 14: Exemplo de expressões (PACHET; ROY, 2007).

3.5.2.1. Regras de Tipagem e Heurísticas

A criação de funções é controlada por dois mecanismos: tipagem e heurística. As regras de tipagem são responsáveis por controlar a combinação dos operadores para que os tipos de dados de *input* e *output* se encaixem adequadamente. Por exemplo, um FFT receberá como entrada um sinal acústico (tempo/amplitude) e transformará em uma saída do tipo frequência/amplitude, ou vice-versa, a operação de média, por sua vez, recebe qualquer sequência de informação e a transforma em um escalar. EDS pode dessa forma gerar:

[1] http://www.csl.sony.fr/

Fft(HpFilter(x)), mas não *Fft(max(x))*[2], uma vez que no primeiro a saída de *HpFilter* é compatível com a entrada de *Fft*, que deve ser um sinal acústico (tempo/amplitude), já no segundo caso o operador *max* recebe uma sequência de valores e retorna o valor máximo entre eles (um escalar), mas *Fft* não trabalha com somente um valor, portanto a expressão *Fft(max(x))* por estar errada não pode ser gerada pelo sistema. As heurísticas representam o conhecimento especializado dos profissionais em processamento de sinais, permitindo apostar a priori em algumas funções interessantes, sem que se precise calcular o seu desempenho. Também é característico do mecanismo impedir formações de funções desnecessárias, como *fft(fft(fft(fft(x))))* (PACHET; ROY, 2007), (PACHET; ZILS, 2003).

3.5.2.2. Operadores Genéricos e Padrões

O sistema de tipagem possibilita a criação de "operadores genéricos" (o conceito é diferente de características genéricas da abordagem *Bag-of-frame*). Esses operadores genéricos são expressões regulares que suportam um ou mais operadores e formam funções cujo tipo de saída esteja forçada (PACHET; ZILS, 2003). Por exemplo: O operador genérico "*_a (x)" aponta para uma combinação composta por vários operadores cujo tipo de saída é um escalar "a", sendo *Square (Mean (x))* uma função válida para satisfazer o operador, pois os operadores *Mean* e *Square* tem saída do tipo "a" ao mesmo tempo que *Square*, por admitir um valor como *input*, pode ser combinado com *Mean*.

Ao todo são três os operadores genéricos que EDS implementa:

- "?_T" aponta para 1 operador cujo tipo de saída é "T".
- "*_T" aponta para vários operadores cujos tipos de saída são todos "T".
- "!_T" aponta para vários operadores cuja apenas a saída final é do tipo "T"

Isso possibilita definir padrões de funções como: " ?_a (!_Va (Split (*_t:a (SIGNAL))))" que incentiva a criação das seguintes funções dentre outras (PACHET; ZILS, 2003):

- *Sum_a (Square_Va (Mean_Va(Split_Vt:a (HpFilter_t:a(SIGNAL_t:a, 1000Hz), 100))))* ou
- *Log10_a (Variance_a (NPeaks_Va(Split_Vt:a (Autocorrelation_t:a(SIGNAL_t:a), 100), 10)))*.

[2] A lista de tipos de dados, operadores e outras características do EDS podem ser encontrados no Apêndice A.

3.5.2.3. Mecanismos do Algoritmo Genético

Através das operações de recombinação, mutação e seleção do algoritmo genético, o sistema busca "evoluir" uma população de indivíduos a fim de que seja encontrado aqueles mais aptos a sobreviver. No contexto dos problemas abordados neste trabalho, isso implica dizer que, a partir de uma população aleatória de características de áudio (indivíduos), aplicando-se sucessivas vezes operações genéticas, é possível melhorar a qualidade dessas expressões mediante objetivo.

Veremos agora quais são essas operações, através dos exemplos que seguem. Partindo da expressão *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))* o sistema executa as seguintes operações:

- Clonagem – Muda os parâmetros de uma função.

  Ex.: Antes: *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*

  Depois: *Sum (Square (Mean (Split (HpFilter (SIGNAL, **430Hz**), **65ms**))))*
- Mutação (cabeça ou cauda) – Muda inteiramente parte da cabeça ou cauda da expressão.

  Ex.: Antes: *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*

  Depois: ***Max (Max** (Split (HpFilter (SIGNAL, 430Hz), 65ms))))*
- Deleção – Retira uma função qualquer da expressão.

  Ex.: Antes: *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*

  Depois: *Sum (Mean (Split (HpFilter (SIGNAL, 500Hz), 50ms)))*.
- Adição – Adiciona na cabeça da expressão uma função qualquer.

  Ex.: Antes: *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*

  Depois: **Log** (Sum (Square (Mean (Split (HpFilter (SIGNAL, 500Hz), 50ms)))))
- Substituição – Troca uma função por outra qualquer de tipagens equivalentes.

  Ex.: Antes: *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*

  Depois: *Sum (Square (Mean (Split (**LpFilter** (SIGNAL, 500Hz), 50ms))))*
- Cross-overs – Recombina dois indivíduos para gerar um novo.

  Ex.: Um possível resultado do cruzamento entre *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))* e a expressão *Mean (**Autocorrelation (SIGNAL)**)*,

pode ser os dois indivíduos apresentados a seguir: 1º - *Sum (Square (Mean (Split (**Autocorrelation (SIGNAL)**, 50ms))))* e 2º -*Mean (HpFilter(SIGNAL, 500Hz))*

Não é o foco deste trabalho explicar o significado de cada operador utilizado nos exemplos que se seguem, entretanto, muitos deles podem ser encontrados em Peeters (2004).

Além dos operadores utilizados, o EDS define outros aspectos importantes tais como: os meta-parâmetros do algoritmo (quantidade de gerações, tamanho da população e etc.), a definição das características inerentes ao problema como a combinação correta de funções elementares de forma que a expressão gerada seja uma expressão válida. Por fim, com a necessidade de determinar quando uma característica em questão é melhor do que outra, o sistema define o *fitness*, para tal se utiliza *Fisher Discriminant Ratio* (FISHER, 1936), uma técnica de análise de discriminância, ou usa-se alguma instância de classificador, assim de acordo com a taxa de acerto do mesmo é atribuída uma aptidão a nova característica concebida.

3.5.2.4. Algoritmo Global

A execução do EDS pode ser abstraída em duas partes: O aprendizado de novas características através do algoritmo genético e a seleção das características relevantes resultantes desse processo. A Figura 15 ilustra bem isso, enquanto a Figura 16 apresenta em pseudocódigo o algoritmo global implementado pelo EDS.

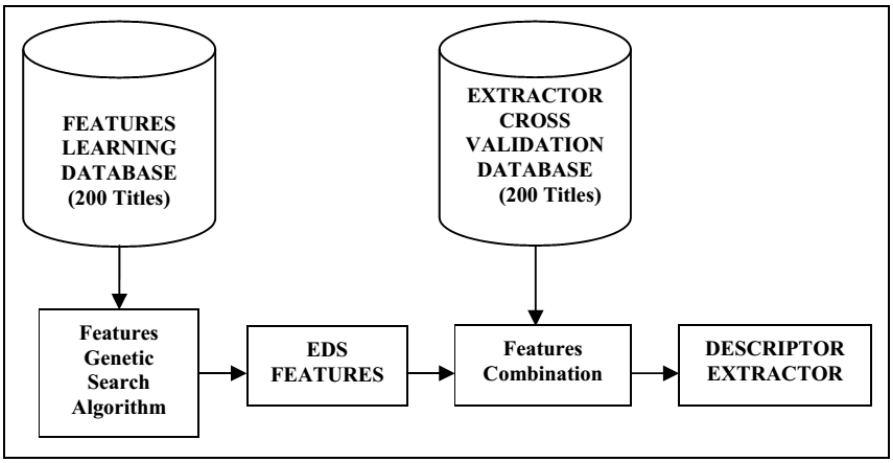

Figura 15: Arquitetura global do Extractor Discovery System (PACHET; ZILS, 2003).

```
- Constrói 1ª população P0,  pela computação ramdômica de N funções
(composição de operadores), cujo tipo de saída é compatível com
o tipo de dado.
- Loop Inicial:
    - Computação das funções para cada audio da base.
    - Computação do fitness de cada característica.
    - SE (fitness >= limiar) ou (numero máximo de interações for atingido),
        PARE e RETORNE as melhores funções
    - Selecão de funções, cruzamento e mutações, para produzir uma nova
    população Pi+1
    - Simplificação da população Pi+1 com regras de reescrita
    - RETORNE para Loop Inicial
```

Figura 16: Algoritmo global do EDS. Traduzido de Pachet e Zils (2003, p.8).

De acordo com os critérios de satisfação para a solução esperada (Capítulo 2, Seção 2.3) o EDS melhora os resultados da classificação. Possui operações evolucionárias e mecanismo de avaliação de fáceis implementações. O algoritmo serve para diferentes problemas de classificação. Apesar de requerer conhecimento de domínio específico na fase de implementação, o EDS economiza conhecimento especializado na execução. Além de explorar em tempo hábil o espaço analítico de funções. No entanto falha no critério de adequação, uma vez que não possibilita o desenvolvimento de soluções com satisfação de restrições.

Sobre essa classe de problemas de CAA que requerem satisfazer restrições uma possível solução seria o emprego de algoritmos genéticos multiobjetivos, onde se pretende evoluir a eficiência da classificação paralelamente à evolução da adequação da condição a ser satisfeita. Por exemplo, em um problema de identificação de voz no controle de acesso de um dado sistema é, até certo ponto, tolerável que o controle de acesso falhe na identificação da voz de uma pessoa cadastrada, negando-lhe permissão (falso negativo), mas é intolerável que o sistema permita acesso a pessoa não cadastrada (falso positivo) por falha em seu processo de identificação de voz. Um algoritmo genético simples não possibilita melhorar dois aspectos importantes de um mesmo problema, nesse caso deve-se escolher direcionar a busca, ou para aumentar a taxa de acerto ou para satisfazer a restrição. A estratégia multiobjetivo, porém, possibilita que os dois aspectos sejam perseguidos durante o processo. Assim, além de buscar melhorar a taxa de acerto da classificação, também se faz necessário diminuir a ocorrência de falsos positivos ou falsos negativos, dependendo da natureza do problema. É nisso que se torna mais adequado, para essa classe de problemas, o aprendizado de características através de um processo de otimização multiobjetivo.

