UNIVERSIDADE DE CAXIAS DO SUL
CENTRO DE COMPUTAÇÃO E TECNOLOGIA DA INFORMAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

LUCIANE CARNEIRO

DESENVOLVIMENTO DE UM SISTEMA DE RECONHECIMENTO AUTOMÁTICO DE MÚSICAS - SRAM

Caxias do Sul
2012

LUCIANE CARNEIRO

DESENVOLVIMENTO DE UM SISTEMA DE RECONHECIMENTO AUTOMÁTICO DE MÚSICAS - SRAM

Trabalho de Conclusão de Curso para obtenção do Grau de Bacharel em Ciência da Computação da Universidade de Caxias do Sul.

André Gustavo Adami
Orientador

Caxias do Sul
2012

AGRADECIMENTOS

Agradeço aos meus pais Irides e Zeferino que sempre me incentivaram e acreditaram no meu trabalho, e principalmente por terem me educado para a vida com bons princípios, além da minha irmã Fabiane por todo o apoio. Ao Maurício, pelo carinho, pelo companheirismo e pela grande ajuda no final desta jornada. Ao meu professor orientador, André Gustavo Adami, pelos ensinamentos passados e pelo desenvolvimento deste projeto.

Agradeço também aos professores e colegas de curso e de centro pelos momentos de estudos e de descontração proporcionados, pelas amizades construídas, pelas discussões, e conhecimentos transmitidos.

RESUMO

Com o aumento do número de arquivos digitais de música, houve também a necessidade de sua organização e fácil acesso, e grandes repositórios são criados para o armazenamento desses arquivos. Este aumento torna operações manuais de busca cada vez mais difíceis de serem realizados. Para tanto, é importante que haja um método automático para o reconhecimento das informações principais de uma música, como o seu artista.

O problema de reconhecimento de músicas pode ser analisado com um problema de reconhecimento de padrões, onde são extraídas as características principais de cada música, que são analisadas, e após classificadas de acordo com modelos previamente analisados. Com isso, é possível o reconhecimento de informações de uma música automaticamente.

O objetivo deste trabalho é desenvolver um sistema que reconheça o artista de uma música a partir do seu áudio. Para tanto, é utilizado um modelo que utiliza modelos de misturas gaussianas e modelagem n-grama para geração dos modelos dos artistas e para a classificação.

Palavras-chave: Reconhecimento de Padrões, Reconhecimento de Artista, Modelo de Misturas Gaussianas, N-grama

ABSTRACT

With the growing number of music digital files, there is also the need of its organization and ease of access, and big repositories are being created to store these files. This growth turns manual operations of search more difficult to perform. Therefore, it is important to have an automatic method for recognition of main information of a song, with its artist.

The problem of recognition of songs can be analyzed as a problem of recognition of patterns, where the main features are extracted from each song. They are analyzed and then, they are classified according to models previously analyzed. Thus, it is possible to recognize informations from a song automatically.

The objective of this work is to develop a system that recognizes the artist of a certain song from its audio. For so, it is utilized a model that uses Gaussian mixture models and n-gram modeling to generate artist models and its classification.

Keywords: Pattern Recognition, Artist Recognition, Gaussian Mixture Model, N-gram

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1 INTRODUÇÃO

Com o crescente número de arquivos digitais de música, procedentes das mais diversas fontes, faz-se necessária uma organização em vastos repositórios (LOGAN, ELLIS e BERENZWEIG, 2003). O problema destes repositórios é que a recuperação das músicas ou de suas informações torna impossível operações manuais de busca. O que é mais comum para a organização desses arquivos é a utilização do recurso de metadados textual, onde podem ser inseridas várias informações sobre o seu conteúdo. Contudo, como a inserção destas informações é manual, é expressiva a quantidade de erros que podem ocorrer, demandando assim uma supervisão constante (CASEY, VELTKAMP, *et al.*, 2008). Por isso, é imprescindível a utilização de métodos automáticos para a recuperação de tais informações.

A utilização de métodos automáticos de recuperação de informações de músicas é importante também para a identificação de músicas que são tocadas em estações de rádio, cuja execução exige o pagamento de direitos autorais aos seus respectivos artistas. Em algumas cidades brasileiras, os processos de captação, gravação e identificação de músicas executadas pelas emissoras de rádios brasileiras, conduzidos pelo Ecad[1], são realizados através de um operador humano que ouve uma determinada rádio e realiza as anotações de horário, música e intérprete de forma manual (somente a lista de músicas é disponibilizada digitalmente). Este tipo de processo pode beneficiar-se da busca e reconhecimento automático da música, pois permite a operação em diversas rádios de forma simultânea e ininterrupta, inclusive para o controle das músicas que estão sendo tocadas, para, após, se fazer o devido pagamento de direitos autorais.

Há diversas informações sobre uma música que podem ser utilizadas para aplicações distintas. Dentre essas informações, há o artista ou banda que interpreta a música, o título da música, o álbum, o ano de lançamento, entre outros. Para sistemas de identificação de músicas, uma das informações mais utilizada é a do seu artista ou banda, pois as características únicas de cada artista ou banda facilitam o seu processo de identificação (KIM e WHITMAN, 2002).

[1] http://www.ecad.com.br

Um sistema que possa identificar automaticamente o artista de uma música a partir do seu áudio (referenciado como Sistema de Reconhecimento Automático de Músicas – SRAM) pode auxiliar no processo de organização dos arquivos de músicas. Com o reconhecimento automático das informações das músicas, a organização pode ser feita de forma automatizada, poupando trabalho manual de verificação de cada arquivo e podendo evitar classificações erradas (LOGAN, ELLIS e BERENZWEIG, 2003). Além disso, o reconhecimento automático de artista pode auxiliar no processo de busca de músicas por um determinado artista.

O processo de reconhecimento automático de música pode ser formulado como um problema de reconhecimento de padrões. Tal processo consiste basicamente em três etapas: pré-processamento dos sinais, que realiza a preparação do sinal da música (como conversão para o formato digital, redução de ruído, normalização do sinal, entre outros), extração de características, que converte o sinal de áudio em uma representação paramétrica do sinal a fim de que seja possível realizar uma classificação do sinal, e classificação das características, que é responsável por estimar uma medida de similaridade (que pode ser uma distância ou probabilidade) entre o sinal de áudio e os padrões de todos os artistas da base (KILL e SHIN, 1996) (WEBB, 1999). Com base nesta medida, é possível definir se o artista da música está ou não no banco de artistas.

Assim, neste trabalho, foi desenvolvido um sistema que permitirá o reconhecimento do artista de uma música a partir do seu conteúdo, para que a música possa ser catalogada e organizada. O atributo de entrada do sistema será o próprio arquivo de música, e nele será feita a análise dos sinais de áudio, para que se componha um padrão de artista que possa, após, ser reconhecido.

1.1 OBJETIVOS

O principal objetivo deste trabalho é desenvolver um sistema de reconhecimento automático de artista, a partir de uma música, em uma base pré-selecionada. Para que seja alcançado o objetivo principal, o trabalho se baseou nos seguintes objetivos específicos:

- Levantamento das técnicas de extração de características e classificação aplicadas a reconhecimento de música;

- Seleção e implementação de algoritmos de extração e classificação das características das músicas;
- Definição e implementação de uma metodologia de avaliação de resultados.

1.2 ESTRUTURA DO TRABALHO

O trabalho está dividido da seguinte maneira: no segundo capítulo, é feita uma revisão geral acerca do assunto de reconhecimento de padrões, onde são citadas as suas etapas e os métodos utilizados em cada etapa; no terceiro capítulo, são apresentados alguns métodos de reconhecimento de música, abordando o reconhecimento de artista; no quarto capítulo, é apresentado o modelo do SRAM, com a sua arquitetura principal, como solução para o problema proposto, além de testes realizados; por fim, no quinto capítulo, são feitas as considerações finais do trabalho.

2 RECONHECIMENTO DE PADRÕES

Reconhecimento de padrões é um campo de estudo cujo objetivo é a classificação de objetos em um número de categorias ou classes a fim de facilitar o processo de identificação (WEBB, 1999) (THEODORIDIS e KOUTROUMBAS, 2003). Esse processo envolve toda a investigação da formulação do problema, coleta de dados e, com isso, a realização da seleção e da classificação de características para tomada de decisões (WEBB, 1999). Ele é utilizado em diversas disciplinas da engenharia e da ciência, como biologia, psicologia, medicina, marketing, visão computacional, inteligência artificial e sensoriamento remoto (JAIN, DUIN e MAO, 2000).

No campo de estudo do reconhecimento de padrões é datado da década de 1960 o início efetivo do seu estudo e desenvolvimento. Antes, era basicamente um estudo teórico da área de estatística (THEODORIDIS e KOUTROUMBAS, 2003). Hoje em dia é um campo multidisciplinar, envolvendo áreas como engenharia, ciência da computação, psicologia e fisiologia, entre outras, além da própria estatística (WEBB, 1999).

O reconhecimento de padrões é utilizado em diversas aplicações. Dentre elas, há os sistemas de identificação de pessoas que, a partir de padrões como impressões digitais, a face e a íris do olho, podem reconhecer um padrão e, através de uma base pré-definida, identificam a pessoa com informações pertinentes para a aplicação. Outra aplicação é o reconhecimento de caracteres, chamado OCR (em Inglês *Optical Character Recognition*), que faz o reconhecimento de documentos manuscritos ou imagens de documentos e os disponibiliza para edição em modo digital (FRIEDMAN e KANDEL, 1999). Além desses, há o reconhecimento de fala, utilizado para a interação usuário-máquina. Nesse sistema, o sinal da fala é captado, transformado para reconhecer, dentro de padrões de fala, o que a pessoa está falando (JAIN e DUIN, 2004) ou então que pessoa está falando (definido como reconhecimento de locutor) (ADAMI, 2004).

O termo “padrão”, segundo (JAIN, DUIN e MAO, 2000), corresponde a uma entidade vagamente definida a qual pode ser atribuída um nome. Matematicamente, como explicam (WEBB, 1999) e (THEODORIDIS e KOUTROUMBAS, 2003), é a denominação

de um vetor de características *p*-dimensional $x = [x_1,...,x_p]^T$, onde ^{T} indica o vetor transposto e cada componente x_i denomina uma medida de uma característica única de um objeto, o qual pode ser classificado. Ainda, dependendo da abordagem utilizada, padrão pode ser um modelo de um objeto que se deseja reconhecer, ou até mesmo ele próprio. Exemplos de objetos que podem ser reconhecidos como padrões são: um sinal de áudio, uma impressão digital, a face de uma pessoa, um código de barras e um caractere manuscrito (vide Figura 2-1) (JAIN e DUIN, 2004).

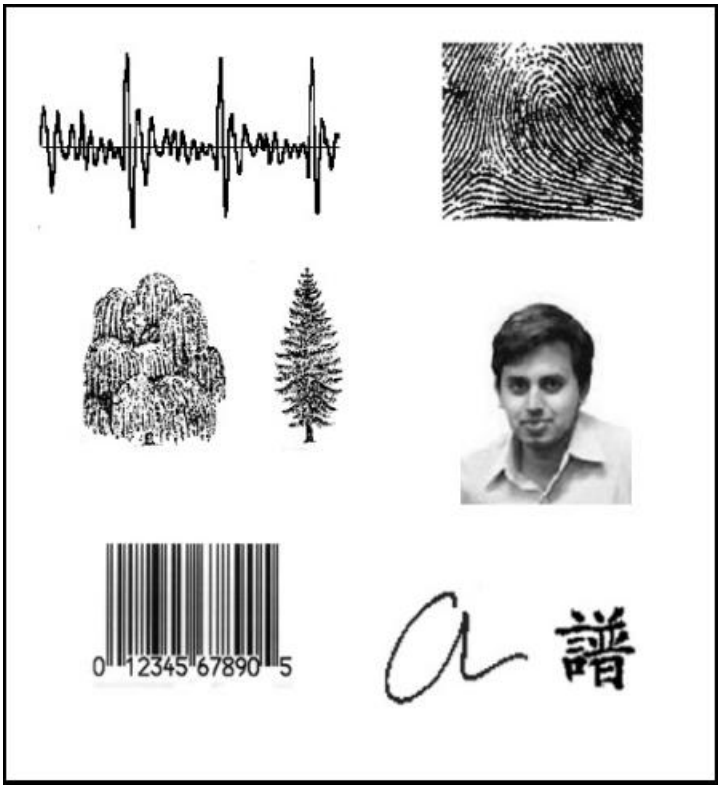

Figura 2-1: Exemplos de padrões (JAIN e DUIN, 2004).

