

**Métodos de
segmentação musical
baseados em
descritores sonoros**

André Salim Pires

DISSERTAÇÃO APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO TÍTULO
DE
MESTRE EM CIÊNCIAS

Programa: Ciência da Computação
Orientador: Prof. Dr. Marcelo Queiroz

São Paulo, junho de 2011

**Métodos de
segmentação musical
baseados em
descritores sonoros**

Esta dissertação contém as correções e alterações sugeridas pela Comissão Julgadora durante a defesa realizada por André Salim Pires em 20/06/2011. O original encontra-se disponível no Instituto de Matemática e Estatística da Universidade de São Paulo.

Comissão Julgadora:

- Prof. Dr. Marcelo Queiroz (orientador) - IME/USP
- Profa. Dra. Airlane Alencar - IME/USP
- Prof. Dr. Miguel Arjona Ramirez - EP/USP

Dedicatória

À Suzana e ao Calvin,
que deram sentido a isto.

Agradecimentos

Agradeço aos professores que lecionaram as disciplinas que participei durante o mestrado: Paulo Feofiloff, Roberto Hirata, Marcelo Finger, Francisco Pelaez, Rodolfo Coelho de Souza, Eduardo Seincman e Fernando Iazzeta. Seus ensinamentos foram muito importantes para meu crescimento. Agradeço também aos colegas que caminharam ao meu lado neste período.

Agradeço imensamente aos professores da banca de qualificação: Nina Hirata e Miguel Ramírez. Seus argumentos me ajudaram a construir melhor os objetivos da pesquisa e a rever deficiências estruturais e conceituais. À Nina Hirata em especial, por me atender em reuniões não agendadas para tirar dúvidas teóricas ou simplesmente para tratar de assuntos ordinários.

Agradeço especialmente à Lane Alencar pelo acolhimento como aluno ouvinte, pela paciência e intenso interesse por um problema de outra área e pelas contribuições com ideias e referências relacionadas ao problema desta pesquisa. Agradeço também ao professor Paulo de Tarso Salles pelo incentivo no ingresso ao mestrado.

Todos os agradecimentos não seriam suficientes ao Marcelo Queiroz, meu orientador durante o mestrado. Seu auxílio e direção durante toda a pesquisa foram excepcionalmente precisos. Dele vieram o posicionamento da pesquisa, a estrutura metodológica, as principais indicações bibliográficas e as principais sugestões de melhoria para os algoritmos. Agradeço também o tempo dedicado durante as reuniões e a paciência com que lidou com minhas limitações.

Aos meus pais, Themis Regina e Sebastião Pires, às minhas irmãs e sobrinhos, que me deram todo o apoio necessário, sou muito grato. Aos meus amigos Daniel Tucci, Mauro Piva, Paulo Penov, George Szégo, Vivian Broge e muitos outros grandes amigos, que me ajudaram e me apoiaram.

E por último, agradeço à Suzana, minha esposa, pelo apoio moral, pelas comemorações a cada etapa conquistada e pela imensa compreensão nos momentos difíceis. Sem ela nada disto seria possível.

Resumo

Esta dissertação apresenta um estudo comparativo de diferentes métodos computacionais de segmentação estrutural musical, onde o principal objetivo é delimitar fronteiras de seções musicais em um sinal de áudio, e rotulá-las, i.e. agrupar as seções encontradas que correspondem a uma mesma parte musical. São apresentadas novas propostas para segmentação estrutural não-supervisionada, incluindo métodos para processamento em tempo real, alcançando resultados com taxas de erro inferiores a 12%. O método utilizado compreende um estudo dos descritores sonoros e meios de modelá-los temporalmente, uma exposição das técnicas computacionais de segmentação estrutural e novos métodos de avaliação dos resultados que penalizam tanto a incorreta detecção das fronteiras quanto o número incorreto de rótulos encontrados. O desempenho de cada técnica computacional é calculado utilizando diferentes conjuntos de descritores sonoros e os resultados são apresentados e analisados tanto quantitativa quanto qualitativamente.

Palavras-chave: recuperação de informação musical, segmentação estrutural musical, segmentação musical em tempo real, geração e seleção de descritores sonoros, processamento de sinal de áudio em tempo real.

Abstract

A comparative study of different music structural segmentation methods is presented, where the goal is to delimit the borders of musical sections and label them, i.e. group the sections that correspond to the same musical part. Novel proposals for unsupervised segmentation are presented, including methods for real-time segmentation, achieving expressive results, with error ratio less than 12%. Our method consists of a study of sound descriptors, an exposition of the computational techniques for structural segmentation and the description of the evaluation methods utilized, which penalize both incorrect boundary detection and incorrect number of labels. The performance of each technique is calculated using different sound descriptor sets and the results are presented and analysed both from quantitative and qualitative points-of-view.

Keywords: music information retrieval, music structural segmentation, real-time music segmentation, generation and selection of sound descriptors, real-time sound processing.

Sumário

Lista de Abreviaturas	xiii
Lista de Figuras	xv
Lista de Tabelas	xxi
1 Introdução	1
2 Metodologia	7
2.1 Geração de dados musicais	7
2.2 Obtenção de descritores	9
2.3 Segmentação	10
2.4 Resultados Finais	12
3 Descritores de Áudio	15
3.1 Análise do Som	17
3.2 Do Sinal ao Descritor	20
3.3 Seleção de Descritores	20
3.3.1 Pré-processamento	22
3.3.2 Seleção Automática de Descritores	23
3.3.3 Análise independente com curvas ROC	26
3.3.4 Seleção com Critérios de Separabilidade entre Multi-Classes	29
3.3.5 Seleção com LDA	32
3.3.6 Seleção com PCA	33
3.3.7 Seleção com IRMFSP	33
3.3.8 Critérios para seleção do subconjunto final	34
3.4 Geração de Descritores Dinâmicos	35
3.4.1 Descritores Dinâmicos Cumulativos	35
3.4.2 Descritores Dinâmicos por Bandas Mel (DDBM)	39
4 Segmentação Musical	41
4.1 Separabilidade entre Classes	46
4.1.1 Divergência	46
4.1.2 Distância de Bhattacharyya	47
4.2 Aglomerados	48
4.2.1 K-Médias	50

4.2.2	Aglomerados Hierárquicos	51
4.3	Segmentação Musical Supervisionada	53
4.3.1	Classificador Ingênuo de Bayes	53
4.3.2	Modelos de Misturas de Gaussianas	54
4.3.3	<i>K</i> -Vizinhos Mais Próximos	55
4.3.4	Árvores de Decisão - J48	55
4.3.5	Modelos Ocultos de Markov	56
4.4	Segmentação Musical Não-supervisionada	58
4.4.1	Segmentação via Matriz de Dissimilaridade*	58
4.4.2	Segmentação via Matriz de Similaridade de Cooper e Foote (2002)	61
4.4.3	Segmentação por Delta de Mahalanobis (Tzanetakis e Cook, 1999)	62
4.4.4	Segmentação por Dissimilaridade e Processamento de Imagem*	63
4.4.5	Segmentação em Multi-passos - Peeters <i>et al.</i> (2002b)	65
4.4.6	Segmentação em Multi-passos com HMM de Misturas de Gaussianas*	68
4.5	Segmentação Musical Não-supervisionada em Tempo Real	77
4.5.1	Critério de Informação Bayesiano e Soma Cumulativa	78
4.5.2	Segmentação com Hiper-elipses*	80
4.5.3	Modelos escondidos de Markov Adaptativos*	81
4.6	Pós-processamento	83
4.6.1	Suavização por moda dos vizinhos	84
4.7	Avaliação dos Segmentadores	84
5	Resultados em Seleção de Descritores e Segmentação	91
5.1	Geração de Dados e Seleção de Descritores	93
5.1.1	Análise das amostras	94
5.1.2	Descritores selecionados	95
5.2	Segmentação Musical	102
5.2.1	Visão de Tipo de Descritor	106
5.2.2	Visão de Tipo de Descritor Dinâmico	110
5.2.3	Visão de Memória Temporal	111
6	Conclusões	119
6.1	Considerações Finais	120
6.2	Sugestões para Pesquisas Futuras	120
A	Descritores de Áudio	123
A.1	Taxonomia do Projeto CUIDADO	123
A.1.1	Descritores Temporais (Globais e Instantâneos)	125
A.1.2	Descritores de Energia (Instantâneos)	129
A.1.3	Descritores Espectrais (Instantâneos e Globais)	129
A.1.4	Descritores Harmônicos (Instantâneos e Globais)	132
A.1.5	Descritores Perceptuais (Instantâneos)	134
A.1.6	Modelagem Temporal	136
A.2	Projeto JAUDIO	137

B Trechos Musicais agrupados por Cluster Hierárquico	139
B.1 Tabelas dos grupos de Trechos Musicais	142
C Estatística	145
C.1 Razão da máxima verossimilhança estatística	145
D Códigos e Scripts	147
D.1 Arquivo .orc Csound	147
D.2 Exemplo de arquivo .sco Csound	148
D.3 Exemplo de arquivo de transição	148
Referências Bibliográficas	149

Lista de Abreviaturas

AHMM	Modelos Escondidos de Markov Adaptativos (<i>Adaptive Hidden Markov Models</i>).
ALLDs	Todos os Descritores de Baixo Nível (<i>All Low Level Descriptors</i>).
BIC	Critério de Informação Bayesiana (<i>Bayesian Information Criterion</i>).
CART	Arvore de Classificação e Regressão (<i>Classification And Regression Tree</i>).
COOPER/FOOTE	Segmentação via Matriz de Similaridade de Cooper e Foote (2002).
CUSUM	Segmentação por Soma Cumulativa .
DD	Descritores Dinâmicos.
DDBM	Descritores Dinâmicos por Bandas Mel.
DDC	Descritores Dinâmicos Cumulativos.
DFT	Transformada discreta de Fourier (<i>Discrete Fourier Transform</i>).
EER	Erro que penaliza Estimação do número de Rótulos.
EPT	Erro de Precisão Temporal.
fdp	Função Densidade de Probabilidade.
FFTCs	Coeficientes da Transformada de Fourier de tempo reduzido (<i>Fast Fourier Transform coefficients</i>).
GMM	Modelos de Misturas Gaussianas(<i>Gaussian Mixture Models</i>).
HIPER-ELIPSES	Segmentação Hiper-elipses.
HMM	Modelos escondidos de Markov (<i>Hidden Markov Models</i>).
IRMFSP	<i>Inertia Ratio Maximization using Feature Space Projection.</i>
K-NN	K-Vizinhos mais próximos (<i>K-Nearest Neighbors</i>).
LDA	Análise de Discriminantes Lineares (<i>Linear Discriminant Analysis</i>).
LLDs	Descritores de Baixo Nível (<i>Low Level Descriptors</i>).
LPC	Coeficientes de Predição Linear (<i>Linear Predictive Coefficients</i>).
MDISS	Segmentação via Matriz de Dissimilaridade.
MDISS-IPROC	Segmentação por Dissimilaridade e Processamento de Imagem.
MFCC	Coeficientes Mel cepstrais (<i>Mel-Frequency Cepstrum Coefficients</i>).
MLP	Perceptron de Multi Camadas (<i>Multi Layer Perceptron</i>).

MPS-GHMM	Segmentação em Multi-Passos de com HMM de Misturas de Gaussianas.
MPS-PBR	Segmentação em Multi-Passos de Peeters <i>et al.</i> (2002b) <i>(Multi-pass segmentation).</i>
NBC	Classificador Ingênuo de Bayes (<i>Naive Bayes Classifier</i>).
PCA	Análise de Componentes Principais <i>(Principal Component Analysis).</i>
RCEPS	Coeficientes cepstrais (<i>Real Cepstral Coefficients</i>).
TZ/COOK	Segmentação por Delta de Mahalanobis de Tzanetakis & Cook (1999).
ZCR	Taxa de Cruzamento de zero (<i>Zero Crossing Rate</i>).

Lista de Figuras

2.1	Criação do banco de dados de trechos musicais	8
2.2	Método de segmentação, com a geração de dados musicais, obtenção de descritores, segmentação e avaliação dos resultados.	9
2.3	Tipos de descritores gerados nos dados de teste.	13
3.1	Amostras de um sinal digital.	18
3.2	Janelamento	19
3.3	Processo de extração de descritores propostos pelo projeto CUIDADO (Peeters, 2004)	20
3.4	Boxplot para visualização de pontos fora da curva para os descritores LPC_3 e Method of Moments ₁ para observações do Grupo 12	23
3.5	Função de Densidade da primeira componente do descritor LPC para as classes de sons agrupados (Grupo 2 e Grupo 9).	24
3.6	Estimativa da curva ROC da primeira componente do descritor LPC para as classes dos Grupo 1 e 9.	24
3.7	Visão geral do procedimento de seleção de descritores.	26
3.8	Duas funções de densidade de probabilidade, um limiar e as áreas falso-positivo (FP), falso-negativo (FN), verdadeiro-positivo (VP) e verdadeiro-negativo (VN). . .	27
3.9	Exemplos (a)(b)(c)(d) de sobreposições de distribuições de probabilidades e (e) as curvas ROC para os pares de distribuições.	28
3.10	Média das áreas sob a curva ROC de 65 descritores e 5 segmentos e os limiares para corte dados pelos algoritmos 2, 3 e 4. Para o algoritmo 2, $\alpha = .9$, no algoritmo 3 $\alpha = .001$ e $\beta = 0.07$, e no algoritmo 4 $\alpha = .001$ e $\beta = 0.05$	31
3.11	Função exponencial para o cálculo do descritor dinâmico por média ponderada. .	38
3.12	Processo de extração de descritores (Peeters <i>et al.</i> , 2002b). Da esquerda para a direita: sinal, banco de filtros, sinal de cada filtro, STFT para cada sinal filtrado . . .	39
4.1	Primeiros compassos de Quarteto de cordas, Op. 28 - Anton Webern	43
4.2	Gráfico de dispersão de um conjunto de dados representados por três funções normais	49
4.3	Mesmo conjunto de dados da figura 4.2 representado por duas funções normais, agrupadas por um critério de medida de dissimilaridade (distância de Bhattacharyya). .	49
4.4	Dendrograma de um agrupamento hierárquico realizado para agrupar 367 vetores de trechos de músicas	52
4.5	Matriz de dissimilaridade de um sinal musical.	59

4.6	Sequência de dissimilaridade utilizando algoritmo 9. Os pontos vermelhos indicam as mudanças reais de seção, adquiridas manualmente.	60
4.7	Representação da matriz de dissimilaridade e a direção de varredura da vizinhança.	60
4.8	Resultado da primeira etapa da segmentação utilizando descritor dinâmico (norma Euclidiana e memória temporal de 5 segundos). Sequência de dissimilaridade (figura superior), sendo os pontos vermelhos os picos encontrados; e matriz de dissimilaridade (figura inferior), onde os pontos vermelhos são os picos encontrados, os pontos verdes são os pontos de mudança reais (gabarito), os pontos amarelos, que coincidem com os pontos verdes, são os pontos depois do ajuste $P + L/2$, e a faixa azul são os pontos da vizinhança.	62
4.9	(a) Matriz de similaridade. (b) Pontos de segmentação estimados através dos picos da pontuação Q de Cooper e Foote (2002)	63
4.10	Matriz de dissimilaridade com distância de cosseno dos descritores dinâmicos	64
4.11	Matriz de dissimilaridade binária após técnicas de processamento de imagem com os pontos de mudança de seção.	64
4.12	Fluxo do algoritmo de segmentação em multi-passos proposto por Peeters <i>et al.</i> (2002b), que se resume em encontrar os estados potenciais através da matriz de similaridade; encontrar os estados iniciais pela redução dos estados encontrando aqueles que são redundantes; encontrar os estados intermediários pelo agrupamento das observações encontradas em cada estado; e por fim, considerar os estados intermediários em um modelo de HMM, onde o tempo é levado em consideração.	66
4.13	Segmentação manual de um sinal musical	67
4.14	Resultado de segmentação com algoritmo MPS-PBR-1 . Estados potenciais (superior-esquerdo); estados iniciais (superior direito); sequência de estados definida por K-Médias (central); e sequência de estados definida por HMM (inferior)	70
4.15	Resultado de segmentação com algoritmo MPS-PBR-2 . Estados potenciais (superior-esquerdo); estados iniciais (superior direito); sequência de estados definida por K-Médias (central); e sequência de estados definida por HMM (inferior)	71
4.16	Resultado de segmentação com algoritmo MPS-PBR-3 . Estados potenciais (superior-esquerdo); estados iniciais (superior direito); sequência de estados definida por K-Médias (central); e sequência de estados definida por HMM (inferior)	72
4.17	Resultado de segmentação com algoritmo MPS-PBR-4 . Estados potenciais (superior-esquerdo); estados iniciais (superior direito); sequência de estados definida por K-Médias (central); e sequência de estados definida por HMM (inferior)	73
4.18	Resultado de segmentação com algoritmo MPS-GHMM-1 . Estados potenciais (superior-esquerdo); estados iniciais (superior direito); e sequência de estados definida por HMM (inferior)	74
4.19	Resultado de segmentação com algoritmo MPS-GHMM-2 . Estados potenciais (superior-esquerdo); estados iniciais (superior direito); e sequência de estados definida por HMM (inferior)	75
4.20	Resultado de segmentação com algoritmo MPS-GHMM-3 . Estados potenciais (superior-esquerdo); estados iniciais (superior direito); e sequência de estados definida por HMM (inferior)	76

4.21 Sequência de estados sem o agrupamento das seções encontradas.	83
4.22 Sequência de estados com agrupamento hierárquico das seções encontradas.	83
4.23 Saída de uma segmentação antes do pós-processamento	84
4.24 Saída de uma segmentação depois do pós-processamento	84
4.25 Exemplo de mapeamento utilizando função bijetora, com $q = p$. Sequência de referência $f(t)$, ou gabarito (canto superior esquerdo); sequência de teste $g(t)$ (canto superior direito); sequência de mapeamento correspondente $h(t)$ (canto inferior esquerdo); e sequência de discrepâncias entre as funções f e h (canto inferior direito)	86
4.26 Exemplo de mapeamento utilizando função bijetora, com $q < p$. Sequência de referência $f(t)$, ou gabarito (canto superior esquerdo); sequência de teste $g(t)$ (canto superior direito); sequência de mapeamento correspondente $h(t)$ (canto inferior esquerdo); e sequência de discrepâncias entre as funções f e h (canto inferior direito)	87
4.27 Exemplo de mapeamento utilizando função bijetora, com $q > p$. Sequência de referência $f(t)$, ou gabarito (canto superior esquerdo); sequência de teste $g(t)$ (canto superior direito); sequência de mapeamento correspondente $h(t)$ (canto inferior esquerdo); e sequência de discrepâncias entre as funções f e h (canto inferior direito)	87
4.28 Exemplo de mapeamento ótimo (função injetora), com $q = p$. Sequência de referência $f(t)$, ou gabarito (canto superior esquerdo); sequência de teste $g(t)$ (canto superior direito); sequência de mapeamento correspondente $h(t)$ (canto inferior esquerdo); e sequência de discrepâncias entre as funções f e h (canto inferior direito)	88
4.29 Exemplo de mapeamento ótimo (função injetora), com $q < p$. Sequência de referência $f(t)$, ou gabarito (canto superior esquerdo); sequência de teste $g(t)$ (canto superior direito); sequência de mapeamento correspondente $h(t)$ (canto inferior esquerdo); e sequência de discrepâncias entre as funções f e h (canto inferior direito)	88
4.30 Exemplo de mapeamento ótimo (função sobrejetora), com $q > p$. Sequência de referência $f(t)$, ou gabarito (canto superior esquerdo); sequência de teste $g(t)$ (canto superior direito); sequência de mapeamento correspondente $h(t)$ (canto inferior esquerdo); e sequência de discrepâncias entre as funções f e h (canto inferior direito)	89
5.1 Descritores selecionados com DDC <i>weighted</i> e memória temporal de 0.5 segundos. Na parte superior esquerda e direita estão os rótulos das seções para cada amostra no tempo. Os gráficos restantes são as amostras para os oito primeiros descritores selecionados.	94
5.2 Descritores selecionados com DDC <i>weighted</i> e memória temporal de 0.5 segundos. Na parte superior esquerda e direita estão os rótulos das seções para cada amostra no tempo. Os gráficos restantes são as amostras para os sete últimos descritores selecionados.	95
5.3 Gráfico de dispersão dos descritores selecionados, considerando que existem 6 rótulos distintos para representar as seções. Na diagonal se encontram os histogramas dos eixos. Os eixos desta matriz estão ordenados da seguinte forma: MFCC ₂ , Momentos ₂ , MFCC ₃ , MFCC ₅ , MFCC ₄ , Spectral Smoothness ₃ , Momentos ₅ , Centróide do Espectro, Momentos ₃ , MFCC ₆ , LPC ₅ , Momentos ₄ , LPC ₃ , MFCC ₁₁ , Compacidade.	96
5.4 Árvore de descritores utilizados nos segmentadores para atingirem, na média, a taxa de erro mínima.	101

5.5	Gráfico de erro EER versus técnicas de segmentação, com ALLDs, DDC <i>weighted</i> e memória temporal de 1 segundo.	104
5.6	Gráfico de erro EPT versus técnicas de segmentação, com ALLDs, DDC <i>weighted</i> e memória temporal de 1 segundo.	105
5.7	Gráfico de erro EPT versus técnicas de segmentação não-supervisionadas, com MFCC, DDC <i>Euclidean</i> e memória temporal de 5 segundos.	106
5.8	Gráfico de erro EPT versus técnicas de segmentação não-supervisionadas, com DDBM e memória temporal de 5 segundos.	107
5.9	Erro médio agrupado por tipo DDC e tipo de descritor (MFCC e ALLDs).	109
5.10	Erro médio agrupado por tipo DDC e tipo de descritor (MFCC e ALLDs).	109
5.11	Gráfico de erro EPT com descritores MFCC e ALLDs.	110
5.12	Gráfico de erro EPT com descritores MFCC e ALLDs, gerados com DDC <i>Euclidean</i> e DDBM, ambos gerados com memória temporal de 0.5 segundos.	111
5.13	Gráfico de erro EPT com descritores MFCC e ALLDs, gerados com DDC <i>moments</i> e DDBM, ambos gerados com memória temporal de 0.5 segundos.	112
5.14	Gráfico de erro EPT com descritores MFCC e ALLDs, gerados com DDC <i>weighted</i> , e DDBM, ambos gerados com memória temporal de 1 segundo.	113
5.15	Gráfico de erro EPT com descritores MFCC e ALLDs, gerados com DDC <i>fft</i> , e DDBM, ambos gerados com memória temporal de 0.5 segundos.	113
5.16	Gráfico de erro EPT para os segmentadores supervisionados, com descritores MFCC e memória temporal de 0.5 segundos, agrupados por tipo de DDC.	114
5.17	Gráfico de erro EPT para os segmentadores não-supervisionados, com descritores MFCC e memória temporal de 0.5 segundos, agrupados por tipo de DDC.	114
5.18	Gráfico de erro EPT para os segmentadores não-supervisionados em tempo real, com descritores MFCC e memória temporal de 0.5 segundos, agrupados por tipo de DDC.	115
5.19	Gráfico de erro EPT para os segmentadores supervisionados, com descritores ALLDs e memória temporal de 1 segundo, agrupados por tipo de DDC.	115
5.20	Gráfico de erro EPT para os segmentadores não-supervisionados, com descritores ALLDs e memória temporal de 1 segundo, agrupados por tipo de DDC.	116
5.21	Gráfico de erro EPT para os segmentadores não-supervisionados em tempo real, com descritores ALLDs e memória temporal de 1 segundo, agrupados por tipo de DDC.	116
5.22	Gráfico de erro EPT para os segmentadores supervisionados, com descritores ALLDs e tipo de DDC <i>weighted</i> , agrupados por memória temporal.	117
5.23	Gráfico de erro EPT para os segmentadores não-supervisionados, com descritores MFCC e tipo de DDC <i>Euclidean</i> , agrupados por memória temporal.	117
5.24	Gráfico de erro EPT para os segmentadores não-supervisionados em tempo real, com descritores DDBM, agrupados por memória temporal.	118
A.1	Organização de Descritores Instantâneos do projeto CUIDADO	124
A.2	Organização de Descritores Globais do projeto CUIDADO	125
A.3	Porcentagem de RMS para trecho musical e limiares de 20% e 90% para início e fim de ataque.	126

A.4 A figura superior contem os esforços medidos para cada instante do sinal de áudio, o esforço médio \bar{w} , e o limiar $\bar{w}M$; e a figura inferior contem o gráfico da energia do sinal com os pontos do início e o fim dos ataques detectados.	127
A.5 Centroide temporal de uma envoltória RMS de instrumentos de corda.	128
A.6 Duração efetiva com limiar de 40% de uma envoltória de instrumentos de corda. .	129

Lista de Tabelas

3.1	Matrizes de similaridade de DD gerados por momentos estatísticos	37
3.2	Matrizes de similaridade de DD gerados pela norma Euclidiana	38
3.3	Matrizes de similaridade de DD gerados pela média ponderada	38
3.4	Matrizes de similaridade de DD gerados pela análise espectral	39
3.5	Matrizes de similaridade de DD (Peeters <i>et al.</i> , 2002b)	40
4.1	Dissimilaridade entre pares de modelos de seção, calculadas com a distância de Bhattacharyya. Na matriz, as tonalidades em branco indicam uma distância pequena, e que é possível um agrupamento.	48
4.2	Pontos de mudança de textura reais, detectados manualmente (em segundos) . .	65
4.3	Pontos de mudança de textura estimados (em segundos)	65
4.4	Configurações para o segmentador MPS-PBR	68
4.5	Configurações para o segmentador AHMM	82
5.1	Exemplos de leitura dos descritores segundo o padrão adotado.	93
5.2	Lista ordenada por frequência de uso dos descritores ALLDs após processo de seleção.	97
5.3	Valores de Memória Temporal utilizados para a geração de descritores dinâmicos.	98
5.4	Lista de descritores que foram selecionados duas ou mais vezes durante o processo de seleção de descritores, considerando o tipo de descritor dinâmico (ver seção 3.4.1) e a memória temporal em sua geração.	99
5.5	Erros EPT para os métodos de segmentação configurados com família de descritor ALLDs, tipo de descritor dinâmico <i>weighted</i> e memória temporal de 1 segundo. .	103
5.6	Erros EPT para os métodos de segmentação configurados com MFCC sem geração de descritor dinâmico.	105
5.7	Erros médios e as configurações de descritores para cada técnica de segmentação. .	108
B.1	Base musical utilizada para recortar os trechos dos dados de treinamento e geração musical.	142
B.2	Trechos musicais do Grupo 1.	142
B.3	Trechos musicais do Grupo 2.	143
B.4	Trechos musicais do Grupo 9.	143
B.5	Trechos musicais do Grupo 12.	143

Capítulo 1

Introdução

Ulisses respondeu: “Sublime deusa,
Não te agraves, portanto; eu sei que em
tudo

À prudente Penélope transcendes,
Nem da morte és escrava ou da velhice;
Mas para os lares meus partir suspiro.
Se um deus me impece, como os já
passados,
Suportarei constante os outros males”.