Uma outra lacuna deixada pelo EDS ao usar uma técnica mono-objetivo está no fato de que características avaliadas isoladamente e que tem baixa aptidão não sobrevivem ao longo das iterações, o que é natural, porém existe a possibilidade dessas características menos aptas, se combinado com outras, levarem a melhores resultados na classificação. Dessa forma, para não desperdiçar características com baixa aptidão pode-se salvar toda a lista gerada de características ao longo das iterações do algoritmo genético e, aplicando-lhes uma das técnicas de seleção de característica, selecionar aquelas mais aptas a produzirem melhores resultados. No entanto esta abordagem não permite que características de baixa aptidão interfiram na evolução de outras durante as iterações do algoritmo. Dessa forma é necessário garantir sua sobrevivência, o que sugere que a medida de aptidão de cada indivíduo não seja a única coisa a ser otimizada pela descoberta de novas características.

A falta de soluções abertas desse tipo constitui um outro motivador para este trabalho. O EDS é um sistema proprietário. Propor uma solução que esteja disponível para a comunidade é algo importante. Além do mais, entendemos que é possível simplificar o procedimento, eliminando a etapa de combinação das características após a busca genética, ilustrada na Figura 15.

Em suma *Extractor Discovery System*, solução muito elegante para o problema de geração *Ad-hoc* de características analíticas, pode ser melhorada a partir dos seguintes motivadores:

- Não empregar heurísticas que visem satisfazer restrições do tipo falso positivo ou negativo;
- Não preservar, durante a programação genética, os atributos de áudio com baixa aptidão, mas que potencialmente influenciam num resultado global;
- Ser código-fechado;
- É possível simplificar o processo, eliminando uma de suas etapas.

Por esses pontos nos propomos oferecer, além do estado da arte, uma solução para classificação automática de áudio, que utilizem atributos analíticos resultantes de uma busca multiobjetivo, a fim de satisfazer os pontos elencados.

4. Proposta de solução

4.1. Princípios da solução

Apesar das técnicas do estado da arte possuírem bons resultados, nós consideramos possível melhorar utilizando técnicas de otimização multiobjetivo, seja na eficiência da classificação, aumentando sua acurácia, seja na natureza do problema que requer múltiplos objetivos, ou simplificando o processo. Neste Capítulo é apresentada uma proposta para os problemas apresentados até agora.

A diferença fundamental entre as otimizações mono-objetivo e multiobjetivo, no que concerne às soluções existentes e a proposta por este trabalho, é que a otimização mono-objetivo leva em conta apenas os aspectos isolados da característica de áudio, enquanto a otimização multiobjetivo pode levar em conta a relevância da característica ao estar contida em um conjunto. Esse é um importante detalhe, pois determina como uma população de indivíduos evolui ao longo das gerações do algoritmo. Quando mono-objetivo, as características com baixa aptidão não sobrevivem ao longo das iterações, no entanto existe a possibilidade dessas características, se combinadas com outras, levarem a resultados globais melhores. Uma abordagem multiobjetivo não permite que características sejam descartadas somente com base em sua baixa aptidão individual. Ao sobreviverem, poderão contribuir com resultados coletivos melhores.

Nas definições do EDS cada indivíduo representa uma característica de áudio. Já em nossa proposta representamos o indivíduo como um conjunto de características de áudio: o indivíduo é o conjunto de funções e seus valores de aptidão para um problema de classificação arbitrário (Figura 17). O *fitness* isolado, o qual é atribuído a cada expressão de forma similar ao EDS e o *fitness* coletivo, o qual é atribuído ao conjunto de expressões. A *n-ésima* expressão tem um fitness de 0,32, (32% é sua taxa de acerto) o qual pode ser facilmente superado por outros, entretanto se combinada com as outras ela contribuirá para uma taxa de

acerto coletiva de 0,88 (88%), que é um valor melhor do que todas as outras medidas de *fitness* isoladas, superando inclusive a da melhor característica, 73%. Essa medida adicional de aptidão acaba sendo muito relevante para a evolução das características, uma vez que a característica ruim não persistiria isoladamente, mas pelo fato de fazer parte de um grupo ela sobrevive e contribui com a evolução do grupo.

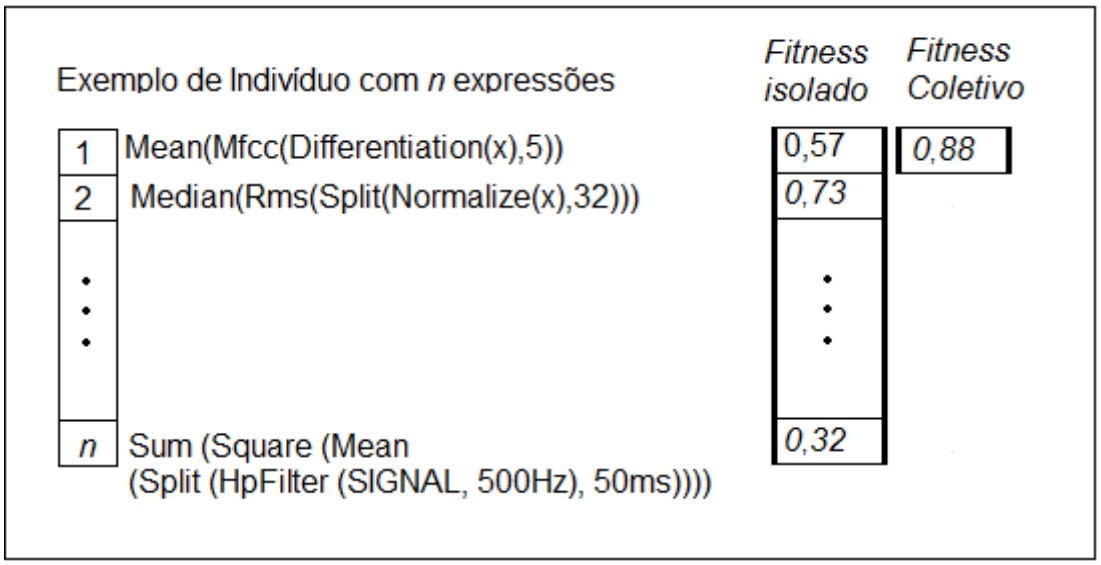

Figura 17: Exemplo ilustrativo de indivíduo de solução multiobjetivo e suas medidas de aptidão (entre 0 – 1). Fonte: O Autor.

Partindo da nova forma de representação de um indivíduo, concebemos mais operações de recombinação e mutação que conciliavam com a forma de representação do indivíduo, como por exemplo, cruzar expressões entre um indivíduo e outro ou mesmo remover ou adicionar expressões. Assim adaptamos as operações mono-objetivo da seguinte forma:

A partir de um indivíduo com três genes (expressões): *1 - Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*; *2 - (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))) e 3 - Mean (Autocorrelation (SIGNAL))*, o sistema executa as seguintes operações:

- Mutação (cabeça ou cauda) – Muda inteiramente parte da cabeça ou cauda do indivíduo trocando por outras expressões quaisquer. Ex.:

  - *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*

  - ***Median(FFT (SIGNAL, 500Hz), 50ms))***

  - ***RMS (Normalize (SIGNAL))***

- Deleção – Retira uma expressão qualquer do conjunto. Ex.:

  - *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*

  - *Mean (Autocorrelation (SIGNAL))*

- Adição – Adiciona uma expressão qualquer ao conjunto. Ex.:

  - *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*

  - *Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms)))*

  - *Mean (Autocorrelation (SIGNAL))*

  - ***RMS (Normalize(SIGNAL))***

- Substituição – Troca uma expressão por outra qualquer. Ex.:

  - ***RMS (Normalize(SIGNAL))***

  - *Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))*

  - *Mean (Autocorrelation (SIGNAL))*

- *Cross-overs* – Recombina dois indivíduos para gerar um novo. Ex.: O cruzamento com o indivíduo de dois genes: *A* - *RMS (Normalize(SIGNAL))* e *B*- *Median(FFT (HpFilter(SIGNAL, 500Hz), 50ms))))*, pode resultar em dois novos filhos:

  Filho 1:

  A - ***RMS (Normalize(SIGNAL))***

  2 - *Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*

  3 - *Mean (Autocorrelation (SIGNAL))*

  Filho 2:

  1 - *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*

  B - ***Median(FFT (HpFilter(SIGNAL, 500Hz), 50ms))))***

Além disso, as operações de mutação da abordagem EDS puderam continuar sendo utilizadas também, como um tipo particular de mutação no novo algoritmo, uma vez que elas são feitas para agir em uma única expressão.

Concebemos também uma restrição. Constatou-se que ao longo das gerações os indivíduos tediam a ficar cada vez maiores, pela ação das recombinações, tornando a solução lenta e custosa. Procuramos contornar essa tendência possibilitando a atribuição de um limite no tamanho dos indivíduos, sendo esses penalizados quando porventura ultrapassassem esse

limite. Dessa forma o método tende a evoluir mantendo o tamanho dos indivíduos até um certo limiar escolhido pelo usuário.

No que compete à medição da aptidão, utilizamos o mesmo mecanismo de avaliação de indivíduo da implementação anterior, a instância de um classificador para verificar os resultados de aptidão. Dois ou mais objetivos podem ser definidos para o problema. Um deles sendo necessariamente "aumentar a taxa de acerto do conjunto". E os demais sendo qualquer um dentre: "aumentar a taxa de acerto da melhor característica do conjunto", "aumentar o *fitness* da 'pior' característica do grupo", "aumento da distância entre os *fitness* da melhor e pior característica do grupo", "diminuição da distância entre os *fitness* da melhor e pior característica do grupo", ou restrições como, "diminuição do número de falsos positivos e/ou negativos na avaliação coletiva", dentre outros objetivos que podem ser concebidos especificamente para cada instância de problema de CAA.

Como trata-se de objetivos que podem estar em conflito, o algoritmo resulta numa fronteira de Pareto (Figura 18). O resultado que interessa, no entanto, não é sempre a fronteira de soluções não dominadas, mas o indivíduo (conjunto de características de áudio) que melhor se adequa à natureza do problema, ou seja, caberá ao profissional decidir, a partir dos resultados alcançados, qual a solução da fronteira mais adequada para seu problema.

```
1  - Constrói 1ª população P0,  pela computação ramdômica de N indivíduos
2  (conjunto de características analíticas)
3  - Loop Inicial:
4      - Computação das funções/indivíduo para cada audio da base.
5      - Computação dos objetivos para cada indivíduo de Pi.
6      - SE (numero máximo de interações for atingido),
7          PARE e RETORNE a Fronteira de Pareto
8      - Selecão de indivíduos, cruzamentos e mutações, para produzir uma nova
9      população Pi+1
10     - Pi <- Pi+1
11     - RETORNE para Loop Inicial
```

Figura 18: Algoritmo global multiobjetivo. Fonte: O Autor.