Há duas fases principais no modelo básico do reconhecimento de padrões, ilustrado na Figura 2-2: fase de treinamento e fase de teste. Na fase de treinamento, o sistema recebe amostras de dados a serem reconhecidos (através da etapa de aquisição), faz o processamento, estima um modelo, ou classe, que é armazenado em uma base de dados. Na fase de teste, um padrão de teste é apresentado ao sistema a fim do mesmo reconhecer ou não o padrão. A fase de teste é composta principalmente dos seguintes passos: extração e seleção de características e classificação (JAIN, DUIN e MAO, 2000).

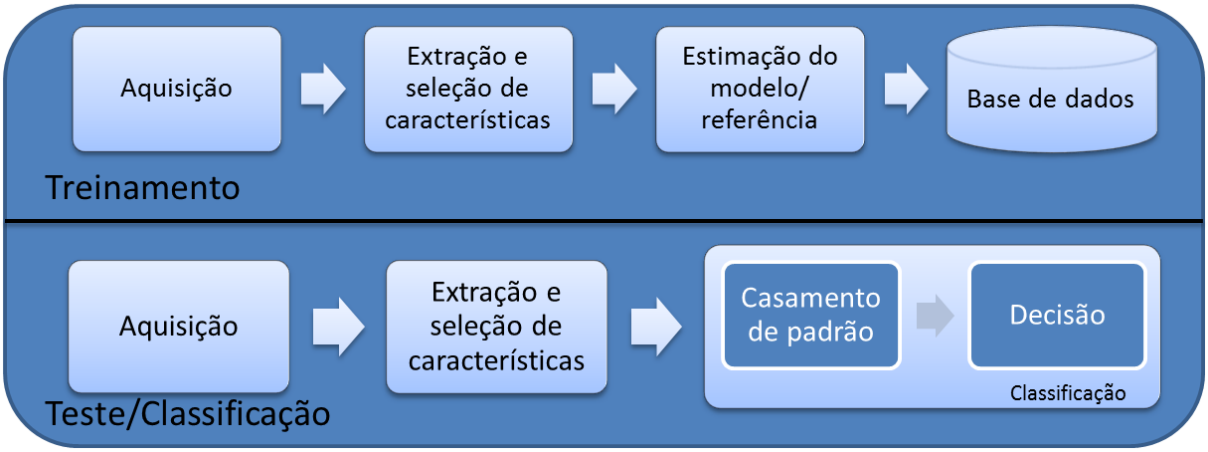

Figura 2-2: Modelo básico de reconhecimento de padrões.

Uma característica do classificador é a generalização, isto é, a sua capacidade de reconhecer objetos além dos fornecidos na fase de treinamento (JAIN, DUIN e MAO, 2000). Alguns fatores que contribuem para que o classificador não seja generalizável são os seguintes (JAIN, DUIN e MAO, 2000) (BISHOP, 2006):

- Maldição de dimensionalidade (em Inglês, *curse of dimensionality*), onde o número de características extraídas é maior que o número de padrões usados no treinamento;
- Sobre-treinamento (em Inglês, *overtraining*), onde o classificador é treinado com um conjunto excessivo de exemplos de padrões que possuem pequena variação em suas características. Neste caso, o reconhecedor aprende uma quantidade demasiada de detalhes desnecessários para o reconhecimento e, com isso, perde a capacidade de reconhecer semelhanças essenciais nos padrões (JAIN e DUIN, 2004);
- Sobre-ajuste (em Inglês, *overfitting*), onde o classificador gera superfícies de decisão mais complexas do que o requisitado pelo problema. Com isso, há um menor poder de generalização.

É importante destacar também que se a redução das características for excessiva o classificador pode perder seu poder de discriminação. Por isso, é necessário adaptar a complexidade do sistema de reconhecimento para a complexidade e o tamanho do conjunto dos dados a serem reconhecidos (CAMPOS e CESAR JÚNIOR, 2000) (BISHOP, 2006) (JAIN e DUIN, 2004).

2.1 ABORDAGENS DO RECONHECIMENTO DE PADRÕES

Existem várias abordagens ao problema de reconhecimento de padrões. As mais utilizadas são quatro: Casamento, Abordagem Estatística, Abordagem Sintática e Redes Neurais. A Tabela 2-1 (JAIN, DUIN e MAO, 2000) mostra as principais características de cada uma.

Tabela 2-1: Abordagens do reconhecimento de padrões.

Abordagem	Representação	Função de Reconhecimento	Critério de Decisão
Casamento	Amostras, *pixels*, curvas	Correlação, medida de distância	Erro de classificação
Abordagem Estatística	Características	Função discriminante	Erro de classificação
Abordagem Estrutural	Primitivas	Regras, gramática	Erro de aceitação
Redes Neurais	Amostras, *pixels*, características.	Função de rede	Erro médio quadrático

2.1.1 Casamento (*Template Matching*)

A abordagem do casamento de padrões (em Inglês, *Template Matching*) é a mais antiga e simples. O processo consiste basicamente em detectar a similaridade entre duas entidades diferentes, mas do mesmo tipo. É utilizado um modelo (*template*) como um protótipo do padrão a ser reconhecido. Na classificação, é feito o casamento deste modelo com o modelo de teste a ser reconhecido, levando em consideração variações como translação e mudanças de escala ou de tempo (JAIN, DUIN e MAO, 2000).

Esse tipo de abordagem demanda um grande custo computacional, e por isso é utilizada somente em casos específicos (JAIN, DUIN e MAO, 2000). Aplicações que utilizam a abordagem de casamento são o reconhecimento de palavras manuscritas (onde as palavras podem não estar escritas sempre no mesmo tamanho) e reconhecimento de palavras isoladas (onde as palavras não são faladas com a mesma duração de tempo). Exemplos de métodos da abordagem de casamento são a distância Euclideana, que reconhece um padrão pela menor distância dentro do espaço de características, e o *Dynamic Time Warping* (DTW), algoritmo que mede a similaridade de duas sequências através de um alinhamento temporal dinâmico

para lidar com as diferenças de tempo e espaço dos padrões (THEODORIDIS e KOUTROUMBAS, 2003).

2.1.2 Abordagem Estatística

Na abordagem estatística, a classificação é baseada em regras de decisão. Para cada padrão de teste, é calculada a probabilidade de ele ocorrer. Através do cálculo da distribuição de probabilidade, são criadas fronteiras de decisão que separam os padrões de classes diferentes. As regras de decisão são formuladas, por exemplo, por meio da transformação de uma probabilidade de ocorrência de uma classe (conhecimento a priori) em medidas de probabilidade condicional (conhecimento a posteriori) do tipo qual a probabilidade de uma classe ocorrer dado que um padrão ocorreu. Elas podem também ser formuladas por meio do cálculo de uma medida de erro de classificação esperado e da escolha de uma regra de decisão que minimiza essa medida. Exemplos de métodos desta abordagem são a estimativa da máxima verossimilhança e os classificadores Bayesianos (WEBB, 1999) (CAMPOS e CESAR JÚNIOR, 2000) (FRIEDMAN e KANDEL, 1999). A abordagem estatística tem sido mais intensivamente estudada e utilizada em diversas áreas (JAIN, DUIN e MAO, 2000).

2.1.3 Abordagem Sintática

A abordagem sintática trabalha com padrões mais complexos, onde eles podem ser adaptados em uma perspectiva de hierarquia. Essa hierarquia é composta por um padrão, que é composto por sub-padrões mais simples, e assim por diante. Os sub-padrões elementares são denominados primitivas, e as inter-relações entre essas primitivas é que define o padrão como um todo. A vantagem da utilização desta abordagem é que provém uma descrição de como os padrões são formados, a partir das primitivas. Além disso, uma larga coleção de padrões complexos pode ser descrita em um pequeno número de primitivas e regras gramaticais, que chega à composição do padrão, o que, nos dois casos, facilita o processo de classificação (JAIN, DUIN e MAO, 2000). Os métodos para se trabalhar com a abordagem sintática são baseadas em gramáticas ou grafos.

2.1.4 Redes Neurais

As redes neurais tem o mesmo princípio de funcionamento das redes de neurônios do cérebro humano, sendo desenvolvidas em cima de modelos matemáticos do processo cognitivo humano. Elas consistem em um grande número de neurônios, também chamados de células, nodos ou unidades, onde há uma ligação direta entre cada neurônio. Essa ligação é feita de acordo com um peso atribuído a cada neurônio, que varia em relação ao tipo de dado a ser analisado. A Figura 2-3 mostra uma rede neural simples, onde três neurônios (X_1, X_2 e X_3) enviam um sinal cada um para um quarto neurônio (Y), e este processa os sinais recebidos e envia um novo sinal para dois outros neurônios (Z_1 e Z_2). Cada um dos neurônios envia um único sinal para vários neurônios ao mesmo tempo (FRIEDMAN e KANDEL, 1999). O modelo de redes neurais pode ser comparado a uma grande rede paralela de pequenos processadores com diversas interligações (JAIN, DUIN e MAO, 2000).

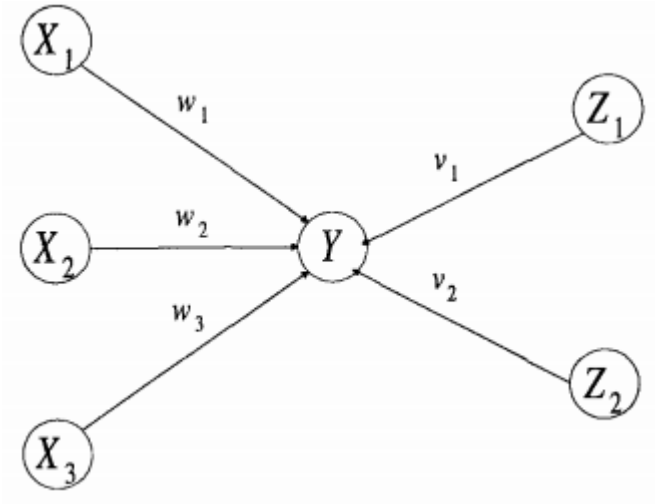

Figura 2-3: Exemplo de rede neural simples (FRIEDMAN e KANDEL, 1999).

Esse tipo de abordagem tem a habilidade de ler entradas complexas não lineares, usar procedimentos sequenciais de treinamento e se adaptar ao tipo de dado. A extração de características e a classificação podem ser mapeadas juntas em arquiteturas de redes neurais para uma maior eficiência (FRIEDMAN e KANDEL, 1999) (JAIN, DUIN e MAO, 2000).

Pode-se citar como exemplos de métodos que trabalham com a abordagem de redes neurais os métodos da chamada família *feed-forward network*, que inclui o *Perceptron* Multi-Camadas (*Multilayer Perceptron*) e as redes *Radial-Basis Function* (RBF), organizadas em

camadas e com conexões unidirecionais entre essas camadas. Além desses, há também o *Self-Organizing Map* (SOM), ou *Kohonen-Network*, normalmente usado para clusterização de dados e mapeamento de características (JAIN, DUIN e MAO, 2000).

2.2 EXTRAÇÃO E SELEÇÃO DE CARACTERÍSTICAS

Nas etapas de extração e de seleção de características, os dados de entrada são analisados e preparados para a etapa de classificação. Após a aquisição dos dados é feito o seu pré-processamento. Essa etapa é importante para as próximas, pois é onde é feita a eliminação de ruídos e de outros tantos fatores de confusão no reconhecimento. Em seguida, inicia-se a extração de características. O processo de extração das características tem como objetivo calcular e coletar os atributos da entrada pré-processada e dispor em um compacto vetor, denominado vetor de características. Com isso, ele elimina as informações irrelevantes e ao mesmo tempo preserva as informações mais relevantes (KILL e SHIN, 1996).

A extração de características tem sua maior importância no auxílio ao bom desempenho do classificador. É responsável pela captura, dentro dos espaços de projeção do sinal pré-processado, dos atributos (ou características) que sejam mais relevantes para a discriminação das classes. Com esses atributos selecionados, o espaço de projeção diminui em relação ao original, isto é, há uma redução na dimensão do vetor de características (KILL e SHIN, 1996) (JAIN, DUIN e MAO, 2000).

Segundo (KILL e SHIN, 1996), boas características devem, preferencialmente:

- Possuir grande distância média entre as classes e pequena variância dentro de cada classe;
- Ser insensíveis ao ruído (e.g., efeitos do sensor de coleta, informações não relevantes);
- Ter baixo custo computacional na sua medição;
- Ter covariância igual a zero entre as suas medidas;
- Ser definidas matematicamente; e
- Ser explicáveis em termos físicos.

A seleção de características, por sua vez, encarrega-se de identificar e selecionar as características com maior poder de discriminabilidade, para que se minimize o problema denominado maldição de dimensionalidade (CAMPOS e CESAR JÚNIOR, 2000). Nesta etapa, é feita uma nova análise das características extraídas anteriormente, para se encontrar as que tenham menos custo de processamento (KILL e SHIN, 1996) (FUKUNAGA, 1990).