*Diálogo de Ulisses com a Deusa Calipso.
Homero, Odisseia. Tradução de Manuel
Odorico Mendes, Edusp, 1996.*

O homem sempre teve a necessidade de estruturar as informações percebidas externamente. Para exemplificar esta afirmação, basta olharmos ao nosso redor. O tempo cronológico (segundos, minutos, horas, dias) e as estações do ano (primavera, verão, outono e inverno) são exemplos em que alguma condição externa ao homem – como, por exemplo, o planeta Terra ter esta dimensão e estar localizado a uma distância particular do sol – forçaram o homem a dividir ou segmentar o tempo de alguma maneira. Evidentemente, a maneira como o homem decidiu dividir o tempo ou as estações não foi de forma aleatória, mas sim baseado em algum raciocínio lógico que explicasse claramente o motivo de uma região estar agora, por exemplo, no verão e outra no inverno.

Outro exemplo pode ser encontrado na literatura, onde o autor de um livro normalmente sugere algumas divisões de sua obra, criando o sumário. Suponha, no entanto, que exista pelo menos um livro que não contenha um sumário. O leitor leria parágrafos atrás de parágrafos até o fim deste livro, e podemos nos perguntar: este leitor seria capaz de dizer, durante ou ao fim da leitura, que um ou mais parágrafos adjacentes do livro fazem parte de um contexto só (podendo chamá-la de seção ou capítulo, não importa), distinto do restante da obra? Será que o leitor poderia inferir alguma estrutura lógica da obra sem que houvesse uma divisão pré-estabelecida? Vejamos por exemplo o caso de Ulisses, quando é acolhido na ilha da Deusa Calipso, onde passa 7 anos e, por fim, nega Calipso escolhendo uma existência mortal, porém cheia de glórias. A saída de Ulisses da ilha de Calipso é um exemplo de corte, de segmento, e o leitor – ou até mesmo Ulisses, se este pudesse sair dos livros e nos dizer – concordaria que as aventuras que Ulisses vivencia em seguida não fazem parte da época em que os dias eram sempre iguais na ilha de Calipso. Ulisses decide retornar e este ato é um momento de corte, de ruptura, do início de uma nova seção.

O exemplo mais claro ocorre na *música*. A música, por ser uma arte temporal (e, pelo menos do ponto de vista do ouvinte, não visual), geralmente nos remete ao seu próprio conteúdo para que possamos desvendar sua estrutura. Quando ouvimos uma música, podemos tentar a todo momento compreender sua estrutura que transparece através das repetições, dos padrões e, consequentemente, dos pontos de mudança, das transições. Estes pontos podem ser determinados

por uma mudança de melodia, uma mudança de tonalidade ou até mesmo, dentre outras possibilidades, por uma mudança na configuração de instrumentos empregados, que modificará aquilo que chamaremos de *timbre global* da música. Este esforço depende de nossa *memória* e de nosso intelecto para armazenar as informações até um certo instante e processá-las, buscando referências, correlações e relações com estruturas conhecidas da própria música ou de outras músicas e inclusive de outras artes.

Das características relacionadas a transições que podemos encontrar em uma música, podemos selecionar pelo menos uma que não envolva um conhecimento semântico da obra: o timbre. O timbre, no sentido acústico da palavra, é uma informação física e uma vez que se tenha aprendido a identificá-lo, é praticamente impossível esquecê-lo, como, por exemplo, no caso da voz de um familiar.

Segmentar é encontrar os pontos de mudança de timbres em uma música e *rotular* é dar nomes às seções encontradas. Outro termo utilizado para denominar estas ações é segmentação estrutural. Sendo assim, seria possível recuperar a estrutura musical ao realizar segmentações e rotulações nos trechos de uma única música de acordo com seu timbre global? Isto é o que vamos investigar. Não é do escopo desta pesquisa entrar no campo cognitivo ou da musicologia para o entendimento da estrutura musical, como, por exemplo, na análise de forma, fraseologia, período, tonalidade ou atonalidade, cadências, e assim por diante. Podemos dizer, no entanto, que esta pesquisa utiliza diretamente conhecimentos da área de inteligência artificial, mais especificamente de reconhecimento de padrões, recuperação de informação musical e, indiretamente, da área de psicoacústica, pois são as características geradas por esta área de conhecimento que auxiliam nos modelos dos segmentadores estudados.

Muitos dos argumentos encontrados em Moore (1990) podem ser interpretados sob o prisma de nosso problema. O que nos motiva a fazer esta pesquisa é a possibilidade de encontrar automaticamente as estruturas do material sonoro para os ouvintes de música em geral, e isto inclui compositores, musicólogos e artistas em geral. Não duvidamos que as tarefas de segmentação que nós humanos executamos são eficientes, mas se existe a possibilidade de desvendar algo que já existe no material sonoro e que possamos perceber (as seções musicais), então podemos assim criar outras relações, sejam elas já estabelecidas por uma análise manual/artesanal ou não. Estas informações podem auxiliar em outras aplicações um pouco mais complexas, como, por exemplo, navegação em conteúdo musical, a sumarização musical e até mesmo em performances musicais que utilizam processos interativos em tempo real que respondem a variações de timbres.

From a musical point of view, computers turn abstractions into concrete perceptions. (...) Many of the most profound problems of computer music lie in the development of new understandings relating what we perceive to what is there. (...) Music is our sense of time. Music draws our attention to a detailed progression of moments, the significance of which is apprehended as an abstraction of the succession of momentary awarenesses that make up a human lifetime. (...) Because music is a temporal art, its proper study necessarily includes a method for capturing, representing, and interpreting information about successive moments of time.

Richard F. Moore, "Elements of Computer Music" Moore (1990)

A solução adotada envolve o estudo de dois assuntos principais que não são necessariamente independentes: o entendimento do material sonoro, ou seja, a extração de características, e os modelos computacionais para resolver o problema da segmentação. O entendimento do material sonoro significa extrair os *descritores* sonoros (que são suas características), realizar um pós-processamento nestes descritores, possivelmente gerar novos descritores que capturem a característica temporal (o que vamos chamar de descritores dinâmicos) e *selecionar* aqueles descritores que melhor representam o material sonoro.

Os modelos computacionais estudados foram divididos em três categorias: segmentadores supervisionados, segmentadores não-supervisionados, e segmentadores não-supervisionados em tempo real. Em nossos experimentos, procuramos comparar diferentes técnicas computacionais,

sendo que algumas delas são técnicas bastante conhecidas, como, por exemplo, K-Vizinhos Mais Próximos, enquanto outras técnicas, encontradas na literatura, são sobreposições de conjuntos de técnicas, como algumas técnicas de segmentação em multi-passos que utilizam Modelos Ocultos de Markov e agrupamento de classes. Encontrar ou propor métodos de segmentação não-supervisionados em tempo real foi um dos principais objetivos desta pesquisa onde, nestes casos, gostaríamos de resolver um problema em que não temos nenhuma informação *a priori* do sinal musical, que é recebido continuamente através de um *stream* de áudio. As técnicas que estão assinaladas com um asterisco (*) são aquelas que nós propomos durante nossa pesquisa.

Nossos experimentos se basearam em um banco de dados gerados a partir de trechos musicais com timbres distintos entre si. A geração destes dados musicais serviu para minimizar o problema da subjetividade na determinação temporal de um segmento. A partir destes dados musicais, ampliamos a base de testes com a geração de descritores dinâmicos, ou seja, descritores que modelam a evolução temporal dos descritores originais.

Uma das limitações de nossa pesquisa é que a maior parte dos modelos computacionais utilizados partem do pressuposto que as seções são estacionárias, ou seja, que dentro de cada seção não há uma grande variação de timbres durante certos períodos de tempo e, em alguns casos, o usuário gostaria de considerar determinadas variações cíclicas ou frequentes como pertencendo a uma mesma seção. De certa forma, o que importa para avaliar corretamente uma segmentação é comparar os pontos de mudanças encontrados com os pontos de alguma referência previamente estabelecida. Entretanto, este não é um problema simples, pois devemos tanto avaliar a posição correta das fronteiras de seções, como também avaliar se os rótulos encontrados para cada seção estão corretos. Métricas tradicionais penalizam tanto a imprecisão temporal das fronteiras quanto o número de rótulos ou categorias utilizadas, o que é útil em alguns contextos e prejudicial em outros, especialmente na detecção de fronteiras em tempo real, onde a rotulação pode ter uma importância secundária. Para resolver este problema, propomos uma nova forma de avaliar o erro do segmentador, que penaliza apenas a localização errada das fronteiras, independentemente do número de rótulos distintos usados para representar uma mesma seção.

Os resultados de nossos experimentos em segmentação não-supervisionada mostram que é viável realizar estas tarefas, mesmo em tempo real. Enquanto as técnicas supervisionadas obtiveram taxas de erro baixas, tendo um mínimo de 1.89% de erro, outras técnicas, incluindo algumas novas propostas, chegaram a uma taxa de erro de até 5.23%. Dos métodos em tempo real que estamos propondo, a taxa de erro foi um pouco mais alta, chegando a 11.9%. Todas as taxas de erro acima foram medidas utilizando a métrica que penaliza somente a localização errada das fronteiras dos segmentos.

Objetivos

O principal objetivo desta pesquisa é comparar quantitativa e qualitativamente diferentes algoritmos de segmentação estrutural musical, feita sem informação *a priori* do sinal de áudio. Os métodos computacionais que montam esta estrutura basicamente realizam duas tarefas: segmentação do objeto sendo analisado, ou seja, definição dos pontos de transição, e rotulação das seções encontradas. Para atingir este objetivo, optamos por avaliar também os diferentes descritores sonoros, incluindo aqueles que modelam a evolução temporal de descritores tradicionalmente usados nesta tarefa. A partir dos resultados com cada família de descritores, vamos avaliar quais são as características destes que forneceram os melhores desempenhos em termos de segmentação estrutural.

Contribuições

As principais contribuições deste trabalho são as seguintes:

- Geração de dados musicais voltados para a área de segmentação musical.

O mesmo material sonoro poderia ser utilizado em outras áreas de pesquisa que necessitem de séries temporais rotuladas.

- Estudo sobre métodos de seleção de descritores.
- Novas propostas de segmentação estrutural em tempo real.
- Nova proposta para avaliar métodos de segmentação não-supervisionados.

Revisão de Literatura

A *Computação Musical* é uma sub-área da Ciência da Computação, e sua principal característica é a interdisciplinaridade, incluindo aspectos de arte, ciência e tecnologia Moore (1990). Um dos tópicos desta área é a *Recuperação de Informação Musical*, onde este estudo está situado. Das conferências internacionais mais importantes sobre este tópico, podemos citar as conferências da *International Society for Music Information Retrieval*¹ (ISMIR) e da *Sound and Music Computing*² (SMC), de onde colhemos grande parte das referências. Como mostra Pratyush e Serra (2010) ao analisar as publicações do SMC, este tópico teve uma tendência positiva na quantidade de contribuições, correspondendo a 11.63% dos artigos publicados em todas as conferências SMC, e a 20.97% dos artigos publicados em 2009 na mesma conferência.

Estudos relacionados também aparecem no Simpósio Brasileiro de Computação Musical³ (SBCM), onde foram apresentados trabalhos sobre descritores de áudio (Cabral *et al.*; Rocamora e Herrera, 2007), detecção de eventos musicais utilizando descritores sonoros (Malt e Jourdan, 2009), e uma metodologia para a geração automática de arquivos de áudio acompanhado de rótulos, descrevendo as amplitudes e frequências presentes em cada parcial (Mart Rocamora, 2009). Este último se aproxima de nossa pesquisa, uma vez que esta pesquisa também contribui para facilitar o trabalho de rotulação em sistemas de recuperação de informação musical.

A segmentação e rotulação musical é objeto de pesquisa em Peeters *et al.* (2002b), Aucouturier e Sandler (2001) e Tzanetakis e Cook (1999), onde diferentes técnicas de segmentação são propostas. Estes autores também são citados nos surveys de Dannenberg e Goto (2009) e Paulus *et al.* (2010), representando o atual estado da arte em segmentação estrutural musical. Sobre a segmentação em tempo real, buscamos referências em outras áreas da ciência da computação, que utilizam sinais de áudio não necessariamente musical, como Chen e Gopalakrishnan (1998), Sainath *et al.* (2007) e Omar *et al.* (2005).

Para a base teórica das técnicas computacionais de aprendizado de máquina, utilizamos principalmente os livros de Theodoridis e Koutroumbas (2008) e Duda *et al.* (2001), exceto para informações teóricas sobre Modelos Ocultos de Markov, para as quais utilizamos Rabiner (1989) e Rabiner e Juang (1993). De modo geral, observamos que, até a presente data, a segmentação musical em tempo real ainda não foi totalmente explorada em sinais de áudio musicais.

A seleção de descritores sonoros também foi objeto de nosso estudo, e as principais referências que utilizamos sobre a organização dos descritores são Kim *et al.* (2005), Peeters (2004) e McEnnis *et al.* (2005). Sobre os métodos de seleção de descritores, buscamos referências em Peeters e Rodet (2002a), Peeters (2003a) e Somol *et al.* (2006).

Organização do Trabalho

O capítulo 2 descreve a metodologia proposta nesta pesquisa, onde suas seções são um contorno dos capítulos seguintes. O estudo da obtenção de descritores sonoros (envolvendo desde o

¹<http://www.ismir.net>

²<http://www.smcnetwork.org>

³<http://compmus.ime.usp.br>

pré-processamento, extração, seleção de descritores e geração de descritores dinâmicos) é apresentado no capítulo 3, enquanto o capítulo 4 apresenta os métodos computacionais e modelos de aprendizado de máquina empregados na segmentação musical. Este último capítulo teórico apresenta as noções de separabilidade entre classes, aglomerados, segmentação supervisionada, segmentação não supervisionada, segmentação não supervisionada em tempo real, pós-processamento da saída da segmentação e avaliação dos segmentadores. Os resultados são apresentados no capítulo 5 e no capítulo 6 são apresentadas as conclusões deste estudo.

Capítulo 2

Metodologia

Dizer que um fenômeno evolui do estado A para o estado B é dizer que entre A e B fervilham detalhes e acidentes que eu negligencio, mas que sempre posso assinalar. Mas se eu considero a estrutura fina, no limite da precisão experimental, é preciso ter em conta um novo postulado: *o detalhe do detalhe não tem sentido experimental*; o detalhe do detalhe recai com efeito no nada absoluto do erro sistemático, do erro imposto pelas necessidades de detecção.

Gaston Bachelard, “A dialética da duração”

O método proposto nesta pesquisa é composto por quatro etapas: a geração automática dos dados musicais, a obtenção dos descriptores a partir das músicas geradas, a segmentação e a avaliação da segmentação. Procuramos, portanto, um método que possibilitasse a avaliação do segmentador, que permitisse a avaliação da segmentação de dezenas de músicas sem que fosse necessário rotulá-las manualmente. Com este método, resolvemos de forma simples um problema recorrente para os pesquisadores da área de recuperação de informação musical: o da obtenção de dados de treinamento e teste, contendo índices e dados musicais rotulados, com informações suficientes para a avaliação final da segmentação, ou seja, o instante no tempo em que ocorrem as transições e os rótulos de cada seção. Geralmente esta etapa de obtenção de dados de teste é realizada manualmente, o que gera um grande volume de trabalho e ainda fica sujeita à interpretação do pesquisador, que determina subjetivamente onde estão localizadas e como são rotuladas as seções musicais. A desvantagem do método que iremos apresentar é justamente o fato de que não estamos realizando experimentos em músicas reais. Entretanto, esta desvantagem nos parece menor do que o benefício alcançado pela geração automática de dados rotulados, pois o que buscamos é identificar pontos de transição entre um timbre musical A e um timbre musical B que o sucede, e, desta forma, o importante é garantir que tenhamos trechos distintos nas músicas.

2.1 Geração de dados musicais

Quando Bachelard diz que “o detalhe do detalhe não tem sentido experimental”, podemos entender que, para o nosso problema, as pequenas variações musicais que ocorrem durante uma seção não são importantes para o reconhecimento dos pontos A e B na música. A redução que propomos, da música como composição e arte para a música de teste gerada automaticamente é

puramente científica, e de fato, o resultado final não é “musical”, é somente um sinal de áudio formado por trechos musicais, onde as pequenas variações dentro de cada trecho são supostamente insignificantes. O detalhe não é importante, pois estamos interessados no contorno mais expressivo da música gerada. Assim, mais do que o método proposto, podemos dizer que é a qualidade da música gerada (permitindo a distinção objetiva entre as seções) que fornece a chave para a validação de nossos modelos computacionais. O primeiro passo foi montar a base de dados musicais, o que foi realizado recortando manualmente trechos de músicas que continham perceptualmente o mesmo timbre musical global (embora diferissem no detalhe do detalhe). A figura 2.1 ilustra esta etapa do processo, e a lista de músicas selecionadas que utilizamos para realizar os cortes de trechos musicais pode ser vista na tabela B.1 do apêndice B.

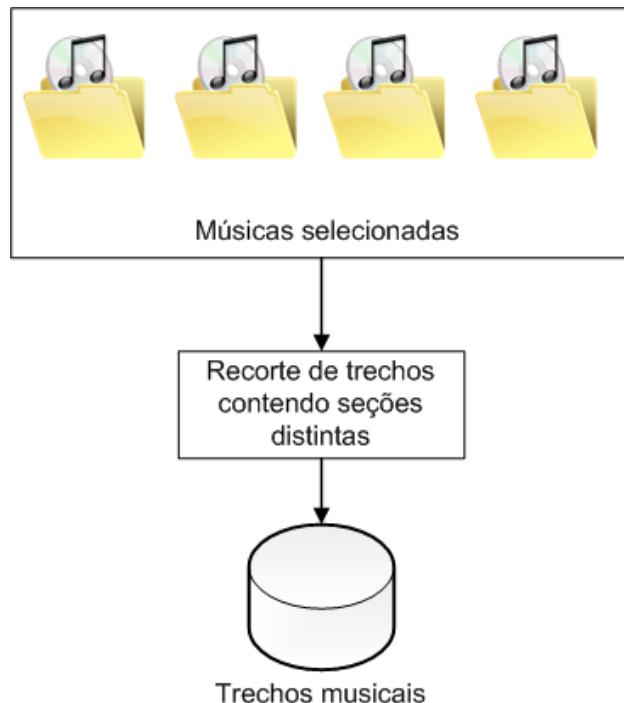


Figura 2.1: Criação do banco de dados de trechos musicais

Após a criação deste banco de dados de trechos musicais pudemos iniciar o procedimento de geração de dados musicais. Este procedimento, escrito em Java, Csound¹ e JAudio², compreende os seguintes passos:

1. Seleciona aleatoriamente um conjunto de trechos musicais do banco de dados;
2. Para cada trecho musical selecionado, determina aleatoriamente a duração de cada seção a ser gerada;
3. Gera um arquivo .wav com o Csound:
 - (a) Gera um arquivo .sco para o Csound;
 - (b) Executa o Csound com o arquivo .sco gerado e o arquivo .orc do apêndice D.1.
4. Gera os vetores de características utilizando o JAudio;
5. Gera o arquivo de transições com as seguintes colunas (veja exemplo no apêndice D.3):
 - (a) Nome do trecho musical

¹<http://csound.sourceforge.net/>

²<http://jaudio.sourceforge.net/>

- (b) Rótulo, que é um valor numérico dado para cada seção, onde o valor é igual para seções geradas com o mesmo trecho musical.
- (c) O instante, em segundos, do início de uma seção musical.
- (d) O instante, em número de amostras, do início de uma seção musical.

Com as informações dos pontos de transição, será possível avaliar os segmentadores, e com isto podemos executar as próximas etapas de nosso método. Veja no diagrama da figura 2.2 o esquema geral do método.

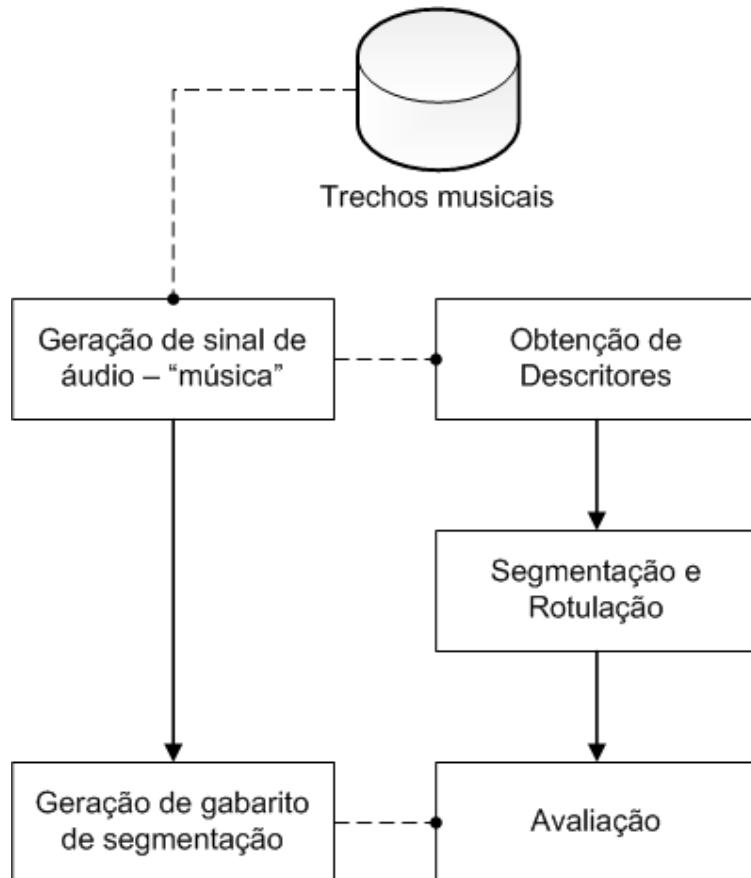


Figura 2.2: Método de segmentação, com a geração de dados musicais, obtenção de descritores, segmentação e avaliação dos resultados.

2.2 Obtenção de descritores

Como veremos com mais detalhes no capítulo 3, a obtenção de descritores de áudio é uma etapa à qual se deve dar bastante atenção, pois a modelagem dos dados influencia diretamente a qualidade dos modelos de classificadores construídos. Na prática de nosso método, a etapa de obtenção de descritores se inicia no item 4 da seção anterior – a geração dos vetores de características. Então, primeiramente, dirigimos a atenção para a importância da escolha dos eixos de descritores e para a qualidade dos dados gerados. Como não era do escopo desta pesquisa construir ferramentas de extração de descritores, optamos por utilizar ferramentas já existentes. Avaliamos duas ferramentas: Jaudio (McEnnis *et al.*, 2005) e Libxtract (Bullock, 2007). As duas apresentam vantagens, no entanto, JAudio se mostrou mais portátil e fácil de usar que Libxtract. Depois de selecionada a ferramenta, devemos, portanto, selecionar um conjunto de descritores a serem extraídos, ou seja, realizar a modelagem dos dados. Esta modelagem é realizada para cada contexto musical. Em nosso caso, por exemplo, eliminamos todos os descritores que têm como

base a frequência fundamental, pois tal informação não tem valor para o timbre musical. Assim, após a extração de descritores, entramos em uma tarefa de seleção de descritores, que envolve o pré-processamento das observações (tratamento de pontos fora da curva, normalização de dados), a eliminação de eixos redundantes ou eixos que não distinguem as diferentes seções musicais, e, por fim, a execução de um algoritmo de seleção que informe quais são os eixos mais relevantes.

O principal objetivo desta fase é utilizar o mínimo de descritores *originais* que maximizam o poder de discriminação entre as seções musicais. Dizemos “*originais*” para enfatizar que o que nos interessa são os eixos criados na etapa 3, e não novos eixos de descritores, gerados, por exemplo, a partir de uma combinação linear dos descritores originais. Na etapa 6 de seleção, utilizamos seis diferentes abordagens, e por fim, um método multi-critério seleciona o conjunto final.

Podemos então, enumerar os passos desta fase da seguinte maneira:

1. Selecionar / Adquirir ferramenta de extração de descritores de áudio;
2. Modelagem de dados: selecionar um conjunto de descritores disponibilizados pela ferramenta;
3. Extração de descritores;
4. Pré-processamento das observações;
5. Eliminação de eixos redundantes;
6. Seleção dos eixos mais relevantes.

O passos descritos até aqui são aqueles que normalmente são realizados em um sistema de classificação de padrões. No caso particular de nossa pesquisa, adicionamos outra etapa no processo, a geração de descritores dinâmicos. O papel destes descritores é captar o perfil global da música, e não somente um instante do tempo, como veremos com mais detalhes na seção 3.4. A consequência disto é que as opções para avaliar um segmentador aumentam, pois não temos somente um tipo de dado para comparar os resultados. Outra consequência é que, por estes descritores dependerem de um parâmetro temporal, que serve como um limitante no horizonte do tempo abrangido pelo descritor, a latência para a geração destes descritores é maior. Então, se estamos falando em um processamento em tempo real, existe ainda o tempo necessário para calcular todas as observações neste horizonte de tempo para só depois ter a observação final gerada.