Se tratando de um problema de satisfação de restrições, na diminuição de falsos positivos e/ou negativos, o resultado deve ser a fronteira de Pareto, ficando a critério do desenvolvedor definir qual dos indivíduos não dominados deve ser aproveitado como solução de seu problema. Mas no caso de o problema não possuir restrições, o que nos interessa é somente a acurácia. Portanto, se o melhor valor for alcançado pelo 1º objetivo (*fitness* coletivo) de um indivíduo então ele deverá ser aproveitado no modelo de classificador, mas se o melhor valor for alcançado pelo 2º objetivo (*fitness* isolado) de algum gene (característica

de áudio), então apenas a função do conjunto que detém esse valor é quem deverá ser aproveitada.

4.2. O Protótipo

Com o projeto dos aspectos apresentados anteriormente, a solução proposta abrange todos os pontos não abrangidos pelo EDS o qual elencamos no Capítulo anterior, subseção 3.5.2, resta, no entanto, saber o quão efetiva pode ser uma implementação que utilize nossa abordagem. Para isso foi desenvolvido um protótipo computacional o qual chamamos de ExpertMIR. Nesta Seção são apresentados os detalhes da implementação, assim como, as tecnologias utilizadas, a arquitetura do sistema, linguagens e algoritmos utilizados.

4.2.1. Arquitetura e fluxo

Inicialmente o ExpertMIR foi desenvolvido com um sistema EDS-*like*, buscando implementar as características do *Extractor Discovery System* a fim de melhor estudar o comportamento de tal abordagem, isso foi necessário devido à solução da Sony ser fechada, ficando impossível de se experimentar. Em um segundo momento foi proposta uma evolução desse sistema, utilizando algoritmos multiobjetivo a fim de alcançar alguma melhora.

Assim, a solução foi construída de forma a operar em duas perspectivas: na otimização mono-objetivo e multiobjetivo de características de áudio analíticas. Ambas operando sobre o mesmo espaço analítico, tendo em comum os operadores e padrões utilizados para gerar os indivíduos nos algoritmos genéticos.

O ExpertMIR poderá explorar características que seguem os seguintes padrões:

- “ !_a (SIGNAL)” – qualquer função cuja saída final é um valor;
- “ Mean(!_t:a (SIGNAL))” – a média de qualquer função cuja saída esteja no domínio do tempo.
- “ Mean(!_f:a (SIGNAL))” – a média de qualquer função cuja saída esteja no domínio da frequência.

Também foram utilizados o mesmo arcabouço tecnológico para ambas: o sistema extrator, o algoritmo classificador dentre outros.

A Figura 19 permite visualizar como os módulos do sistema foram projetados para interagir.

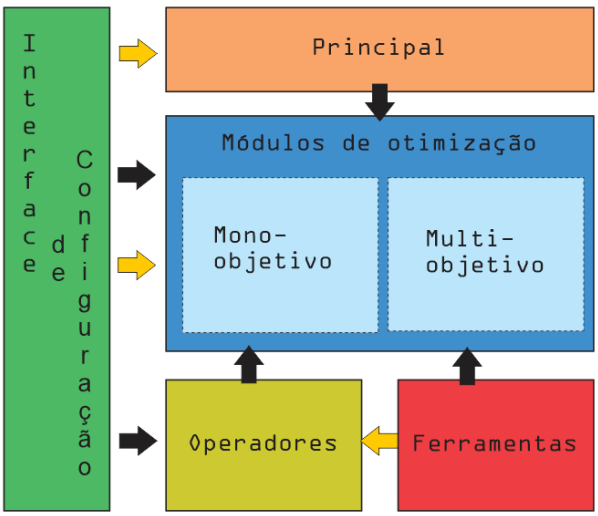

Figura 19: Arquitetura dos módulos do protótipo. Fonte: O Autor.

Interface de configuração: Conjunto de classes que permitem ao usuário não especialista configurar um novo problema (classes, base de áudio, restrições ou objetivos e etc.), definir a perspectiva na qual o sistema operará, dispor de arquivos e relatórios para registrar o resultado do processo.

Principal: Reúne todas as possibilidades de execução da ferramenta, controlando o que pode ser feito e como dever ser feito. Aqui são instanciados os problemas e são realizadas as chamadas dos algoritmos evolucionários, fazendo o devido controle.

Módulos de otimização mono e multiobjetivo: São os principais módulos do sistema, responsáveis por executar o *feature learning*, nunca são executados ao mesmo tempo. Devido à complexidade e custo de processamento dos algoritmos genéticos, é importante que cada abordagem possa ser executada de maneira separada. Além de não sobrecarregar o recurso computacional também permite melhor avaliar o desempenho de cada algoritmo.

Pacote de operadores: Aqui são implementados cada operador matemático, de processamento de sinais e específicos de música, os quais formam o conjunto de operadores que determinam o espaço analítico de expressões que as buscas genéticas irão percorrer. Esse módulo do sistema é escalável, podendo, sempre que se achar necessário adicionar tantos operadores quanto se queira, bem como ativar e desativar um operador.

Pacote de ferramentas: Para executar bem os algoritmos genéticos de acordo com a nosso interesse e necessidade foi necessário desenvolver um pacote de suporte aos algoritmos, neste pacote é possível recorrer a diversas ferramentas inerentes ao processo, como: algoritmo classificador (utilizado no cálculo do fitness de indivíduos), conversor de áudio (transformando arquivos de áudio digital em objetos do sistema), extrator de características (extrai os valores a partir das amostras de áudio convertidas), validador de características (responsável por verificar se a característica encontrada na busca está sendo corretamente construída), dentre outras.

Para compreender as diversas transformações dos dados ao longo de uma execução do sistema, o diagrama do fluxo da execução é apresentado na Figura 20.

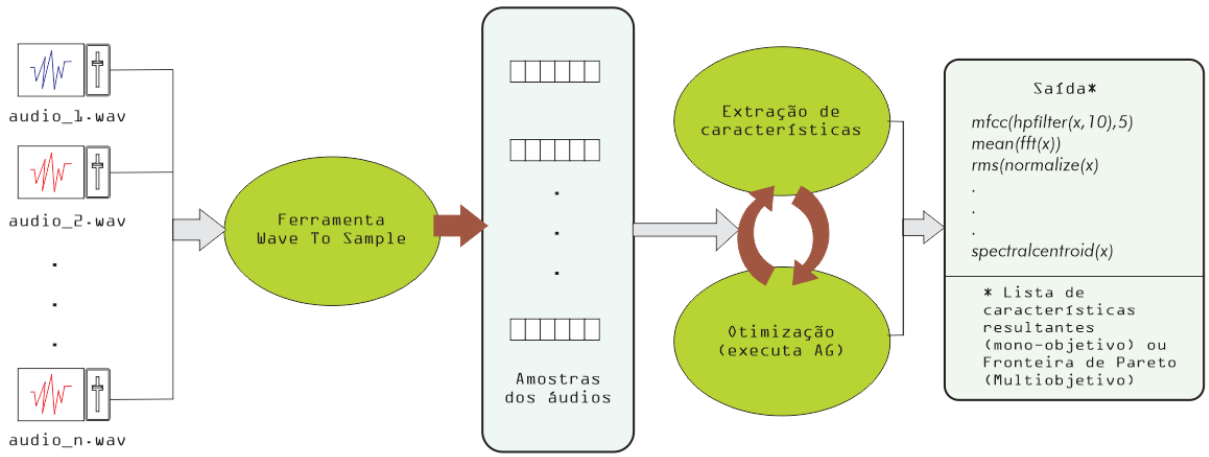

Figura 20: Fluxo dos dados manipulados na execução do protótipo. Fonte: O Autor.

Ao se buscar novas características de áudio a solução requer uma base de áudio anotada, uma vez que o aprendizado se dará por supervisão. Arquivos no formato WAVE representando instâncias das classes de um dado problema são pré-processados e representados como objetos do sistema. Isso se faz necessário para tornar o processo mais rápido, uma vez que, ao longo das interações genéticas do aprendizado, as informações contidas nas amostras serão requisitadas frequentemente.

O segundo momento é a etapa mais importante do processo e ocorre com dois módulos trocando informações constantemente. O módulo de otimização, a cada geração de uma população, requisitará ao extrator de características que calcule o valor das características geradas (indivíduos da população naquela geração) para cada amostra da base. Essa informação deve ser utilizada no cálculo do *fitness* do indivíduo, que utiliza uma instância de

classificador para realizar validação cruzada, na base de dados anotada. Isso permite descobrir o quanto as características encontradas são adequadas para a solução do problema.

Por fim o resultado do processo é o conjunto de características aprendidas pelo algoritmo mono-objetivo, ou o conjunto de soluções não dominadas (fronteira de Pareto) do algoritmo multiobjetivo.

4.2.2. Tecnologias utilizadas

O ExpertMIR foi desenvolvido em Java 8 através do Eclipse IDE. A escolha se deu devido às boas opções de ferramentas complementares nessa linguagem e que podiam ser aproveitadas pelo ExpertMIR. Abaixo estão relacionadas essas ferramentas:

- jAudio/jMIR versão 2.4: Utilizado para auxiliar na extração de características por possuir várias delas já implementadas.
- ACE/jMIR versão 2.4: Necessário para instanciar o classificador utilizado no cálculo do *fitness*.
- jMetal versão 4.5 (DURILLO; NEBRO, 2011): Framework Java para desenvolvimento de aplicações multiobjetivos com meta-heurísticas. O jMetal também dá suporte a heurísticas mono-objetivo e foi fundamental para abstrair atividades específicas dos algoritmos evolucionários. Cabendo a nós o projeto de representação de indivíduo, a definição dos operadores e do método de avaliação dos indivíduos. O jMetal ainda dispõe de versatilidade na alternância de seus algoritmos (NSGA II, SPEA, etc.), podendo estes serem facilmente substituídos e testados para quando se queira inferir o impacto de cada um na qualidade da geração das características.

5. Avaliação

Este capítulo tem como finalidade apresentar e discorrer a respeito dos resultados obtidos com a experimentação da solução proposta. A seguir é descrito como se deu a realização dos testes e a validação da hipótese de que "é possível melhorar o desempenho de um conjunto de características de áudio utilizado em atividades de CAA através da otimização multiobjetivo das características desse conjunto" (Capítulo 1, Seção 1.2).

5.1. Metodologia

O problema da CAA para classificar áudios muito semelhantes, necessita de um conjunto de características de tal modo que esses áudios possam ser diferenciados da melhor forma. Como apontado em nosso estado da arte, técnicas de otimização mono-objetivo tem sido promissoras na geração de características analíticas. Entretanto técnicas multiobjetivo, que não tem sido propostas na literatura, sugerem uma melhoria na eficiência do aprendizado de características, além de possibilitar melhor adequação para problemas com restrições.