Os passos básicos da seleção de características, cuja ordem de execução está ilustrada na Figura 2-4, são os seguintes (DASH e LIU, 1997):

a. Geração da nova característica selecionada;
b. Avaliação, que mede a qualidade da característica gerada, comparando-a com a melhor encontrada anteriormente. Caso a nova seja melhor que a anterior, ela a substitui.
c. Critério de parada, que define o término do ciclo de geração e avaliação; e
d. Validação do resultado final.

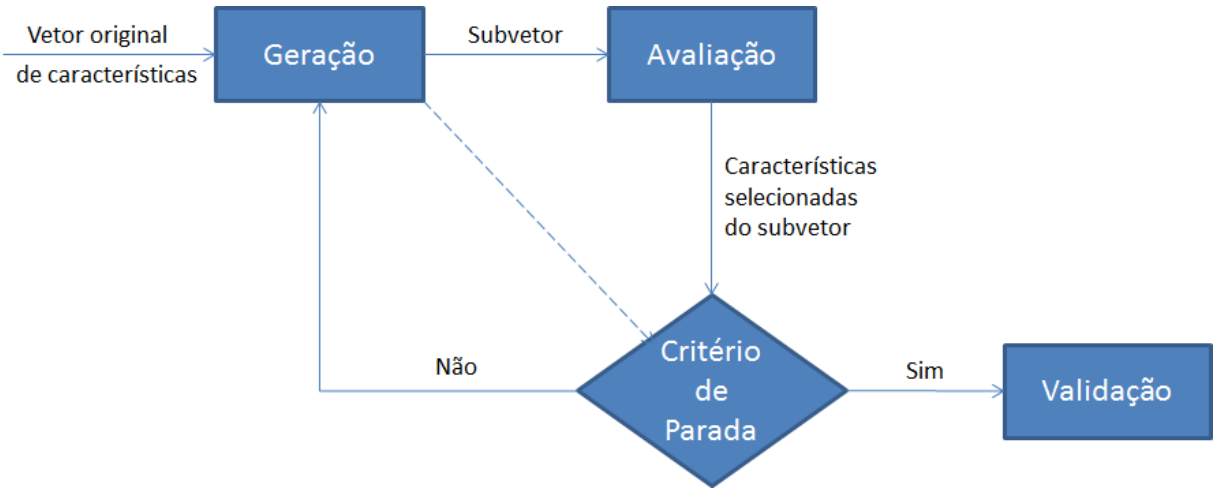

Figura 2-4: Processo de seleção de características.

Há diversos métodos utilizados para a extração de características, para a seleção de características ou para ambas. Um dos exemplos é o Principal Component Analysis (PCA), método não-supervisionado que implementa a transformação de Karhunen-Loève e minimiza o erro médio quadrático da representação. Outro exemplo é o *Linear Discriminant Analysis*, método supervisionado, baseado na função discriminante linear de Fisher, que utiliza a informação da classe associada a cada padrão para extrair as características mais diferentes umas das outras e, assim, maximiza o critério de separabilidade entre classes (WEBB, 1999)

(JAIN, DUIN e MAO, 2000). Ambos os métodos projetam o espaço de características original em outro tal que aumenta a capacidade discriminatória das características.

2.3 CLASSIFICAÇÃO

O classificador é uma das principais etapas do reconhecimento de padrões, cujo principal objetivo é identificar a qual classe cada padrão pertence. As possíveis classes utilizadas pelo classificador são normalmente melhor definidas durante o projeto do reconhecedor, com base nas medidas do objeto a ser reconhecido (KILL e SHIN, 1996).

A etapa de classificação é dividida em duas sub-etapas: o casamento de padrão e a decisão. Em sua base, o casamento de padrão é a comparação entre um padrão de referência e o padrão que se deseja classificar. Há diversos métodos que fazem o casamento de padrões, os quais envolvem técnicas como de cálculo de distância entre os padrões, cálculo de probabilidades de o padrão ocorrer, ou ainda cálculo de funções que identificam fronteiras de decisão (JAIN, DUIN e MAO, 2000). Dependendo da margem de erro que for definida, se houver alguma classe a qual o padrão se assemelha, isso significa que o padrão foi reconhecido (WEBB, 1999). Este é o princípio da sub-etapa de decisão.

Quanto ao tipo de distribuição assumida para o modelo, a classificação se divide em paramétrica e não paramétrica. Quanto à forma de treinamento do classificador, a classificação se divide em supervisionada e não supervisionada.

2.3.1 Classificador Paramétrico e Não Paramétrico

Classificadores paramétricos trabalham em função da distribuição das classes. Os métodos de classificação paramétrica utilizam a estimação da máxima verossimilhança, ou seja, calculam o escore de verossimilhança para cada classe e selecionam a classe com maior escore. Podem ser lineares ou quadráticos (FUKUNAGA, 1990) (KILL e SHIN, 1996). Exemplos de classificadores paramétricos são os métodos de análise de discriminante, que formulam critérios de separabilidade através da análise da dispersão de dados dentro das classes, entre as classes, ou ainda através de matrizes de misturas de dispersão (FUKUNAGA,

1990). Além destes, há o classificador linear de Bayes, que utiliza a teoria de decisão Bayesiana para minimizar a probabilidade de erro (JAIN, DUIN e MAO, 2000).

Os classificadores não paramétricos são utilizados quando não se tem conhecimento exato da distribuição estatística das classes. Eles definem os parâmetros de classificação pelos dados das próprias classes (JAIN, DUIN e MAO, 2000) (KILL e SHIN, 1996). Exemplos da classificação não paramétrica são as funções discriminantes, as quais identificam fronteiras de decisão entre as classes, a função de vizinhança mais próxima (em Inglês, *K-nearest neighbor*), que constrói a fronteira de decisão baseando-se nos próprios vetores de treinamento, utilizando normalmente a distância Euclidiana para encontrar o padrão mais próximo, e as árvores de decisão, que selecionam, iterativamente, características mais destacadas em cada nó de uma árvore, normalmente binária, formada pelas características dos padrões (JAIN, DUIN e MAO, 2000).

2.3.1.1 Modelo de Mistura Gaussiana

O modelo de mistura Gaussiana, ou *Gaussian Mixture Model* (GMM), é um modelo estatístico usado geralmente como modelo paramétrico da distribuição de probabilidade de medidas contínuas ou características em sistemas biométricos, como em um sistema de reconhecimento de locutor, devido à sua capacidade de representar uma grande classe de distribuições de amostra (JAIN, DUIN e MAO, 2000) (KILL e SHIN, 1996) (REYNOLDS, 2008). É uma técnica de classificação composta por um número finito de funções de probabilidade Gaussianas multivariadas, cada uma parametrizada por um vetor de médias *d*-dimensional e uma matriz de covariância *d x d*. Cada Gaussiana possui ainda um peso, que determina o grau de pertinência da modelagem. A matriz de covariância pode ser representada de forma completa ou somente através de sua diagonal (REYNOLDS, QUATIERI e DUNN, 2000). O GMM, segundo (REYNOLDS, QUATIERI e DUNN, 2000), pode ser visto como um híbrido das técnicas paramétricas e não paramétricas.

Os parâmetros do GMM são estimados através do algoritmo de iteração chamado maximização da expectativa, ou *Expectation-Maximization* (EM), ou ainda pela estimativa *maximum a posteriori* a partir de um modelo anterior bem treinado (REYNOLDS, 2008). A idéia básica do algoritmo de maximização da expectativa é estimar, a partir de um modelo λ, um novo modelo $\bar{\lambda}$, de modo que a verossimilhança do novo modelo seja maior que a do primeiro. A cada iteração, o modelo estimado passa a ser o primeiro, e um novo modelo é

estimado. Esse processo se repete até que um limiar de convergência seja alcançado. Os pesos das misturas, as médias e as covariâncias são reestimados a cada iteração do algoritmo, para o aumento da verossimilhança do modelo. A estimativa *maximum a posteriori* é dividida em duas etapas: na primeira, é feito o cálculo das estimativas de estatísticas suficientes, como no algoritmo de maximização da expectativa; na segunda etapa, a adaptação, ocorre a combinação das novas estimativas de estatísticas suficientes com as antigas (REYNOLDS, 2008).

2.3.2 Classificação Supervisionada e Não Supervisionada

Na classificação supervisionada, o padrão de entrada é identificado como membro de uma classe definida pelos padrões de treinamento, que são rotulados com suas classes. Esses padrões são definidos de forma supervisionada, através da fase de treinamento (JAIN, DUIN e MAO, 2000) (WEBB, 1999). Assim, qualquer método que necessita construir um modelo da classe a ser reconhecida, será classificado como supervisionado. Por exemplo, o modelo de misturas Gaussianas necessita de dados de cada classe para construir os modelos. Suas etapas na fase de treinamento são as seguintes:

a. Escolha de um conjunto de treinamento;
b. Escolha dos parâmetros relevantes a serem medidos;
c. Obtenção da função discriminante, que pode ser obtida por método paramétrico ou não paramétrico;
d. Eliminação dos parâmetros não relevantes;
e. Testes com objetos fora do conjunto de treinamento.

Na classificação não supervisionada o padrão é associado a uma classe que é aprendida com base na similaridade entre os padrões de treinamento. São estabelecidos agrupamentos naturais no espaço de características, a partir da medida de diferentes parâmetros dos objetos, e tenta-se buscar os limites naturais destes agrupamentos, sem o conhecimento de suas classes (JAIN, DUIN e MAO, 2000) (BISHOP, 2006) (FUKUNAGA, 1990). Exemplos de classificação não supervisionada são o classificador de Bayes não supervisionado, e o agrupamento por *k*-médias, onde cada classe é representada por um vetor,

que é a média de todos os vetores daquela classe (JAIN, DUIN e MAO, 2000) (WEBB, 1999).

2.4 MEDIDAS DE DESEMPENHO

Medidas de desempenho são utilizadas para avaliar quantitativamente o desempenho de um classificador. As medidas de desempenho podem variar com o domínio do problema. As medidas de desempenho podem incluir a taxa de acerto ou de erro. Entretanto, estas medidas devem ser estimadas sobre dados não utilizados na fase de treinamento, para se conhecer a real eficiência em aplicações futuras (THEODORIDIS e KOUTROUMBAS, 2003) (OLSON e DELEN, 2008).

Matriz de Confusão

A matriz de confusão mostra o número de classificações corretas versus as classificações previstas para cada classe, sobre um conjunto de exemplos. O número de acertos, para cada classe, também chamado de positivo verdadeiro (em Inglês, *true positive* – TP) localiza-se na diagonal principal da matriz e os demais elementos, também chamados de falso positivo (em Inglês, *false positive* – FP), representam erros na classificação. A matriz de confusão de um classificador ideal possui todos esses elementos iguais a zero uma vez que ele não comete erros (KILL e SHIN, 1996) (OLSON e DELEN, 2008).

A Figura 2-5 (OLSON e DELEN, 2008) mostra um exemplo de matriz de confusão de duas classes, onde se pode reconhecer, além dos TP e FP, os seguintes termos:

- Verdadeiro negativo (em Inglês, *true negative* – TN): quantidade de classes classificadas corretamente como negativas;
- Falso negativo (em Inglês, *false negative* – FN): quantidade de classes classificadas incorretamente como negativas;
- Acurácia (em Inglês, *Accuracy*): estimativa do número de classes classificadas corretamente sobre a soma de todas as classes. Sua fórmula é: $\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$;

- Precisão (em Inglês, *Precision*): estimativa do número de classes positivas classificadas corretamente sobre todas as classes classificadas como positivo. Sua fórmula é: $\text{Precision} = \frac{TP}{TP + FP}$;
- Revocação (em Inglês, *Recall*): estimativa do número de classes positivas classificadas corretamente sobre todas as classes que são realmente positivas. Sua fórmula é: $\text{Recall} = \frac{TP}{TP + FN}$.

		Classe Verdadeira	
		Positiva	Negativa
Classe Estimada	Positiva	Contagem de verdadeiros positivos (TP)	Contagem de falsos positivos (FP)
	Negativa	Contagem de falsos negativos (FN)	Contagem de verdadeiros negativos (TN)

Figura 2-5: Exemplo de matriz de confusão de duas classes.

2.4.1 Métodos para Treinamento e Testes

Quando se mede o desempenho de um classificador, é necessário que se tenham dados para utilização tanto na fase de treinamento quanto na de teste, e que preferencialmente estes dados não sejam os mesmos. Caso contrário, a estimativa de erro é mínima ou zero, pois os padrões que estão sendo testados já foram igualmente classificados, e o classificador, neste caso, não é generalizável. Existem diversos métodos que definem como os dados serão utilizados para as fases de treinamento e de teste (THEODORIDIS e KOUTROUMBAS, 2003), dentre eles:

- Método Holdout: neste método, divide-se a base de dados em duas partes, onde normalmente 2/3 dos dados são utilizados na fase de treinamento e 1/3 na fase de teste. No caso de utilizar a abordagem de redes neurais, normalmente divide-se a base de dados em três partes, onde uma é utilizada na fase de treinamento,

outra para validação e a última na fase de teste. É utilizado em bases com grande quantidade de dados. (OLSON e DELEN, 2008).