2.3 Segmentação

As fases anteriores desta metodologia são necessárias para o entendimento do material com que estamos lidando, e por isso são muito importantes. No entanto, a segmentação musical é o foco principal desta pesquisa. O termo segmentação musical pode causar, no início, pouco entendimento de qual é o real objetivo a ser alcançado por esta pesquisa, pois o segmento musical é afinal algo subjetivo. De alguma forma o escopo desta pesquisa depende de como determinaremos quais são as características desejáveis de um segmento musical. As principais questões técnicas sobre a caracterização de segmentos serão respondidas no capítulo 4, onde também discutiremos as possíveis aplicações musicais de um método que possa segmentar e rotular uma música.

Os algoritmos estudados para resolver este tipo de problema exigem algumas ferramentas emprestadas de técnicas de reconhecimento de padrões, como, por exemplo, distâncias entre modelos normais (ver seção 4.1) e construção de aglomerados (ver seção 4.2); ambas serão utilizadas nos algoritmos apresentados.

De forma abrangente, as etapas da segmentação são as seguintes:

1. **Segmentar e rotular.** Considere dada uma sequência temporal O de vetores de características, tal que

$$O = O_1, \dots, O_i, \dots, O_T \quad (2.1)$$

onde $O_i \in \mathbb{R}^d$. Como veremos mais adiante no capítulo 3, cada observação é um vetor de características que é extraído de uma janela temporal do sinal de áudio. Segmentar e rotular esta sequência significa identificar os pontos onde existem transições entre diferentes seções da música, e rotular cada seção, identificando seções similares através dos mesmos rótulos. Em outras palavras, se a música tem um certo número de segmentos s , então gostaríamos de criar s partições de O , e $r \leq s$ rótulos associados. A noção de “similaridade” entre seções será melhor discutida na seção 4.1. A duração dos segmentos é algo com o que também devemos nos preocupar, e neste modelo, a duração da seção é determinada pelo número de observações que estão contidas nela.

2. **Pós-processamento.** Depois que sabemos o perfil da segmentação, o resultado pode ser visto como uma função de rotulação $f : \mathbb{N} \rightarrow \{1, \dots, q\}$. O objetivo deste pós-processamento é minimizar as transições muito rápidas entre seções, pois supomos que seções musicais possuem uma certa estabilidade temporal (definida por algum limiar de origem perceptual). Esta “suavização” da função f pode ser feita, por exemplo, substituindo os valores de cada observação rotulada pela moda das observações vizinhas. A quantidade dos vizinhos a ser verificada depende somente da duração mínima de uma seção que gostaríamos de identificar.
3. **Avaliação.** A última etapa do nosso método visa a avaliação dos segmentadores em termos do erro médio. O erro da segmentação merece atenção especial, principalmente quando queremos avaliar os resultados de uma segmentação não-supervisionada, pois não temos informações a priori de qual rótulo do resultado deveria ser mapeado em qual rótulo do gabarito. Por exemplo, se uma música foi gerada a partir de seções com rótulos $\{1, 2, 3\}$, e o resultado de um segmentador devolve os rótulos $\{3, 1, 2\}$, comparar ingenuamente observação a observação forneceria um erro máximo, quando na verdade esta rotulação possui um erro mínimo. Para contornar este problema, deve-se mapear os resultados em uma lista de símbolos mais apropriada, e só então produzir a avaliação do erro. Isto será melhor discutido na seção 4.7.

Os algoritmos de segmentação apresentados estão divididos em três categorias: supervisionados, não-supervisionados e não-supervisionados em tempo real. Ao nosso ver, esta categorização ajuda a distinguir quais são as aplicações para as diferentes técnicas, além de deixar claro para o leitor a quantidade de informação *a priori* que temos em cada momento. No primeiro caso, temos os dados de treinamento e, por consequência, o número de seções, mas não necessariamente conhecemos todo o sinal de antemão; no segundo caso, o requisito é que todo o sinal já esteja disponível de antemão, mas não se supõe conhecimento *a priori* nem da quantidade de seções nem de suas características; e no terceiro caso, da segmentação não-supervisionada em tempo real, não temos nenhuma informação *a priori* e o sinal só é conhecido até o instante presente (correspondente ao tempo presente de execução da segmentação). Ao longo do texto, o leitor perceberá que as técnicas apresentadas nas primeiras seções acabam sendo utilizadas para montar técnicas mais complexas nas seções seguintes. Isto quer dizer que algumas das técnicas dos métodos supervisionados acabam sendo utilizadas, de alguma forma, nos métodos não-supervisionados, e estas sendo utilizadas nos métodos não-supervisionados em tempo real.

A exposição dos métodos em tempo real aborda somente questões teóricas da segmentação estrutural e deixa em aberto as questões teóricas de sistemas em tempo real, como complexidade computacional das técnicas e do consumo de CPU. Primeiramente o sistema em tempo real deve aguardar a leitura da janela de análise, que será utilizada para a extração dos descritores sonoros. Em seguida o sistema pode gerar descritores dinâmicos para depois repassar a observação para um método de segmentação, que deve construir os modelos probabilísticos e retornar com uma solução para a segmentação. A extração dos descritores sonoros juntamente com a geração de descritores dinâmicos e o tempo computacional necessário para a construção dos modelos probabilísticos devem ocorrer em um tempo inferior ao tempo de leitura da próxima janela, caso

contrário o sistema não será capaz de devolver uma informação em tempo real. Uma forma de resolver este problema é aumentar a janela de análise.

2.4 Resultados Finais

Podemos agora definir nossos objetivos de forma mais prática, uma vez que temos uma visão geral de nosso problema. O *objetivo* de nosso método é avaliar não somente cada segmentação através dos métodos propostos, mas tentar responder às seguintes perguntas:

1. Sobre os descritores:
 - (a) É mais vantajoso realizar uma seleção automática de descritores, ou podemos simplesmente utilizar os coeficientes MFCC³?
 - (b) O custo/latência para a geração de descritores dinâmicos compensa em termos de probabilidade de erro dos segmentadores? Ou seja, o segmentador tem um melhor desempenho se descritores dinâmicos forem utilizados? Em caso afirmativo, qual seria um valor de memória temporal adequada para a geração destes descritores?
 - (c) Na média, quais são os descritores mais selecionados dentre um conjunto fixo de descritores?
2. Sobre os segmentadores:
 - (a) O desempenho dos segmentadores supervisionados é realmente muito superior aos dos segmentadores não-supervisionados, a ponto de justificar o custo de obtenção dos dados de treinamento?
 - (b) Se não temos os dados de treinamento, mas sabemos o número total de seções, podemos construir um segmentador com um melhor desempenho?
 - (c) A segmentação em tempo real é possível? As restrições impostas a ela acarretam em algum tipo de erro no segmentador? Este erro é relevante?

Para responder a estas perguntas, montamos um conjunto de testes contendo três diferentes grupos de configurações de descritores. Isto quer dizer que para cada sinal musical gerado, extraímos diferentes descritores para serem avaliados pelos mesmos segmentadores. O diagrama da figura 2.3 ilustra o processo de geração de dados de teste desta nossa pesquisa. Os grupos estão divididos da seguinte forma:

1. **Descritores simples.** São aqueles gerados pela ferramenta JAUDIO, sendo que os descritores finais foram ou não selecionados automaticamente do grupo original. Basicamente executamos duas extrações para formar os descritores, uma utilizando somente os coeficientes MFCC e outra utilizando um conjunto grande de descritores disponibilizados pela ferramenta, que chamaremos de Todos os Descritores de Baixo Nível (**ALLDs**⁴).
2. **Descritores Dinâmicos Cumulativos (DDC).** Estes descritores são gerados a partir dos descritores simples. Como veremos mais adiante no capítulo 3, o método que extrai estes descritores depende de dois parâmetros: o tempo em segundos, que corresponde a uma memória temporal do descriptor (e busca imitar uma memória musical), e o tipo de suavização do descriptor dinâmico (usando momentos estatísticos, norma Euclidiana, média ponderada e coeficientes da FFT).
3. **Descritores Dinâmicos por Bandas Mel (DDBM).** Estes descritores são gerados a partir do sinal de áudio, e o método de extração depende de um parâmetro: o tempo em segundos, que também corresponde a uma memória temporal. Veremos mais detalhes sobre este descriptor na seção 3.4.2.

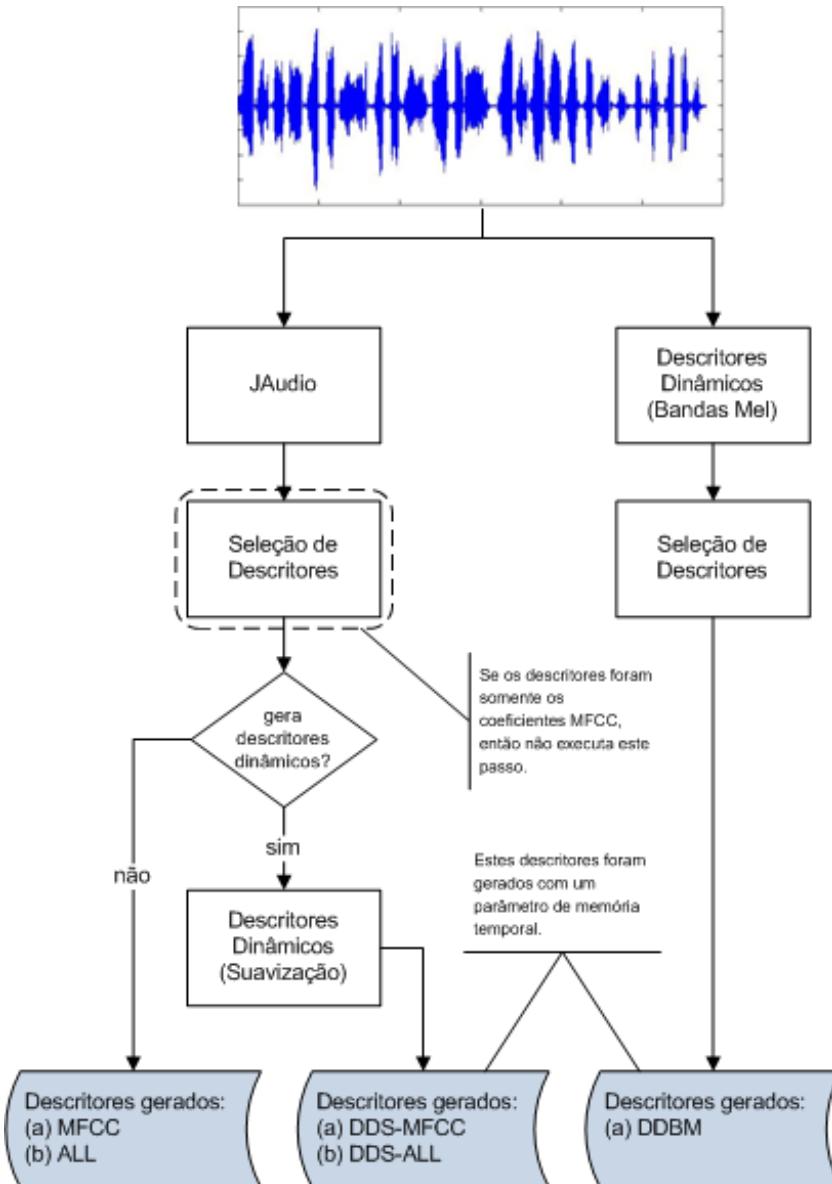


Figura 2.3: Tipos de descritores gerados nos dados de teste.

Assim, nossos resultados podem ser analisados sob pontos de vista distintos. Separamos os resultados em três grupos de gráficos:

- Visão de Tipo de Descriptor.** Nesta visão queremos analisar o desempenho dos classificadores em termos de conjunto de descritores, ou seja, para uma combinação específica de tipo de descritor dinâmico cumulativo (caso seja aplicável) e um valor de memória temporal associada à sua geração, queremos comparar o desempenho do conjunto MFCC com o conjunto de descritores selecionados por nosso método de seleção de descritores (ALLDs) e os descritores dinâmicos por bandas Mel (DDBM).
- Visão de Tipo de Descritor Dinâmico Cumulativo.** Nesta visão queremos comparar o desempenho dos diferentes tipos de descritores dinâmicos cumulativos. Neste caso, podemos compará-los se agrupamos os resultados por conjunto de descritores (MFCC ou ALLDs) e memória temporal.

³Para mais detalhes sobre MFCC, veja apêndice A

⁴ALLDs, do inglês All Low Level Descriptors.

3. Visão de Memória Temporal. Esta visão permite analisar o desempenho dos segmentadores em termos do tempo em segundos do descriptor dinâmico utilizado. Para isto, agrupamos os resultados por conjunto de descritores (MFCC, ALLDs ou DDBM) e tipo de descriptor dinâmico cumulativo (e.g. norma Euclidian), caso seja aplicável⁵.

Com estas visões poderemos extrair as informações necessárias para responder às perguntas de nossa pesquisa.

⁵Lembramos ao leitor que o descriptor DDBM não possui parâmetros de suavização, tendo somente o parâmetro de tempo (memória temporal).

Capítulo 3

Descritores de Áudio

Não é porque ignoramos o que irá intervir que deixamos de prever a eficácia absoluta de uma causa dada; é porque, da causa ao efeito, há uma intervenção totalmente probabilística de acontecimentos que não estão, de maneira nenhuma, ligados ao dado causal. Em particular, não teremos nunca o direito de tomar o intervalo *como dado*. Na ciência, podemos construir alguns fenômenos, podemos proteger o intervalo diante de certas perturbações, mas não poderíamos eliminar toda intervenção de fenômenos imprevistos no intervalo da causa ao efeito.

Gaston Bachelard, "A dialética da duração"

UM descritor de áudio é qualquer aspecto qualitativa ou quantitativamente mensurável do som (Bullock, 2007), ou, como o próprio nome diz, um descritor é uma característica que de alguma forma descreve o som (Peeters, 2004). Por nós humanos, o som pode ser percebido, por exemplo, como forte ou fraco - no aspecto de intensidade -, e musicalmente percebemos sua curva melódica, ritmo, textura e timbre. Estas extrações de características do som são feitas em conjunto por nosso aparelho auditivo e o cérebro.

Grosso modo, o aparelho auditivo é dividido em três seções principais: o ouvido externo, o ouvido médio e o ouvido interno. O ouvido externo amplifica as vibrações de entrada do ar; o ouvido médio converte as vibrações do ar em vibrações mecânicas; e o ouvido interno, por sua vez, executa um processamento nas vibrações de modo a convertê-las em sinais eletroquímicos que são transmitidos através dos nervos para nosso cérebro, atingindo o córtex auditivo (Roads, 2004).

O cérebro avalia os aspectos qualitativos e quantitativos da informação sonora, interpretando, reconhecendo, memorizando e processando toda a informação. E faz de tal forma que ainda não nos é possível simular através de modelos computacionais, dada a potencialidade e plasticidade de nosso córtex auditivo.

No contexto de processamento de sinais, o som é descrito por vetores ou por conjunto de vetores, formando uma série temporal multivariada; e os eixos destes vetores são os descritores sonoros extraídos diretamente do sinal de áudio.

Eixos de Percepção ou Taxonomia de Descritores

Diversos estudos foram realizados na tentativa de classificar os aspectos mensuráveis do som. Alguns destes estudos estão voltados à percepção do som, ou seja, tentam explicar os aspectos do som através da percepção do homem, e outros estão interessados em rotular um som de acordo com uma taxonomia previamente estabelecida. Estudos baseados na percepção são realizados sistematicamente desde Helmholtz (von Helmholtz, 1865) e são estritamente práticos, pois dependem de testes aplicados em seres humanos. Em todos estes estudos, os sujeitos são questionados sobre a dissimilaridade entre um par de sons, em aspectos tais como “ataque”, “brilho”, etc., derivando o que se chama de “*espaço de timbres*”, que é uma forma de representar os julgamentos em eixos de percepção. Estudos recentes tentaram descrever quantitativamente os eixos de percepção, ou seja, relacionar os eixos de percepção com descritores derivados diretamente do sinal sonoro. Não faz parte do escopo deste trabalho detalhar os resultados destes estudos, e em Herrera-Boyer *et al.* (2003), o leitor poderá encontrar um panorama dos mesmos e das conclusões obtidas.

Uma taxonomia padronizada para os descritores serve a dois propósitos: descrever e representar características do sinal de áudio da música em um nível superior de abstração, e permitir que usuários especifiquem os descritores para busca e/ou classificação, daí a importância de uma nomenclatura padronizada. Buscamos, todavia, nesta etapa do trabalho, descritores que são numericamente quantificáveis, para que, de forma determinística, possamos extrair os eixos, sejam ou não relativos à percepção humana, que melhor auxiliem o processo de segmentação. De modo geral, a escolha dos descritores depende do tipo de material sonoro sendo utilizado. Por esta razão, um método de extração de descritores que tenta explicar de forma numérica os eixos de percepção se difere de um método baseado em uma taxonomia.

Neste contexto, o de sinais de áudio digital, os descritores são extraídos diretamente do sinal, ou de transformações do sinal, como a Transformada de Fourier aplicada a janelas de tempo de alguns milissegundos, de forma a apreender a evolução micro-temporal destes descritores. Outros descritores dependem de uma janela de tempo maior ou de uma sumarização dos valores micro-temporais, através de momentos estatísticos ou da derivada dos mesmos no domínio do tempo, como por exemplo, o logaritmo do tempo do ataque (Herrera-Boyer *et al.*, 2003).

Em nossos estudos identificamos algumas tentativas de organizar estes descritores. A primeira foi o padrão *MPEG-7* (Multimedia Content Description Interface) iniciado em 1997, com o principal objetivo de permitir que usuários e agentes possam procurar, identificar, filtrar e navegar em conteúdos audiovisuais. Foram adotados 17 descritores de baixo nível, divididos em 6 categorias:

- *descritores básicos*, que são os descritores no domínio do tempo, e.g., sinal em forma de onda e a energia do sinal;
- *descritores espectrais básicos*, que são os descritores extraídos do espectro, e.g., envelope, centroide, spread e flatness;
- *descritores de parâmetros do sinal*, que se aplicam a sons periódicos, e.g., frequência fundamental e harmonicidade;
- *descritores timbrísticos temporais*, e.g., logaritmo do tempo de ataque e centroide temporal;
- *descritores timbrísticos espectrais*, e.g., centroide do espectro, desvio do espectro harmônico, espalhamento do espectro harmônico e variação do espectro harmônico;
- *representações da base do espectro*, que são descritores utilizados para classificação, e.g., base do espectro e projeção do espectro (Kim *et al.*, 2005).

Como podemos ver pelo contorno que o padrão *MPEG-7* oferece, existe um déficit em descritores de níveis superiores. Outra tentativa de organização dos descritores é o obtido no projeto *CUIDADO* (Peeters, 2004), que estende o padrão *MPEG-7* e oferece um total de 72 descritores de áudio para classificação e recuperação musical. Em um estudo mais recente do grupo de Música e Tecnologia da Universidade McGill (Montreal - Canadá), foi criado o *jAudio*, uma ferramenta

de extração de descritores de sinais musicais, tendo ao todo 27 descritores de baixo, médio e alto nível. Um dos objetivos deste grupo com esta ferramenta é o de fornecer descritores de alta qualidade e que a mesma possa ser estendida facilmente, possibilitando a criação de novos descritores sem muito esforço (McEnnis *et al.*, 2005). O leitor interessado poderá encontrar mais detalhes sobre as taxonomias do projeto *CUIDADO* e do projeto *jAudio* no apêndice A.

Análise, Extração e Seleção

No contexto de aprendizado de máquina, a boa escolha dos descritores é determinante no resultado final da classificação – ou segmentação, em nosso caso. Usualmente o número de descritores em um sistema de classificação é muito grande, e dois problemas associados a isto é a chamada “maldição de dimensionalidade” (ver Theodoridis e Koutroumbas (2008)) e o custo computacional associado à adição de novos eixos.

Fiebrink e Fujinaga (2006) demonstram que a utilização do método Encapsulado de Seleção de Descritores (do inglês *Wrapper Feature Selection*), no contexto de classificação musical, não garante sempre um melhor resultado na classificação; isto pôde ser comprovado através dos resultados de uma classificação supervisionada com redução por PCA – utilizados como referência-, onde houve uma redução de 74 pra 36 descritores, cujas variâncias correspondiam a 95% dos descritores originais, atingindo uma precisão similar ao reduzir a dimensão através da Seleção Encapsulada com o mesmo classificador. Embora a técnica de seleção descrita pelos autores difira da apresentada na seção 3.3 no que se refere à inclusão de um classificador (treinamento e teste) durante o procedimento, os mesmos testes e argumentos elaborados pelo grupo de pesquisadores poderiam ser colocados a respeito das técnicas utilizadas neste trabalho. Liu e Yu (2005) descrevem a técnica Encapsulada como sendo mais robusta que a técnica utilizada em nossos estudos, e, portanto, os resultados da seleção neste trabalho pode ter um resultado ainda pior que utilizando uma redução de dimensão pelo método Encapsulado. Uma justificativa para não incorporar um classificador na etapa de seleção corresponde à intenção de testar um conjunto de técnicas de classificação, e não somente uma, o que pode dificultar a escolha de qual classificador utilizar neste método. Como veremos no capítulo 5, a combinação de técnicas com parâmetros de seleção e descritores dinâmicos já forma um conjunto bastante expressivo para uma comparação.

A motivação que nos leva a incluir o procedimento de seleção em nosso trabalho segue da necessidade de obter um conjunto de descritores relativamente pequeno, mas eficiente, tendo em vista o problema de segmentação musical em tempo real – o que tornaria impossível considerar a combinação linear (PCA) de todas as observações. Para testar se estamos no melhor caminho, partimos da hipótese que selecionar um conjunto de descritores fornece um desempenho de segmentação melhor que utilizar somente o descritor MFCC, veja no capítulo 5 os resultados de desempenho em segmentação utilizando os descritores selecionados.

Esta breve introdução ao tema da seleção esclarece um ponto importante: que apesar da *seleção de descritores* não ser o foco principal de nosso trabalho, ela é sim uma etapa importante para o sucesso da segmentação final. Assim, nas próximas seções vamos fazer uma introdução dos procedimentos básicos de extração de descritores, desde a análise do sinal de áudio e seu recorte até a obtenção dos descritores finais.

3.1 Análise do Som

Nesta seção vamos expor os conceitos fundamentais da análise do som, de forma que o leitor que não esteja familiarizado com este assunto possa recorrer a este material. Não é, porém, o intuito deste material exaurir todo o assunto em tão poucas páginas, mas somente oferecer um ponto de referência sobre o tema. Para o leitor que estiver interessado em mais detalhes, encorajamos fortemente a leitura de Oppenheim *et al.* (1989), e o leitor que se sentir mais à vontade sobre este assunto, poderá avançar diretamente para a seção 3.2.

Sinais Digitais Sonoros

O conceito central da digitalização de sinais é a amostragem, ou seja, converter sinais analógicos contínuos em sinais discretos amostrados no tempo. Uma forma de conversão analógica para digital é a chamada *Pulse Code Modulation* (PCM), onde o sinal analógico é substituído por uma sequência de códigos binários, chamadas amostras (*samples*). O processo ocorre em três estágios:

1. **filtro passa-baixa**, para eliminar frequências acima da frequência de Nyquist;
2. **amostragem**, que mede as amplitudes das vibrações a cada intervalo fixo de tempo; e
3. **quantização**, que converte cada medida em um valor numérico correspondente.

O número de bits utilizados para representar cada amostra determina o nível de ruído e o alcance da amplitude suportado pela conversão – para mais detalhes, ver *quantização* em Moore (1990).

A taxa de amostragem – *sampling frequency* – na qual as amostras são medidas é comumente expressa em amostras por segundo, ou simplesmente Hertz (Hz), e tem uma relação direta com a largura de banda passível de representação (Teorema de Nyquist). Por exemplo, a taxa de amostragem de um CD é normalmente 44.1 KHz, o que equivale a 44100 amostras por segundo, e com esta taxa de amostragem é possível representar frequências de até pouco mais de 20 KHz, o que corresponde ao limiar superior médio do ouvido humano.

O sinal de áudio digital y é, portanto, uma função no tempo, e pode ser representado matematicamente como uma sequência de números, e visualmente por um gráfico no domínio do tempo versus sua amplitude (veja figura 3.1). Formalmente,

$$y = \{y(n)\}, -\infty \leq n \leq \infty. \quad (3.1)$$

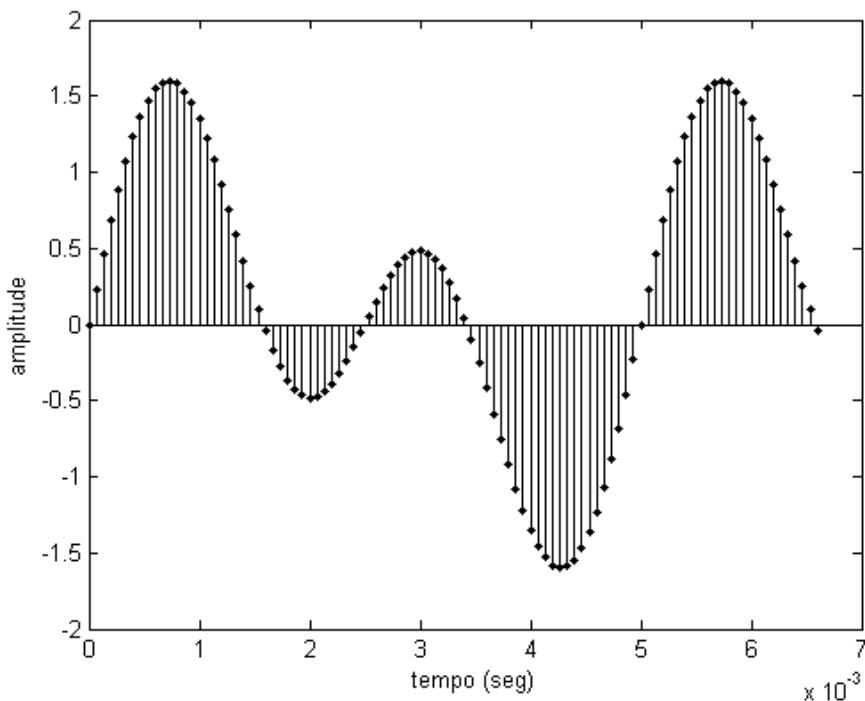


Figura 3.1: Amostras de um sinal digital.