Surge assim nossa questão de pesquisa: É possível melhorar o desempenho do conjunto de características de áudio utilizadas em atividades de CAA através da otimização multiobjetivo das características desse conjunto?

O desempenho das características foi determinado com base em dois aspectos: 1 – a acurácia obtida através do conjunto; 2 – a medida de sensibilidade ou de especificidade quando se pretende minimizar falsos negativo ou positivo, respectivamente (Seção 3.1.2).

Para responder tal questão organizamos dois experimentos que abordassem diferentes casos da problemática da classificação de áudio. O critério para a escolha desses problemas se deu por duas razões fundamentais: estar relacionado com uma situação real difícil de ser

resolvida devido à grande semelhança das classes do problema e a disponibilidade das bases de áudio.

Espera-se mostrar que o uso de algoritmos genéticos multiobjetivo tem um forte indício de efetividade mediante outras técnicas de *feature learning*, o que não significa dizer que a solução desse trabalho deva superar sempre as outras, mas despertar o interesse da comunidade científica para a exploração de novas possibilidades viáveis já seria uma excelente contribuição.

As técnicas em comparação foram três: a busca no espaço analítico com algoritmo evolucionário mono-objetivo (algoritmo EDS-Like), a busca no espaço analítico com algoritmo evolucionário multiobjetivo (ExpertMIR) e o *Feature Learning* com PCA (jMIR). Embora estejamos particularmente interessados nas estratégias evolucionárias, achamos importante comparar com alguma alternativa fora desse grupo a fim de situar a solução entre as opções da área. Como dito anteriormente, a escolha do PCA se deu principalmente pela sua implementação estar presente no arcabouço tecnológico suporte desse trabalho, o jMIR.

Para que houvesse coerência na comparação das abordagens foi necessário igualar alguns pontos. Primeiramente a quantidade e tipo de operadores utilizados por ambos. Foram utilizados um total de 9 operadores entre os grupos matemáticos e de processamento de sinais que seguem listados abaixo:

- MFCC
- FFT de frequências binárias
- Normalização
- *Power Spectrum*
- *Magnitude Spectrum*
- RMS
- *Zero Crossing*
- *Spectral Centroid*
- *Spectral Roll-off*

Esses operadores são as características genéricas das quais resultarão as características analíticas de cada processo de aprendizado. A descrição desses operadores pode ser encontrada juntamente com muitos outros em Peeters (2004).

Em segundo lugar foi necessário fixar o tamanho máximo de uma característica para os processos que envolvem algoritmos genéticos. Achamos conveniente adotar o tamanho empregado por Pachet (2007), até dez operações.

Por fim, para a avaliação das técnicas foi necessário utilizar o mesmo tipo de algoritmo de classificação, o K-NN com k = 1 e a mesma técnica de validação, o *cross-validation* (validação cruzada), o qual Kohavi (1995) sustenta ser o melhor método de amostragem de dados a ser utilizado na avaliação de um modelo de classificação.

Definidos esses aspectos, podemos supor uma comparação justa entre os três modelos de solução para os problemas.

Tabela 1: Métodos utilizados que incorporam as abordagens analisadas.

Tag	Método
PCA	PCA com alg. de classificação 1-NN
MO	AG mono-objetivo com alg. de classificação 1- NN
MT1	AG multiobjetivo com alg. de classificação 1- NN (Obj 2 = reduz falso negativo)
MT2	AG multiobjetivo com alg. de classificação 1- NN (Obj 2 = aumentar a acurácia da melhor característica)

A Tabela 1 apresenta os algoritmos utilizados para comparação das técnicas. Podemos observar que MT1 é o algoritmo evolucionário cujo segundo objetivo é reduzir a incidência de falso negativos, ou seja, melhorar a sensibilidade da solução. MT2 é o algoritmo cujo segundo objetivo é aumentar a acurácia da característica mais apta de um indivíduo multiobjetivo. Ambos os métodos implementam a solução proposta nesse trabalho, entretanto essas escolhas distintas quanto ao segundo objetivo se deram para poder constatar se há diferença ao escolher como objetivo secundário a diminuição dos falsos negativos (ou positivos) quando a natureza do problema assim o exige. Para tal, é necessário comparar com outro método que opere com outro objetivo secundário qualquer (nesse caso MT2 buscando otimizar a acurácia da característica mais apta).

De acordo com o Teorema do Limite Central, em que a distribuição das médias amostrais tende a uma distribuição normal a medida que o tamanho *n* da amostra aumenta. Por causa do tamanho das bases de dados envolvidas, executamos 10 vezes o primeiro experimento ($n = 10$) e 30 vezes o segundo ($n = 30$), obtendo quatro amostras com dados acerca da acurácia e sensibilidade dos métodos e executamos uma série de testes através da ferramenta R Studio (R Core Team, 2015).

Para os métodos com algoritmos evolucionários (MO, MT1 e MT2) fez-se *p = 15* e *e = 1500*, onde *p* e *e* são, respectivamente, o tamanho da população e a quantidade de avaliações a serem feitas. As taxas de recombinação e de mutação foram respectivamente 90% e 5%. MO implementa um algoritmo genético simples, enquanto MT1 e MT2 executam o NSGAII.

5.2. Experimento I: Monitoramento de Ambientes e Segurança (MAS)

Um sistema de segurança para monitoramento de ambientes é capaz de reconhecer situações de alerta como disparos de arma de fogo, colisão de veículos, vidro em estilhaço, gritos e etc. A base utilizada nesse experimento é uma base real fechada, utilizada para fins comerciais e que conta com até 18.000 exemplos rotulados dos mais variados tipos de som para alerta e segurança.

Uma solução do tipo requer uma série de conhecimentos que abrangem geolocalização, captura de som e de imagem, codificação, transmissão de dados, tratamento de imagem, compressão, armazenamento, etc. e dentre eles a classificação de sons encontra-se como uma atividade fundamental.

Para este trabalho, o problema abordado contou com um subconjunto balanceado de 1.900 instâncias desta base já fragmentadas. A situação escolhida consiste em distinguir sons entre duas classes: 1 – Disparo de arma de fogo; 2 – Estouro de fogos de artifício. A forte semelhança entre essas classes torna o problema bastante difícil de ser resolvido e isso sugere o uso de otimização de características para melhorar os resultados das técnicas empregadas para solucionar o problema. Questão semelhante foi abordada por Valenzise et al. (2007) e Gerosa et al. (2007) com a diferença de que buscam distinguir entre som de arma de fogo e outro som qualquer. No nosso caso, abordamos um problema mais específico escolhendo fogos de artifício como a segunda classe do problema, pela semelhança entre eles, que é tamanha ao ponto de a classificação não ser fácil nem para o ouvido humano. Valenzise e Gerosa percorrem um caminho tradicional, selecionando as características genéricas mais relevantes de um grupo.

Neste experimento, foram reunidas 950 instâncias de disparo de vários tipos de armas de fogo (revolveres, pistolas, metralhadoras, fuzis, espingardas, etc.) de diversos calibres em

oposição a 950 instâncias de estouros de fogos de artifício. Cada uma com taxa de amostragem de 48 kHz e duração média de 0.085 segundos.

5.2.1. Resultados Obtidos

5.2.1.1. Acurácia

No que se refere a acurácia das soluções, executou-se o teste T para cada uma das amostras e constatou-se que MT1 apresentou a melhor média (90,67%) sendo também ele o responsável pelo maior valor de acurácia encontrado (92,68%). Os resultados obtidos são ilustrados no gráfico da Figura 21, podemos observar que apesar de MT1 possuir os melhores resultados, MT2 assemelha-se bastante com ele (acurácia média = 90,57%), melhor acurácia = 91,74%). E apesar da máxima obtida em MO (91,05%) se aproximar dos algoritmos multiobjetivos, percebe-se que sua média está logo abaixo (85,83%).

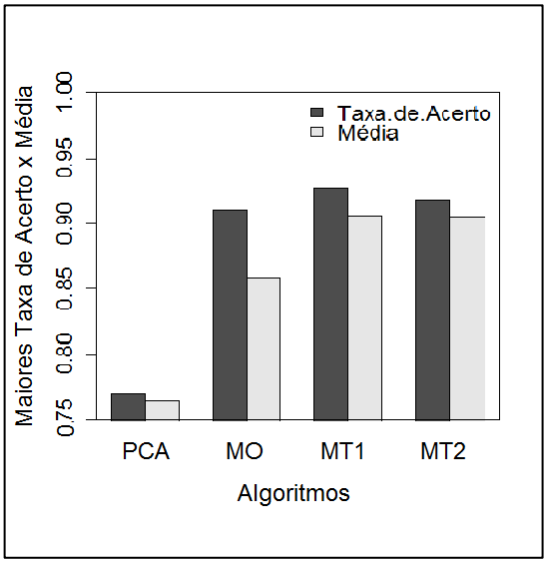

Figura 21: M.A.S. – Acurácia máxima registrada x média da acurácia dos métodos. Fonte: O Autor (2016).

Tabela 2: M.A.S. – Testes T para cada método + máxima acurácia registrada.

	Maior acurácia (%)	Média (%)	Intervalo de confiança (%)	Nível de significância (%)	p-valor
PCA	77	76,46	76,34– 76,6	5	2.2e-16
MO	91,05	85,83	83,67 – 87,99	5	1.34e-16
MT1	92,68	90,67	89,51 – 91,83	5	2.2e-16
MT2	91,74	90,57	89,99 – 91,15	5	2.2e-16

O *boxplot* (Figura 22) permite visualizar de forma mais clara a distribuição empírica dos dados. Podemos observar onde a parte mais relevante dos dados amostrais está concentrada e o quão semelhante são os métodos estudados. Observamos que MT1 tem variabilidade pouco maior do que MT2, e também intervalo, média e mediana ligeiramente distintas, sendo essas soluções semelhantes. Nos parâmetros escolhidos a variabilidade ilustrada no gráfico aproxima-se bastante dos intervalos de confiança dos testes, sendo possível orientar-se por ele para compreender o teste. É possível também, observando o intervalo da *box* dos métodos, entender comportamentos como por exemplo, a possibilidade de um dado estar fora da média e o quanto este pode se distanciar da mesma. Por exemplo: o teste em MO indica que em 95% dos casos sua acurácia estará entre 83,67 e 87,99 (quarta e quinta coluna da Tabela 1), mas o intervalo de MO no *boxplot* é maior ainda e isso explica o aparecimento de 91,05% (a sua melhor acurácia registrada) como evento previsível. Valores fora desse intervalo seriam discrepantes e nesse caso não seria garantido obtê-los repetindo o experimento.