- *Bootstrap*: utilizado em bases menores de dados, o método de *bootstrap* gera novos dados a partir de uma amostragem aleatória com substituição dos dados já existentes para serem utilizados na fase treinamento, enquanto que os dados originais são utilizados na fase de teste. A estimativa da taxa de erro é calculada por contagem de todos os erros e dividindo a soma do número total de amostras de teste utilizadas (THEODORIDIS e KOUTROUMBAS, 2003) (OLSON e DELEN, 2008).
- Validação cruzada (*K-fold*): no método de validação cruzada, os dados são divididos aleatoriamente em *k* conjuntos disjuntos, e os procedimentos de treinamento e aferição são repetidos durante *k* rodadas, nas quais o conjunto *k* é usado para aferição do erro e os demais para treinamento do algoritmo (WEBB, 1999).

3 RECONHECIMENTO DE MÚSICA

A música é hoje um dos tipos de informação mais populares da Internet e, por isso, é importante que haja métodos de reconhecimento automático, tanto para busca quanto para organização (CASEY, VELTKAMP, *et al.*, 2008). Várias pesquisas já foram realizadas, e há diversos métodos para reconhecimento de música. O princípio básico do reconhecimento de música está no reconhecimento de padrões, sendo este basicamente adaptado para a música.

A área que trata da recuperação das informações de arquivos de música é chamada recuperação de informação de música, em Inglês *Music Information Retrieval* (MIR) (TYPKE, WIERING e VELTKAMP, 2005). As pesquisas nesta área envolvem técnicas efetivas de organização e recuperação de áudio digital, que pode ser representado simbolicamente (notações, arquivos MIDI), através do próprio arquivo de áudio, de partituras ou de metadados (FUTRELLE e DOWNIE, 2002). Neste trabalho, será abordada a recuperação de informação de música baseada em conteúdo, em Inglês, *Content-Based Music Information Retrieval*.

Há diversas informações sobre uma música que podem ser utilizadas para diferentes aplicações. A informação do artista ou banda normalmente é a mais utilizada para a identificação de uma música, pois as características únicas de cada artista ou banda facilitam o seu processo de identificação (KIM e WHITMAN, 2002). Há também as seguintes informações que são utilizadas (CANO, BATLLE, *et al.*, 2005):

- Título da música;
- Informações sobre o andamento da música, como ritmo, timbre, melodia e harmonia;
- Informações sobre a composição da música como nome do compositor, ano de composição, gravadora ou performance ao vivo;
- Informações sobre o trabalho do artista, como nome do álbum, imagem da capa e biografia do artista.

Existem duas abordagens para o reconhecimento de música: a convencional e a de métodos *fingerprint*. A abordagem convencional trabalha de acordo com os modelos de

reconhecimento de padrões, onde um conjunto de características, isto é, *n* vetores de características, é extraído para que seja classificado. Já na abordagem de métodos *fingerprint*, é gerada uma única característica que será comparada, conforme ilustra a Figura 3-1.

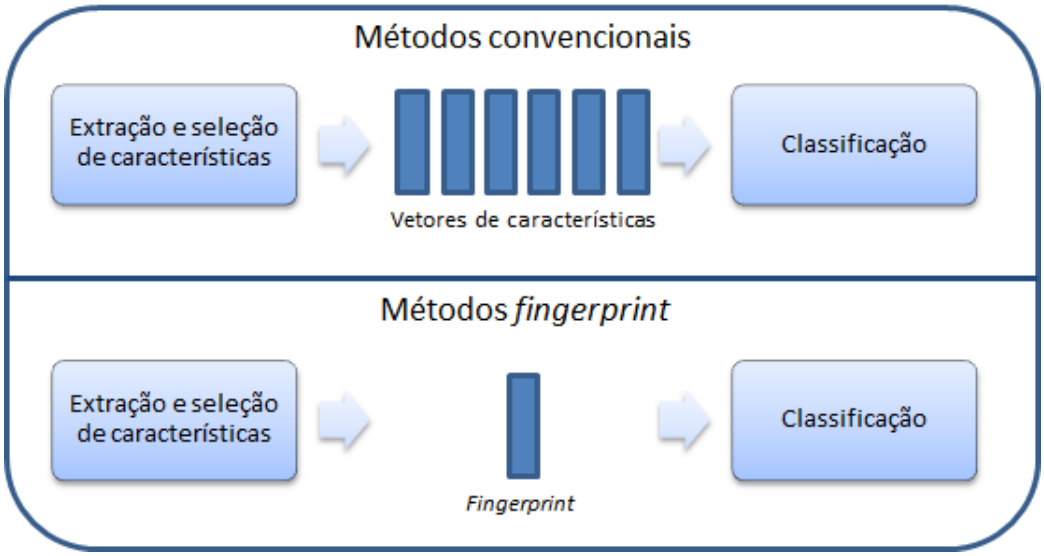

Figura 3-1: Comparação entre métodos convencionais e métodos *fingerprint*.

3.1 EXTRAÇÃO DE CARACTERÍSTICAS

A extração de características na música é uma representação de segmentos de áudio em vetores de características formados por números. Esse processo é feito usando a análise de tempo e frequência, através de técnicas como a Transformada de Fourier (TF), ou aplicação de filtros e escalas.

Diferentes descritores de áudio vêm sendo estudados e utilizados como vetores de características do sinal. Alguns dos mais citados, por exemplo, são os descritores do padrão MPEG-7 e os descritores de características psicoacústicas como o timbre, envelope temporal e espectral e Coeficientes Mel-Cepstrais (MFCCs). Em diversos trabalhos de reconhecimento de música, as características extraídas são os coeficientes mel-cepstrais (em Inglês, *Mel-Frequency Cepstral Coefficients* – MFCCs) (BERENZWEIG, LOGAN, *et al.*, 2004). Originalmente utilizados para reconhecimento de fala, os MFCCs foram desenvolvidos para representar o processo auditivo humano para o reconhecimento de fala. O sucesso do uso de MFCC deve-se ao fato de ter sido desenvolvido para modelar o envelope espectral do sinal de uma forma muito compacta, suprimindo a frequência fundamental, levando em conta a

percepção não linear do som pelo ouvido humano. (JENSEN, CHRISTENSEN, *et al.*, 2009). Os MFCCs são obtidos através da *Discrete Cosine Transform* (DCT), ou Transformada Discreta de Cosenos nos logaritmos de energia dos espectros de cada quadro extraído do sinal de áudio (LOGAN, 2008).

3.2 MÉTODOS CONVENCIONAIS

Há diversos trabalhos na área de reconhecimento de música, sendo que todos possuem entre si algumas características em comum, como a sequência da sua execução, que se assemelha com os processos de reconhecimento de padrões. Alguns métodos utilizados também são comuns, tanto de extração e seleção de características, quanto de classificação e de medidas de desempenho.

Segundo (MÜLLER, ELLIS, *et al.*, 2011), o sinal pode ser processado em sua totalidade ou pode ser separado em componentes individuais (como notas de um instrumento em particular) para serem processados da mesma forma que os sinais monofônicos. Dentre elas, pode-se citar a extração da melodia principal ou extração da voz do cantor, que faz inicialmente uma representação do sinal através da transformada rápida de Fourier, ou *fast Fourier transform* (FFT), que é uma versão da transformação discreta de Fourier, ou *discrete Fourier transform* (DFT), que decompõe uma sequência de sinais em componentes de diversas frequências.

O modelo apresentado por (JENSEN, CHRISTENSEN, *et al.*, 2009), extrai do sinal do áudio os MFCCs, que são representados no espaço de características através do modelo de mistura de Gaussianas (GMM). Para medir a distância entre os modelos de mistura Gaussiana, é utilizado o método da distância de *Kullbak-Lieber*. Após, é utilizado o método do vizinho mais próximo para a classificação. Para avaliação do modelo, através da quantidade de MFCCs utilizados, é medida a sua acurácia, de acordo com os resultados obtidos.

No trabalho de (KIM e WHITMAN, 2002), um dos conceitos trabalhados é o da utilização do modelo de mistura de Gaussianas (GMM) para o reconhecimento do artista da música. É utilizado juntamente o algoritmo chamado *Expectation Maximization* (EM), um

algoritmo interativo que converge para os parâmetros que mais identificam as Gaussianas, de acordo com a função de verossimilhança.

3.3 MÉTODOS *FINGERPRINT*

Sistemas de reconhecimento de música baseados em *fingerprint* extraem determinadas características dos sinais de áudio que permitam gerar uma informação única, isto é, um único vetor de características, chamada de *fingerprint*, que identifica exclusivamente uma música, independentemente do seu formato. O termo *fingerprint* é utilizado, pois é similar à impressão digital humana, que identifica uma única pessoa. O *fingerprint* gerado é comparado depois com uma base de dados para fazer o reconhecimento da música, processo este que pode ser visto como um método de casamento de padrões (*template matching*) (CANO, BATLLE, *et al.*, 2002).

Sua arquitetura básica, apresentada na Figura 3-2, possui dois processos distintos: extração do *fingerprint* e algoritmo de casamento. Na extração, é criado um vetor de características a partir de pequenos segmentos do áudio original. É importante que estas características sejam exclusivas para cada áudio e robustas contra distorções. O algoritmo de casamento, por sua vez, pesquisa na base de dados e faz o casamento com o *fingerprint* a ser reconhecido, para encontrar o que mais se assemelha ou o que seja idêntico. Utilizando este tipo de arquitetura, versões distorcidas de uma gravação podem ser reconhecidas como sendo a mesma música (CANO, BATLLE, *et al.*, 2005) (TYPKE, WIERING e VELTKAMP, 2005).

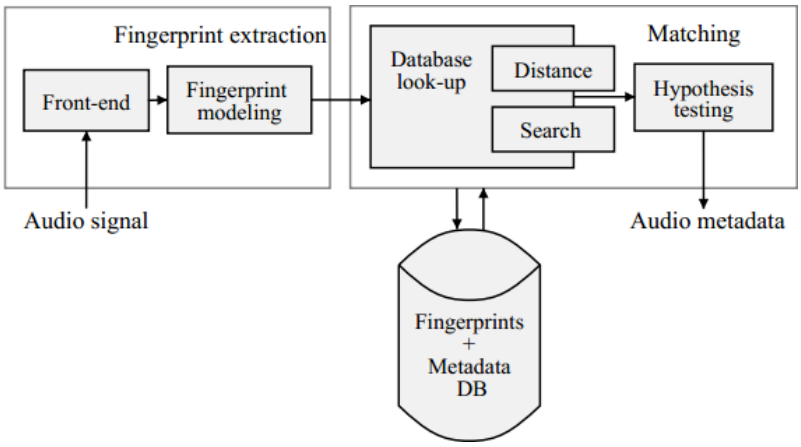

Figura 3-2: Arquitetura básica de métodos *fingerprint.*

Para extração do fingerprint e para o algoritmo de casamento, há alguns métodos que são mais utilizados. Na extração, são utilizados métodos que geram MFCCs. O algoritmo de casamento é desenvolvido através de medidas de similaridade, ou seja, casamento de padrões, que utiliza cálculos como a distância Euclidiana, a classificação pelo vizinho mais próximo ou a distância de Manhattan (CANO, BATLLE, *et al.*, 2005).

Outra característica importante nos métodos *fingerprint* é o modo como essas informações são buscadas e comparadas na base de dados. Podem ser utilizadas medidas simples de similaridade, ou então a técnica de índices invertidos. É possível, inclusive, dividir a base entre *fingerprints* que possuem mais chance de serem buscados e os restantes, para que a busca sempre inicie pela base dos que possuem mais chance. Após a busca, ainda são utilizados as medidas de desempenho *Precision* e *Recall* para que se tenha conhecimento da precisão da escolha (CANO, BATLLE, *et al.*, 2005).

Há diversos programas que utilizam métodos *fingerprint* para reconhecimento de músicas. O Gracenote[2] identifica, em uma base de dados de mais de 130 milhões de músicas (segundo dados do site oficial), os metadados de músicas tocadas no computador a partir de um CD. Na área de dispositivos móveis (e. g., celulares e tablets), o Shazam[3] gera um *fingerprint* a partir da gravação do trecho de uma música que estiver sendo reproduzida em um ambiente como um bar, e o compara com uma base de dados atualizada diariamente. Com o resultado, é possível fazer a compra e o download da música reconhecida. Neste programa, há também a opção de buscar diretamente na base de músicas, através de informações como nome do artista, título da música e nome do álbum. (WANG, 2003) descreve em linhas gerais o funcionamento do reconhecimento do programa Shazam, comenta que o mésmo método de geração de *fingerprint* é utilizado para as músicas a serem adicionadas à base e para as amostras a serem reconhecidas. A base de dados é identificada através de índices, o que facilita a busca dos *fingerprints*. Outro programa que utiliza o mesmo tipo de reconhecimento do Shazam é o SoundHound[4], com o diferencial de que também reconhece a música cantada pelo próprio usuário.