Janela de Análise

O janelamento (do inglês *windowing*) é uma operação comum em análise de sinais. Normalmente, em sistemas de recuperação de informação musical, a análise do sinal de áudio ocorre em

janelas de poucos milissegundos, tipicamente cerca de 50 ms. Podemos listar pelo menos duas razões desta restrição de tempo reduzido: o custo computacional de executar uma análise do espectro com um trecho de sinal muito grande, e a latência na resposta, quando a mesma faz parte dos requisitos do sistema – quanto maior for a janela de análise, maior será o atraso para uma resposta. Uma visão geral do procedimento pode ser encontrada na figura 3.2, onde o sinal é recortado de acordo com o tamanho da janela e multiplicado por uma função de janelamento (hamming, triangular, quadrada, etc.), gerando assim um novo sinal de áudio de curta duração. A etapa de multiplicar o sinal pela janela é necessária quando se deseja extrair descritores do espectro do sinal, e por isto é que se deve preparar o sinal para a análise do espectro. Quando o objetivo é extrair descritores do sinal no domínio do tempo, esta etapa geralmente não é necessária.

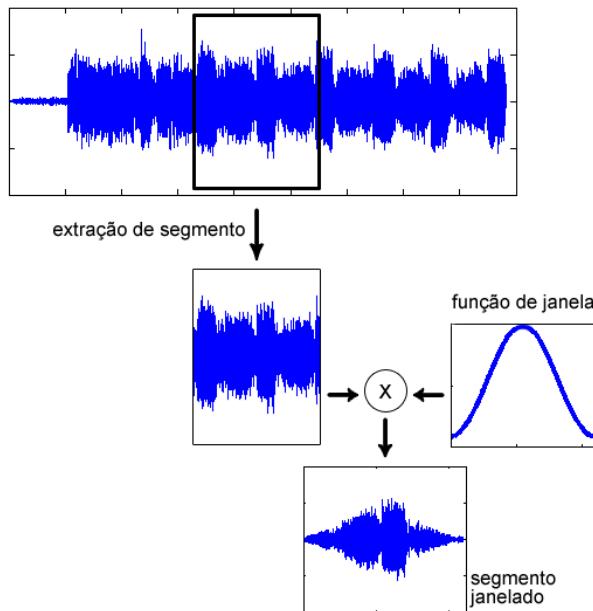


Figura 3.2: Janelamento

Aplicando a Transformada de Fourier

Outra operação comum é analisar o sinal janelado através da *transformada de Fourier de tempo reduzido* (STFT), normalmente através da transformada rápida de Fourier (FFT). Esta operação ocorre para cada janela do sinal, e fornece como saída o espectro do sinal, contendo a magnitude e as fases das componentes de frequência. Uma dificuldade é que o espectro sendo analisado não é puramente do sinal de áudio, mas sim o espectro do produto do sinal e da função de janelamento. Além disto, a FFT de um sinal muito grande – calculada ao aumentar a resolução de frequência – poderá dizer todas as frequências (ou pelo menos com uma resolução muito maior) que ocorreram em um sinal, mas não poderá dizer, aos olhos do analista ou sistema, precisamente quando as frequências ocorreram, sendo que a informação temporal está escondida na combinação matemática das amplitudes e fases do espectro. Assim, o janelamento de curta duração ajuda na visualização e análise do espectro, onde menos componentes são necessárias para representar o espectro, e pode-se saber exatamente quando os eventos de determinada frequência ocorreram.

Sobreposição de janelas

A sobreposição de janelas é importante para suavizar a distorção que ocorre devido ao janelamento, em decorrência da inserção do ruído da função da janela. Em termos de aprendizado de máquina, isto equivale a minimizar a diferença das componentes do espectro de observações

conjuntas, e, portanto, pode-se supor que observações conjuntas têm grandes probabilidades de pertencer a um mesmo modelo de classe.

3.2 Do Sinal ao Descriptor

Na seção anterior abordamos a forma como as amostras do sinal são recortadas e janeladas, permitindo inclusive a sobreposição de janelas. Desta forma, o que temos é um conjunto de $T = \lfloor \frac{s-w}{r} + 1 \rfloor$ janelas, onde s é a duração total do sinal (em amostras ou segundos), w é o tamanho da janela e r é a distância entre cada janela (em amostras ou segundos). Por exemplo, se um sinal de áudio tem $s = 5$ segundos, com janelas de $w = 100$ ms e sobreposição de 50%, então $r = 50$ milissegundos, e $T = 99$ janelas. Assim, dada a duração do sinal e o fator de ocorrência de janelas, o sistema gera T observações contendo d descritores em cada uma.

$$X = \{x_n\}, x_n \in \mathbb{R}^d, n = 1 \dots T \quad (3.2)$$

O número de descritores d depende do sistema a ser construído. Existem descritores que são univariados, como é o caso da *taxa de cruzamento de zeros* (ver A.1.1), e descritores multivariados, como é o caso, por exemplo, do MFCC (ver A.1.5), mas este não é o único aspecto a ser considerado. Como foi dito anteriormente, um descritor é necessário para descrever a informação, do ponto de vista perceptivo e sistêmico. Assim, para organizá-los foram propostas diferentes taxonomias. Existem diversas maneiras de organizar uma taxonomia, mas uma taxonomia de descritores de áudio deve ser concebida como um instrumento operacional relacionado ao áudio, deve se preocupar com questões como quais conceitos serão utilizados e como eles se relacionam com as características do som, e como estes conceitos estão definidos no sistema de recuperação musical. A figura 3.3 resume o processo de extração dos descritores propostos pelo projeto CUIDADO, e como os mesmos estão organizados. Para mais detalhes sobre taxonomias de descritores, ver apêndice A.

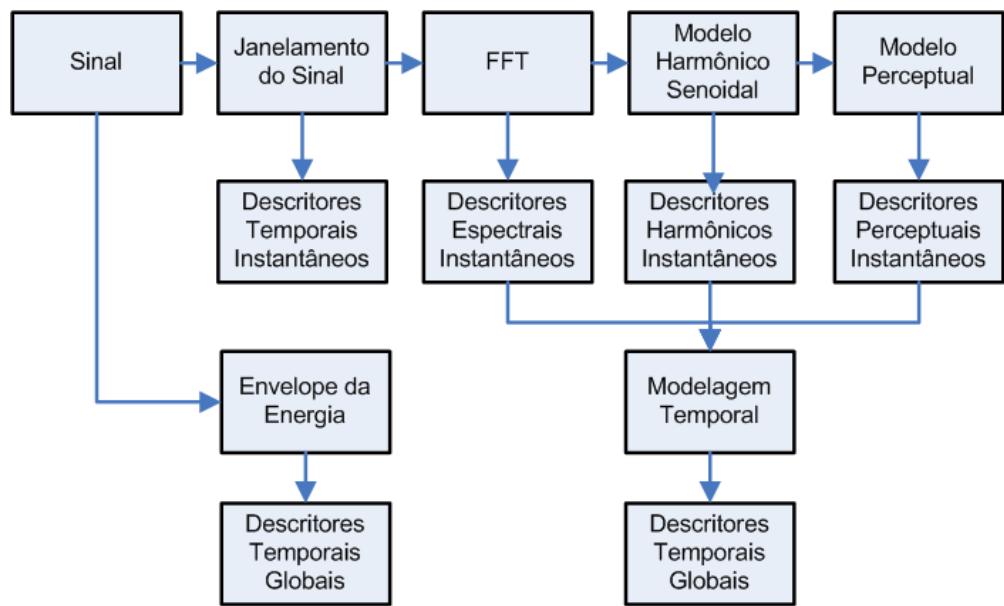


Figura 3.3: Processo de extração de descritores propostos pelo projeto CUIDADO (Peeters, 2004)

3.3 Seleção de Descritores

A seleção de descritores é uma tarefa necessária para eleger o subconjunto mais apropriado dentre um grande conjunto de descritores, de forma que os poucos descritores selecionados man-

tenham a informação de discriminação entre as classes. Uma das razões para selecionar descriptores, e reduzir o espaço dimensional, é o de eliminar descriptores que são irrelevantes, ruidosos ou que, devido à correlação mútua entre as variáveis, não forneçam uma boa informação de discriminação. Outro efeito positivo de selecionar descriptores é a redução do custo computacional de extrair, armazenar e processar os mesmos, o que é muito interessante para uma aplicação de segmentação musical em tempo real. Na maioria das vezes, é verdade que aplicações de classificação sejam consideradas mais apuradas no espaço dimensional reduzido do que no espaço original (Somol *et al.*, 2006), e isto está relacionado com a razão entre o número de amostras de treinamento N e o número de parâmetros do modelo de aprendizado de máquina d , onde os parâmetros são dados por $\Theta = \theta_1 \dots \theta_d$. Segundo Theodoridis e Koutroumbas (2008), “quanto maior esta razão, melhor são as propriedades de generalização do classificador resultante”, e o número de descriptores está diretamente relacionado ao número de parâmetros do modelo de aprendizado de máquina. No nosso caso, o sinal de áudio musical fornece uma abundância de amostras de treinamento, mas isto não nos garante que qualquer modelo de classificação utilizado terá um viés positivo ou uma variância pequena¹, no que diz respeito ao erro do classificador pois, como veremos no próximo capítulo, alguns modelos de segmentação (principalmente segmentação não-supervisionada) são construídos com poucas observações, e sendo assim, a razão N/d tem um valor muito baixo, o que aumenta a tendência de *overfitting* (quando o modelo está muito ajustado aos dados de treinamento), se o número de parâmetros d for muito grande.

Vários dos descriptores sonoros são um legado de estudos de reconhecimento de voz, processamento digital de sinais, estatística e estudos psicoacústicos. A escolha de quais descriptores de sinal utilizar deve ser específica para cada caso de classificação, segmentação ou busca, e pode ser executada de diferentes maneiras, como por exemplo, usar informação a priori sobre o sinal (se é harmônico ou ruidoso, etc.) e a partir desta informação, definir quais descriptores utilizar; ou ainda, utilizar algum método automático de pré-seleção de descriptores. De qualquer maneira, deve-se responder à seguinte pergunta: dado um problema (busca, classificação, segmentação, etc.) e um contexto (transmissão radiofônica, música popular, música contemporânea, sinal ruidoso, etc.), qual é o subconjunto ótimo de descriptores para o problema em questão? Alguns descriptores podem ser irrelevantes para uma determinada classe e arruinar o resultado da aplicação; por exemplo, Peeters e Rodet (2002a) sugerem que o desritor de inharmonicidade é inútil para discriminar entre sons puramente harmônicos. É interessante notar que no nosso caso ocorre um fenômeno parecido, pois como estamos interessados em segmentar músicas, no sentido mais amplo possível, é pouco frequente encontrar sons puramente harmônicos, a menos em passagens de um instrumento solo, e portanto utilizar descriptores que consideram a parte harmônica do sinal pode dificultar o processo de segmentação, uma vez que estaríamos forçando uma condição harmônica em um sinal complexo, composto por misturas de vários instrumentos.

Diversos autores optaram por uma seleção manual de descriptores: Velivelli *et al.* (2003) selecionam os coeficientes *Mel-frequency Cepstral Coefficients* (MFCC) e a energia do sinal para segmentar sons utilizando HMM justapostas, enquanto Bergstra *et al.* (2006) selecionam os descriptores *Fast Fourier Transform coefficients* (FFTCs), *Real Cepstral Coefficients* (RCEPS), MFCC, *Zero-crossing Rate* ZCR, *Spectral spread*, *Spectral centroid*, *Spectral rolloff* e *Linear Predictive Coefficients* (LPC). Entretanto, não é de nosso conhecimento que exista um estudo que diga qual é o melhor subconjunto de descriptores para uma tarefa de segmentação baseada no timbre musical. Em nossos estudos foi comum encontrar uma preferência entre os autores para o desritor MFCC (Aucouturier e Pachet, 2004; Aucouturier *et al.*, 2005), e faz parte de nossos objetivos avaliar a qualidade do desritor MFCC para a segmentação, comparando-o com outros conjuntos de descriptores, selecionados por algum método automático.

Algumas técnicas foram propostas para resolver este problema, como Informação Mútua, Análise de Componentes Principais, Análise de Discriminantes Lineares, Algoritmos Genéticos e Redes Neurais. Peeters e Rodet (2002a) puderam avaliar a pré-seleção de descriptores com dois

¹O erro de um modelo de aprendizado de máquina é composto do viés do modelo e da variância da estimativa. Para mais informação, ver Theodoridis e Koutroumbas (2008), pp.90-91

métodos: Análise de Discriminantes Lineares (*LDA*) e Informação Mútua (*MI*), preferidos por um critério de disponibilidade em tempo real. Os descritores foram selecionados dentre todos os definidos no projeto CUIDADO, obtendo melhores resultados com *MI* do que com *LDA*, que se resume em maiores taxas de acerto e uma redução de descritores maior: 20 descritores com *MI* contra 27 com *LDA*. Outros autores como Agostini *et al.* (2003) utilizaram *LDA*, reduzindo para 8 de um conjunto total de 18 descritores, em uma pesquisa para classificação de instrumentos musicais. São eles: média de inarmonicidade (A.1.4), média e desvio padrão do centroide do espectro (A.1.3), média da energia harmônica (A.1.3), zero-crossing rate (A.1.1), média e desvio padrão da largura de banda e desvio padrão da assimetria harmônica (A.1.4). Para demonstrar a importância da seleção de descritores, Herrera-Boyer *et al.* (2003) fizeram um estudo mostrando vários resultados de redução de dimensão no contexto de classificação de instrumentos musicais, reforçando que alguns descritores podem ter comportamentos não-uniformes, ou seja, alguns descritores são melhores para classificar somente alguns instrumentos ou famílias de instrumentos, mas não outros, e que não somente os descritores temporais e espectrais instantâneos devem ser considerados, mas também a evolução temporal deles.

3.3.1 Pré-processamento

Existem ainda alguns aspectos na seleção de descritores que devem ser considerados. São aspectos mais relacionados ao pré-processamento do que à seleção de características, mas que estão ligados pela qualidade do dado de treinamento fornecido para a solução de seleção de descritores. São eles: a remoção de pontos fora da curva (*outlier removal*), normalização dos dados e o tratamento de dados faltantes (*missing data*). O que vamos ver nesta subseção é uma breve descrição destes métodos de “limpeza” dos dados e as justificativas da utilização ou não dos mesmos. Vale lembrar que esta etapa de pré-processamento não é a mesma que precede a extração dos descritores, quando o pré-processamento se aplica no próprio sinal de áudio.

Tratamento de pontos fora da curva

Os pontos fora da curva são observações que estão a uma distância muito grande de outros valores em uma amostra aleatória de uma população, e o tratamento deles é importante para a remoção de ruídos nos dados, ajudando a melhorar a qualidade dos modelos. Veja na figura 3.4 o gráfico Boxplot para dois descritores de trechos musicais do grupo 12 (B.5). Nesta imagem, a linha central é a mediana, e os limites da caixa são os 25º e 75º percentis, e os pontos fora da curva são exibidos individualmente.

Normalização dos dados

A normalização dos dados se torna necessária quando diferentes descritores têm intervalo de valores muito diferentes, pois valores altos podem ter mais influência na função de custo (a que determina a importância relativa de cada descriptor), mas isso não significa necessariamente que estes sejam os descritores com maior poder discriminante (Theodoridis e Koutroumbas, 2008). A normalização dos dados é um aspecto do pré-processamento que não vamos considerar nos dados de treinamento, pois se buscamos uma segmentação em tempo real, não poderíamos saber de antemão quais seriam os parâmetros para normalizar um determinado descriptor na fase de segmentação. Ademais, não temos conhecimento de quais seriam as consequências de realizar uma seleção de descritores com seus valores normalizados, e posteriormente utilizar descritores com valores brutos, não-normalizados.

Tratamento de dados faltantes

O tratamento de dados faltantes no nosso contexto não é algo que precisamos nos preocupar, pois como existe uma abundância de dados, se houver um descriptor que não tenha valor, podemos

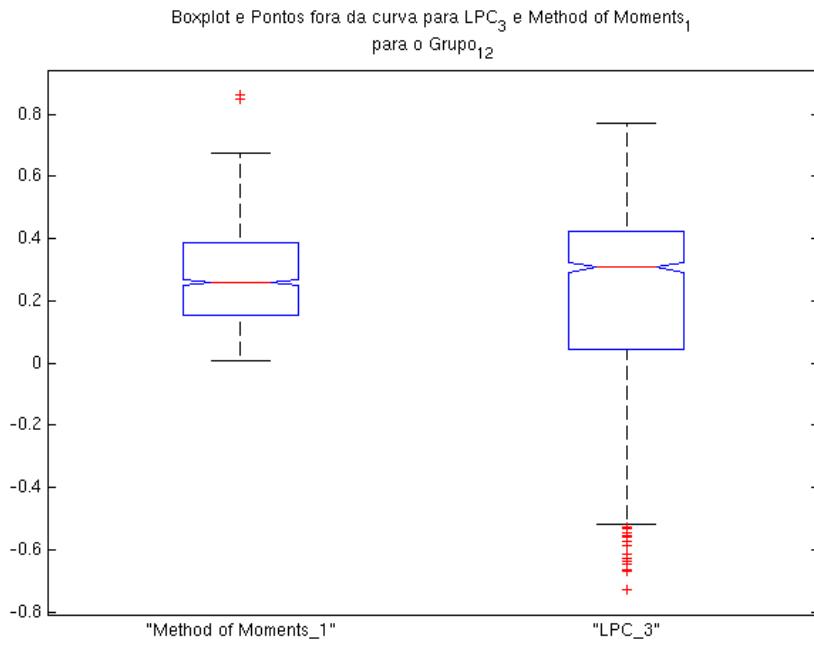


Figura 3.4: Boxplot para visualização de pontos fora da curva para os descritores LPC_3 e $Method\ of\ Moments_1$ para observações do Grupo 12

descartar a observação. Em nossos experimentos, percebemos que dificilmente um descritor tem um valor faltante, fato este relacionado ao tipo de sinal e à qualidade dos algoritmos de extração de características.

3.3.2 Seleção Automática de Descritores

Geralmente um algoritmo de seleção automática de descritores é composto por quatro etapas: análise independente de cada descritor, geração do subconjunto, critério de avaliação do subconjunto e um critério de parada. Na primeira etapa, o objetivo é identificar de antemão os descritores que não fornecem uma boa discriminação entre as classes para o problema em questão. Isto é realizado de forma independente, sem considerar a correlação entre outras variáveis aleatórias, como observado por Theodoridis e Koutroumbas (2008):

A first step in feature selection is to look at each of the generated features *independently* and test their discriminatory capability for the problem at hand. Although looking at the features independently is far from optimal, this procedure helps us to discard easily recognizable “bad” choices and keeps the more elaborate techniques (...) from unnecessary computational burden.

e mais adiante,

(...) However, such methods neglect to take into account the correlation that unavoidably exists among the various features and influences the classification capabilities of the feature vectors that are formed.

Análise independente

Theodoridis e Koutroumbas (2008) propõem dois métodos para analisar independentemente as variáveis: um baseado em teste de hipóteses, que basicamente considera as diferenças das médias amostrais de uma variável aleatória para diferentes classes, verificando se as diferenças absolutas são significativamente maiores que zero; e outro baseado em curvas ROC (*Receiver Operating*

Characteristics), que avalia a sobreposição das funções de densidade de probabilidade de duas classes distintas. O último método é mais robusto, pois no primeiro, mesmo que as médias amostrais sejam bastante diferentes, a variância pode ser grande o suficiente de tal forma que as classes estejam na realidade sobrepostas. Na figura 3.5, vemos um exemplo de uma função de densidade da primeira componente do descriptor LPC para duas classes de sons, e a figura 3.6 representa a estimativa da curva ROC para as densidades da figura 3.5. Veja subseção 3.3.3 para mais detalhes sobre curvas ROC.

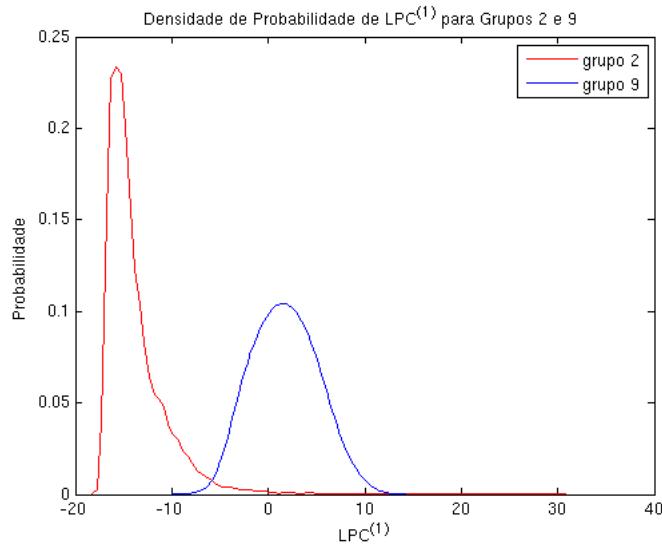


Figura 3.5: Função de Densidade da primeira componente do descriptor LPC para as classes de sons agrupados (Grupo 2 e Grupo 9).

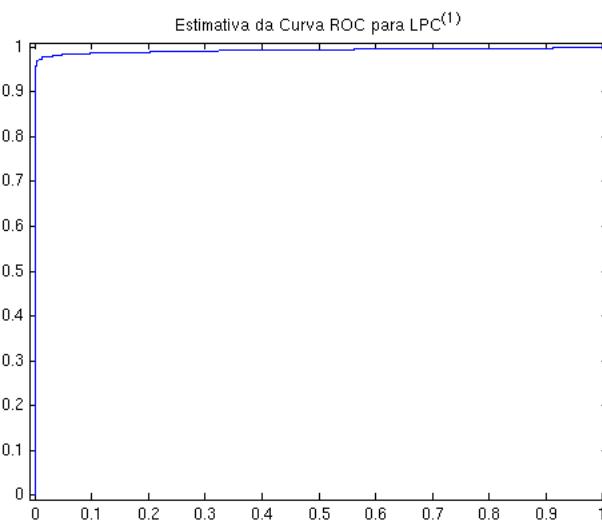


Figura 3.6: Estimativa da curva ROC da primeira componente do descriptor LPC para as classes dos Grupo 1 e 9.

Subconjunto de descritores

Liu e Yu (2005) (referenciando Blum e Rivest (1993)) observam que encontrar o subconjunto ótimo de descritores é um problema NP-Difícil, e para um conjunto de N descritores, onde existem 2^N subconjuntos candidatos, a busca exaustiva é inviável. Portanto, para resolver a segunda etapa do algoritmo algumas estratégias de busca foram propostas: (1) busca completa, que elimina alguns ramos e que garante uma seleção de um subconjunto ótima, dependendo do critério

de avaliação, (2) busca sequencial, que não garantem a seleção de um subconjunto ótimo e utilizam algoritmos gulosos para atingir um subconjunto. Estes algoritmos são simples de implementar e produzem um resultado na ordem do espaço de busca, que é $O(N^2)$ ou menos, e (3) busca aleatória, que inicia com um subconjunto aleatório e pode explorar o espaço de descritores tanto através de uma busca sequencial, que adiciona aleatoriamente descritores ao subconjunto, quanto através da escolha de outro subconjunto de forma aleatória.

Os algoritmos para o critério de avaliação podem ser categorizados em *Filtro*, *Encapsulado* (do inglês *Wrapper*) e *Híbrido*. Na abordagem por *Filtro*, o algoritmo se apoia nas características dos dados de treinamento para encontrar os descritores, independentemente do modelo de aprendizado de máquina a ser utilizado em sequência e, portanto, é um método computacionalmente mais eficiente. Já nos *Encapsulados*, estes precisam de algum método de aprendizado de máquina, e utilizam a informação gerada, como o desempenho em porcentagem de erro, para avaliar o subconjunto de descritores. É um método computacionalmente menos eficiente, mas que permite uma melhor avaliação do subconjunto que o método *Filtro*, por exemplo, e os métodos *Híbridos* foram criados para combinar a eficiência e o desempenho dos dois métodos.

Em nossos estudos, optamos por uma estratégia de geração de subconjuntos pela busca sequencial ordenada (subconjunto inicial $S_0 = \{\}$), pois esta tem a característica de ter um espaço de busca menor que a busca completa, e para o critério de avaliação, optamos por utilizar a estratégia de *Filtro*. É válido lembrar que buscamos opções para uma segmentação em tempo real, e que o procedimento automatizado de escolher descritores deve ser executado sempre que uma nova informação musical² se apresenta, uma vez que a escolha do subconjunto de descritores depende estritamente dos dados de treinamento, o que torna a justificativa do custo computacional relevante.

Outra questão surge quando temos diferentes técnicas para o critério de avaliação do subconjunto. Usualmente, as técnicas disponíveis procuram realizar uma análise das variâncias, procurando encontrar o subconjunto que maximiza as dispersões entre as classes e minimiza a dispersão dentro das classes, o que fornece um grau de discriminação maior. Entretanto, não é de nosso conhecimento que exista um método ideal que resolva este problema para observações de trechos musicais. O que vamos apresentar em seguida são os métodos que encontramos para o critério de seleção de subconjunto, e as estratégias para selecionar um único subconjunto dadas as diferentes técnicas.