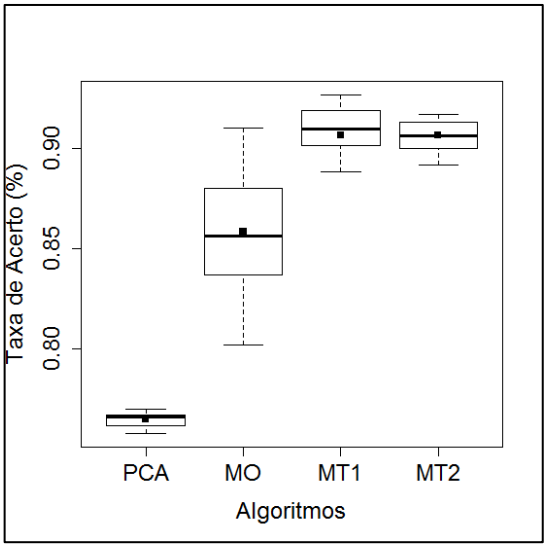

Figura 22: M.A.S. – Boxplot da acurácia dos métodos. Fonte: O Autor (2016).

O teste binomial unilateral à direita (GAMERMAN; MIGON, 1993) foi aplicado para saber se, de fato MT1 é melhor que MT2. Obtivemos os seguintes valores:

- Limiar de 50%
- Nível de significância $\alpha = 0,05$ (5%)
- p-valor = 0.05469

O teste revela que possivelmente em mais da metade dos casos MT1 supera MT2 e ainda dá a probabilidade de 80% de chances de MT1 ser melhor. A Figura 23 dispõe esses dados ordenados para que possa visualizar o comportamento indicado no teste binomial.

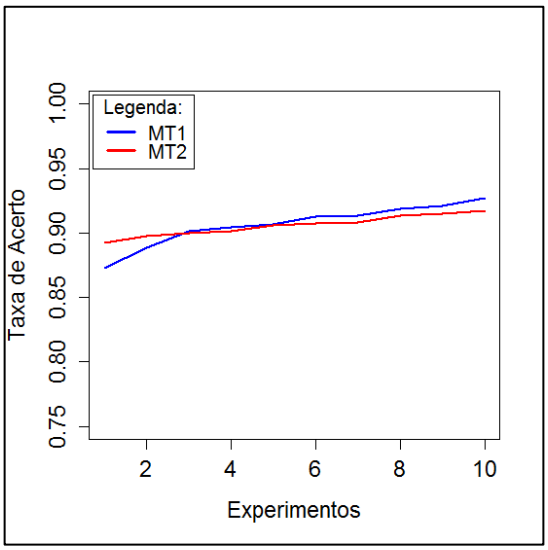

Figura 23: M.A.S. – Acurácia dos métodos MT1 e MT2. Fonte: O Autor (2016).

5.2.1.2. Sensibilidade

Para a base de dados, a Sensibilidade descreve a probabilidade de um legítimo disparo de arma de fogo ser classificado como disparo de arma de fogo.

Foi executado o teste T para cada amostra e constatou-se que a diferença entre MT1 e MT2 fica mais acentuada apresentando as melhores médias de sensibilidade entre os métodos, 89,46% e 87,83 respectivamente. Os outros métodos, por sua vez, não possuem um resultado tão interessante quanto os alcançados pela otimização multiobjetivo (Figura 24). A Tabela 3 dispõe os dados da ilustração.

Tabela 3: M.A.S. – Testes T para cada método + máxima sensibilidade registrada.

	Maior sensibilidade (%)	Média (%)	Intervalo de confiança (%)	Nível de significância (%)	p-valor
PCA	76,63	72,55	75,58 – 76	5	2.2e-16
MO	89,79	83,57	81,15 – 85,99	5	4.684e-14
MT1	92	89,46	87,83 – 91,1	5	7.575e-16
MT2	91,37	87,83	86,42 – 89,24	5	2.316e-16

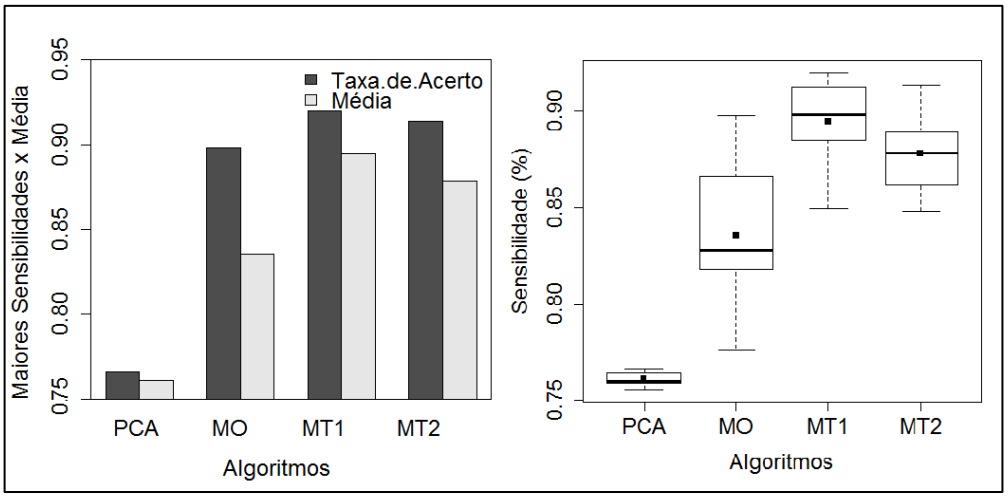

Figura 24: M.A.S. – Testes de sensibilidade obtidos. Fonte: O Autor (2016).

Realizando-se o teste binomial para as vezes que MT1 supera MT2 é obtido o seguinte resultado:

- Limiar de 50%
- Nível de significância $\alpha = 0,05$ (5%)
- p-valor = 0,0009766

O teste é conclusivo quanto a hipótese de MT1 superar MT2, o mesmo teste apresenta uma probabilidade de sucesso de 100%. Na Figura 25 é possível ver o comportamento deles na redução de falsos negativos. Os menores valores de MT1 e MT2 são 4% e 4,3% respectivamente e representam o percentual de falsos negativos na matriz de confusão da melhor instância de cada amostra.

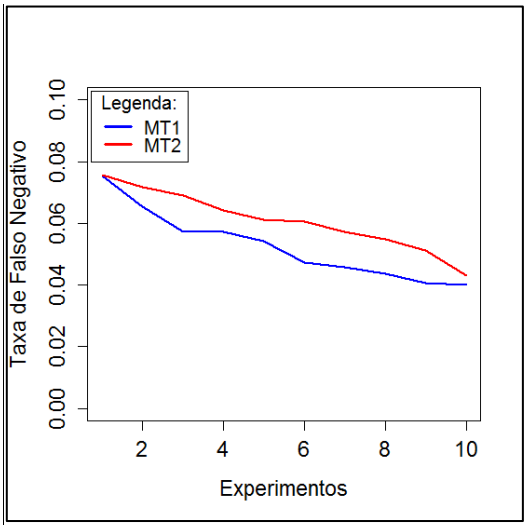

Figura 25: M.A.S. – Taxa de Falso Negativo dos métodos MT1 e MT2. Quanto menor melhor. Fonte: O Autor (2016).

5.2.2. Análise dos resultados

Voltamos a nossa questão de pesquisa: "É possível melhorar o desempenho do conjunto de características de áudio utilizadas em atividades de CAA através da otimização multiobjetivo das características desse conjunto? ".

Para tentar atender devidamente esta questão, temos as seguintes hipóteses que se reportam ao objetivo geral explanado na introdução deste trabalho: "Analisar o poder de algoritmos evolucionários multiobjetivo na concepção de características analíticas de áudio para problemas de classificação automática de áudio " e que dizem respeito as medidas de desempenho (acurácia e sensibilidade para o referido problema) da questão de pesquisa.

Hipótese 1: Acurácia da solução.

- H0: O uso do modelo de otimização multiobjetivo não melhora a acurácia da classificação.
- H1: O uso do modelo de otimização multiobjetivo melhora a acurácia da classificação.

Hipótese 2: Sensibilidade da solução.

- H0: O uso do modelo de otimização multiobjetivo não melhora a sensibilidade da classificação.
- H1: O uso do modelo de otimização multiobjetivo melhora a sensibilidade da classificação.

Tomamos como base o método MT1 que obteve o melhor desempenho entre os métodos multiobjetivo e comparamos com MO, pois este claramente foi o melhor método a não empregar a técnica multiobjetivo. A Tabela 4 apresenta os resultados de cada teste de hipótese.

Tabela 4: Resultados dos testes binomial (MT1 e MO) para cada hipótese.

	Limiar	Nível de significância	Intervalo de confiança	p-valor
Hipótese 1	50 %	5 %	60,58 % - 100 %	0,01074
Hipótese 2	50 %	5 %	60,58 % - 100 %	0,01074

5.2.2.1. Sustentação

Como o p-valor é menor que o nível de significância de ambos os problemas as hipóteses nulas são descartadas. Isso quer dizer que estaticamente o método de otimização multiobjetivo com diminuição de falsos negativos (MT1) é superior a otimização mono-objetivo em mais da metade dos casos. De acordo com o mesmo teste essa efetividade é estatisticamente superior a 60,58% dos casos para ambas as hipóteses e é provavelmente 90%.

5.3. Experimento II: Identificação de Nasalidade

Atividades de transcrição de áudio requerem a clara distinção entre sons de vogais, sendo esses sons nasais ou orais. O problema de nasalidade consiste em identificar quando alguma formante do som é proveniente da vibração nas cavidades nasais (nasal) ou é simplesmente constituído da vibração das cordas vocais (oral). O problema de nasalidade costuma ser resolvido por uma abordagem convencional *bag-of-frame*, entretanto seguindo a proposta do EDS de que é possível melhorar os resultados através das características analíticas, realizamos um experimento que abordasse o problema.

Para este experimento contou-se com um conjunto balanceado de 60 instâncias de sons da vogal *a* já fragmentados, com taxa de amostragem de 44,1 kHz, e incorreu nos resultados apresentados a seguir.

5.3.1. Resultados Obtidos

5.3.1.1. Acurácia

No que se refere a acurácia das soluções, executou-se o teste T para cada uma das amostras e constatou-se que MT1 e MT2 apresentaram as melhores médias (91,83% e 88,61% respectivamente), sendo o maior valor de acurácia registrado por MT1 (96,67%). Nenhuma das outras técnicas possuíram resultado semelhante ao alcançado pelos métodos multiobjetivo.

Tabela 5: Nasalidade – Resultado dos testes T para cada método + máxima acurácia registrada.