Uma vantagem dos métodos baseados em *fingerprint* é a sua rápida busca, que é otimizada por ser somente uma característica a ser comparada. Contudo, há um custo maior

[2] http://www.gracenote.com/
[3] http://www.shazam.com/
[4] http://www.soundhound.com/

na extração do *fingerprint*, devido à complexidade da sua formação (CANO, BATLLE, *et al.*, 2005) (TYPKE, WIERING e VELTKAMP, 2005). Outra desvantagem dos métodos *fingerprint* é o custo de armazenamento na base de todos os *fingerprints* já gerados, e a necessidade de constante atualização do conhecimento desta base assim que um *fingerprint* de uma nova música é inserido (GOMES, CANO, *et al.*, 2003).

3.4 RECONHECIMENTO DE ARTISTA

A área de reconhecimento de artista tem recebido considerável atenção. Sistemas de reconhecimento de artista são normalmente desenvolvidos para identificar músicas com instrumentação, ritmo ou melodia similares, que identificam um artista, mesmo que isso não seja totalmente garantido (JENSEN, 2009).

No trabalho de (LOGAN, 2005), foi apresentado um modelo de identificação de artista através da classificação pelo método do vizinho mais próximo. MFCCs foram extraídos de cada áudio e, a partir deles, foi criada uma assinatura para cada música, onde após foi feito o treinamento de cada artista.

No trabalho de (WHITMAN, FLAKE e LAWRENCE, 2001), foram utilizadas redes neurais para treinamento dos artistas, combinadas com o método *Support Vector Machine* (SVM). Foi utilizada também uma *engine* que determinou os metadados de uma música, e auxiliou na classificação.

4 SISTEMA DE RECONHECIMENTO AUTOMÁTICO DE MÚSICAS

O Sistema de Reconhecimento Automático de Músicas (SRAM) é proposto para que seja possível o reconhecimento do artista de uma música a partir do seu áudio. Para o processamento do sistema foi utilizada a abordagem estatística, cuja decisão é baseada nas probabilidades calculadas para cada padrão gerado.

O sistema é dividido em duas etapas. Na etapa de extração de características, o sinal da música é convertido em uma sequência discreta de símbolos que descrevem as dinâmicas acústicas da mesma utilizando uma representação do espaço de músicas. Na etapa de classificação, o sistema calcula, através de um Modelo Universal de Músicas (MUM) as probabilidades da música de teste ter sido gerada por cada um dos modelos de música a serem reconhecidas. Em seguida, o sistema decide pela música cuja probabilidade for a maior de todas. A Figura 4-1 ilustra a arquitetura principal do SRAM.

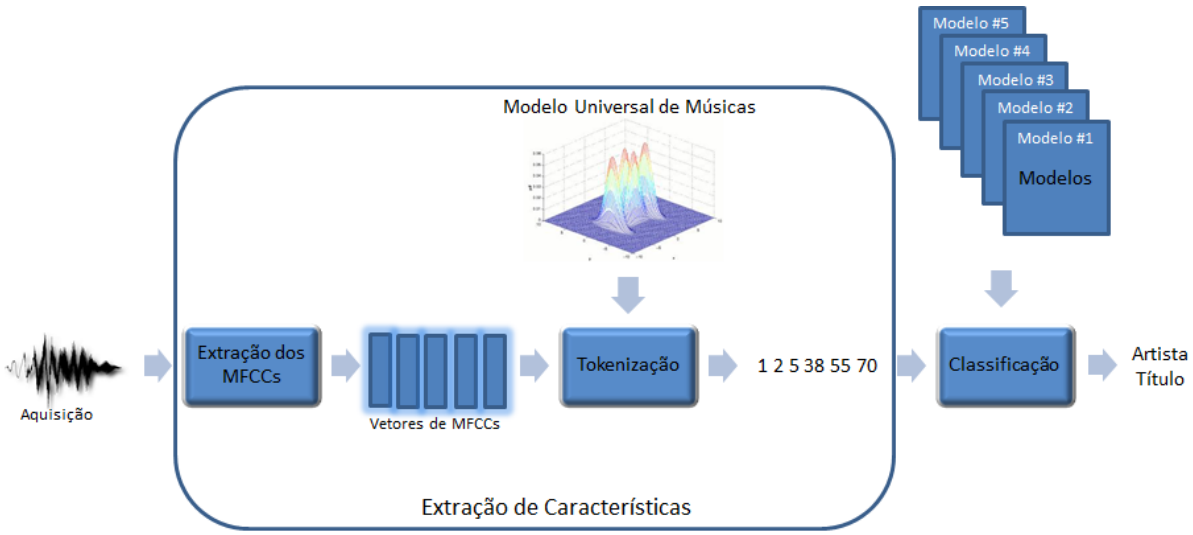

Figura 4-1: Arquitetura principal do SRAM.

4.1 EXTRAÇÃO DE CARACTERÍSTICAS

O objetivo da extração de características é gerar uma sequência de símbolos a partir do sinal de áudio da música. O processo pode ser dividido em duas etapas. Na primeira etapa, um modelo do espaço de características das músicas, chamado de Modelo Universal de Músicas, é estimado. Este modelo é construído utilizando misturas Gaussianas estimadas a partir das características espectrais (isto é, os coeficientes Mel Cepstrais) de um conjunto de músicas (REYNOLDS, QUATIERI e DUNN, 2000). O objetivo deste modelo é produzir uma sequência de símbolos (XIANG, 2002), (XIANG, 2003). Na segunda etapa, a sequência de símbolos é estimada para cada música a partir do Modelo Universal de Músicas onde, para cada característica da música é identificado o componente gaussiano que produziu a maior verossimilhança. A sequência de símbolos será então definida pelos índices dos componentes do Modelo Universal de Músicas conforme a sequência de características. Este processo é chamado de tokenização.

4.2 CLASSIFICAÇÃO

No processo de classificação, utiliza-se a modelagem *n*-grama para gerar o Modelo Universal de Músicas e os modelos de cada artista. Após, é aplicado o teste de detecção para determinar qual o artista de cada música de teste.

4.2.1 Modelagem N-grama

A modelagem *n*-grama, muito utilizada em reconhecimento de fala (GOLD e MORGAN, 2000), utiliza a co-ocorrência de *n* símbolos adjacentes (onde *n* é um número inteiro) para caracterizar a distribuição de probabilidade em sequências de símbolos.

Em problemas estocásticos, uma vez vista uma quantidade significativa de sequências de símbolos é possível calcular a probabilidade de *n* símbolos ocorrerem em sequência. Assim, é possível saber qual a probabilidade de um símbolo ocorrer dado que um símbolo (ou símbolos) ocorreu imediatamente antes (MANNING e SCHÜTZE, 1999). Assim,

a distribuição de probabilidade Pr(S) de que uma sequência de símbolos *S* ocorra em uma determinada ordem pode ser descrita através de

$$\Pr(S) = \prod_{i=1}^{m} P(s_{i-n+1}, s_{i-n+2}, \cdots, s_{i-1}, s_i)$$

onde *m* é o número de símbolos da sequência S e $P(s_{i-n+1}, s_{i-n+2}, \cdots, s_{i-1}, s_i)$ é a probabilidade da sequência de *n* símbolos ocorrerem nesta ordem. Por exemplo, para o caso onde *n* = 2 (chamado de bigrama), a probabilidade da sequência 1, 2 ,2, 3, 4 (num espaço onde existem somente 4 tipos de símbolos) será calculada da seguinte forma:

P(1, 2, 2, 3, 4) = *P*(2 | 1) *P*(2 | 2) *P*(3 | 2) *P*(4 | 3).

No caso onde n = 3 (chamado de trigrama), são necessários dois predecessores para constituir a probabilidade da próxima palavra, que, no exemplo acima, ficaria da seguinte maneira:

P(1, 2, 2, 3, 4) = *P*(2 2 | 1) *P*(2 3 | 2) *P*(2 3 | 4).

A construção de um n-grama é um processo dividido em três estágios. Primeiramente, a sequência de símbolos é treinada e escaneada, e os n-gramas são contados e armazenados em uma base de n-gramas. No segundo estágio, algumas palavras podem ser mapeadas para fora de uma classe do vocabulário ou outra classe de mapeamento pode ser aplicada. No terceiro estágio, é utilizada a contagem dos n-gramas para calcular as distribuições de probabilidade, que podem ser armazenadas em um arquivo definido como modelo de linguagem (YOUNG, EVERMANN, *et al.*, 2006).

Em modelos de linguagens é importante que não existam probabilidades de ocorrências iguais a zero. Devido ao treinamento ser restrito podem haver ocorrências cuja probabilidade seja igual a zero, por não estarem presentes no conjunto de treinamento. Contudo, essas ocorrências podem aparecer na fase de testes. Para evitar essa situação existem métodos de compensação que atribuem probabilidades dentro do modelo, de tal forma que esses eventos não tenham probabilidade igual a zero mas que também não sejam relevantes dado o modelo como um todo (YOUNG, EVERMANN, *et al.*, 2006). Dentre os métodos de compensação, pode-se citar:

- Linear: também chamado de interpolação linear é definido pela fórmula:

$d(r) = 1 - (n(1)/C)$

onde C é o número total de eventos (NEY, ESSEN e KNESER, 1994);

- Good turing: atribui probabilidade a situações não previstas no conjunto de treinamento, enquanto diminiu a probabilidade de situações que ocorrem em quantidade inferior a *r*, que é um número previamente determinado. É definido pela fórmula:

$d(r) = (r+1)n(r+1) / rn(r)$

onde *n(r)* é o número de eventos que ocorrem *r* vezes (KATZ, 1987);

- Witten bell: este método é semelhante ao Good Turing, com a diferença que, ao invés de depender da contagem de eventos, depende da quantidade *t* de tipos que seguem um contexto específico. É definido pela fórmula:

$d(r,t) = n/(n + t)$

onde n é o tamanho do conjunto de treinamento (WITTEN e BELL, 1991).

4.2.2 Processo de Decisão

O reconhecimento de uma música pode ser formulado como um problema de detecção (também chamado de teste de hipótese), onde o objetivo é determinar se um modelo *X* representa um artista *Y* (REYNOLDS, QUATIERI e DUNN, 2000). A decisão, neste caso, deve escolher entre duas hipóstes mutuamente exclusivas:

- **H_0:** o modelo produziu a sequência de símbolos de teste (isto é, o artista da música de teste e o utilizado para treinamento são os mesmos); e

- **H_1:** o modelo não produziu a sequência de símbolos de teste (isto é, os artistas não são os mesmos).

Nestes tipos de problemas, o teste da razão da verossimilhança (em Inglês, *log-likelihood ratio test*) é considerado um classificador ótimo quando as funções de verossimilhança são conhecidas (REYNOLDS, QUATIERI e DUNN, 2000). O teste é dado por

$$\frac{P(X|H_0)}{P(X|H_1)} \begin{cases} \geq \theta & H_0 \, foi \, aceito, artista \, reconhecido \\ < \theta & H_0 \, não \, foi \, aceito, artista \, não \, reconhecido \end{cases}$$

onde $P(X|H_0)$ e $P(X|H_1)$ são as funções densidade probabilidade das hipóteses H_0 e H_1 estimadas sobre o conjunto de vetores de características X. O limiar de decisão é dado por θ. Assim, o objetivo do sistema de reconhecimento é determinar os métodos para estimar $P(X|H_0)$ e $P(X|H_1)$. A hipótese H_0 é representada pelo modelo do artista $\lambda_{artista}$que se deseja verificar. A hipótese alternativa H_1 é representada pelo modelo $\lambda_{\overline{artista}}$, o qual representa o universo de todos os demais artistas, referenciado neste trabalho por Modelo Universal de Músicas. Assim, o teste da razão pode ser estimado utilizando o logaritmo das verossimilhanças

$$\Lambda(X) = \log P(X|\lambda_{artista}) - \log P\left(X\middle|\lambda_{\overline{artista}}\right),$$

onde $P(X|\lambda_{artista})$ é estimado para cada música a ser reconhecida através do modelo de *n*-gramas estimados a partir da sequência de símbolos extraídos da mesma e $P\left(X\middle|\lambda_{\overline{artista}}\right)$ é estimado da mesma maneira mas de um conjunto representativo das músicas (e que não pertençam ao conjunto de músicas a serem reconhecidas). A decisão do sistema é realizada através de

$$\Lambda(X) \begin{cases} \geq \theta & H_0 \, foi \, aceito, artista \, reconhecido \\ < \theta & H_0 \, não \, foi \, aceito, artista \, não \, reconhecido \end{cases}$$

Há dois tipos de erros que podem ocorrer em um problema de detecção. O primeiro ocorre quando a hipótese H_0 é rejeitada, quando na verdade é verdadeira. Esse erro é conhecido como *miss detection*, ou falso negativo, o que significa que o artista não foi reconhecido. O segundo erro ocorre quando a hipótese é classificada incorretamente como verdadeira. Esse erro é conhecido como *false alarm*, ou falso positivo, o que significa que o artista foi reconhecido, mas não pelo correto (ADAMI, 2004).