Na figura 3.7 temos uma visão geral do procedimento adotado. Depois da etapa de extração de características, utilizando um conjunto de descritores S , temos os dados de treinamento $X = X_1 \dots X_N \in \mathbb{R}^d$. O conjunto de descritores iniciais tem algum de seus eixos eliminados na primeira etapa da seleção (indicada pela análise independente), resultando uma redução na dimensão de X , o que é representado pelo subconjunto S' d' -dimensional, $d' < d$. Em seguida, utilizamos diferentes critérios para *ordenar* o conjunto S' em ordem decrescente de poder de discriminação - e não diminuir o número de elementos do subconjunto. Utilizamos seis critérios para esta etapa: LDA, PCA, IRMFSP (Peeters, 2003a), J_1 , J_2 e J_3 (Theodoridis e Koutroumbas, 2008), e que forneceram seis ordenações diferentes do subconjunto S' . O último passo deste procedimento é decidir qual das ordenações, ou qual combinação das ordenações, fornece, com o menor número de descritores, a melhor discriminação entre as classes, que é o que veremos na seção 3.3.8. Como resultado desta última etapa, temos o subconjunto S'' , que determina quais são os descritores que serão utilizados na etapa de treinamento e construção dos modelos de classificação e segmentação. O pseudocódigo da etapa de seleção do subconjunto pode ser visto no algoritmo 1. Este algoritmo executa dois outros métodos, *ANALISE_INDEPENDENTE*(X), que analisa independentemente cada descritor, e *AVALIA*(S , X), que avalia a aderência do subconjunto S em representar os dados de treinamento, e que devemos comentar nas seções 3.3.3 e 3.3.4, respectivamente.

Em seguida vamos apresentar o método de análise independente por curvas ROC e seis métodos para avaliar o subconjunto de descritores dadas as observações de treinamento: LDA, PCA,

²Informação musical, neste contexto, se refere a qualquer conjunto de sons musicais polifônicos que alguém gostaria de identificar.

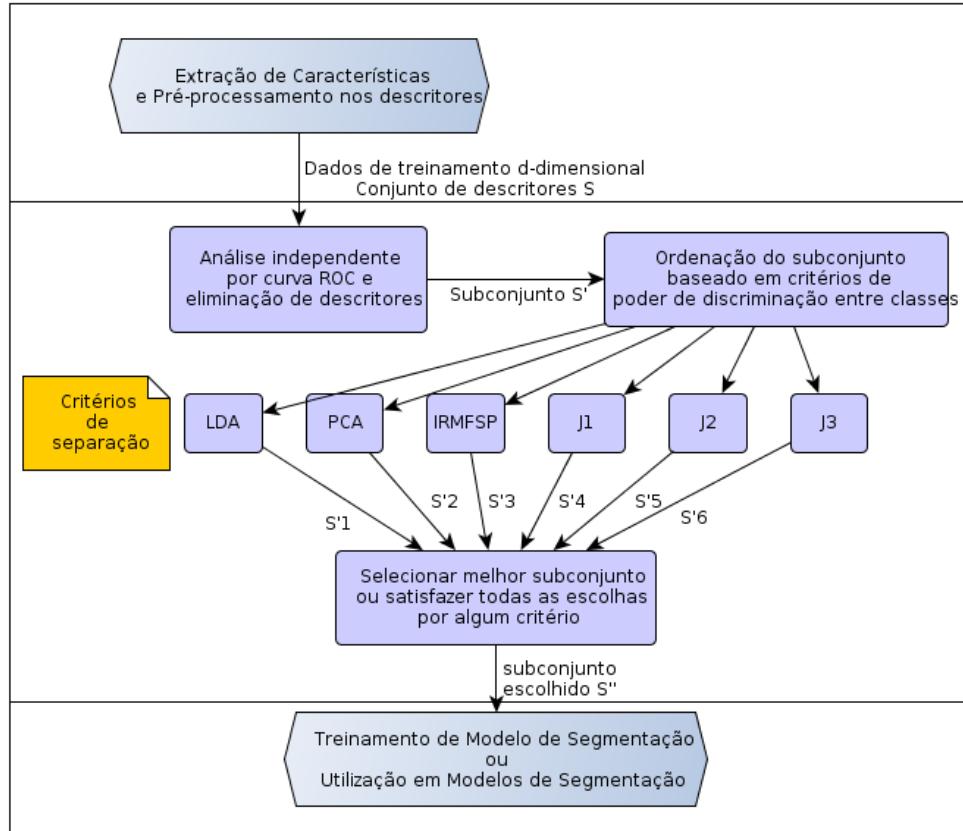


Figura 3.7: Visão geral do procedimento de seleção de descritores.

Algoritmo 1 Seleciona Descritores

FILTRO_SEQUENCIAL(X)

Entradas:

$X = X_1^d \dots X_n^d$; dados de treinamento d -dimensional

Saída: S ; subconjunto, contendo os índices das dimensões selecionadas

X' ; dados de treinamento filtrados d' -dimensional, $d' < d$

1: $X' = \text{ANALISE_INDEPENDENTE}(X)$; analisa e remove independentemente cada eixo

2: $d' = \text{LENGTH}(X')$; d' recebe o número de dimensões das observações X'_i

3: $S \leftarrow \{\}$; inicializa subconjunto

4: $R \leftarrow \{1, \dots, d'\}$; conjunto com as d' dimensões de X' , que encolhe ao longo das iterações

5: **repete**

6: **para** $i \in R$ **faz**

7: $\gamma_i \leftarrow \text{AVALIA}(S \cup R(i), X')$; avalia o subconjunto

8: **fim para**

9: $j \leftarrow \arg \max \gamma$

10: $S \leftarrow S \cup R(j)$

11: $R \leftarrow R \setminus R(j)$; remove o elemento da posição j

12: **enquanto** $R \neq 0$

IRMFSP, J_1 , J_2 e J_3 .

3.3.3 Análise independente com curvas ROC

A análise independente dos descritores com curva ROC é realizada considerando somente um dos eixos do espaço das características por vez, juntamente com as sobreposições das densidades

de probabilidade de cada classe w . No caso mais simples, em uma classificação binária - onde existem somente duas classes w_1 e w_2 e pretendemos avaliar o desempenho de classificar uma das classes -, esta técnica avalia a sobreposição das duas densidades de probabilidade para estimar o desempenho da classificação, que é expressa pela área sob a curva ROC. A curva ROC é um gráfico da probabilidade de verdadeiros-positivos (VP) em função da probabilidade de falsos-positivos (FP), determinados por um limiar. A figura 3.8 exemplifica as áreas dado o valor real e o valor previsto para uma determinada classe. Movendo este limiar sobre todos os pontos possíveis, podemos obter diferentes valores para as probabilidades $Pr(VP)$ e $Pr(FP)$, onde

$$Pr(VP|limiar) = \frac{VP}{VP + FN} \quad (3.3)$$

e

$$Pr(FP|limiar) = \frac{FP}{FP + VN}. \quad (3.4)$$

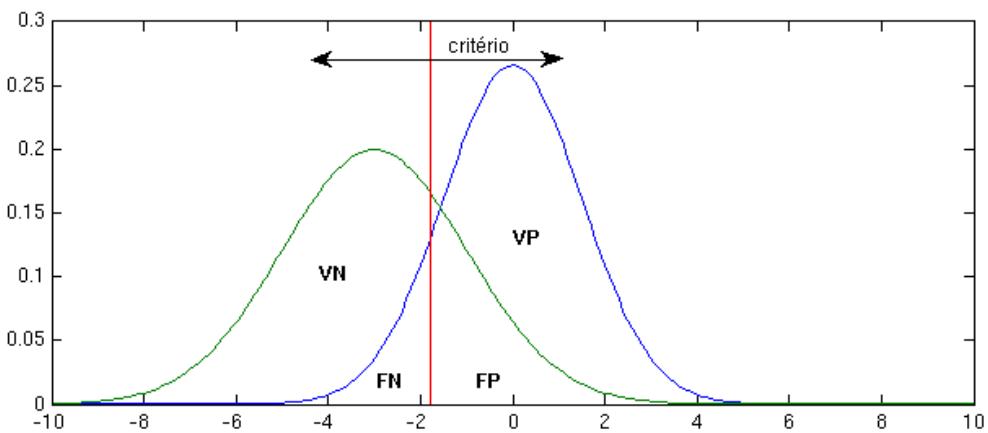


Figura 3.8: Duas funções de densidade de probabilidade, um limiar e as áreas falso-positivo (FP), falso-negativo (FN), verdadeiro-positivo (VP) e verdadeiro-negativo (VN).

Podemos perceber que se as densidades de probabilidade se sobrepõem totalmente, então a curva ROC é uma linha reta na diagonal do gráfico, e a área sob a curva ROC é igual a 0, se considerarmos a área acima da diagonal principal. No caso oposto, quando não existe nenhuma sobreposição, a área sob a curva ROC é igual a 1/2, que exemplifica o classificador ideal. Desta forma, o valor da área sob a curva varia entre 0 (densidades totalmente sobrepostas, ver figura 3.9 (d)) e 1/2 (sem nenhuma sobreposição), isto é uma “medida de capacidade de discriminação de classes de uma característica específica” (Theodoridis e Koutroumbas, 2008). Veja na figura 3.9 alguns exemplos de pares de densidades e suas respectivas curvas ROC.

No caso em que estamos avaliando uma característica f onde existem K classes w_1^f, \dots, w_K^f , podemos calcular a média da área sob a curva de f para todas as combinações de sobreposições dos pares de densidade de probabilidade das classes. Apesar de ser somente uma estimativa, a média da área sobre a curva ROC pode ajudar a determinar um conjunto de descritores, que na média, tem a maior área, e que por fim pode indicar os melhores descritores o problema. Assim, formalmente, a área para um descritor f fica definida da seguinte forma:

$$AUC_f = \frac{K!}{2(K-2)!} \sum_{i=1}^{K-1} \sum_{j=i+1}^K AUC_PDF(w_i^f, w_j^f), \quad (3.5)$$

onde $AUC_PDF(w_i^f, w_j^f)$ é uma função que calcula a área sob a curva ROC para os pares de densidades de probabilidade.

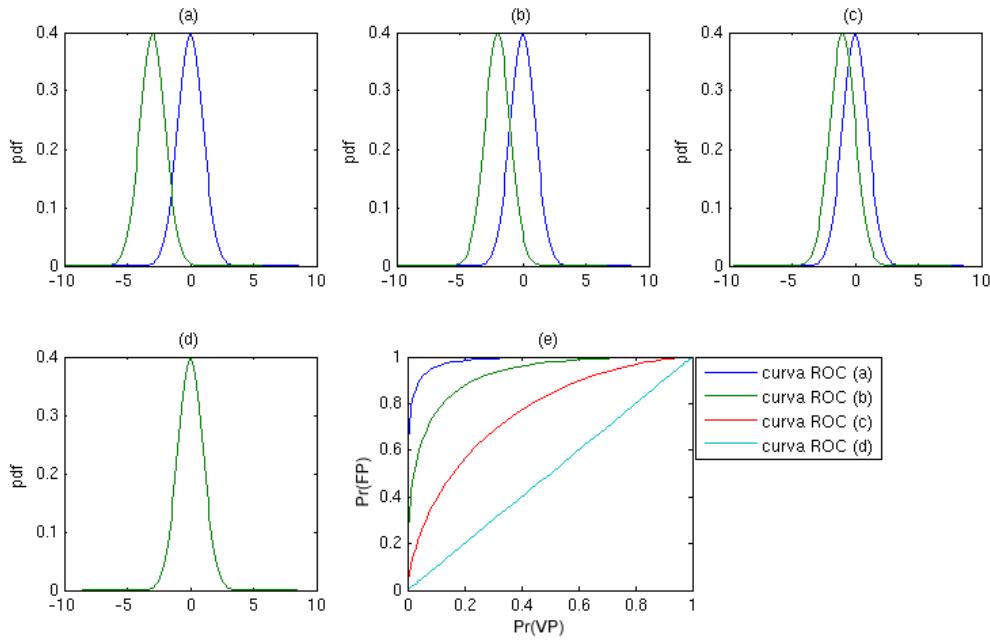


Figura 3.9: Exemplos (a)(b)(c)(d) de sobreposições de distribuições de probabilidades e (e) as curvas ROC para os pares de distribuições.

Estratégias de corte

Depois de ter calculado AUC_f para $f = 1 \dots d$, onde d é o número total de descritores extraídos do sinal, temos que determinar um limiar que elimine aqueles descritores cujas áreas sob a curva ROC sejam pequenas. Como “pequena” é um conceito subjetivo, não nos parece viável determinar um valor qualquer para este limiar. Em uma estratégia simples pode-se recuperar os $\alpha\%$ dos descritores cujas áreas são as maiores. O problema com esta estratégia é que o conjunto selecionado não contém necessariamente descritores com boas capacidades de discriminação entre as classes. A seguir vamos apresentar três novas estratégias para determinar automaticamente os limiares para a eliminação dos descritores que pouco discriminam as classes.

Limiar adaptativo usando a média dos maiores. Nesta estratégia, percorremos os valores em ordem crescente e determinamos um valor de corte quando o percentual α da média dos maiores for menor que o elemento corrente. Veja o algoritmo 2.

Algoritmo 2 Encontra valor de corte pela porcentagem da média dos maiores
LIMIAR_ROC1(AUC, α)

Entradas:

$AUC \leftarrow AUC_1 \dots AUC_d$; d áreas sob a curva ROC
 α ; porcentagem de corte

Saída: v ; valor de corte

- 1: $S \leftarrow \text{ORDENA}(AUC)$; ordena em ordem crescente o vetor AUC
 - 2: $i \leftarrow d - 1$
 - 3: **enquanto** $S(i) > \alpha * \text{MÉDIA}\{S(i + 1), \dots, S(d)\}$ **faça**
 - 4: $i \leftarrow i - 1$
 - 5: **fim enquanto**
 - 6: $v \leftarrow S(i)$; aqui i aponta para o primeiro valor descartado
-

Maximizar a diferença das médias. Nesta estratégia, o objetivo é encontrar o valor de limiar que maximiza a diferença das médias dos valores de AUC acima e abaixo do limiar, em um dado intervalo $[\alpha, \beta]$ de AUC. Veja o algoritmo 3.

Algoritmo 3 Encontra valor de corte que maximiza a diferença das médias

LIMIAR_ROC2(AUC, α , β)

Entradas:

AUC \leftarrow AUC₁ ... AUC_d; d áreas sob a curva ROC

α ; valor mínimo

β ; valor máximo

Saída: v ; valor de corte

```

1:  $S \leftarrow$  ORDENA(AUC); ordena em ordem crescente o vetor AUC
2:  $i \leftarrow 1$ 
3: enquanto  $S(i) < \alpha$  faça
4:    $i \leftarrow i + 1$ 
5: fim enquanto
6:  $\text{maxdiff} \leftarrow$  MÉDIA{ $S(i+1), \dots, S(d)$ } – MÉDIA{ $S(1), \dots, S(i)$ }
7:  $k \leftarrow i$ 
8: enquanto  $i < d$  e  $S(i) < \beta$  faça
9:   aux  $\leftarrow$  MÉDIA{ $S(i+1), \dots, S(d)$ } – MÉDIA{ $S(1), \dots, S(i)$ }
10:  se aux > maxdiff então
11:    maxdiff  $\leftarrow$  aux
12:     $k \leftarrow i$ 
13:  fim se
14:   $i \leftarrow i + 1$ 
15: fim enquanto
16:  $v \leftarrow S(k)$ ; aqui  $k$  aponta para o primeiro valor descartado

```

Maximizar o vazio. Nesta estratégia, o objetivo é encontrar o valor de limiar que maximiza o vazio entre os valores de AUC acima e abaixo do limiar, em um dado intervalo $[\alpha, \beta]$ de AUC. Veja o algoritmo 4.

A figura 3.10 demonstra os limiares encontrados a partir da execução destes algoritmos para um conjunto de 65 descritores e 5 segmentos (que determinam o número de classes). Os parâmetros utilizados na execução deste exemplo são os seguintes: para o algoritmo 2, $\alpha = .9$, no algoritmo 3 $\alpha = .001$ e $\beta = 0.07$, e no algoritmo 4 $\alpha = .001$ e $\beta = 0.05$.

Voltando ao algoritmo 1, podemos discutir agora os métodos de avaliação dos descritores considerando a correlação entre os diferentes eixos – computado pela função AVALIA(S, X). Os métodos descritos abaixo fornecem um critério para determinar quanto um eixo de descritor é melhor que o outro em termos de separabilidade entre as classes, e assim, devolvem um vetor numérico para a continuação do procedimento.

3.3.4 Seleção com Critérios de Separabilidade entre Multi-Classes

Os critérios de separabilidade abaixo são construídos de acordo com a forma como os vetores de observação estão espalhados no espaço d -dimensional, e não precisam supor necessariamente que os dados têm uma distribuição normal (neste sentido, estes critérios deveriam ter um peso menor na avaliação final, pois a maioria dos métodos de segmentação utilizados - assunto que veremos a seguir - assumem normalidade dos dados). Entretanto, não gostaríamos de descartar os resultados destes métodos, e sim levá-los em consideração no conjunto de descritores final. Para a construção destes critérios, é necessário definir algumas matrizes:

Algoritmo 4 Encontra valor de corte que maximiza o vânio
LIMIAR_ROC3(AUC, α , β)

Entradas:

AUC \leftarrow AUC₁ ... AUC_d; d áreas sob a curva ROC

α ; valor mínimo

β ; valor máximo

Saída: v ; valor de corte

- 1: $S \leftarrow$ ORDENA(AUC); ordena em ordem crescente o vetor AUC
 - 2: $i \leftarrow 1$
 - 3: **enquanto** $S(i) < \alpha$ **faça**
 - 4: $i \leftarrow i + 1$
 - 5: **fim enquanto**
 - 6: $inicio \leftarrow i$
 - 7: **enquanto** $i < d$ e $S(i) < \beta$ **faça**
 - 8: $D(i) \leftarrow \min \{S(i+1) \dots S(d)\} - \max \{S(1) \dots S(i)\}$
 - 9: $i \leftarrow i + 1$
 - 10: **fim enquanto**
 - 11: $fim \leftarrow i - 1$
 - 12: $k \leftarrow \text{argmax}\{D(inicio), \dots, D(fim)\}$
 - 13: $v \leftarrow S(k)$; aqui k aponta para o primeiro valor descartado
-

- Matriz de dispersão dentro das classes³,

$$S_w = \sum_{i=1}^M P_i \Sigma_i \quad (3.6)$$

onde Σ_i é a matriz de covariância para a classe (do inglês *Within-class scatter matrix*) ω_i

$$\Sigma_i = E[(x - \mu_i)(x - \mu_i)^T] \quad (3.7)$$

e P_i é a probabilidade *a priori*, ou seja $P_i = n_i/N$, onde n_i é o número de observações da classe ω_i sobre o total de observações N .

- Matriz de dispersão entre classes (do inglês *Between-class scatter matrix*),

$$S_b = \sum_{i=1}^M P_i (\mu_i - \mu_0)(\mu_i - \mu_0)^T \quad (3.8)$$

onde μ_0 é o vetor de médias globais

$$\mu_0 = \sum_{i=1}^M P_i \mu_i. \quad (3.9)$$

- Matriz de dispersão total (do inglês *Total scatter matrix*), que é a matriz de covariância das observações dada a média global,

$$S_t = E[(x - \mu_0)(x - \mu_0)^T]. \quad (3.10)$$

Destas definições derivam os critérios J₁, J₂ e J₃ (Theodoridis e Koutroumbas, 2008):

³Utilizamos o termo classe neste contexto, pois na literatura encontramos este termo mais frequentemente. Entretanto, podemos encarar a classe ou categoria como sendo um segmento musical delimitado.

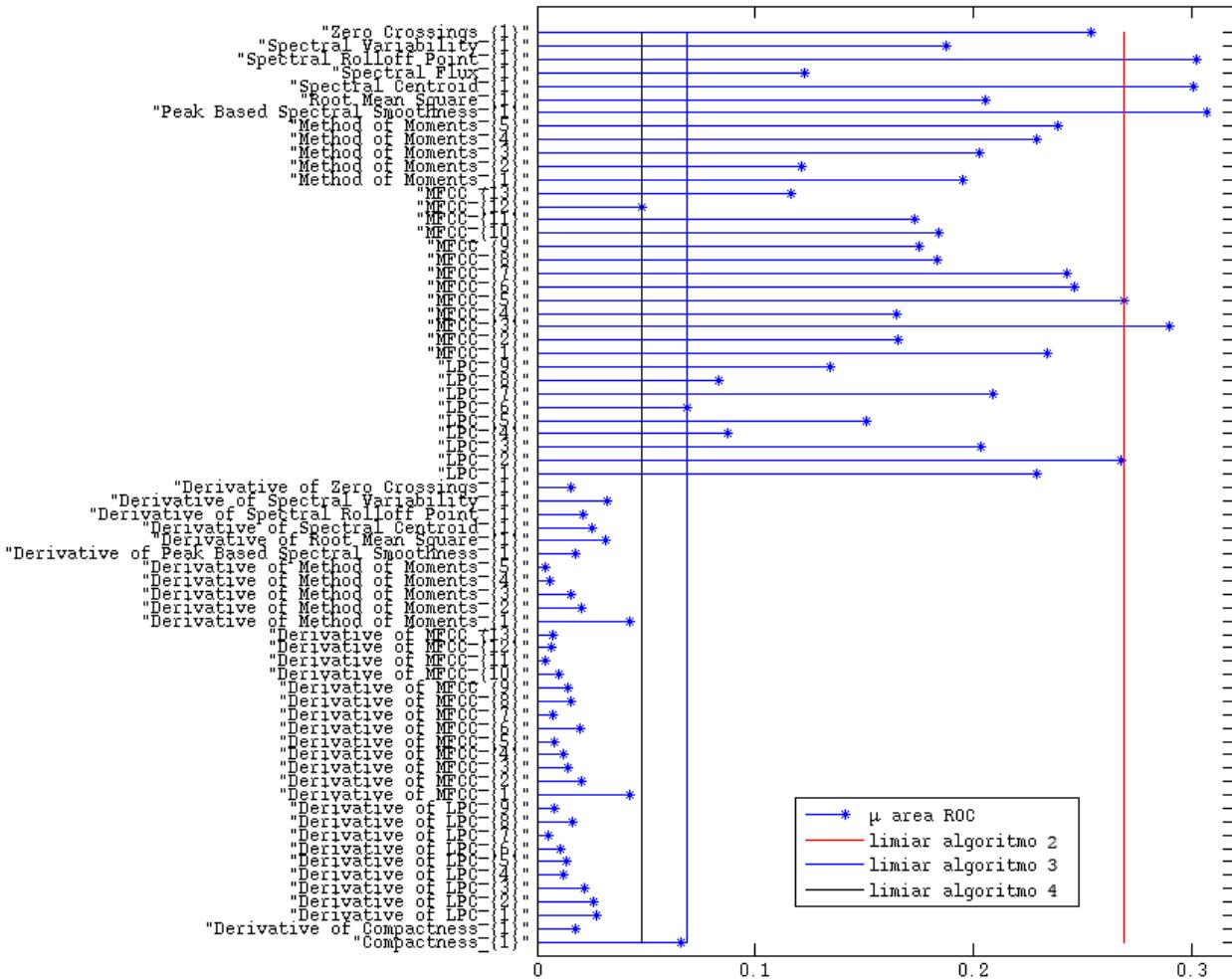


Figura 3.10: Média das áreas sob a curva ROC de 65 descritores e 5 segmentos e os limiares para corte dados pelos algoritmos 2, 3 e 4. Para o algoritmo 2, $\alpha = .9$, no algoritmo 3 $\alpha = .001$ e $\beta = 0.07$, e no algoritmo 4 $\alpha = .001$ e $\beta = 0.05$.

Critério J₁

O critério J₁ tem valores altos quando as observações no espaço d -dimensional estão bem agrupadas em torno de suas médias, dentro de cada classe, e os grupos de cada classe estão bem separados.

$$J_1 = \frac{\text{traço}(S_t)}{\text{traço}(S_w)} \quad (3.11)$$

Critério J₂

A substituição do traço pelo determinante das matrizes é uma variante do critério J₁, e é justificado para matrizes que são simétricas positivas definidas, e, portanto, com autovalores positivos. Assim, valores altos para J₁ também correspondem a valores altos para J₂.

$$J_2 = \frac{|S_t|}{|S_w|} = |S_w^{-1} S_t| \quad (3.12)$$

Critério J_3

Uma variante de J_2 , bastante encontrada na prática (Theodoridis e Koutroumbas, 2008), é o critério J_3 :

$$J_3 = \text{traço}(S_w^{-1} S_t) \quad (3.13)$$

Com estes três critérios, a integração com o algoritmo de seleção de descritores sequencial (algoritmo 1) é imediata, uma vez que o argumento máximo devolve o eixo que permite a maior discriminização entre as classes, e que contém as amostras mais próximas de sua média.

3.3.5 Seleção com LDA

O objetivo da técnica LDA (*Linear Discriminant Analysis*) é reduzir o espaço dimensional preservando o máximo de informação discriminante entre as classes. Basicamente se deseja projetar um conjunto de amostras d dimensionais pertencentes às classes $\omega_1 \dots \omega_n$ em um subespaço de dimensão $l < d$, onde os itens de uma mesma classe ω_i fiquem mais próximos uns dos outros e o mais separados o possível de outras classes (Duda *et al.*, 2001). O método é originalmente baseado no trabalho de Fisher em discriminantes lineares. Para uma apresentação mais didática sobre o assunto, ver Theodoridis e Koutroumbas (2008). O que vamos apresentar aqui são os conceitos básicos para o entendimento do método na seleção de descritores.

Definição Seja x o vetor d -dimensional de observações gerados por K classes, a tarefa é transformá-lo em um vetor l -dimensional y , $l < d$, tal que um critério de separabilidade entre as classes seja utilizado,

$$y = A^T x \quad (3.14)$$

onde A^T é uma matriz $l \times d$. Os critérios de separabilidade da subseção 3.3.4 podem ser utilizados para a otimização da redução dimensional, como, por exemplo, o critério J_3 . Seja S_{xw}, S_{xb} (ver equações 3.6 e 3.8) as matrizes de dispersão de x , então as matrizes S_{yw}, S_{yb} correspondentes de y são:

$$S_{yw} = A^T S_{xw} A, \quad S_{yb} = A^T S_{xb} A. \quad (3.15)$$

Portanto, o critério J_3 no espaço de y é dado por

$$\text{traço}[(A^T S_{xw} A)^{-1} (A^T S_{xb} A)]. \quad (3.16)$$

Para se obter o tipo de separabilidade entre classes pretendido pela técnica LDA, a matriz A é definida pelos autovetores de $S_{xw}^{-1} S_{xb}$ associados aos l maiores autovalores desta última matriz. Entretanto, a matriz $S_{xw}^{-1} S_{xb}$ é uma matriz $d \times d$, e a questão principal, para sistemas de redução de dimensão, é quais l autovalores devem ser escolhidos.