	Maior acurácia (%)	Média (%)	Intervalo de confiança (%)	Nível de significância (%)	p-valor
PCA	76,66	72,55	71,89 – 73,22	5	2.2e-16
MO	83,33	73,61	72,03 – 75,19	5	2.2e-16
MT1	96,67	91,83	90,67 – 92,99	5	2.2e-16
MT2	95	88,61	87,58 – 89,65	5	2.2e-16

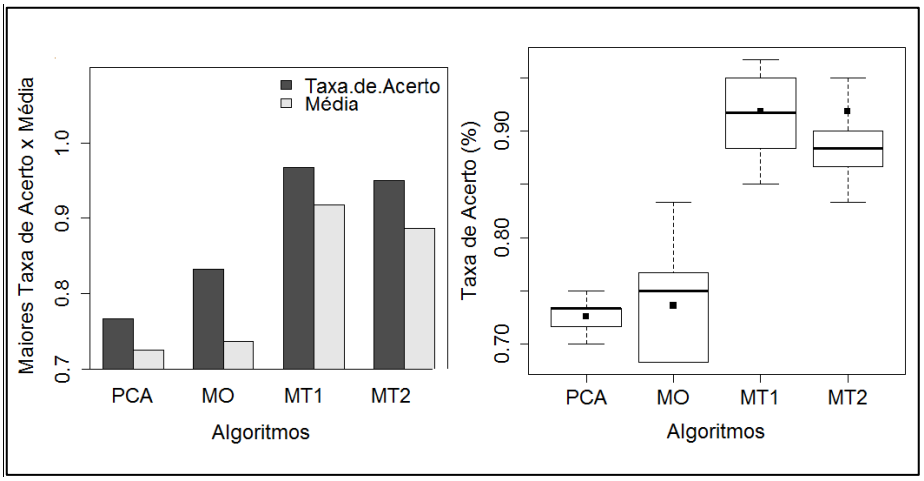

Figura 26: Nasalidade – Acurácia máxima registrada x média da acurácia dos métodos. Fonte: O Autor (2016).

O *Boxplot* apresenta a distribuição empírica dos dados. Nele pode-se perceber apesar da semelhança, constatada na Tabela 5, entre os métodos PCA e MO, esse último por possuir maior variabilidade alcança melhor valor máximo da acurácia. Enquanto os valores de desempenho para os métodos multiobjetivo apresentam-se como a favor de MT1. Aplicando-se o teste binomial conclui-se que MT1 é de fato melhor que MT2:

- Limiar de 50%
- Nível de significância $\alpha = 0,05$ (5%)
- p-valor = 9.313e-10.

O mesmo teste infere que MT1 superará MT2 em mais de 90,5% dos futuros casos, sendo, dos casos apresentados nas duas amostras, obtido 100% de sucesso. A Figura 27 dispõe esses dados ordenados para visualização do comportamento das amostras.

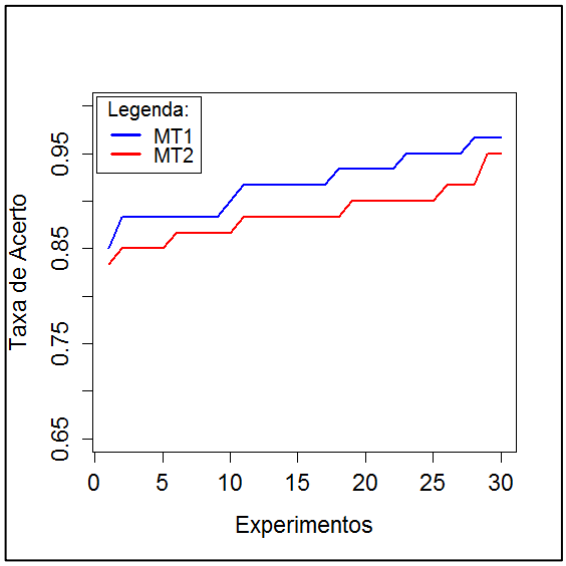

Figura 27: Nasalidade – Acurácia dos métodos MT1 e MT2. Fonte: O Autor (2016).

5.3.1.2. Sensibilidade

O problema de nasalidade não requer nenhuma exigência sobre a incidência particular de falsos positivos ou de falsos negativos, a importância da ocorrência de um ou de outro é a mesma. Espera-se somente que a solução utilizada faça boa distinção entre um som nasal ou oral. Entretanto, como é objetivo desta experimentação averiguar a capacidade de determinado método corresponder melhor a restrições, escolhemos investigar o comportamento da sensibilidade. Nesse caso a medida corresponde a probabilidade de uma solução classificar um som como "nasal" quando este de fato é.

Os resultados obtidos pelo teste de sensibilidade foram equilibrados. O método MO chegou a alcançar o melhor resultado (100%), entretanto ao visualizar a distribuição empírica dos dados (*boxplot* da Figura 28) percebe-se que se tratou de um resultado discrepante. Quanto aos demais algoritmos, aqueles que obtiveram melhor desempenho foram multiobjetivos. MT1 parece ser mais constante. Na realização do teste T é constatado que ele possui um intervalo de confiança melhor (Tabela 6).

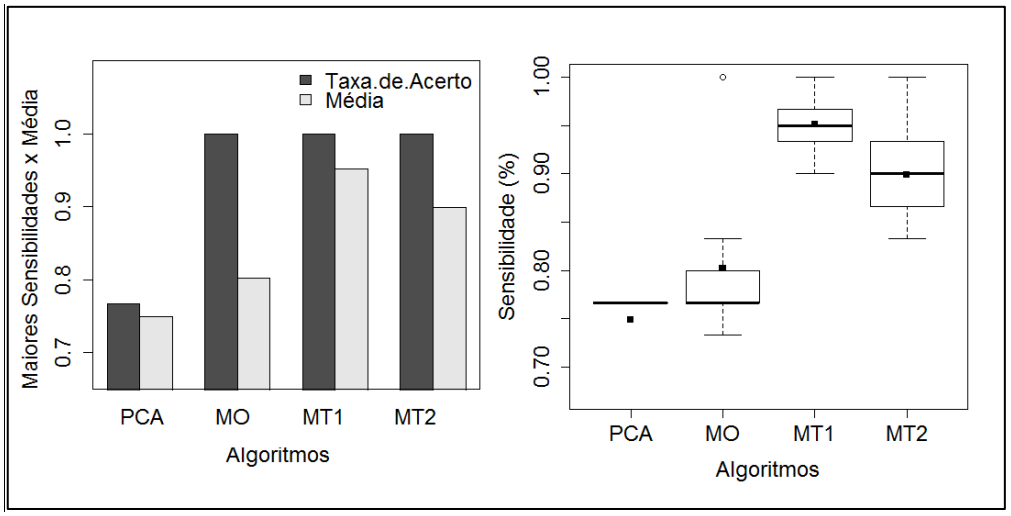

Figura 28: Nasalidade – Ilustração do teste de sensibilidade obtidos. Fonte: O Autor (2016).

Tabela 6: Nasalidade – Resultado dos testes T para cada método + máxima sensibilidade registrada.

	Maior sensibilidade (%)	Média (%)	Intervalo de confiança (%)	Nível de significância (%)	p-valor
PCA	76,67	74,89	73,59 – 76,19	5	2.2e-16
MO	100	80,22	76,76 – 83,68	5	2.2e-16
MT1	100	95,11	94,26 – 95,96	5	2.2e-16
MT2	100	89,88	88,27 – 91,51	5	2.2e-16

Aplicando o teste binomial para comprovar estatisticamente a superioridade de MT1 em relação a MT2 obtivemos:

- Limiar de 50%
- Nível de significância 5%
- p-valor = 4.34e-07,

A probabilidade de MT1 superar MT2 é de 93,33%. Esse comportamento pode ser visualizado no gráfico da Figura 29, onde o percentual de falsos negativos para ambos algoritmos está ilustrado.

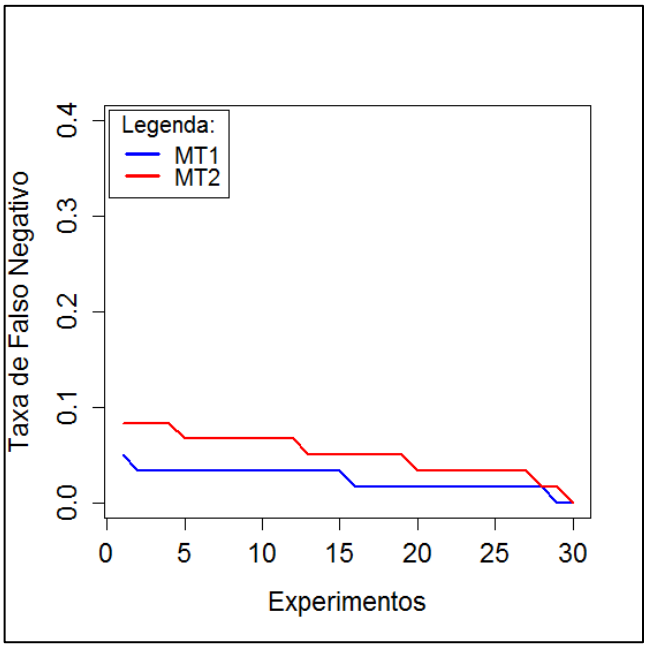

Figura 29: Nasalidade – Taxa de Falso Negativo para cada instância de MT1 e MT2. Quanto menor, melhor. Fonte: O Autor (2016).

5.3.2. Análise dos resultados

Para esta análise utilizamos as mesmas hipóteses do primeiro experimento, recordemo-las a seguir:

Hipótese 1: Acurácia da solução.

- H0: O uso do modelo de otimização multiobjetivo não melhora a acurácia da classificação.
- H1: O uso do modelo de otimização multiobjetivo melhora a acurácia da classificação.

Hipótese 2: Sensibilidade da solução.

- H0: O uso do modelo de otimização multiobjetivo não melhora a sensibilidade da classificação.
- H1: O uso do modelo de otimização multiobjetivo melhora a sensibilidade da classificação.

O teste tomou como base o método MT1 que obteve melhor desempenho do que MT2 em ambos os critérios e, comparou-o com MO, já que alguns de seus resultados intercedem

com os resultados de MT1, enquanto PCA fica atrás do mesmo. A Tabela 7 apresenta os resultados ao testar as hipóteses.

Tabela 7: Resultados dos testes binomial (MT1 e MO) para cada hipótese.

	Limiar	Nível de significância	Intervalo de confiança	p-valor
Hipótese 1	50 %	5 %	90,5 % - 100 %	9.313e-10
Hipótese 2	50 %	5 %	72,04 % - 100 %	2.974e-05

5.3.2.1. Sustentação

A respeito da acurácia do método o p-valor é menor que o nível de significância e nesse caso a hipótese nula é descartada. Isso quer dizer que estaticamente o método de otimização multiobjetivo com diminuição de falsos negativos (MT1) é superior aos otimização mono-objetivo em mais da metade dos casos. E de acordo com o mesmo teste a probabilidade de isso ocorrer é de maior que 90,5%. Quanto a sensibilidade de MT1, o p-valor do teste também é menor do que a significância, portanto não cabe a hipótese nula. A probabilidade de MT1 superar MO no critério da sensibilidade é maior que 72,04%, para os dados manipulados ela foi de 86,67%.