Neste trabalho, a comparação é feita através da taxa *equal error rate* (EER), que é o ponto onde o número (ou probabilidade) dos erros de *false alarm* é igual ao número (ou probabilidade) dos erros de *miss detection* (ADAMI, 2004). Quando há mais de uma situação de teste, é necessário calcular também a significância estatística entre duas taxas EER, que pode ser feita através de testes de hipóteses paramétricos ou não-paramétricos.

Para a avaliação de desempenho e escolha do limiar são utilizados gráficos representativos da distribuição dos sinais através das curvas DET. A Curva DET (*Detection-Error Tradeoff Curve)* (MARTIN, DODDINGTON, *et al.*, 1997) traça um gráfico com as taxas de erro em ambos os eixos, dando tratamento uniforme aos tipos de erro (*false alarm* e *miss detection*). Quanto mais à esquerda e na porção inferior estiver a curva, melhor o desempenho do método empregado.

4.3 DESENVOLVIMENTO DO SRAM

O desenvolvimento do SRAM ocorreu em três fases. Na primeira fase foram desenvolvidos scripts que foram executados no software MATLAB[5] para a leitura dos arquivos HTK, geração do modelo de misturas gaussianas e geração das sequências de tokens. Na segunda fase foram desenvolvidos scripts em Python[6] para as chamadas dos programas de geração dos modelos de n-grama e de cálculo dos scores de cada música de teste. Já na terceira fase, o software MATLAB foi novamente utilizado para a avaliação dos scores gerados na fase anterior. A Figura 4-2 ilustra as fases de desenvolvimento do SRAM.

Figura 4-2: Fases do desenvolvimento do SRAM.

[5] http://www.mathworks.com/products/matlab/
[6] http://www.python.org/

4.3.1 BASE DE DADOS

A base que foi utilizada para o desenvolvimento do sistema provem de dados cedidos pelo *Laboratory for the Recognition and Organization of Speech and Audio - LabROSA*[7], da Universidade de Columbia (EUA), da base de músicas *uspop2002* (ELLIS, BERENZWEIG e WHITMAN, 2003) do projeto de *Music Similarity*[8] (BERENZWEIG, LOGAN, *et al.*, 2004). A base é formada por arquivos (no formato HTK) com as características MFCC extraídas, cujo processo foi feito a partir da biblioteca *feacalc*, do pacote SPRACHcore[9]. Cada arquivo HTK contem 20 vetores de características extraídas de cada uma das músicas da base.

A base possui 8752 músicas de 400 artistas, divididas em 706 álbuns. Segue outras informações:

- Média de 1,76 álbuns por artista;
- Média de 12,39 músicas por álbum;
- 247 artistas com 1 álbum;
- 86 artistas com 2 álbuns;
- 67 artistas com 3 álbuns ou mais;
- Artista com menos músicas: Blackstreet, com 4 músicas;
- Artista com mais músicas: Queen, com 126 músicas.

Dos 400 artistas, dois não puderam ser aproveitados, pois os seus arquivos de características gerados estavam corrompidos. Com isso, a base utilizada foi de 398 artistas, com 8717 músicas, divididas em 700 álbuns.

4.3.1.1 Ambiente de Desenvolvimento

O sistema desenvolvido utilizou a classificação supervisionada, onde foram gerados modelos de artistas na fase de treinamento. Para cada um dos 180 artistas com menos de 14 músicas na base, foi selecionada a primeira música da sua respectiva listagem para a geração do Modelo Universal de Músicas. Dos 218 artistas restantes, com 14 músicas na base ou mais,

[7] http://labrosa.ee.columbia.edu/
[8] http://labrosa.ee.columbia.edu/projects/musicsim/
[9] http://www1.icsi.berkeley.edu/~dpwe/projects/sprach/sprachcore.html

a seleção das músicas utilizadas para treinamento e para teste foi dividida da seguinte maneira: do conjunto total de músicas de cada artista, dez foram selecionadas aleatoriamente para compor o modelo de treinamento do artista; do restante das músicas, quatro foram selecionadas aleatoriamente para teste.

4.3.2 Geração do Modelo de Misturas Gaussianas e Sequências de Tokens

A primeira fase do desenvolvimento do SRAM trata desde a leitura dos arquivos HTK até a geração dos arquivos de tokens. Primeiramente, através das listas dos artistas, dos álbuns e das músicas existentes, disponíveis na página referente à base[10], foram geradas duas listas:

- nummusicas: quantitativo de músicas por artista, com as colunas artista e quantitativo, ordenada pelo quantitativo;
- numalbuns: quantitativo de álbuns por artista, com as colunas artista e quantitativo, ordenada pelo quantitativo. Essa lista gerou um arquivo, *lista dos artistas do treinamento e teste*, que foi utilizada na fase de teste.

Essas listas auxiliaram na seleção de artistas para a geração do MUM, além de definir quantas músicas foram utilizadas dos artistas restantes para treinamento e quantas foram utilizadas para teste.

Com as listas geradas, foram selecionadas 180 músicas, sendo uma música de cada um dos 180 primeiros artistas da lista *nummusicas*, para a geração do Modelo Universal de Músicas (MUM). O MUM é primeiramente formado por um modelo de misturas gaussianas, gerado através dos scripts *mixturefit*.m e *pxi.m*, adaptações desenvolvidas por André Gustavo Adami aos scripts de Todd Leen. A matriz de covariância utilizada foi a diagonal pois, segundo (REYNOLDS, QUATIERI e DUNN, 2000), essa matriz possui um desempenho superior ao da sua forma completa. Na etapa seguinte da geração da MUM, para cada uma das 180 músicas anteriormente selecionadas foi realizada a leitura de cada característica para encontrar a componente gaussiana do modelo de misturas gaussianas que tivesse a maior probabilidade, através do script *pxi.m*. O resultado ao final da leitura foi uma sequência de

[10] http://labrosa.ee.columbia.edu/projects/musicsim/uspop2002.html

símbolos, onde cada símbolo representa uma característica da música em relação ao MUM. Ao final, a sequência de símbolos gerada foi salva em um arquivo texto. Esse mesmo processo de geração de sequências de símbolos foi realizado com as músicas selecionadas para treinamento e para teste, salvando os arquivos criados em diretórios separados.

Os padrões de nomes de arquivo utilizados foram diferentes para cada tipo de arquivo. Os arquivos pertencentes ao MUM e os arquivos de teste tinham como padrão *nome_do_artista-numero_da_musica-nome_da_musica.txt*. Já os arquivos de treinamento dos modelos de artista foram nominados por dois números sequenciais: um para o artista (definido pelo número da linha na lista *numalbuns*), que ia de 1 a 218, e um para a música, na ordem em que iam sendo gerados. Assim, esses arquivos tinham como padrão *número_do_artista-numero_da_musica.txt*.

4.3.3 Modelos de N-grama e Geração dos Scores

Na segunda fase do desenvolvimento do SRAM, os arquivos de tokens gerados na primeira fase foram utilizados para a geração dos modelos de n-gramas e dos scores, através dos programas do *toolkit* CMU-Cambridge Statistical Language Modeling[11].

O script de preparação dos arquivos e de chamada dos métodos foi desenvolvido na linguagem Python. Nesse script, os arquivos de tokens foram processados na seguinte maneira: os arquivos do MUM foram concatenados em um, denominado *ubm.2*, onde a sequência de cada arquivo ocupava uma linha; já os arquivos de treinamento foram concatenados por artistas, onde a sequência de cada arquivo também ocupava uma linha. Esses arquivos de treinamento foram nomeados pelo respectivo número sequencial gerado na fase anterior, de *001.2* a *218.2*.

Com os arquivos de tokens concatenados, foram gerados o vocabulário e os modelos de n-grama do MUM e de cada artista, processo este da fase de treinamento do sistema. Foram utilizados os programas *text2wfreq*, que calcula a frequência de cada um dos símbolos, e *wfreq2vocab* que, a partir do arquivo de frequências, relaciona os símbolos existentes e gera o arquivo de vocabulário. O arquivo de vocabulário é formado pelos números da quantidade

[11] http://www.speech.cs.cmu.edu/SLM/toolkit.html

de componentes do Modelo Universal de Músicas gerado na primeira fase: no modelo com 64 componentes, o vocabulário possui os números de 1 a 64; já no Modelo Universal de Músicas de 128 componentes, o vocabulário possui os números de 1 a 128.

Já os modelos de n-grama tanto do modelo universal quanto de cada artista foram gerados a partir de dois outros programas: *text2idngram*, que possui como entrada o arquivo a ser lido e, dentre outros parâmetros, o arquivo do vocabulário e a quantidade de n-gramas a ser calculado; e o *idngram2lm*, que possui, dentre outros parâmetros, o método de compensação a ser utilizado. No trabalho foram utilizados os métodos de compensação linear, de Good Turing e de Witten Bell. O final da execução dos dois comandos resulta em um arquivo com extensão ARPA, que é um modelo de n-grama e possui informações como a quantidade de n-gramas para cada tamanho de n-grama contido no modelo, além das probabilidades dos n-gramas propriamente ditos. Um trecho de um arquivo ARPA é ilustrado na Figura 4-3. Neste trecho, é possível identificar a quantidade de n-gramas para cada tamanho (no caso unigrama, bigrama e trigrama), através da palavra-chave *data*\. Após, é iniciada uma seção do arquivo para as sequências de unigrama, identificado através da palavra-chave *1-grams:*. Para cada uma das sequências de unigrama, é definida a sua probabilidade, onde <UNK> representa as sequências de unigrama desconhecidas do vocabulário.

```
\data\
ngram 1=129
ngram 2=9757
ngram 3=55009

\1-grams:
-99.0000 <UNK>  0.0000
-2.2633 1   -0.9626
-2.6140 10  -1.1775
-2.1672 100 -0.7688
-2.3948 101 -1.2822
-2.3321 102 -1.2127
-1.8713 103 -0.9772
-1.8966 104 -0.6001
-1.9618 105 -0.7861
-2.3344 106 -1.2213
-2.2885 107 -1.0585
-2.2623 108 -0.9976
-1.8429 109 -0.9652
-2.1832 11  -0.8918
```

Figura 4-3: Trecho de modelo de n-grama.

Após a geração dos modelos, é iniciada a etapa de teste. Através de um script gerado na linguagem Python, para cada arquivo de teste, é gerado outro com scores. Esses scores são calculados através de dois programas: *perplexity* e *evallm*, que possuem como entrada o arquivo de teste e o modelo a ser comparado e cuja saída é denominada score. Toda a execução da geração dos modelos e dos testes segue o exemplo de fluxo de execução ilustrado na Figura 4-4, onde os itens em elipse são as entradas e saídas e os itens em retângulos são os programas do *toolkit*.

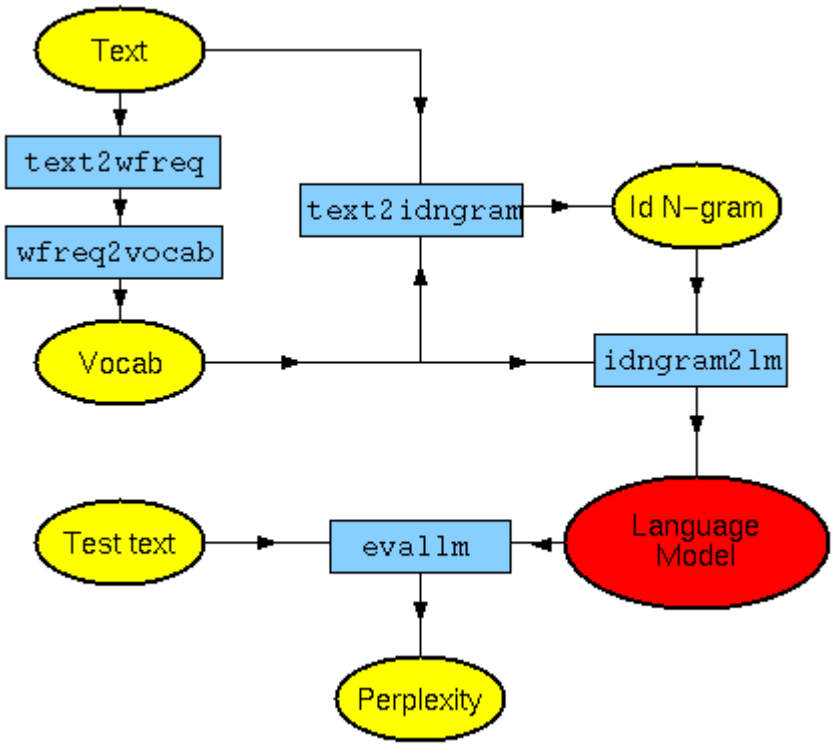

Figura 4-4: Exemplo de fluxo de execução do toolkit CMU-Cambridge Statistical Language Modeling[12].