Utilização Relembrando que o que buscamos são os eixos originais que melhor discriminam as classes e não novos eixos gerados a partir de uma combinação linear, a solução de qual eixo original f melhor explica o autovetor \vec{e} de máximo autovalor é direta através da projeção de \vec{e} em todos os eixos originais \vec{f}_i , $i = 1 \dots d$, onde calcula-se a discrepância M_i da projeção em relação a \vec{e} :

$$M_i = \left| \vec{e} - \frac{\vec{e} \cdot \vec{f}_i}{\|\vec{f}_i\|^2} \vec{f}_i \right|. \quad (3.17)$$

A projeção escalar de \vec{e} em \vec{f}_i é a magnitude da projeção do vetor \vec{e} em \vec{f}_i , e devemos escolher o eixo i que fornece a menor magnitude M_i , pois este é o que mais se aproxima do eixo gerado.

Para integrar este método com o algoritmo de seleção de descritores sequencial (algoritmo 1) a função deve devolver $1/M$, para que o argumento que maximiza a função seja aquele que tem a menor discrepância da projeção de \vec{e} em \vec{f} .

3.3.6 Seleção com PCA

Na subseção anterior, uma transformação linear (*LDA*) foi aplicada sobre as observações com o objetivo de reduzir a dimensionalidade (ver equação 3.14). Naquele contexto, os rótulos das classes (ou segmentos) eram conhecidos, e essa informação pôde ser utilizada para otimizar a escolha da matriz de transformação. PCA (*Principal Component Analysis*), também busca uma transformação ortogonal, que transforma um conjunto de variáveis em um conjunto de novas variáveis não correlacionadas entre si, com a tarefa de redução de dimensionalidade, mas de uma perspectiva diferente, não-supervisionada.

Definição⁴ Se x é uma variável aleatória vetorial com média amostral μ e covariância Σ , então a transformação de componentes principais é a transformação

$$x \rightarrow y = \Gamma^T(x - \mu), \quad (3.18)$$

onde Γ é uma matriz ortogonal, $\Gamma^T\Sigma\Gamma = \Lambda$ é uma matriz diagonal, e $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d \geq 0$. Os autovalores λ_i são garantidamente positivos se a matriz Σ é positiva definida. A i -ésima componente principal de x pode ser definida como sendo o i -ésimo elemento do vetor y , ou seja,

$$y_i = \gamma_{(i)}^T(x - \mu), \quad (3.19)$$

onde $\gamma_{(i)}$ é a i -ésima coluna de Γ . A decomposição espectral (autovalor e autovetor) da matriz de covariância revela a dimensionalidade do hiperplano no qual os dados estão espalhados, e assim, PCA projeta eixos nas direções de máxima variância⁵.

Utilização Como foi dito anteriormente, o que buscamos são os eixos originais, e não os novos eixos gerados. À semelhança do método anterior (ver equação 3.17), o eixo \vec{f} que contém a máxima variância é aquele que tem a menor distância, em termos de magnitude, do autovetor de Γ com o maior autovalor, e assim como em LDA, a função deve devolver $1/M$, para que o argumento que maximiza a função seja aquele que tem a menor discrepância da projeção em relação a $\vec{\gamma}$.

3.3.7 Seleção com IRMFSP

Esta técnica de seleção de descritores, cuja sigla em inglês corresponde a *Inertia Ratio Maximization using Feature Space Projection* e foi proposta por Peeters (2003a), se baseia em dois critérios. No primeiro critério, supondo um classificador com modelos gaussianos, observações de classes de sons pertencentes a uma classe específica devem estar distantes das observações de outras classes. Uma forma de resolver esta restrição é considerar a razão r entre a matriz de dispersão entre classes (eq. 3.8) e a matriz de dispersão total (eq. 3.10). Para um descritor f_i , r é então definido como

$$r_i = \frac{S_b}{S_t} = \frac{\sum_{k=1}^K \frac{n_k}{N} (\mu_{i,k} - \mu_i)(\mu_{i,k} - \mu_i)^T}{\frac{1}{N} \sum_{n=1}^N (f_{i,n} - \mu_i)(f_{i,n} - \mu_i)^T}, \quad (3.20)$$

⁴Para mais detalhes da definição de PCA, ver Mardia *et al.* (1979) e Theodoridis e Koutroumbas (2008).

⁵Por vezes, a transformação PCA nem sempre é satisfatória, uma vez que a escolha do autovetor com maior autovalor, e, portanto, maior variância, pode causar a sobreposição das classes (Theodoridis e Koutroumbas, 2008)

onde N é o número total de observações, n_k é o número total de observações da classe k , m_i é a média amostral do descriptor f_i , e $m_{i,k}$ é a média amostral do descriptor f_i para a classe k . Assim, um descriptor f_i com um valor alto para r_i é um descriptor cujas classes estão bem separadas com relação à dispersão dentro das classes.

O segundo critério relevante é que um método deveria levar em consideração o fato de que um descriptor f_i com um valor alto da razão r_i carregue em si a mesma informação que outro descriptor previamente selecionado. Para resolver esta questão, o método IRMFSP aplica um processo de ortogonalização depois da seleção do descriptor f_i , da seguinte forma: Seja F o espaço de descriptores, \vec{f}_i o último descriptor selecionado e \vec{g}_i sua versão normalizada

$$\vec{g}_i = \frac{\vec{f}_i}{\|\vec{f}_i\|}, \quad (3.21)$$

o espaço de descriptores F é projetado em g_i , mantendo o descriptor f_i . O processo é repetido até que o ganho de adicionar um novo descriptor seja muito pequeno. O critério de parada adotado por Peeters foi $\frac{r_l}{r_1} < 0.01$, onde r_l é a razão da iteração corrente.

Utilização Para o método IRMFSP ser integrado no algoritmo 1, faltaria somente a adição do segundo critério antes de reter o descriptor f_i . Outra modificação do método acima proposto é o critério de parada, onde o processo deve repetir até que tenha processado todos os descriptores.

3.3.8 Critérios para seleção do subconjunto final

Na subseção anterior expomos a técnica de seleção de descriptores baseado no algoritmo de seleção por filtro sequencial (algoritmo 1) e seis técnicas para a avaliação do subconjunto. O que vamos apresentar nesta seção são dois métodos para selecionar o subconjunto final de descriptores, sem descartar qualquer dos critérios apresentados. Não estamos interessados neste momento em avaliar qualquer uma destas técnicas e eleger uma para que nos informe o melhor subconjunto, mas sim que, de forma ponderada, todos os critérios tenham um peso na decisão final, caracterizando assim, uma escolha baseada em multi-critérios. Existem dois motivos para a escolha desta heurística, o primeiro é porque acreditamos que todos os métodos oferecem bons critérios de separabilidade entre as classes, sem um vencedor claro. A segunda motivação é simplesmente para manter o foco de nossa pesquisa: apesar de sabermos que esta etapa é muito importante para a construção dos modelos de segmentação, estamos mais interessados em comparar os diferentes métodos de segmentação que propriamente os métodos de seleção, o que por fim, multiplicaria os resultados comparativos, e poderia, por fim, dar outro entendimento para o problema proposto.

Método da soma dos primeiros selecionados

O primeiro método (algoritmo 5) dá a precedência para aqueles descriptores que aparecem mais vezes nas primeiras posições. Isso é feito contabilizando-se o número de ocorrências que um descriptor ocupou em todos os subconjuntos. Assim, se um descriptor foi sempre selecionado entre as primeiras posições em todos os subconjuntos, este algoritmo contabiliza um valor baixo para este descriptor, o que o tornará um dos primeiros a ser selecionado.

Método da união de conjuntos

Este método (algoritmo 6) propõe unir os primeiros elementos de cada subconjunto para formar um único conjunto. Neste algoritmo, além dos subconjuntos de cada critério, deve-se ainda fornecer um corte $l < d$, para a extração dos l primeiros elementos de cada subconjunto.

Uma das características deste método é que pode acontecer do subconjunto final ter um número menor de elementos, tendo no mínimo l descriptores e no máximo $\min\{N * l, d\}$, onde N , neste caso, é o número de subconjuntos a serem unidos.

Algoritmo 5 Define subconjunto final pela soma dos primeiros
SUCONJUNTO_FINAL_SOMA(S_1, \dots, S_N)

Entradas:

$S_1, \dots, S_j, \dots, S_N$, tal que $S_j = \{f_{j,1} \dots f_{j,i} \dots f_{j,d}\}$, onde $f_{j,i} \in \{1 \dots d\}$

Saída: S ; subconjunto final

```

1:  $R \leftarrow \underbrace{\{0, \dots, 0\}}_d$ 
2: para  $i \leftarrow 1$  até  $d$  faz
3:   para  $j \leftarrow 1$  até  $N$  faz
4:      $R[S_j[i]] \leftarrow R[S_j[i]] + i$ ; acumula a soma da posição que cada critério deu para ele
5:   fim para
6: fim para
7:  $S \leftarrow \text{ORDENA_INDICE}(R)$ ; devolve os índices de  $R$  na ordem em que foram ordenados pelos seus
valores.
```

Algoritmo 6 Define subconjunto final pela união dos primeiros elementos

SUCONJUNTO_FINAL_UNIAO(l, S_1, \dots, S_N)

Entradas:

$S_1, \dots, S_j, \dots, S_N$, tal que $S_j = \{f_{j,1} \dots f_{j,i} \dots f_{j,d}\}$, onde $f_{j,i} \in \{1 \dots d\}$
 l ; número de elementos de corte para cada subconjunto

Saída: S ; subconjunto final

```
1:  $S \leftarrow S_1[1 \dots l] \cup S_2[1 \dots l] \cup \dots \cup S_N[1 \dots l]$ 
```

3.4 Geração de Descritores Dinâmicos

A escolha das características do som para a segmentação automática é o que pode determinar seu sucesso ou fracasso, e um dos aspectos do vetor de observações extraídos diretamente do sinal de áudio é que os mesmos só contêm a informação da janela de análise (normalmente entre 30 ms e 100 ms) em torno de um instante de tempo definido. Por esta razão, Peeters *et al.* (2002b) os chamam de *descritores estáticos*, em contraste com os *Descritores Dinâmicos* (DD), que tentam de alguma forma representar a evolução dos descritores no tempo. Se, por um lado, aumentamos o tamanho da janela de análise (supondo que os descritores requeridos dependem do cálculo do espetro) e esperamos que assim as componentes do espetro armazenem esta informação temporal de alguma maneira, por outro lado, o sistema de extração de descritores teria uma latência muito maior em sua resposta.

Outra forma, um pouco menos custosa, de obter estes descritores, é usar a transformada de *Fourier* de tempo reduzido como anteriormente e depois executar uma “suavização” em cada eixo de descritor extraído, como veremos na subseção 3.4.1. A subseção 3.4.2 mostra ainda uma outra alternativa, proposta por Peeters *et al.* (2002b), onde os descritores convencionais não são calculados, e somente as componentes da FFT são levadas para o vetor de observações após uma filtragem em sub-bandas de frequência *Mel*.

3.4.1 Descritores Dinâmicos Cumulativos

A geração de descritores cumulativos requer que os descritores convencionais, como *MFCC* ou *ZCR*, sejam previamente calculados. Além disto, é preciso definir um horizonte de tempo L , que corresponderá ao intervalo temporal, ou à memória temporal, que cada descritor vai condensar.

Seja então $X = \{x_n\}$, $x_n \in \mathbb{R}^d$, $n = 1 \dots T$, o vetor de observações extraído do sinal de áudio, onde n é um índice de uma janela. Assim, definimos a sequência de descritores dinâmicos $F = \{f_n\} \subset \mathbb{R}^\alpha$, $n = L + 1 \dots N$ como

$$f_t = D(x_t, x_{t-1}, \dots, x_{t-L}),$$

onde D é uma função de suavização que leva em conta L observações e α é a nova dimensão após a transformação.

A função de suavização D pode ser definida de diversas maneiras, e nós consideramos quatro diferentes estratégias. A primeira, os gera através dos momentos estatísticos, ou seja, considera que a memória temporal pode ser armazenada se considerarmos a média, variância, obliquidade (medida de assimetria da fdp) e curtose (medida de “achatamento” da fdp). A segunda os gera através da norma Euclidiana, ou seja, o tempo é um vetor e a memória temporal é a magnitude deste vetor. A terceira os gera através da média ponderada da série por uma curva exponencial, ou seja, considera que a memória neste caso tem pesos menores para observações mais distantes, e pesos maiores para observações mais próximas. A última maneira pretende armazenar as variações lentas do espectro, ao registrar as primeiras componentes da FFT .

Para demonstrar as diferenças entre cada método de geração de descritores, mostraremos três matrizes de similaridade para cada método de geração: a primeira, construída com os descritores originais, e a segunda e a terceira, construídas com os descritores dinâmicos gerados, com memória temporal de 1 e 8 segundos, respectivamente. Nestas imagens, os pontos vermelhos são as transições reais entre as diferentes seções musicais, definidas manualmente.

Matriz de similaridade

A matriz de similaridade, contém medidas de discrepância entre pares de observações $x_i, x_j \in \mathbb{R}^d$. Uma simples medida de discrepância que pode ser utilizada é a distância Euclidiana

$$d_e^2(x_i, x_j) = (x_i - x_j)^T (x_i - x_j). \quad (3.22)$$

Outra medida de similaridade comumente utilizada é o cosseno do ângulo entre os vetores, que pode atingir valores altos, mesmo se os vetores forem pequenos em magnitude. A medida de similaridade do cosseno é dada por

$$d_c(x_i, x_j) = \frac{x_i^T x_j}{\|x_i\| \cdot \|x_j\|}. \quad (3.23)$$

Dependendo do contexto, o custo computacional de calcular esta matriz pode ser alto. Como as aplicações de segmentação musical *normalmente* consideram somente uma vizinhança de observações, algumas heurísticas podem ser aplicadas para minimizar este custo.

O propósito de exibir a matriz de similaridade – ou de dissimilaridade, dependendo da medida adotada–, é que podemos avaliar visualmente o quanto as observações de um mesmo segmento estão próximas de outras observações. Nas imagens, o que buscamos são os quadrados em preto, ou seja, onde conseguimos visualizar as formas quadradas pretas, é porque as observações dentro do quadrado têm uma distância pequena, e a geração de descritores foi benéfica. O ideal seria se fosse possível gerar matrizes binárias, com regiões bem definidas entre segmentos contíguos. Os pontos em *vermelho* nas imagens condizem com os pontos de mudança de segmento, que foram estipulados por um método automático (ver seção 2.1).

Estatísticas baseadas em momentos

Neste método, para cada descritor $f_k(t)$ (i.e. o eixo k de f no instante de tempo t), são gerados quatro descritores, ou seja, no final da operação o espaço de descritores aumenta, $F = \{f_n\}$, $f_n \in \mathbb{R}^{4*d}$. Os descritores são:

- Média

$$\hat{\mu}_k(t) = \frac{1}{L} \sum_{n=t}^{t-L+1} x_k(n) \quad (3.24)$$

- Variância

$$\hat{\sigma}_k^2(t) = \frac{1}{L} \sum_{n=t}^{t-L+1} (x^k(n) - \mu_k(t))^2 \quad (3.25)$$

- Assimetria

$$\hat{\gamma}_k(t) = \frac{1}{L} \frac{\sum_{n=t}^{t-L+1} (x^k(n) - \mu_k(t))^3}{\sigma_k^3(t)} \quad (3.26)$$

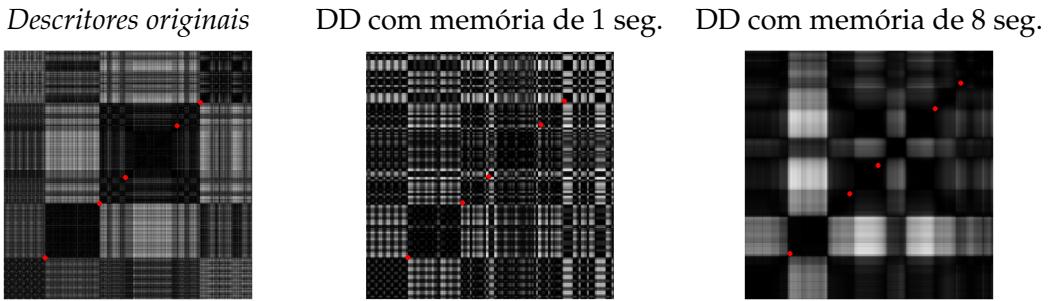
- Curtose

$$\hat{\kappa}_k(t) = \frac{1}{L} \frac{\sum_{n=t}^{t-L+1} (x^k(n) - \mu_k(t))^4}{\sigma_k^4(t)} \quad (3.27)$$

onde $t = 1, \dots, N - L$, e $k = 1, \dots, d$.

A tabela 3.1 mostra as matrizes de similaridade para os diferentes tipos de descritores calculados com este método.

Tabela 3.1: Matrizes de similaridade de DD gerados por momentos estatísticos



Norma Euclidiana

Neste método, considera-se que a série temporal traçada por um descritor k pode ser representada pela norma de L observações consecutivas, e assim, para cada instante, podemos calcular o descritor como:

$$N_k(t) = \sqrt{\sum_{n=t}^{t+L-1} T_k(n)^2} \quad (3.28)$$

onde $t = 1, \dots, N - L$, e $k = 1, \dots, d$.

A tabela 3.2 mostra as matrizes de similaridade para os diferentes tipos de descritores calculados com este método.

Média Ponderada

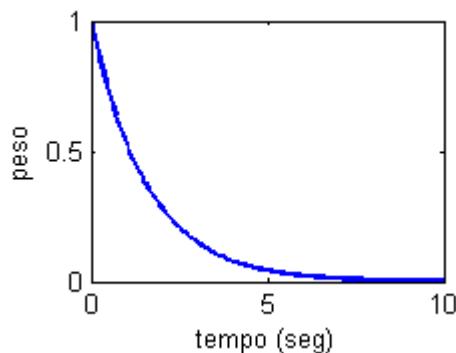
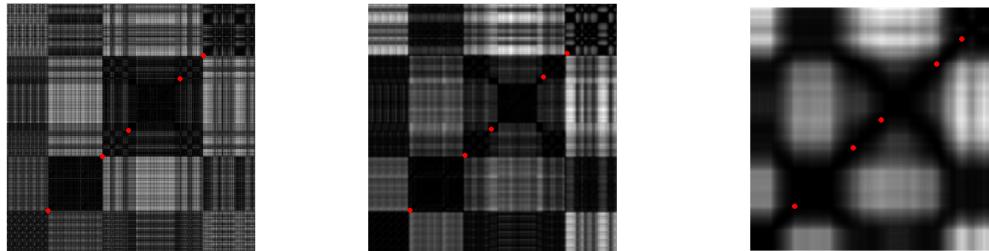
Neste método, o valor do descritor no instante t é substituído pela média dos elementos da série temporal ponderada por uma função de janela exponencial $w(n) = e^{-2\pi n}$ (ver figura 3.11).

$$P_k(t) = \sum_{n=t}^{t+L-1} T_k(n) w(\theta(n)) \quad (3.29)$$

onde $t = 1, \dots, N - L$, $k = 1, \dots, d$ e $\theta(n) = \frac{t-n}{L-1}$.

Tabela 3.2: Matrizes de similaridade de DD gerados pela norma Euclidiana

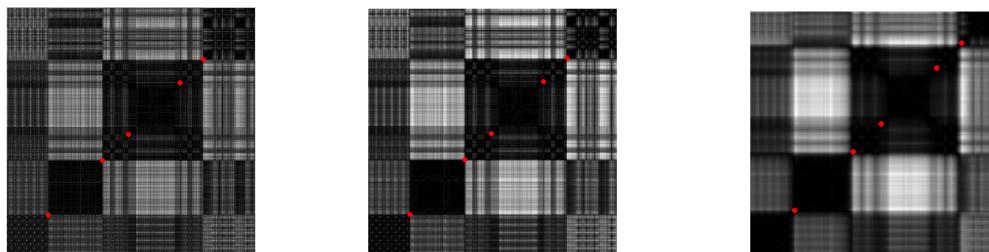
Descriptores originais DD com memória de 1 seg. DD com memória de 8 seg.

**Figura 3.11:** Função exponencial para o cálculo do descritor dinâmico por média ponderada.

A tabela 3.3 mostra as matrizes de similaridade para os diferentes tipos de descriptores calculados com este método.

Tabela 3.3: Matrizes de similaridade de DD gerados pela média ponderada

Descriptores originais DD com memória de 1 seg. DD com memória de 8 seg.



Coeficientes da Transformada de Fourier

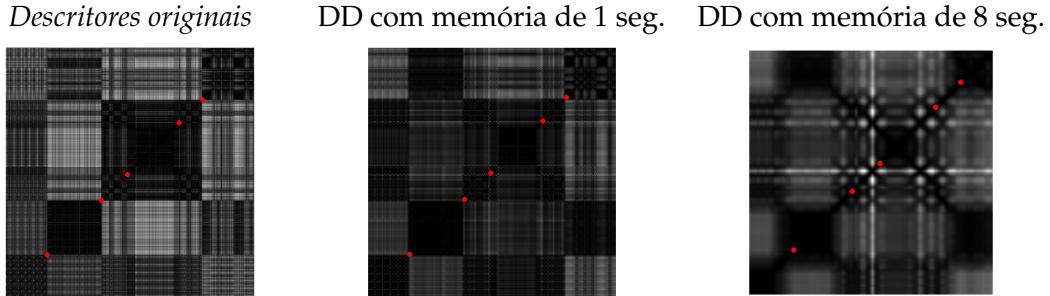
Neste método, cada descritor k foi considerado como um sinal independente, e novos coeficientes foram calculados através da transformada de Fourier. Foram selecionados somente os primeiros 15 coeficientes de cada transformação, aqueles que representam a variação lenta do espectro. Neste caso foi utilizada a Transformada Discreta de Fourier (DFT), e assim, os novos coeficientes $j = 1, \dots, 15$ podem ser calculados da seguinte forma:

$$X_k^j(t) = \sum_{n=t}^{t+L-2} T_k(n) e^{-\frac{2\pi i}{L} j(n-t)}, \quad (3.30)$$

onde $t = 1, \dots, N - L$, e $k = 1, \dots, d$.

A tabela 3.4 mostra as matrizes de similaridade para os diferentes tipos de descritores calculados com este método.

Tabela 3.4: Matrizes de similaridade de DD gerados pela análise espectral



3.4.2 Descritores Dinâmicos por Bandas Mel (DDBM)

A proposta desta técnica é modelar a evolução do tempo através da energia espectral e de um banco de filtros auditivos (filtros *Mel*). Inicialmente proposto por Peeters *et al.* (2002b), o método modela a evolução do sinal através da transformada de Fourier aplicada na energia do espectro ao longo de um intervalo de tempo de longa duração (chegando até 10 segundos). O método pode ser resumido da seguinte forma:

- O sinal de áudio $x(t)$ passa por N filtros passa-banda em intervalos regulares na escala *Mel*.
- A evolução lenta (frequências entre 0 e 50 Hz) da energia do sinal $x_n(t)$ que sai do n -ésimo filtro, $n = 1 \dots N$, é analisada pela transformada de Fourier com uma janela de tamanho L .

Desta forma, para cada instante t , é gerada uma matriz $X_{n,t}(\omega)$, com o espectro da banda de frequência n observada. Segundo os autores, uma “escolha apropriada de ω , n e L , pode facilitar a procura por padrões repetitivos”, então uma outra etapa seleciona os coeficientes (n, ω) que maximizam a Informação Mútua (nos nossos experimentos, utilizamos o seletor de descritores descrito na seção 3.3). A figura 3.12 demonstra os passos do processo de extração de descritores dinâmicos, e a tabela 3.5 demonstra a matriz de similaridade gerada a partir deste processo.

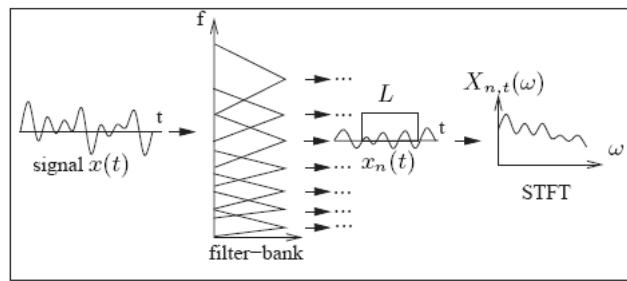


Figura 3.12: Processo de extração de descritores (Peeters *et al.*, 2002b). Da esquerda para a direita: sinal, banco de filtros, sinal de cada filtro, STFT para cada sinal filtrado



Tabela 3.5: Matrizes de similaridade de DD (Peeters et al., 2002b)

Capítulo 4

Segmentação Musical

ANTES de entrar em detalhes mais técnicos e fórmulas matemáticas é necessário esclarecer o que vem a ser, no sentido desta pesquisa em particular, um *segmento musical*. Devemos, pois, descrevê-lo para que o leitor tenha ao menos uma intuição subjetiva, e, a partir disto, definiremos o processo de *segmentação*. De modo amplo, “*segmentar* dados acústicos é identificar e rotular suas diferentes seções de interesse” (Aucouturier e Sandler, 2001). Para outros autores (Dannenberg e Goto, 2009; Peeters *et al.*, 2002b), o termo utilizado é *sumarização* ou *descoberta de estrutura*, onde ambas as definições apontam para diferentes tipos de aplicação, como veremos mais adiante. O termo *segmento musical* é muito subjetivo, mas é consenso que a música de modo geral é formada por estruturas, onde nós humanos conseguimos perceber as mudanças, tais como: as diferentes linhas melódicas, os efeitos, a harmonia, os padrões rítmicos, melodias similares, cadências similares, enfim, todo o jogo da composição musical. Infelizmente esta é uma tarefa difícil para as técnicas computacionais que conhecemos hoje, que “não conseguiram chegar a um nível de desempenho próximo ao dos humanos” (Dannenberg e Goto, 2009), como por exemplo, a separação de instrumentos em um sinal de áudio com vários instrumentos ao mesmo tempo.