5.4. Consolidação dos Resultados

Ao serem analisados os métodos em termos de acurácia e sensibilidade obteve-se importantes indicativos de que a solução multiobjetivo tem um peso significante na efetividade da solução de problemas de CAA. A Tabela 8 resume o método que obteve melhor desempenho em cada critério dos dois experimentos realizados.

Tabela 8: Resumo do desempenho de cada método por base de áudios.

	Acurácia	*Sensibilidade*
M.A.S.	MT1 apresenta-se melhor	MT1 apresenta-se melhor
Nasalidade	MT1 apresenta-se melhor	MT1 apresenta-se melhor

Nesse sentido pode-se afirmar que MT1, algoritmo genético multiobjetivo em que o segundo objetivo é reduzir falsos negativo combinado com 1-NN, foi a melhor solução encontrada.

Os problemas tratados nos experimentos são reais o que vem evidenciar a aplicabilidade do método. O indicativo particularmente interessante do quanto ele pode ser promissor é a aproximação com a solução comercial original, comentada em Araújo (2014). Em um problema de duas classes: 1 – disparo de arma de fogo; 2 – outro barulho, a solução já comercializada obteve uma redução de erro para 3,57% (ARAÚJO, 2014) já a proposta nesse trabalho chegou a reduzir o erro para 7,32% (equivalente a acurácia de 92,68% do método MT1 na Tabela 2), entretanto havendo uma diferença na base de dados quanto a quantidade e tipo de sons, já que o problema original está distinguindo disparos de arma de fogo de qualquer outro som, enquanto o problema tratado aqui é distinguir entre disparos de arma de fogo e fogos de artifício, que são classes mais semelhantes e portanto a distinção é mais difícil. É possível que a solução proposta ofereça resultados mais interessantes na mesma base do problema original. Uma oportunidade de estudo futuro.

Além das conclusões estatísticas alguns aspectos podem ser destacados a respeito de cada método. A otimização de característica com o PCA e o algoritmo genético simples são as mais rápidas, chegando aos resultados em minutos, enquanto que a abordagem multiobjetivo costumou dar resultados em horas. No entanto levando em conta que é um processo automatizado o resultado é mais interessante do que soluções manuais como a *ad-hoc*. No fim das contas, a eliminação da etapa de seleção no fim da execução do EDS (Capítulo 3, Seção 3.5.2.4), embora simplifique o processo não deixa a solução mais ágil.

A solução proposta neste trabalho foi aquela implementada pelos dois métodos MT1 e MT2 e mostra-se estatisticamente superior às demais. É claro, isso é dito somente a respeito da capacidade de otimização de características de áudio. Não houve preocupação em otimizar o algoritmo classificador, que foi fixado.

Remetendo-se a solução esperada do Capítulo 2 pode-se perceber que a solução proposta corresponde à expectativa da pesquisa, por assim dizer:

- Encontrar em tempo viável um conjunto de características de áudio que possam representar os dados.
- Corretude: A solução influenciou na melhora dos resultados da classificação.
- Adequação: A solução levou em conta as particularidades dos problemas, buscando a satisfação de restrições.
- Disponibilidade de código: Solução aberta.
- Reusabilidade: A técnica pode ser empregada em vários tipos de problema.

- Economia de conhecimento especializado: Não é necessário conhecer a natureza das características de áudio, basta somente o manuseio das tecnologias de classificação.

6. Conclusões

A área de classificação de áudio é enorme e diversificada, sendo o aprendizado de características um aspecto muito importante. Apesar dos avanços relevantes alcançados com o *Deep Learning*, esse trabalho mostrou que ainda há espaço para melhorias em outros sentidos.

O principal intuito da solução proposta é melhorar o desempenho do conjunto de características de áudio nas atividades de classificação. Inspirado no *Extractor Discovery System* ousou-se extrapolar o espaço de características genéricas, que normalmente são as empregadas nas fases de extração e seleção, buscando construir soluções mais dinâmicas, que não são facilmente pensadas por um especialista.

Essas soluções estão intrinsecamente ligadas ao algoritmo de classificação utilizado, uma vez que não foi realizada nenhuma busca ou tentativa de melhorar o classificador através do ajuste de seus parâmetros. Mesmo assim a classificação produziu resultados interessantes. Conclui-se que as características analíticas exploradas otimizam o classificador para aquela base de dados utilizada. Ou seja, apesar de um tipo de classificador não ser adequado para um dado problema de Classificação Automática de Áudio é possível melhorar seu desempenho através do desenvolvimento de características analíticas. Isto não quer dizer que outro tipo de classificador configurado com outros parâmetros não possa superá-lo, mas que, dentro de suas especificações ele será otimizado. Portanto ExpertMIR pode ser capaz de otimizar todo tipo de classificador de áudio.

A necessidade de ferramentas que auxiliem a comunidade no desenvolvimento de características de áudio pode ser suprida por implementações da abordagem proposta por este trabalho.

Levando-se em conta a dificuldade de processo e de custo na concepção de características analíticas e, visando uma melhor adaptação a natureza de alguns problemas de CAA, buscou-se desenvolver uma alternativa que se adequasse às necessidades da área. Com

os resultados dos experimentos consolidados foi possível afirmar a utilidade da proposta. ExpertMIR é uma solução aberta e útil na otimização de características acústicas principalmente quando não se pode contar com especialistas, ou não se dispõe de recursos e tempo, ou até mesmo quando uma solução for difícil de ser concebida analiticamente.

É importante ampliar a comparação com outras abordagens e fazer testes com outras bases a fim de identificar eventuais limitações do modelo ou evoluí-lo. Sendo assim nossos materiais e métodos ficam disponíveis para que outros possam realizar novos estudos e pesquisas complementares.

6.1. Trabalhos futuros

É possível se pensar em um método de otimização que esteja simultaneamente preocupado em otimizar o conjunto de características e o tipo de classificador. Aplicar métodos específicos de meta-aprendizagem como *Sequential Model Algorithm Selection* (SMAC) durante o processo apresenta-se como perspectiva atraente para trabalhos futuros.

Entendemos também que o diferencial da técnica em relação a solução já existente do EDS é a multiobjetividade de seu algoritmo. Nesse campo há diversas possibilidades: utilização de mais de dois objetivos, identificação do melhor segundo objetivo, otimização dos parâmetros do mesmo algoritmo e etc. A necessidade de preencher a lacuna deixada por essas opções também abre novos caminhos para pesquisas. Também é possível crescer na performance da técnica utilizando CUDA (*Compute Unified Device Architecture*) para paralelização.

Uma outra sugestão atraente está no fato de Pachet (2007, p.6) demonstrar que é possível aproximar-se de características genéricas desconhecidas através da exploração analítica. Ou seja, mesmo sem conhecer seu funcionamento é possível imitar seu comportamento. Se é possível imitar tais características, seria concebível também imitar modelos de classificação?

Por fim, existe também a possibilidade de expansão da pesquisa para outros domínios de conhecimento, como processamento de imagens ou classificações diversas.

APÊNDICE A – Extractor Discovery System: Alguns aspectos

1. Lista de operadores básicos:

Abs	HFC	Pitch
Arcsin	HMean	PitchBands
AttackTime	HMedian	Power
Autocorrelation	HMax	Range
Bandwidth	HMin	RemoveSilentFrames
BarkBands	HpFilter	RHF
Bartlett	Integration	Rms
Blackman	Inverse	SpectralCentroid
BpFilter	Iqr	SpectralDecrease
Centroid	Length	SpectralFlatness
Chroma	Log10	SpectralKurtosis
Correlation	LpFilter	SpectralRolloff
dB	Max	SpectralSkewness
Differentiation	MaxPos	SpectralSpread
Division	Mean	Split
Envelope	Median	SplitOverlap
Fft	MelBands	Sqrt
FilterBank	Min	Square
Flatness	Mfcc0	Sum
Hamming	Mfcc	Triangle
Hann	Multiplication	Variance
Hanning	Normalize	Zcr
HarmSpectralCentroid	Nth	Harmonicity(Praat)
HarmSpectralDeviation	NthColumns	Ltas(Praat)
HarmSpectralSpread	PeakPos	
HarmSpectralVariation	Percentile	

Figura 30: Operadores usados em Pachet e Roy (2007).

2. Representação dos tipos de dados utilizados:

Representação	Significado
« **a** »	amplitude
« **t** »	tempo
« **f** ».	frequência
« **t : a** »	Sinal de áudio (tempo/amplitude)
« **f : a** »	Frequência/amplitude
« **Vt:a** »	Vetor de sinais de áudio
« **Vf:a** »	Vetor de sinais de frequências
« **Va** »	Vetor de valores

3. Tipos de operadores:

MEAN: « TESTWAV » → « **Mean** (TESTWAV) »
« t:a » → « a »
« x:y » → **« y »**
MEAN: «MFCC(TESTWAV, 10)» → « **Mean** (MFCC (TESTWAV, 10)) »
« Va » → « a »
« Vx » → **« x »**
SPLIT: « TESTWAV » → « **Split** (TESTWAV, 1000) »
« t:a » → « Vt:a »
« X » → **« VX »**
FFT: « TESTWAV » → « Fft (TESTWAV) »
« t:a » → « f:a »
FFT: « Fft (TESTWAV) » → « Fft (Fft (TESTWAV)) »
« f:a » → « t:a »
« x:y » → **« x^-1:y »**
HPFILTER: « TESTWAV, 1000 » → « **HpFilter** (TESTWAV, 1000) »
« t:a », « f » → « t:a »

4. Operadores Genéricos

➔ **?_a (TESTWAV): 1 operator**

- Mean (TESTWAV): « t:a » → « a »
- Variance (TESTWAV): « t:a » → « a »

➔ ***_a (TESTWAV): several operators of « a » output type**

- *« Square (Mean (TESTWAV)) »*

Square (Mean (TESTWAV))
« a » ← « a » ← « t:a »

➔ **!_a (TESTWAV): several operators of any type**

- *« Square (Max (Fft (HpFilter (TESTWAV, 1000))) »*

Square (Max (Fft (HpFilter (TESTWAV, 1000)))
« a » ← « a » ← « f:a » ← « t:a » ← « t:a »

Referências

ALPAYDIN, Ethem. **Introduction to machine learning**. MIT press, 2014.