Com o score gerado, é calculado o teste da razão da verossimilhança, utilizando o logaritmo das verossimilhanças, através da seguinte linha do script:

score = -math.log(x/xubm)/math.log(2)

onde *x* é o score gerado para o modelo que está sendo comparado e *xubm* é o score gerado para o modelo universal de músicas. O resultado é inserido em um novo arquivo texto, com extensão SCORES.

Para fins de desenvolvimento e avaliação do sistema na sua etapa de treinamento, cada arquivo de teste foi comparado com o modelo do artista verdadeiro e com outros 20 impostores, além do modelo universal. Na primeira linha do arquivo SCORE, é gravado o score do artista verdadeiro, e nas linhas seguintes é gravado o score de cada um dos 20 impostores. Com isso, cada arquivo SCORES gerado tem o total de 21 linhas.

[12] http://www.speech.cs.cmu.edu/SLM/toolkit_documentation.html#typical_use

A validação inicial para o artista verdadeiro foi feita através da comparação do nome do artista (nome do arquivo até o primeiro caractere "-") com a *lista dos artistas do treinamento e teste*, gerada no início do desenvolvimento do sistema. Com o seu número correspondente na lista era selecionado o modelo correspondente e feita a comparação. Já os 20 impostores eram escolhidos aleatoriamente dentre os 217 modelos restantes.

4.3.4 Avaliação

Na terceira fase do desenvolvimento do SRAM, os arquivos de score gerados na fase anterior são avaliados. O script, desenvolvido no software MATLAB, seleciona os scores verdadeiros de todos os arquivos de teste e os reúne em um vetor, fazendo o mesmo com os scores falsos em outro vetor. Para a decisão do limiar, que decide se a música é de um dado artista ou não, foram calculadas as taxas de *miss detection* para cada um dos scores verdadeiros, e *false alarm* para cada um dos scores falsos. O cálculo foi feito a partir das linhas apresentadas na Figura 4-5. Neste trecho do script, todos os scores (tanto os verdadeiros quanto os falsos) são unidos em um único vetor, chamado de *limiares*. Após, para cada valor de *limiares*, é feito o cálculo de quantos scores verdadeiros entram como *miss detection*, sendo esse valor inserido no vetor *md*, e de quantos scores falsos entram como *false alarm*, sendo esse valor inserido no vetor *fa*. Este laço é executado, pois qualquer um dos valores contidos no vetor *limiares* pode ser o limiar escolhido para a aplicação.

```
51  %Geração das Taxas de False Alarm e Miss Detection
52  limiares = unique([scorestrue; scoresfalse]);
53  md = zeros(size(limiares));
54  fa = md;
55  ltrue = length(scorestrue);
56  lfalse = length(scoresfalse);
57  for i = 1:length(limiares)
58      md(i) = length(find(scorestrue<limiares(i)))/ltrue;
59      fa(i) = length(find(scoresfalse>=limiares(i)))/lfalse;
60  end
```

Figura 4-5: Geração das Taxas de *False Alarm* e *Miss Detection.*

Para a decisão do melhor limiar foi utilizada a taxa EER, calculada através do programa desenvolvido pelo NIST, e para calcular a significância estatística entre duas taxas

EER foi utilizado o teste de significância da diferença entre proporções (KANJI, 1999). Ambos os scripts foram executados através do software MATLAB.

4.4 RESULTADOS

O SRAM possui diversos parâmetros que devem ser definidos a fim de produzir o resultado. Dentre eles, podemos citar:

1. Número de Gaussianas do MUM
2. Ordem do Modelo de N-Gramas
3. Método de Compensação do N-grama

Para os experimentos foram gerados dois modelos de misturas Gaussianas: um com 64 componentes, e outro com 128 componentes. Modelos com maior número de componentes não foram utilizados devido às restrições de gerenciamento de memória por parte do software utilizado.

A fim de modelar a dinâmica da trajetória dos tokens, e tentar diminuir o custo de execução da geração dos modelos de artistas, a repetição de tokens foi eliminada substituindo por um único token. Por exemplo, a sequência “1 2 2 3 4 3 3 3 5 5” seria modificada para “1 2 3 4 3 5”. A Figura 4-6 mostra o desempenho do reconhecimento para bigrama e trigrama, com método de compensação linear. Pode-se notar que o melhor desempenho é obtido pelo sistema com modelos de bigramas estimados a partir de um MUM com 64 componentes (EER = 17.32%). Apesar do bigrama estimado do MUM com 128 componentes ter um EER maior (EER = 17.43%), não é possível dizer que esta diferença é estatisticamente significante (α = 0.05). O desempenho dos modelos trigrama pode ser atribuído ao problema de estatística suficiente para cada trigrama, já que com 64 possíveis símbolos, o modelo poderá ter 262144 trigramas.

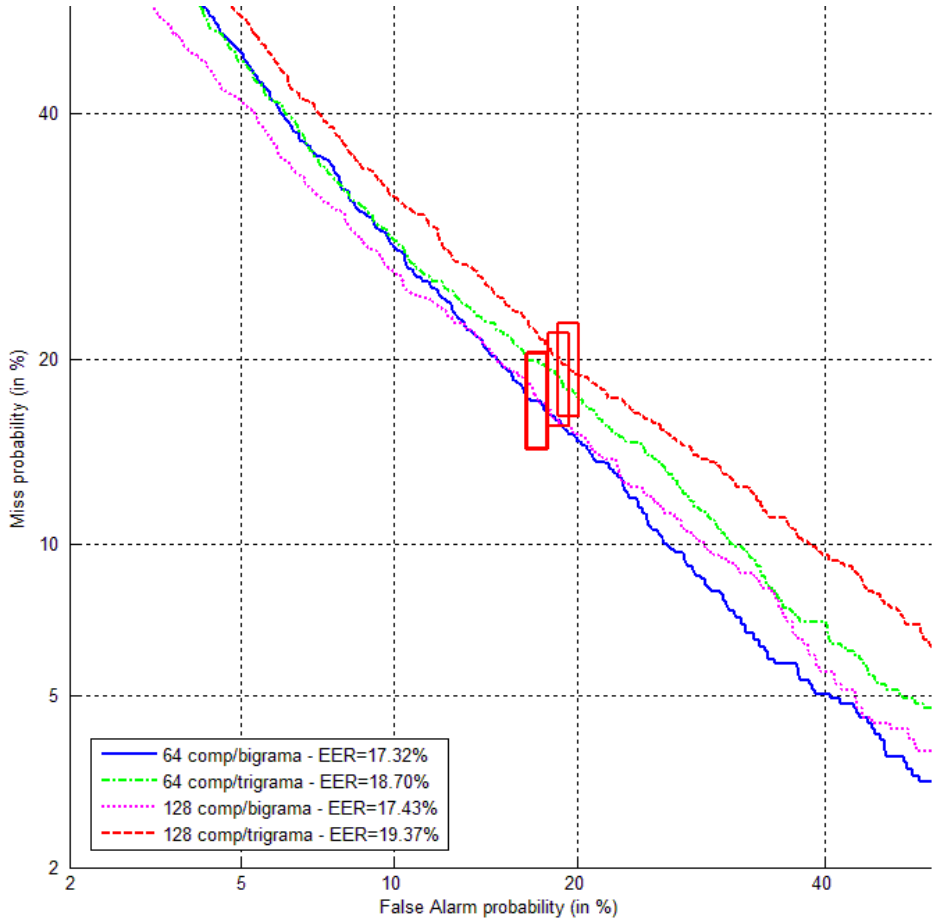

Figura 4-6: Desempenho do SRAM para MUM com 64 e 128 componentes com bigramas e trigramas.

Devido ao grande número de n-gramas produzido e pela possibilidade de existir n-gramas que não ocorrem nos arquivos de treinamento foram avaliados outros métodos de compensação de n-gramas. A Figura 4-7 mostra o desempenho do reconhecimento para MUM com 64 e com 128 componentes, com bigramas e métodos de compensação Good Turing e Witten Bell. Neste caso, o melhor desempenho obtido foi com MUM de 64 componentes e método de compensação Good Turing. O bigrama estimado com MUM de 128 componentes e método de compensação Good Turing possui EER maior, porém essa diferença não é estatisticamente significante.

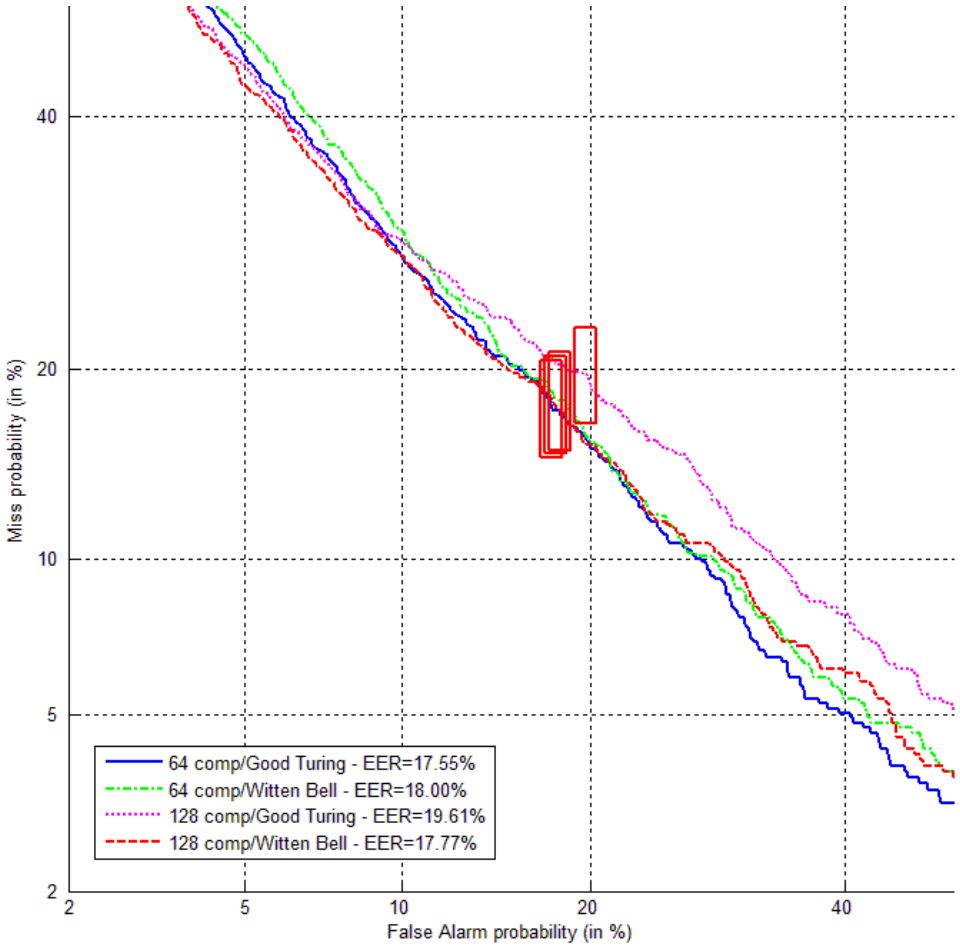

Figura 4-7: Desempenho do SRAM para MUM com 64 e 128 componentes com bigramas e métodos de compensação Good Turing e Witten Bell.

Foram também avaliados os modelos gerados a partir de tokens sem repetições. A Figura 4-8 mostra os modelos de tokens sem repetições que obtiveram melhores desempenhos, modelos esses gerados para tokens com repetições. Nesta situação, o melhor desempenho é obtido pelo sistema de bigramas com MUM de 64 componentes e método de compensação linear (EER de 16.63%). Contudo, a diferença não é estatisticamente significante para o sistema com MUM de 128 componentes e método de compensação Witten Bell, que possui EER maior.