Como foi visto na seção 2.1, os dados musicais foram gerados automaticamente através de trechos de sons recortados, formando uma colagem de trechos sonoros distintos, mas estes dados musicais não têm nenhuma relação com a música a não ser pelo fato de terem sido recortadas de músicas. Possivelmente, nenhum músico irá concordar que o que foi gerado é de fato música, e que existe uma continuidade ou estrutura lógica que justifique a segmentação de um dado musical apresentado desta forma¹. Entretanto, é inegável que existe alguma estrutura (formada pelos trechos), e acreditamos que as mesmas técnicas utilizadas para segmentar estas “músicas *frankenstein*” poderiam ser utilizadas para segmentar verdadeiras músicas, de verdadeiros compositores. Então como podemos interpretar o que é um segmento na música? De forma bem simples, podemos começar pela definição de *forma* de Schoenberg (1993):

O termo *forma* é utilizado em muitas acepções: nas expressões “*forma binária*”, “*forma ternária*” ou “*forma rondó*”, ele se refere, substancialmente, ao número de partes; a expressão “*forma sonata*” indica, por sua vez, o tamanho das partes e a complexidade de suas inter-relações; quando nos referimos ao “*minueto*”, ao *scherzo*, e a outras “*formas de dança*”, pensamos nas características rítmicas, métricas e de andamento, que identificam a dança. Em um sentido estético, o termo *forma* significa que a peça é “*organizada*”, isto é, que ela está constituída de elementos que funcionam tal como um *organismo vivo*.

Neste sentido, da *forma* relacionada ao “número de partes”, o *segmento* seria, analogamente, uma das partes da música. Por exemplo, supondo uma música com forma ternária A–B–A, diríamos que existem dois segmentos: A e B, sendo que o segmento B só vem depois de um segmento

¹É válido reforçar que optamos por uma geração automática de dados musicais simplesmente para facilitar a localização das mudanças de segmento.

A e que o mesmo se repete depois do segmento B. A complexidade de encontrar os segmentos é proporcional à complexidade de estruturas da música, não somente no aspecto do número de partes, mas principalmente pelo conteúdo do material sonoro. Desta forma, como vamos explicar mais adiante, comparar a *forma* (como número de partes) e o *segmento* (como parte), não passa de uma analogia, e não é suficiente para caracterizá-lo. Consideraremos importantes duas questões sobre o material sonoro para caracterizar o segmento: o *timbre* e o *gênero musical*.

O timbre. O timbre² desempenha um importante papel na identificação de um segmento, uma vez que a altura (frequência) e intensidade não são aspectos que estamos preocupados, e a duração faz sentido somente de forma global. O timbre foi objeto de diversos estudos (Catanzaro, 2003; DALBAVIE, 1991; Sethares, 2005) sob diferentes perspectivas, e Boulez (1985) summariza as duas maneiras de se considerar o timbre: a primeira, “de forma objetiva, científica, sem linguagem, sem critérios estéticos”; e a outra, “de uma maneira subjetiva, artística, de abordar o timbre como uma componente de linguagem com os critérios estéticos e formais”. Pensar no timbre objetivamente pode não ter nenhum valor para a arte musical, no entanto, são as características objetivas do timbre que distinguem, idealmente, um segmento de outro.

Definimos o timbre, portanto, como a evolução temporal das componentes de frequência de um ou mais instrumentos musicais, cuja definição é também considerada de maneira abrangente, podendo incluir instrumentos musicais de orquestra tradicional (fagote, violoncelo, etc.), instrumentos musicais populares (bateria, baixo elétrico, guitarra, etc.) ou mesmo instrumentos e ruídos digitais, e todas as *misturas e sobreposições* destes instrumentos existentes em uma música. E esses timbres, simples ou complexos, presentes na música em geral, servirão como base para o que queremos segmentar. Esta definição de timbre está de acordo com a de Aucouturier *et al.* (2005), onde os autores não estão preocupados com o timbre de um instrumento, mas com o “timbre polifônico”, a descrição do “timbre global”, para enfim comparar os modelos de timbre criados.

O gênero musical. O gênero musical pode parecer um assunto menos importante no processo de segmentação, mas consideramos a hipótese de que encontrar um segmento em uma música popular é mais fácil do que encontrá-lo em uma música clássica ou erudita. Podemos encontrar argumentos a favor desta hipótese nos trabalhos de Adorno (1994), que discute as diferenças entre os dois tipos de música, e o primeiro aspecto relevante da música popular é o efeito da relação dos detalhes com a estrutura geral da música, onde o ouvinte “fica inclinado a ter reações mais fortes para a parte do que para o todo”. E na música clássica, “o detalhe contém virtualmente o todo e leva à exposição do todo, ao mesmo tempo em que é produzido a partir da concepção do todo”. Segundo Adorno, os detalhes da música popular se encontram em posições estratégicas na estrutura geral, e por terem esta padronização da estrutura, seriam mais facilmente reconhecidas por um ouvinte, que sabe de antemão que não haverá surpresas na estrutura geral da música. Veja que o mesmo poderia ser dito em termos de localização de segmentos por um ouvinte.

²Na música, o timbre é considerado a propriedade do som que caracteriza ou identifica o som do instrumento ou voz. É pelo timbre que as pessoas distinguem o som de um instrumento de outro. Algumas definições do timbre o explicam pelo que não é, como por exemplo a definição da *American Standards Association*(1960), onde “o timbre é o atributo de sensação musical no qual o ouvinte pode julgar a similaridade de dois sons, tendo estes intensidades e alturas diferentes”. Tal afirmação sugere que o timbre não é altura e nem mesmo intensidade. Neste caso, fica a pergunta: e se o som não tem uma altura definida, não existe timbre? Muitas das explicações tendem a simplificar a descrição, conforme definição proposta pelo dicionário *The New Grove Dictionary of Music and Musicians*, editado por Sadie e Tyrrell (2001), que tenta ir mais a fundo na questão: “Um termo que descreve a qualidade tonal de um som; um clarinete e um oboé soando na mesma altura e mesma intensidade produzem timbres diferentes. O timbre é um atributo mais complexo que a altura e a intensidade, que podem ser representadas por uma escala unidimensional; a percepção do timbre é a síntese de vários fatores, e em músicas geradas por computadores um esforço considerável tem sido dedicado para a criação e a exploração de espaços de timbre multi-dimensionais. O espectro em frequência de um som, em particular as formas em que diferentes parciais crescem em amplitude durante o ataque, são de grande importância para se determinar o timbre.”. Para Roads (1996), “um timbre comum agrupa tons de um instrumento a diferentes alturas, intensidades e durações”, e assim torna clara que a preocupação em categorizar o timbre está mais vinculada à percepção que ao sinal sonoro.

Do ponto de vista da música clássica, ocorrem com mais frequência formas mais complexas, ou formas tão dependentes do contexto harmônico (tonal ou atonal) que somente um ouvinte atento poderia compreender sua estrutura. Vale lembrar que, na seleção de descritores (capítulo 3), evitamos calcular aqueles que são gerados pela parte harmônica do sinal, ou seja, não poderíamos descobrir a estrutura de uma música se a movimentação harmônica tiver papel fundamental na própria estrutura. Em algumas músicas (principalmente no pós-romantismo), o problema se torna ainda maior, por exemplo, no *Quarteto de cordas*, Op. 28 de Anton Webern (ver trecho na figura 4.1), em que a série dodecafônica, com uma estrutura interna simétrica (contendo uma sequência intervalar que se repete na própria estrutura; e invariante em transposições, retrógrados, inversões e retrógrados das inversões), é utilizada durante toda a peça e a estrutura da mesma se dá de maneira tão viscosa que não é possível determinar onde começa e onde termina cada seção. Neste exemplo, a estrutura só se revela mediante a análise de uma transcrição simbólica (i.e. em partitura) da música, e não pelo “timbre global”. No caso da música popular existem ainda as fórmulas de “embelezamento atrás do qual o esquema sempre pode ser percebido” (Adorno, 1994), muitas vezes estes “embelezamentos” se resumem em efeitos e distorções, e que seriam mais facilmente identificáveis, pois possibilitaria uma distinção mais evidente do timbre.

Figura 4.1: Primeiros compassos de *Quarteto de cordas*, Op. 28 - Anton Webern

Como último argumento da importância do gênero musical, podemos citar o estudo de seg-

mentação de Aucouturier e Sandler (2001), em que os autores discutem a necessidade dos Modelos Escondidos de Markov dados três tipos de música: o primeiro, e mais simples, é o das músicas populares, onde os autores observam que na maioria das vezes os timbres estavam bem definidos e separados no espaço de descritores, e que a segmentação poderia ser realizada somente com um agrupamento estático; o segundo tipo, de complexidade intermediária, é o das músicas orquestrais, e os autores observam que os timbres estão sobrepostos no espaço de descritores, e, portanto, mais difíceis de desenredar, e que os agrupamentos estáticos não oferecem uma boa precisão da localização das seções; no último tipo, e mais complexo, estão as músicas contemporâneas, em que os autores confessam a falha na segmentação, por exemplo, em obras de Gyorgy Ligeti (compositor húngaro, 1923-2006). Em Peeters *et al.* (2002b), os autores também sugerem que a geração de um sumário musical “só é aplicável a certos tipos de gêneros musicais, baseados em algum tipo de repetição”. Como não é objetivo deste trabalho aprofundar esta discussão, e nem mesmo comparar os resultados de segmentação com músicas populares, clássicas ou contemporâneas, é desnecessário acrescentar outros argumentos para esta hipótese, mas ficam aqui, no entanto, informações que podem ajudar o leitor a compreender as características de uma seção, cuja detecção pode se tornar mais difícil dependendo de seu gênero, até mesmo para nós humanos.

Caracterizações desejáveis da segmentação. É uma tarefa difícil dar uma definição formal do segmento, visto que existe a questão da temporalidade associada ao espaço de timbres. Entretanto, podemos definir, de forma abrangente, quais são as características desejáveis da segmentação.

1. Transições claras nas fronteiras de seções

Uma maneira de exigir isso é obrigar que cada seção S formada pelas observações x_i, x_{i+1}, \dots, x_j (indexadas pelo frame de análise correspondente) satisfaça a condição que nem x_{i-1} nem x_{j+1} pertencem ao casco convexo dos pontos x_i, \dots, x_j . Outra maneira possível seria construir um modelo gaussiano para as observações da seção S e exigir que os pontos imediatamente antes e imediatamente depois da seção tenham baixa probabilidade de acordo com o modelo gaussiano.

2. Economia do número de seções

Deseja-se evitar a tendência à pulverização das seções, que no pior caso tenderia à associação de uma nova seção a cada frame. Uma maneira de garantir isso, por exemplo, é exigir durações mínimas das seções (parametrizáveis conforme o contexto) e exigir que as regiões do espaço de características ocupadas por seções adjacentes sejam claramente distinguíveis (por exemplo através da existência de um separador linear, ou por algum critério estatístico como a existência de uma distância mínima entre os modelos gaussianos correspondentes às seções adjacentes).

3. Economia no tamanho das seções

Deseja-se evitar a tendência à construção de seções muito grandes, que no pior caso tenderia à associação de uma única seção à música inteira. Isso pode ser feito exigindo-se que cada seção não possa ser dividida em duas sub-seções mantendo-se as propriedades (1) e (2).

Aplicações. Na literatura encontramos diversas aplicações que utilizam a segmentação musical, sendo a maioria direcionada por necessidades comerciais, provavelmente devido ao crescente número de vendas de músicas pela internet. Outras aplicações estão interessadas na descoberta da estrutura musical (Peeters *et al.*, 2002b) (tópico da área de recuperação de informação musical), mas os pesquisadores não deixam claro quem é o público alvo (se são compositores, musicólogos ou simplesmente ouvintes) e muito menos qual é a finalidade (comercial, acadêmica ou didática).

Das aplicações que encontramos na literatura, podemos citar:

- **Navegação rápida em músicas.** A navegação rápida seria uma opção para o ouvinte, que poderia controlar, indo para frente ou para trás na música, sem necessariamente ter que

passar por todos os instantes entre um ponto e outro na música. Seria uma forma de “busca de títulos” dentro de uma música.

- **Sumário visual.** Para Dannenberg e Goto (2009), um ouvinte poderia usufruir da construção de sumários musicais (que fornecem um curto resumo dos principais elementos de um trabalho musical), possibilitando-os buscar um pedaço em particular de uma música que eles conhecem, ou localizar músicas não-familiares que eles poderiam gostar baseado na similaridade destes sumários.
- **Separação voz/música.** Em uma aplicação de transcrição de fala, a segmentação representa uma redução na utilização de recursos computacionais, uma vez que somente o que é fala é que seria transcrito (Berenzweig e Ellis, 2002).
- **Estrutura musical.** Encontrar a estrutura da música pode ajudar em análises musicais. Na base dos trabalhos realizados nesta área está o conceito que a estrutura só pode ser induzida pelas repetições de materiais similares: “A segmentação de uma peça musical e o agrupamento destes segmentos em aglomerados é uma forma de análise ou ‘explicação’ da música” (Dannenberg e Goto, 2009).
- **Recomendação automática de músicas e Reconhecimento de gênero.** A segmentação e rotulação podem ser utilizadas por como uma etapa intermediária para tarefas mais complexas, como por exemplo a recomendação automática de músicas e reconhecimento de gênero.

Outro ponto é que, considerando dados musicais, todas as aplicações mencionadas partem do princípio que o dado já é conhecido em sua totalidade, ou seja, nenhum autor promove qualquer aplicação a partir de uma segmentação em tempo real em que existe um fluxo contínuo de áudio musical. Fato que não ocorre quando voltamos a atenção para pesquisas em reconhecimento de voz (Chen e Gopalakrishnan, 1998; Omar *et al.*, 2005; Sainath *et al.*, 2007). Neste sentido, podemos citar algumas possibilidades de aplicações utilizando a segmentação musical em tempo real:

- **Automatização de gatilhos em performances musicais.** Supondo que em uma composição houvesse a necessidade de iniciar processamentos ou eventos associados a cada seção, a segmentação automática auxiliaria a disparar gatilhos em performances musicais de acordo com a seção corrente. O gatilho poderia, por exemplo, disparar automaticamente qualquer evento musical ou visual previamente desenvolvido, sendo que este acompanhamento poderia ainda utilizar todas as informações da seção corrente, como o rótulo associado e o modelo estatístico da seção (e.g. média e covariância de cada eixo de descritores). Este aplicativo poderia ser utilizado por compositores, artistas multimídia e *performers*.
- **Busca de músicas em dispositivos móveis.** Hoje existem aplicações que fazem reconhecimentos de músicas em tempo real³, e a segmentação automática poderia melhorar a precisão do classificador, que enveria, como consulta, não observações de um trecho qualquer da música sendo executada, mas observações de uma ou mais seções, possibilitando a utilização de modelos markovianos para a classificação musical. O público alvo desta aplicação seriam os usuários de aparelhos móveis em geral.

Em sinais de áudio que não são estritamente musicais, como em transmissões radiofônicas, a segmentação automática é um importante passo de pré-processamento para reconhecimento de fala e mineração de áudio, assim como para sinais de áudio musicais onde, segundo Dannenberg e Goto (2009), a segmentação é uma etapa inicial para outras tarefas, embora o que os autores chamam de segmentação seja somente a identificação da mudança de timbres:

³Reconhecimento de músicas é uma aplicação comum em aparelhos móveis de hoje em dia, como, por exemplo, o aplicativo Shazam (www.shazam.com).

"A segmentação pode ser utilizada como uma etapa inicial para um grande número de outras tarefas mais complicadas, incluindo sumarização musical, análise musical, busca musical e classificação de gêneros musicais (...) pode também auxiliar na navegação do áudio, uma tarefa que pode ser reforçada através de algum tipo de sumarização visual da música e dos segmentos de áudio."

Neste capítulo vamos ver todos os assuntos relacionados ao nosso estudo em segmentação e rotulação de sinais de áudio musicais. Começaremos por expor, na seção 4.1, as medidas de separabilidade entre classes para fins de comparação entre modelos de timbre. Essas medidas de separabilidade servem ao propósito de criar medidas de dissimilaridade entre pares de modelos das seções encontradas (que mais tarde chamaremos de estados da cadeia de Markov), com a finalidade de agrupar modelos que são similares. Em seguida, na seção 4.2, apresentaremos duas técnicas de aglomerados, que têm como objetivo agrupar modelos das seções encontradas. Nos capítulos seguintes veremos as técnicas de segmentação, agrupadas de acordo com as seguintes categorias:

1. **Segmentação supervisionada** (seção 4.3). Apresentaremos técnicas clássicas de classificação de padrões, que consideram um modelo de classificador através de dados de treinamento. Optamos por utilizar alguns modelos de classificador supervisionado somente para comparar os resultados com as outras técnicas.
2. **Segmentação não-supervisionada** (seção 4.4). São as técnicas que não precisam de amostras *a priori* para o treinamento, porém requerem a totalidade das observações do sinal musical para o início da segmentação.
3. **Segmentação não-supervisionadas em tempo real** (seção 4.5). Estas técnicas realizam a segmentação baseando-se apenas nas observações passadas, e, portanto, podem ser executadas em tempo real.

Por fim, na seção 4.6, apresentaremos formas de suavizar o erro de segmentação (válidas somente nos casos das técnicas de segmentação 1 e 2), e na seção 4.7, um método de avaliação dos resultados de segmentação, que dependem de dois fatores: o ponto de mudança e a rotulação dada para cada segmento.

4.1 Separabilidade entre Classes

A questão de separabilidade entre classes vem à tona quando queremos comparar dois modelos de segmentos encontrados. Esta comparação tem como propósito determinar o quanto um modelo de seção ω_i é similar a outro modelo de seção ω_j e, caso a "distância" entre eles seja pequena, gostaríamos de rotulá-los igualmente. Para efeito didático, vamos expor as fórmulas considerando somente dois modelos de segmento: ω_i e ω_j ; mas os conceitos apresentados devem ser extrapolados para um conjunto de M segmentos, onde são calculadas todas as distâncias d_{ij} dos pares de modelos ω_i e ω_j , $i, j \in \{1 \dots M\}$, podendo ser representadas por uma matriz triangular. As medidas aqui apresentadas supõem uma distribuição normal dos dados (assunto que vamos aprofundar no capítulo 5).

4.1.1 Divergência

Esta medida parte do princípio da regra do classificador de Bayes, em que se escolhe um determinado modelo (ou classe) ω_i sobre ω_j se $P(\omega_i|x) > P(\omega_j|x)$, onde x é o vetor de observações, e, que portanto, a razão $\frac{P(\omega_i|x)}{P(\omega_j|x)}$ fornece uma informação útil sobre as capacidades discriminantes associadas ao vetor de observações x em respeito às classes ω_i e ω_j . O mesmo vale para a fração $\ln \frac{p(x|\omega_i)}{p(x|\omega_j)} \equiv D_{ij}$, onde

$$D_{ij} = \begin{cases} 0 & , \text{ quando as distribuições estão totalmente sobrepostas;} \\ < 0 & , \text{ quando } p(x|\omega_j) > p(x|\omega_i); \\ > 0 & , \text{ caso contrário.} \end{cases}$$

A medida de divergência é a soma dos valores médios sobre a classe ω_i e os valores médios sobre a classe ω_j , ou seja:

$$d_{ij} = \int_{-\infty}^{\infty} (p(x|\omega_i) - p(x|\omega_j)) \ln \frac{p(x|\omega_i)}{p(x|\omega_j)} dx \quad (4.1)$$

Supondo que as funções de densidades para $p(x|\omega_i)$ e $p(x|\omega_j)$ são normais $\mathcal{N}(\mu_i, \Sigma_i)$ e $\mathcal{N}(\mu_j, \Sigma_j)$, a divergência d_{ij} fica da seguinte forma:

$$d_{ij} = \frac{1}{2} \text{traço} \left[\Sigma_i^{-1} \Sigma_j + \Sigma_j^{-1} \Sigma_i - 2\mathbf{I} \right] + \frac{1}{2} (\mu_i - \mu_j)^T (\Sigma_i^{-1} + \Sigma_j^{-1}) (\mu_i - \mu_j), \quad (4.2)$$

assim, a divergência d_{ij} pode ainda tomar valores altos mesmo se na média os modelos coincidem, pois pode haver ainda diferenças com relação à matriz de covariância Σ (Theodoridis e Koutroumbas, 2008).

4.1.2 Distância de Bhattacharyya

A distância de Bhattacharyya, também conhecida como limite de Bhattacharyya, foi desenvolvida como uma forma de calcular os limites dos erros de um classificador de Bayes – considerando densidades normais e a Teoria de Decisão de Bayes⁴. Assim como o limite de Chernoff, o limite de Bhattacharyya fornece um limite superior para este erro, que também é utilizado como uma medida de separabilidade entre classes (Theodoridis e Koutroumbas, 2008).

Assim, para os modelos de seção ω_i e ω_j , a distância é dada pela seguinte equação:

$$B_{i,j} = \frac{1}{8} (\mu_i - \mu_j)^T \left(\frac{\Sigma_i + \Sigma_j}{2} \right)^{-1} (\mu_i - \mu_j) + \frac{1}{2} \ln \frac{\left| \frac{\Sigma_i + \Sigma_j}{2} \right|}{\sqrt{|\Sigma_i||\Sigma_j|}}. \quad (4.3)$$

Uma vez calculada a distância para cada par de modelo de seção, aqueles que tiverem uma distância muito pequena serão agrupados, e este é o propósito destas medidas. Nas figuras da tabela 4.1, os estados representam as seções. Na primeira imagem (a), vemos uma matriz de dissimilaridade com uma representação das seções e duas “distâncias” calculadas pelo método de Bhattacharyya. Visualmente, poderíamos dizer que os estados 2 e 3 são candidatos a um agrupamento, assim como os estados 6, 7, e 8. A imagem seguinte (b) representa os estados após um agrupamento, e é notável que os agrupamentos não devam ser realizados somente com estados adjacentes (no que se refere ao tempo), e sim com relação a todos os estados, pois estamos também interessados em saber se um segmento distante na música também se refere a um mesmo rótulo, e, portanto, não devemos fazê-lo considerando somente seções contíguas.

⁴A teoria de decisão de Bayes tem como objetivo determinar um limiar onde a probabilidade de erro $P(\text{erro}) = P(x \in \mathcal{R}_j, \omega_i) + P(x \in \mathcal{R}_i, \omega_j)$ é mínima, onde as regiões \mathcal{R}_i e \mathcal{R}_j foram determinadas por um limiar fixo, e os erros ocorrem quando x cai em uma região \mathcal{R}_i quando na verdade sua classe é ω_j , ou x cai em uma região \mathcal{R}_j quando sua classe é, na verdade, ω_i (Duda et al., 2001)

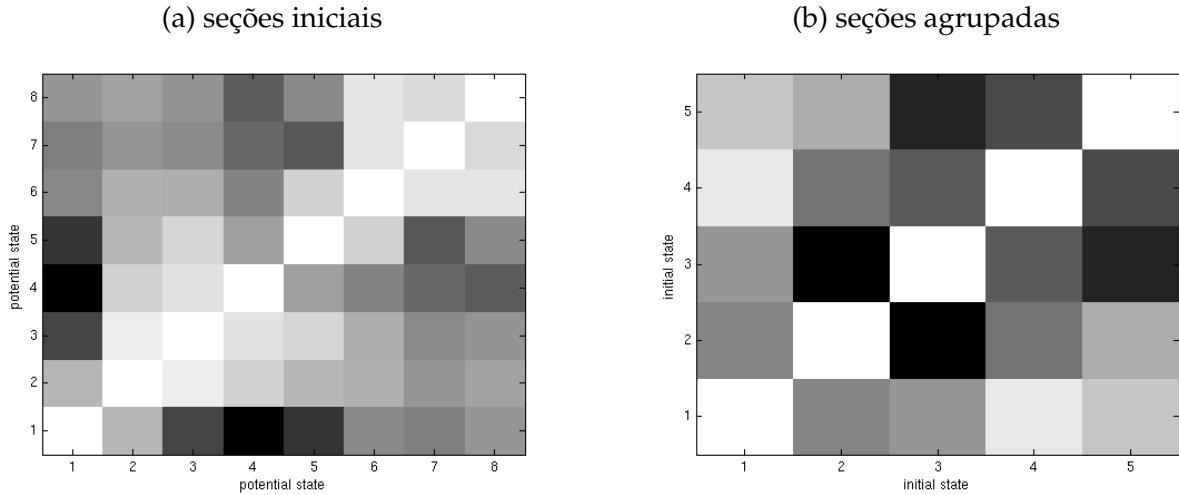


Tabela 4.1: Dissimilaridade entre pares de modelos de seção, calculadas com a distância de Bhattacharyya. Na matriz, as tonalidades em branco indicam uma distância pequena, e que é possível um agrupamento.

4.2 Aglomerados

A análise de aglomerados é um conjunto de técnicas não-supervisionadas de classificação, onde os rótulos dos dados de treinamento não estão disponíveis, e o objetivo é desvendar a organização dos dados em grupos, permitindo descobrir similaridades e dissimilaridades em um conjunto finito de observações, e, a partir disto, poder tomar decisões.

Definição de Análise de Aglomerados. Suponha que X é o conjunto de dados, ou seja,

$$X = \{x_1, x_2, \dots, x_i, \dots, x_N\}, \quad x_i \in \mathbb{R}^d$$

onde N é o número de vetores aleatórios e d é a dimensão do vetor.