ARAÚJO, D. F. D. E. Busca como sistema de apoio à melhoria de classificadores automáticos de áudio. 2014.

AUCOUTURIER, Jean-Julien; DEFREVILLE, Boris; PACHET, François. The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music. **The Journal of the Acoustical Society of America**, v. 122, n. 2, p. 881-891, 2007.

BENGIO, Yoshua; COURVILLE, Aaron; VINCENT, Pascal. Representation learning: A review and new perspectives. **IEEE transactions on pattern analysis and machine intelligence**, v. 35, n. 8, p. 1798-1828, 2013.

BLUM, Avrim L.; LANGLEY, Pat. Selection of relevant features and examples in machine learning. **Artificial intelligence**, v. 97, n. 1, p. 245-271, 1997.

BOULANGER-LEWANDOWSKI, Nicolas; BENGIO, Yoshua; VINCENT, Pascal. Audio Chord Recognition with Recurrent Neural Networks. In: **ISMIR**. 2013. p. 335-340.

BURKA, Zak. Perceptual audio classification using principal component analysis. 2010.

CABRAL, Giordano; PACHET, François; BRIOT, Jean-Pierre. Recognizing chords with EDS: Part one. In: **International Symposium on Computer Music Modeling and Retrieval**. Springer Berlin Heidelberg, 2005. p. 185-195.

COATES, Adam; LEE, Honglak; NG, Andrew Y. An analysis of single-layer networks in unsupervised feature learning. **Ann Arbor**, v. 1001, n. 48109, 2010.

CORRÊA, Débora Cristina. **Inteligência artificial aplicada à análise de gêneros musicais**. 2012. Tese de Doutorado. Universidade de São Paulo.

COSTA, Luciano da Fontoura Da; CESAR JR, Roberto Marcondes. **Shape analysis and classification: theory and practice**. CRC Press, Inc., 2001.

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

DEB, Kalyanmoy. Multi-objective optimisation using evolutionary algorithms: an introduction. In: **Multi-objective evolutionary optimisation for product design and manufacturing**. Springer London, 2011. p. 3-34.

DEB, Kalyanmoy et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. **IEEE transactions on evolutionary computation**, v. 6, n. 2, p. 182-197, 2002.

DEB, Kalyanmoy. **Multi-objective optimization using evolutionary algorithms**. John Wiley & Sons, 2001.

DUDA, Richard O.; HART, Peter E.; STORK, David G. **Pattern classification**. John Wiley & Sons, 2012.

DURILLO, Juan J.; NEBRO, Antonio J. jMetal: A Java framework for multi-objective optimization. **Advances in Engineering Software**, v. 42, n. 10, p. 760-771, 2011.

EGGINK, Jana; BROWN, Guy J. A missing feature approach to instrument identification in polyphonic music. In: **Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on**. IEEE, 2003. p. V-553-6 vol. 5.

EIBEN, Agoston E.; SMITH, James E.**Introduction to evolutionary computing**. Heidelberg: springer, 2003.

ERONEN, Antti. Musical instrument recognition using ICA-based transform of features and discriminatively trained HMMs. In: **Signal Processing and Its Applications, 2003. Proceedings. Seventh International Symposium on**. IEEE, 2003. p. 133-136.

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

FISHER, Ronald A. The use of multiple measurements in taxonomic problems. **Annals of eugenics**, v. 7, n. 2, p. 179-188, 1936.

GAMERMAN, Dani; DOS SANTOS MIGON, Helio.**Inferência estatística: uma abordagem integrada**. Instituto de Matemática, Universidade Federal do Rio de Janeiro, 1993.

GEROSA, Luigi et al. Scream and gunshot detection in noisy environments. In: **Signal Processing Conference, 2007 15th European**. IEEE, 2007. p. 1216-1220.

GOLBERG, David E. Genetic algorithms in search, optimization, and machine learning. **Addion wesley**, v. 1989, p. 102, 1989.

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

GUO, Y. B.; AMMULA, S. C. Real-time acoustic emission monitoring for surface damage in hard machining. **International Journal of Machine Tools and Manufacture**, v. 45, n. 14, p. 1622-1627, 2005.

HOLMES, Geoffrey; DONKIN, Andrew; WITTEN, Ian H. Weka: A machine learning workbench. In: **Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on**. IEEE, 1994. p. 357-361.

HUMPHREY, Eric J.; BELLO, Juan P.; LECUN, Yann. Feature learning and deep architectures: new directions for music informatics. **Journal of Intelligent Information Systems**, v. 41, n. 3, p. 461-481, 2013.

HUMPHREY, Eric J.; BELLO, Juan Pablo; LECUN, Yann. Moving Beyond Feature Design: Deep Architectures and Automatic Feature Learning in Music Informatics. In: **ISMIR**. 2012. p. 403-408.

HUTTER, Frank; HOOS, Holger H.; LEYTON-BROWN, Kevin. Sequential model-based optimization for general algorithm configuration. In:**International Conference on Learning and Intelligent Optimization**. Springer Berlin Heidelberg, 2011. p. 507-523.

HYVÄRINEN, Aapo; OJA, Erkki. Independent component analysis: algorithms and applications.**Neural networks**, v. 13, n. 4, p. 411-430, 2000.

KINNEAR, Kenneth E. **Advances in genetic programming**. MIT press, 1994.

KLAPURI, Anssi; DAVY, Manuel (Ed.). **Signal processing methods for music transcription**. Springer Science & Business Media, 2007.

KNOWLES, Joshua; CORNE, David. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In: **Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on**. IEEE, 1999.

KOHAVI, Ron et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: **Ijcai**. 1995. p. 1137-1145.

KOZA, John R. **Genetic programming: on the programming of computers by means of natural selection**. MIT press, 1992.

KOZA, John R. Genetic programming II: Automatic discovery of reusable subprograms. **Cambridge, MA, USA**, 1994.

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

KWON, Oh-Wook et al. Emotion recognition by speech signals. In:**INTERSPEECH**. 2003.

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.

LIPPMANN, R. P. Review of neural networks for speech recognition. **Neural Computation**, MIT Press, v. 1, n. 1, p. 1–38, 2016/07/27 1989. Disponível em: http://dx.doi.org/10.1162/neco.1989.1.1.1.

MCENNIS, Daniel; MCKAY, Cory; FUJINAGA, Ichiro; DEPALLE, Philippe. jAudio: A feature extraction library. In: **Proceedings of the International Conference on Music Information Retrieval**. 2005. p. 600-3.

MCKAY, Cory. **Automatic music classification with jMIR**. 2010. Tese de Doutorado. McGill University.

MCKAY, Cory et al. ACE: A Framework for Optimizing Music Classification. In: **ISMIR**. 2005. p. 42-49.

MCKAY, Cory; FUJINAGA, Ichiro. Automatic music classification and the importance of instrument identification. In: **Proceedings of the Conference on Interdisciplinary Musicology**. 2005.

MIERSWA, Ingo; MORIK, Katharina. Automatic feature extraction for classifying audio data. **Machine learning**, v. 58, n. 2-3, p. 127-149, 2005.

MIETTINEN, Kaisa. **Nonlinear multiobjective optimization**. Springer Science & Business Media, 2012.

MITCHELL, T. M. **Machine learning**. [S.l.]: McGraw Hill, 1997

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. **Sistemas Inteligentes-Fundamentos e Aplicações**, Manole, p. 89–114, 2003.

MUTHUSAMY, Yeshwant K.; BARNARD, Etienne; COLE, Ronald A. Reviewing automatic language identification. **IEEE Signal Processing Magazine**, v. 11, n. 4, p. 33-41, 1994.

PACHET, François; ROY, Pierre. Analytical features: a knowledge-based approach to audio feature generation. **EURASIP Journal on Audio, Speech, and Music Processing**, v. 2009, n. 1, p. 1, 2009.

PACHET, François; ROY, Pierre. Exploring billions of audio features. In: **2007 International Workshop on Content-Based Multimedia Indexing**. IEEE, 2007. p. 227-235.

PACHET, François; ZILS, Aymeric. Evolving automatically high-level music descriptors from acoustic signals. In: **International Symposium on Computer Music Modeling and Retrieval**. Springer Berlin Heidelberg, 2003. p. 42-53.

PEETERS, Geoffroy. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Tech. Rep., IRCAM, 2004.

POTAMITIS, Ilyas; GANCHEV, Todor; FAKOTAKIS, Nikos. Automatic acoustic identification of insects inspired by the speaker recognition paradigm. In: **INTERSPEECH**. 2006.

POTAMITIS, Ilyas; GANCHEV, Todor. Generalized recognition of sound events: Approaches and applications. In: **Multimedia Services in Intelligent Environments**. Springer Berlin Heidelberg, 2008. p. 41-79.

R Core Team (2015). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

RITTHOF, O. et al. A hybrid approach to feature selection and generation using an evolutionary algorithm. In: **2002 UK workshop on computational intelligence**. 2002. p. 147-154.

RUSSEL, Stuart J.; NORVIG, Peter. Inteligência Artificial: uma abordagem moderna. 3ª edição. **Rio de Janeiro, Brasil. Editora Elsevier**, 2013.

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.

SIEDLECKI, Wojciech; SKLANSKY, Jack. A note on genetic algorithms for large-scale feature selection. **Pattern recognition letters**, v. 10, n. 5, p. 335-347, 1989.

SOHN, Jongseo; KIM, Nam Soo; SUNG, Wonyong. A statistical model-based voice activity detection. **IEEE signal processing letters**, v. 6, n. 1, p. 1-3, 1999.

VALENZISE, Giuseppe et al. Scream and gunshot detection and localization for audio-surveillance systems. In: **Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on**. IEEE, 2007. p. 21-26.

VAN VELDHUIZEN, David A.; LAMONT, Gary B. Evolutionary computation and convergence to a pareto front. In: **Late breaking papers at the genetic programming 1998 conference**. 1998. p. 221-228.

VARILE, Giovanni Battista; ZAMPOLLI, Antonio. **Survey of the state of the art in human language technology**. Cambridge University Press, 1997.

YASLAN, Yusuf; CATALTEPE, Zehra. Audio music genre classification using different classifiers and feature selection methods. In: **18th International Conference on Pattern Recognition (ICPR'06)**. IEEE, 2006. p. 573-576.

WEST, Kristopher; COX, Stephen. Features and classifiers for the automatic classification of musical audio signals. In: **ISMIR**. 2004.

ZHOU, Xinquan; LERCH, Alexander. Chord Detection Using Deep Learning. In: **Proceedings of the 16th ISMIR Conference**. 2015.

ZITZLER, Eckart et al. SPEA2: Improving the strength Pareto evolutionary algorithm. In: **Eurogen**. 2001. p. 95-100.