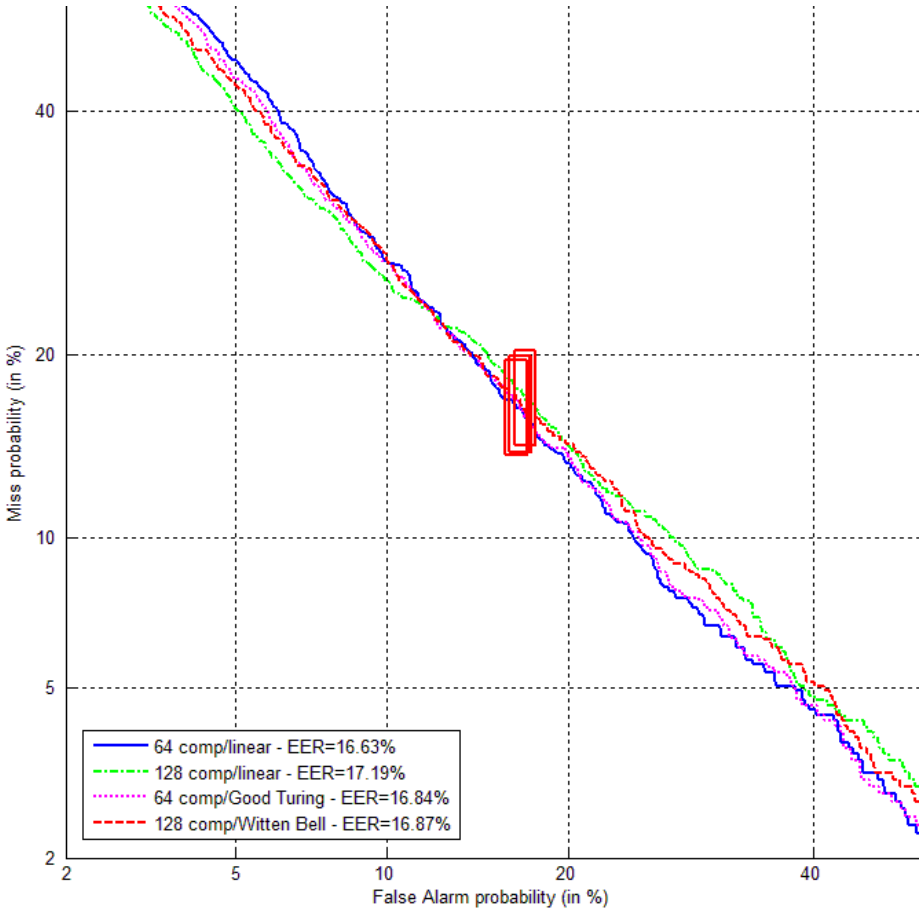

Figura 4-8: Desempenho do SRAM para tokens com repetição e bigramas.

Através dos resultados obtidos é possível concluir que o melhor sistema, com relação ao ambiente de teste desenvolvido, é o sistema com MUM de 64 componentes, bigrama, método de compensação linear e com repetição de tokens. Com EER de 16.63%, a diferença desse sistema é estatisticamente significante com relação ao de MUM de 64 componentes, bigrama, método de compensação linear e sem repetição de tokens.

5 CONCLUSÃO

A necessidade de automatizar a organização de bases de músicas, bases estas que podem ser cada vez maiores, devido à facilidade de geração e transmissão de arquivos de música, foi a principal motivação deste trabalho, juntamente com a facilidade de busca que essa organização pode oferecer. Como o trabalho manual de busca em uma grande quantidade de músicas é impossível, é importante que se tenham disponíveis ferramentas através das quais seja possível o reconhecimento automático de certas informações de uma música, como o seu artista. O Sistema de Reconhecimento Automático de Músicas foi proposto para essa finalidade.

O reconhecimento de padrões é utilizado em sistemas com finalidade em diversas áreas, inclusive na área de áudio, como o reconhecimento de fala e o próprio reconhecimento de música. O funcionamento destes sistemas é totalmente baseado no princípio do reconhecimento de padrões, que possui como etapas principais a extração de características e a classificação.

Sendo assim, o SRAM foi desenvolvido, com base no reconhecimento de padrões, totalmente adaptado para a música. Desde a aquisição das músicas, até as etapas de extração de seleção de características e classificação, foram utilizados métodos que tem se mostrado mais adequados para o tratamento de sinais de áudio. Foram utilizados os conceitos de modelo de mistura de Gaussianas, juntamente com a modelagem N-grama, onde um Modelo Universal de Músicas e modelos de artistas foram gerados para a classificação, etapa esta que foi desenvolvida através da modelagem N-grama.

A análise dos resultados foi feita através da detecção, utilizando o teste da razão da verossimilhança, com a geração da taxa EER para cada sistema. Após os testes realizados, o sistema que produziu melhores resultados foi o que possui os seguintes parâmetros: MUM com 64 componentes, bigrama, método de compensação linear, e com repetição de tokens, com taxa EER de 16.63%.

5.1 TRABALHOS FUTUROS

Trabalhos futuros podem ser desenvolvidos para testar outras situações, tanto nas etapas de treinamento quanto de teste. Uma das situações que pode ser analisada é a utilização de uma quantidade de artistas na validação dos dados de treinamento. Com isso, a base seria dividida em uma quantidade de artistas para a MUM, e outra para o treinamento, a validação e os testes.

Outro teste que pode ser feito com base no sistema apresentado é a utilização de menos componentes na geração da MUM, como por exemplo, com 32 ou ainda 16 componentes. Com os resultados obtidos nos testes, foi possível concluir que, no geral, a geração da MUM com menos componentes também produz taxas menores de erro, além de possuir menor custo de processamento.

Ainda como ideia para trabalhos futuros, pode ser trabalhada a normalização dos dados gerados, como o arquivo de scores. Isso pode auxiliar na execução do sistema, já que a normalização reduz a redundância de dados e as chances dos dados se tornarem inconsistentes.

REFERÊNCIAS

ADAMI, A. G. **Modeling Prosodic Differences for Speaker and Language Recognition.** 2004. 304 f. Tese de Doutorado – Electrical Engineering, Oregon Health & Science University, 2004.

BERENZWEIG, A. et al. A large-scale evaluation of acoustic and subjective music-similarity measures. **Computer Music Journal**, v. 2, n. 28, p. 63-76, Junho 2004.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. 1ª. ed. New York: Springer, 2006.

CAMPOS, T. E. D.; CESAR JÚNIOR, R. M. Técnicas de Seleção de Atributos e de Classificação para Reconhecimento de Faces, São Paulo, 18 setembro 2000.

CANO, P. et al. Robust Sound Modeling for Song Detection in Broadcast Audio. **Proceedings of the 112th Audio Engineering Society Convention**, Munich, Germany, 2002.

CANO, P. et al. A Review of Audio Fingerprinting. **J. VLSI Signal Process. Syst.**, p. 271-284, Novembro 2005.

CASEY, M. et al. Content-Based Music Information Retrieval: Current Directions and Future Challenges. **Proceedings of the IEEE**, v. 96, n. 4, p. 668-696, Abrl 2008.

DASH, M.; LIU, H. Feature Selection for Classification. **Intelligent Data Analysis**, p. 131-156, 1997.

ELLIS, D.; BERENZWEIG, A.; WHITMAN, B. The "uspop2002" Pop Music data set, 2003. Disponivel em: <http://labrosa.ee.columbia.edu/projects/musicsim/uspop2002.html>. Acesso em: 27 Junho 2012.

FRIEDMAN, M.; KANDEL, A. **Introduction to Pattern Recognition:** Statistical, Structural, Neural, and Fuzzy Logic Approaches. [S.l.]: World Scientific, 1999.

FUKUNAGA, K. **Introduction to Statistical Pattern Recognition**. 2ª. ed. West Lafayette: Academic Press, 1990.

FUTRELLE, J.; DOWNIE, J. S. Interdisciplinary Communities and Research Issues in Music Information Retrieval. **Proceedings on the Third International Conference on Music**, Paris, p. 215-221, 2002.

GOLD, B.; MORGAN, N. **Speech and Audio Signal Processing:** Processing and Perception of Speech and Music. 1ª. ed. New York: John Wiley, 2000.

GOMES, L. D. C. T. et al. Audio Watermarking and Fingerprinting: For Which Applications? **Journal of New Music Research**, v. 23, p. 65-81, 2003.

JAIN, A. K.; DUIN, R. P. W.; MAO, J. Statistical pattern recognition: A review. **IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE**, v. 22, n. 1, p. 4-37, Janeiro 2000.

JAIN, K. A.; DUIN, P. W. R. Introduction to Pattern Recognition. In: GREGORY, R. L. **The Oxford Companion to the Mind**. 2ª. ed. Oxford: Oxford University Press, 2004.

JEHSEN, J. H. **Feature extraction for music information retrieval.** 2010. Tese de Doutorado, Aalborg University, 2010.

JENSEN, J. H. et al. Quantitative Analysis of a Common Audio Similarity Measure. **IEEE Transactions on Audio, Speech, and Language Processing**, v. 17, n. 4, p. 693-703, Maio 2009.

KANJI, G. K. **100 Statistical Tests**. 2ª. ed. [S.l.]: SAGE Publications, 1999. 224 p.

KATZ, S. M. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. **IEEE Transactions on Acoustics, Speech and Signal Processing**, v. ASSP-35, n. 3, p. 400-401, Março 1987.

KILL, D. H.; SHIN, F. B. **Pattern Recognition and Prediction With Applications to Signal Characterization**. 1ª. ed. Woodbury: American Institute of Physics, 1996.

KIM, Y. E.; WHITMAN, B. Singer Identification in Popular Music Recordings Using Voice Coding Features. **Proceedings of the 3rd International Conference on Music Information Retrieval**, Paris, Outubro 2002.

LOGAN, B. Nearest Neighbor Artist Identification. **Proceeding of Music Information Retrieval Evaluation eXchange**, Londres, p. 192-194, 2005.

LOGAN, B. Mel frequency cepstral coefficients for music modeling. **Proceedings of International Symposium on Music Information Retrieval (ISMIR)**, Plymouth, 2008.

LOGAN, B.; ELLIS, D. P. W.; BERENZWEIG, A. Toward Evaluation Techniques for Music Similarity. **Proceedings of the 4th International Symposium on Music Information Retrieval**, Bartimore, p. 81-85, 2003.

MANNING, C. D.; SCHÜTZE, H. **Foundations of Statistical Natural Language Processing**. 1ª. ed. Cambridge: MIT Press, 1999.

MARTIN, A. et al. The DET Curve in Assesment of Detection Task Perfonnance. **Proceedings of EUROSPEECH**, p. 1895-1898, 1997.

MÜLLER, M. et al. Signal Processing for Music Analysis. **IEEE Journal Of Selected Topics In Signal Processing**, v. 5, n. 6, p. 1088-1110, Outubro 2011.

NEY, H.; ESSEN, U.; KNESER,. On Structuring Probabilistic Dependences in Stochastic Language Modeling. **Computer, Speech, and Language**, v. 8, n. 1, p. 1-28, 1994.

OLSON, D. L.; DELEN, D. **Advanced Data Mining Techniques**. 1ª. ed. [S.l.]: Springer, v. XII, 2008. 180 p.

REYNOLDS, D. A. Gaussian Mixture Models. **Encyclopedia of Biometric Recognition**, Fevereiro 2008.

REYNOLDS, D. A.; QUATIERI, T. F.; DUNN, R. B. Speaker Verification Using Adapted Gaussian Mixture Models. **Digital Signal Processing**, v. v. 10, p. 19-41, 2000.

THEODORIDIS, S.; KOUTROUMBAS, K. **Pattern Recognition**. 2ª. ed. Elsevier: Academic Press, 2003.

TYPKE, R.; WIERING, F.; VELTKAMP, R. C. A Survey of Music Information Retrieval Systems. **ISMIR**, p. 153-160, 2005.

WANG, A. L.-C. An Industrial-Strength Audio Search Algorithm. **Proceedings of the Fourth International Conference on Music Information Retrieval**, Baltimore, 2003.

WEBB, A. **Statistical Pattern Recognition**. 1ª. ed. New York: Oxford University Press Inc., 1999.

WHITMAN, B.; FLAKE, G.; LAWRENCE, S. Artist Detection in Music with Minnowmatch. **Proceedings of the 2001 IEEE Workshop on Neural Networks for Signal Processing**, Falmouth, p. 559-568, Setembro 2001.

WITTEN, I. H.; BELL, T. C. The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression. **IEEE Transactions on Information Theory**, v. 37, n. 4, Julho 1991.

XIANG, B. Speaker verification using Gaussian component strings in dynamic trajectory space. **Proceedings of International Conference on Spoken Language Processing**, Denver, p. 1365-1368, 2002.

XIANG, B. Text-independent speaker verification with dynamic trajectory model. **IEEE Signal Processing Letters**, v. 10, p. 141-143, Maio 2003.

YOUNG, S. et al. **The HTK Book (for HTK Version 3.4)**. 8ª. ed. Cambridge: Cambridge University Engineering Department, 2006.