Pela definição de Theodoridis e Koutroumbas (2008), um K -agrupamento de X é a partição de X em K conjuntos, C_1, \dots, C_K , que satisfazem as seguintes condições:

- $C_i \neq \emptyset, \quad i = 1, \dots, K$
- $\cup_{i=1}^K = X$
- $C_i \cap C_j = \emptyset, \quad i \neq j, i, j = 1 \dots, K$

Além disto, podemos dizer que em um agrupamento, os vetores de C_i são mais similares entre si e menos similares entre vetores de outro grupo. O termo *similar* ou *dissimilar* depende do tipo de grupos que se deseja associar (grupos compactos, alongados ou contornos de formas geométricas), mas estamos interessados nos grupos compactos, ou que estão mais próximos no espaço de dados. As medidas de similaridade são aquelas que medem o grau de similaridade entre vetores, sendo que existem diversos tipos de medida. Uma medida de dissimilaridade δ em X é definida como

$$\delta : X \times X \rightarrow \mathcal{R}$$

onde \mathcal{R} é um conjunto de números reais, tal que

$$\exists \delta_0 \in \mathcal{R} : -\inf < d_0 \leq d(x, y) < +\inf, \quad \forall x, y \in X,$$

$$\delta(x, x) = d_0, \quad \forall x \in X$$

e

$$\delta(x, y) = \delta(y, x), \quad \forall x, y \in X.$$

Desta forma, d_0 atinge um valor mínimo quando x e y são idênticos.

Medidas de dissimilaridade. Existem diversas medidas de dissimilaridade, e uma delas é aquela apresentada na seção 4.1, onde, por exemplo, a distância de Bhattacharyya pode ser utilizada para medir a dissimilaridade entre modelos de distribuições normais. Um exemplo de agrupamento com esta distância pode ser visualizado nas figuras 4.2 e 4.3, onde a primeira demonstra um conjunto de dados que podem ser representados por três funções normais $\mathcal{N}(\mu_1, \Sigma_1)$, $\mathcal{N}(\mu_2, \Sigma_2)$ e $\mathcal{N}(\mu_3, \Sigma_3)$, e a segunda demonstra o mesmo conjunto de dados após o agrupamento com esta medida de dissimilaridade, resultando em dois modelos $\mathcal{N}(\mu_1, \Sigma_1)$ e $\mathcal{N}(\mu_2, \Sigma_2)$.

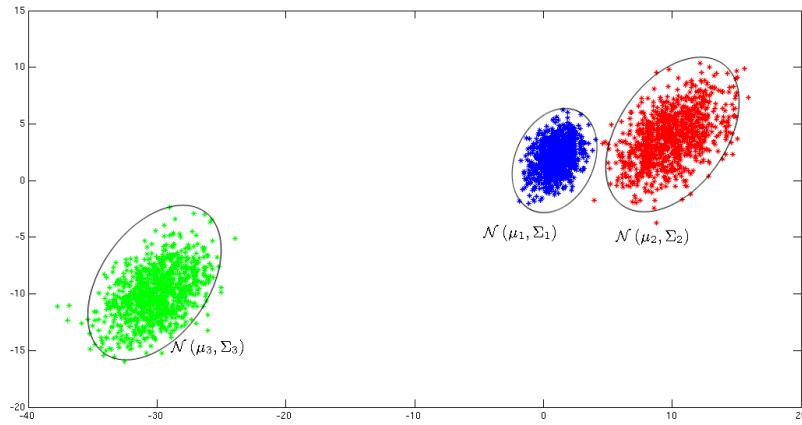


Figura 4.2: Gráfico de dispersão de um conjunto de dados representados por três funções normais

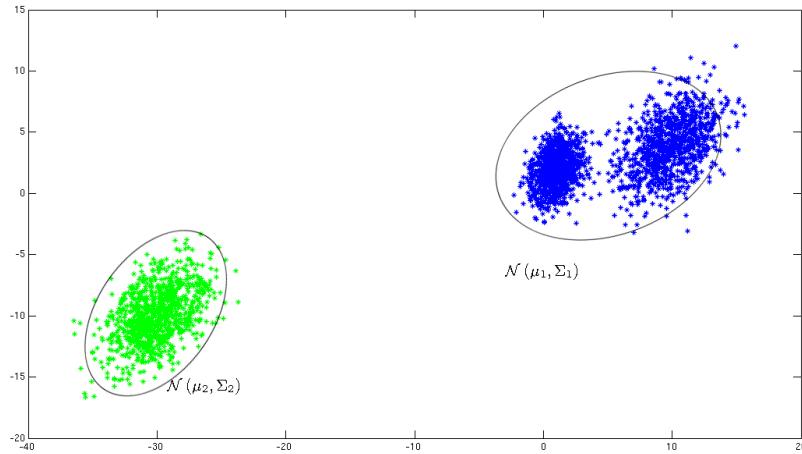


Figura 4.3: Mesmo conjunto de dados da figura 4.2 representado por duas funções normais, agrupadas por um critério de medida de dissimilaridade (distância de Bhattacharyya).

Outras medidas comumente utilizadas, somente para citar algumas, são:

- **Medida ponderada** l_p , também conhecida como medida de Minkowski:

$$\delta_p(x, y) = \left(\sum_{i=1}^d w_i |x_i - y_i|^p \right)^{\frac{1}{p}} \quad (4.4)$$

onde x_i e y_i são os eixos de x e y , $i = 1, \dots, l$ e $w_i \geq 0$ é o coeficiente de peso associado ao

índice i . Esta é uma generalização da distância em δ_p , e três casos particulares se originam a partir dela:

- **Distância Euclidiana**, definindo $p = 2$ e $w_i = 1$, $i = 1, \dots, d$,

$$\delta(x, y) = \sqrt{(x - y)^T(x - y)} \quad (4.5)$$

- **Distância Euclidiana normalizada**, definindo $p = 2$

$$\delta(x, y) = \sqrt{(x - y)^T V^{-1} (x - y)} \quad (4.6)$$

onde V^{-1} é uma matriz $d \times d$ simétrica, positiva-definida, normalmente uma matriz diagonal onde o i -ésimo elemento é o desvio padrão $S(i)^2$ do i -ésimo eixo de características.

- **Distância de Mahalanobis**, definindo $p = 2$

$$\delta(x, y) = \sqrt{(x - y)^T \Sigma^{-1} (x - y)} \quad (4.7)$$

onde Σ^{-1} é a matriz de covariância.

- **Medida do cosseno:**

$$\delta(x, y) = 1 - \frac{x^T y}{\|x\| \|y\|} \quad (4.8)$$

onde $\|x\| = \sqrt{\sum_{i=1}^l x_i^2}$ e $\|y\| = \sqrt{\sum_{i=1}^l y_i^2}$ são as normas dos vetores x e y , respectivamente.

Encontramos na literatura (Duda *et al.*, 2001; Theodoridis e Koutroumbas, 2008) outras distâncias, como, por exemplo, a distância de Pearson, a medida de *City Block* (que é uma variante de medida ponderada), as distâncias de Chebychev, Tanimoto, entre outros. A distância mais simples e imediata de ser utilizada é a distância Euclidiana (eq. 4.6), porém “esta medida não é invariante com relação a transformações lineares ou qualquer transformação que de alguma forma possa distorcer a relação de distância”(Duda *et al.*, 2001), o que pode ocasionar em agrupamentos totalmente distintos. Uma forma de contornar este problema é utilizar a distância de Mahalanobis (eq. 4.7), que normaliza os dados, forçando os eixos a terem média zero e variância unitária, mas esta normalização pode ser uma etapa anterior da extração de descritores, como vimos na subseção 3.3.1. Porém, existem casos em que esta normalização não é totalmente necessária, tornado-se até mesmo prejudicial para o agrupamento, como, por exemplo, quando o espalhamento de um certo grupo é ocasionado pela presença de subgrupos, e que seria desejável identificá-los. No restante desta seção veremos duas técnicas de agrupamento utilizadas nos algoritmos de segmentação.

4.2.1 K-Médias

K-médias faz parte de uma categoria de algoritmos cujo agrupamento é baseado em uma otimização por função de custo J . Nesta categoria de algoritmos, o número de grupos é mantido fixo, e iterativamente o algoritmo produz agrupamentos enquanto não alcança um ótimo local de J . Ademais, K-médias faz parte de uma subcategoria de algoritmos que produz agrupamentos onde cada vetor pertence somente a um grupo, chamados de algoritmos de agrupamentos rígidos (*hard clustering*).

A função de otimização J no algoritmo de K-médias é uma função dos vetores do conjunto de dados X parametrizado por Θ e U , e normalmente utiliza o quadrado da norma Euclidiana como medida de dissimilaridade, ou seja,

$$J(\Theta, U) = \sum_{i=1}^N \sum_{j=1}^K u_{ij} \|x_i - \theta_j\|^2, \quad (4.9)$$

onde N é o número de observações de X , K é o número fixo de grupos, θ_j é a média dos vetores do j -ésimo grupo, ou seja, é o representante do grupo C_j , e $u_{ij} \in \{0, 1\}$ representa a relação de pertinência $X_i \in C_j$, tendo valor 1 somente para um dos grupos C_j e 0 para o restante, ou seja,

$$\sum_{j=1}^K u_{ij} = 1. \quad (4.10)$$

O algoritmo 7 converge para um mínimo local da função de custo J (eq. 4.9).

Algoritmo 7 Algoritmo de Agrupamento por K-Médias
 $[\Theta, U] = \text{K-MEDIAS}(X, m)$

Entradas:

$X = X_1 \dots X_N$ # conjunto de dados
 m # número de grupos

Saída:

$\Theta = \{\theta_j\}, j = 1, \dots, m$ # onde θ_j é o representante do grupo C_j
 $U = \{u_i\}$ # Vetor de N posições de associação de cada vetor x_i com um grupo C_j

- 1: Inicialização aleatória de θ_j para $j = 1, \dots, m$
- 2: **repete**
- 3: **para** $i \leftarrow 1$ até N **faça**
- 4: $j \leftarrow \arg \min_j (\|x_i - \theta_j\|^2)$ # determina o representante mais próximo, digamos j
- 5: $u_i \leftarrow C_j$
- 6: **fim para**
- 7: **para** $j \leftarrow 1$ até m **faça**
- 8: Atualização dos parâmetros: θ_j é a média dos vetores $x_i \in X$ com $u_i = C_j$;
- 9: **fim para**
- 10: **enquanto** não exista mudança em θ_j entre duas iterações consecutivas

4.2.2 Aglomerados Hierárquicos

Esta categoria de algoritmos de agrupamento produz uma sequência de agrupamentos com números de grupos decrescentes a cada nível, onde o agrupamento produzido a cada passo t é resultado da fusão de dois grupos do nível anterior.

O ponto de partida desta técnica é a matriz de dissimilaridade $N \times N$, $P_0 = P(X)$, onde X são os dados de entrada e P é a função que calcula a matriz de dissimilaridade. A cada nível t , o tamanho da matriz P_t se torna $(N-t) \times (N-t)$, e o procedimento para gerá-la depende somente de P_{t-1} , C_i e C_j , da seguinte forma:

1. remove as duas linhas e as duas colunas da matriz P_t que correspondem aos grupos sendo fundidos C_i, C_j , e
2. adiciona uma nova linha e uma nova coluna à matriz P_t que correspondem às distâncias entre o novo grupo formado C_q e os grupos antigos que não foram alterados.

Uma forma simples de calcular a distância deste novo grupo C_q para os grupos antigos é o algoritmo de vínculo único (do inglês *single link algorithm*):

$$P_t(q, s) = \min\{P_{t-1}(i, s), P_{t-1}(j, s)\}, \quad s = \{1, \dots, N-t\} \quad (4.11)$$

O pseudo-algoritmo 8 demonstra a construção dos grupos hierárquicos. O resultado da execução deste algoritmo pode ser visualizado por um dendrograma⁵, como, por exemplo, o da

⁵do grego dendron (árvore), -gramma (desenho)

Algoritmo 8 Algoritmo de Agrupamento Hierárquico

 $[\mathfrak{R}_0, \dots, \mathfrak{R}_{N-1}] = \text{HCLUST}(X)$

Entradas:
 $X = X_1 \dots X_N$ # conjunto de dados
Saída:
 $[\mathfrak{R}_0, \dots, \mathfrak{R}_t, \dots, \mathfrak{R}_{N-1}]$ # onde \mathfrak{R}_t é o agrupamento no nível t

1: # Inicialização

2: $\mathfrak{R}_0 \leftarrow \{C_i = x_i, i = 1, \dots, N\}$ # O agrupamento inicial é dado por cada elemento de X 3: $P_0 \leftarrow P(X)$ # matriz de dissimilaridade $N \times N$ 4: $t \leftarrow 0$ 5: **repete**6: $t \leftarrow t + 1$ 7: $(r, s) \leftarrow \underset{i \neq j}{\operatorname{argmin}}\{P_{t-1}(i, j), (i, j) \in \mathfrak{R}_{t-1}\}$ 8: $C_q = \{C_r, C_s\}$ 9: $\mathfrak{R}_t \leftarrow (\mathfrak{R}_{t-1} - C_r - C_s) \cup C_q$ 10: Define P_t a partir de P_{t-1} , C_i e C_j , como descrito no texto acima.11: **enquanto** $t < N - 1$ # o agrupamento \mathfrak{R}_{N-1} contém um único grupo, com todos os elementos de X

figura 4.4, cujos dados para o agrupamento são as médias dos vetores de descritores de 367 trechos de músicas, que foram agrupados, neste exemplo, em 30 grupos (os grupos gerados podem ser visualizados no apêndice B).

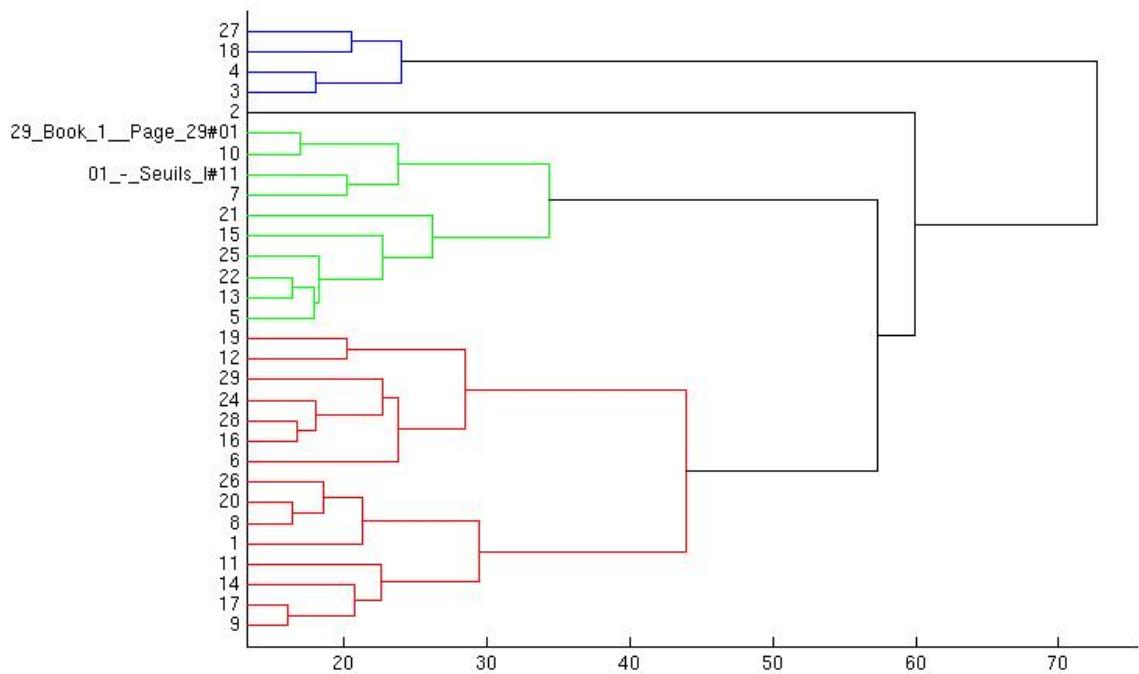


Figura 4.4: Dendrograma de um agrupamento hierárquico realizado para agrupar 367 vetores de trechos de músicas

Existem muitas formas de calcular o vínculo entre os grupos. No exemplo da figura 4.4, o algoritmo de vínculo utilizado para a atualização da matriz de dissimilaridade foi o vínculo de Ward, que é definido por:

$$P_t(q, s) = \sqrt{\frac{2n_q n_s}{n_q + n_s} \|\bar{x}_q - \bar{x}_s\|} \quad (4.12)$$

onde \bar{x}_q e \bar{x}_s são os centros dos grupos C_q e C_s , e n_q e n_s são o número de elementos nos grupos C_q e C_s , respectivamente.

4.3 Segmentação Musical Supervisionada

Dados de treinamento correspondem ao conjunto de dados cujos rótulos das classes (no caso da segmentação musical, os rótulos das seções de interesse) são conhecidos, e os dados de teste são os dados que não são utilizados para a construção dos modelos de classificação, ou segmentação. A segmentação supervisionada implica que temos em mãos um conjunto de dados de treinamento, e podemos construir segmentadores explorando esta informação *a priori*. Um sistema como este, onde os modelos de segmentação são construídos a partir de dados de treinamento, tem uma aplicação restrita a um contexto específico, e não serviria, por exemplo, para a descoberta de novas seções e muito menos para ser utilizado como ferramenta de segmentação de um catálogo musical extenso. Desta forma, o problema de segmentação se reduz a um problema de classificação supervisionada, e podemos avaliar a probabilidade de erro de cada modelo de classificador da mesma forma.

Validação cruzada k -fold A validação cruzada k -fold é um método simples de avaliação de classificadores, tanto para ajudar nas estimativas dos parâmetros dos modelos quanto para avaliar o quanto bem o modelo escolhido generaliza a classificação. Em nossos estudos, a validação cruzada foi utilizada com o propósito de medir o desempenho em termos de probabilidade de erro dos classificadores de forma a cobrir mais cenários de teste. O fato de que existe uma temporalidade associada às amostras não faz diferença para alguns classificadores, como por exemplo, o *Classificador Ingênuo de Bayes*, e uma validação cruzada com amostras aleatórias poderia ser utilizada, mas para classificadores de séries temporais, como *Modelos Ocultos de Markov*, a sequência de observações é importante por manter a integridade temporal dos dados. Entretanto, por questões de simplificação, os mecanismos para a validação nos dois casos são os mesmos.

A ideia da validação cruzada k -fold é partitionar a amostra original em k conjuntos obedecendo a ordem das observações, sendo que destes k conjuntos somente um é retido para ser utilizado como amostra de validação, e os $k-1$ restantes são utilizados para treinamento. O processo é repetido k vezes com cada amostra de validação sendo utilizada somente uma vez, e o resultado final, a taxa de acerto, é a média das taxas de acerto de cada teste nas amostras de validação.

Avaliação no caso supervisionado. No caso supervisionado é conhecido o número de seções e temos os dados de treinamento para cada seção. Assim, podemos simplificar a questão da avaliação da probabilidade de erro nestes casos. Suponha que X é conjunto de dados de teste, e foram construídos K modelos de classes $\omega_i = \{1, \dots, K\}$ através dos dados de treinamento. A probabilidade de erro é dada por:

$$P_{erro}(\omega) = 1 - \sum_{n=1}^K P(x \in \omega_i, \omega_i). \quad (4.13)$$

Além das técnicas aqui apresentadas, utilizamos ainda outras duas técnicas em nossos experimentos, que são: Perceptron de Multicamadas e Regressão Logística (ambos assuntos podem ser vistos em Witten e Frank (2005)). Não vamos apresentar os detalhes destas técnicas aqui, pois o assunto é extenso e o desempenho destas técnicas nos experimentos foi muito baixo.

4.3.1 Classificador Ingênuo de Bayes

O Classificador Ingênuo de Bayes (NBC) é chamado de ingênuo por considerar que as variáveis são independentes. NBC incorpora um passo de aprendizado em que as probabilidades para as classes e as probabilidades condicionais para um dado descritor e classe são estimadas. Essas

probabilidades estimadas são baseadas em suas frequências encontradas nos dados de treinamento. Tais estimativas são chamadas de hipótese de aprendizado, formadas apenas contando as ocorrências de várias combinações nos dados de treinamento. Para classificar uma nova amostra x_i , são contados quantos elementos de cada classe estão na vizinhança de x_i , e a amostra é classificada considerando a hipótese de aprendizado e a probabilidade de vizinhança de cada classe sobre o total de elementos em cada classe.

Dados $\{\omega_1, \dots, \omega_K\}$ o conjunto finito de classes – ou segmentos – e x , o vetor de descritores d -dimensional, a fórmula de Bayes (eq. 4.14) permite encontrar a probabilidade *a posteriori* $P(\omega_i|x)$ (a probabilidade de ω_i dado o vetor de descritores), uma vez que sabemos de antemão a probabilidade *a priori* $P(\omega_i)$, a verossimilhança (ou probabilidade condicional) $p(x|\omega_i)$, e a evidência $p(x)$, que serve como fator de escala garantindo que a soma das probabilidades *a posteriori* seja igual a 1:

$$P(\omega_i|x) = \frac{p(x|\omega_i)P(\omega_i)}{p(x)} \quad (4.14)$$

onde

$$p(x) = \sum_{j=1}^K p(x|\omega_j)P(\omega_j) \quad (4.15)$$

A probabilidade condicional $p(x|\omega_i)$ mais comumente utilizada é a normal multivariada $\mathcal{N}(\mu, \Sigma)$,

$$p(x|\omega_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i (x-\mu_i)}, \quad (4.16)$$

onde

$$\mu_i = E[x] \quad (4.17)$$

é o vetor de d componentes da média das amostras da classe i , e

$$\Sigma_i = E[(x - \mu_i)^T (x - \mu_i)] \quad (4.18)$$

é a matriz de covariância $d \times d$ das amostras da classe i . Desta forma, pela fórmula de Bayes, cada observação é classificada como pertencente a uma classe ω que tem a maior probabilidade *a posteriori*, ou seja,

$$\omega_k = \arg \max_i p(\omega_i|x), \quad i = 1, \dots, K. \quad (4.19)$$

4.3.2 Modelos de Misturas de Gaussianas

Uma maneira diferente de modelar a probabilidade condicional $p(x|\omega_i)$ é através de uma combinação linear de funções de densidades como uma soma de M densidades Gaussianas, chamadas componentes ou estados da mistura:

$$p(x|\omega_i) = \sum_{m=1}^M c_{i,m} \mathcal{N}(x, \mu_{i,m}, \Sigma_{i,m}) \quad (4.20)$$

onde

$$\sum_{m=1}^M c_{i,m} = 1, \quad i = 1, \dots, K \quad (4.21)$$

são os coeficientes da mistura ou probabilidade do estado, x é o vetor do descritor observado, e N é a função de densidade de probabilidade normal da classe ω_i para cada componente da mistura m , com média $\mu_{i,m}$ e matriz de covariância $\Sigma_{i,m}$.

O objetivo de utilizar misturas de gaussianas é aproximar ao máximo qualquer função de densidade contínua com um número adequado de misturas M e um modelo apropriado para o

conjunto das componentes de densidade $\mathcal{N}(x, \mu_{i,m}, \Sigma_{i,m})$. Uma forma de estimar os parâmetros das misturas é utilizar o algoritmo de Esperança e Maximização (*Expectation Maximization*)⁶.

4.3.3 K-Vizinhos Mais Próximos

O algoritmo de *K*-vizinhos mais próximos (*K-NN* ou *K-Nearest Neighbors*) é um dos mais populares algoritmos de aprendizado de máquina. Primeiramente armazena os vetores de descriptores da fase de aprendizado e depois, para classificação de uma nova instância, encontra um conjunto de k amostras mais próximas no espaço de descriptores, e marca a nova instância para a classe que tem mais elementos no conjunto. A distância Euclidiana ou a distância de Mahalanobis são normalmente utilizadas para determinar a similaridade. Um esquema do algoritmo pode ser descrito da seguinte maneira. Dado um vetor de descriptores x , uma medida de distância e os dados de treinamento X_o , então:

- Dentre as N amostras de treinamento X_o , identifica os k vizinhos mais próximos de x , independentemente das classes a que pertencem.
- Destas k amostras, identifica o número de vetores k_i que pertencem à classe ω_i , $i = 1, \dots, K$
- Atribui x à classe ω_i com o número máximo k_i de amostras.

Com os passos acima, podemos determinar qual é a classe, ou no nosso caso, a seção ω_i , para cada observação x . Existem algumas desvantagens no uso deste classificador:

- requer que todos os vetores de treinamento estejam na memória de forma a fornecer uma decisão de classificação de uma nova instância;
- é altamente sensível a descriptores irrelevantes que podem dominar as distâncias métricas; e
- pode requerer uma alta carga computacional a cada nova pesquisa.

4.3.4 Árvores de Decisão - J48

Este classificador é baseado em árvores de classificação e regressão (*CART*) (Duda *et al.*, 2001), e é construído de cima para baixo (*top-down*) em uma estrutura derivada da divisão e conquista, começando pelo desritor mais significativo. Em casos de descriptores com valores não binários, um procedimento de partição deve ser definido por um limiar através dos dados de treinamento. As amostras de treinamento são ordenadas de acordo com uma ordem apropriada, e todo o processo é repetido recursivamente para os nós descendentes. Basicamente um classificador baseado em árvores de decisão deve considerar os seguintes problemas na construção do modelo:

- Qual é o número de divisões – cada resultado de decisão é chamado de divisão.
- Que tipo de pergunta ou teste de propriedade deve ser executada a cada nó.
- Quando parar de dividir. Prefere-se decisões que conduzem a uma árvore simples e compacta com poucos nós.
- Impureza do nó. Um nó tem uma impureza baixa se todas as padronas que atingem um determinado nó carregam uma mesma classe, e, no caso contrário, tem uma impureza alta se ao atingir um nó, as classes são igualmente representadas.
- Poda e o efeito do horizonte. A decisão de uma divisão ótima em um nó N não é influenciada pelas decisões de seus descendentes, e a poda é uma alternativa para a decisão de parar de dividir.

⁶Para mais detalhes, veja Theodoridis e Koutroumbas (2008), p. 45

Este algoritmo efetua uma poda baseada em regras derivadas da árvore de treinamento, e cada nó tem uma regra associada, que é uma conjunção de decisões desde o nó raiz até aquele nó. Assim, após o treinamento da árvore, é possível determinar qual é a classe, ou seção, a que pertence uma observação x .

4.3.5 Modelos Ocultos de Markov

Os modelos ocultos de Markov (HMM ou *Hidden Markov Models*) foram desenvolvidos para solucionar o problema de uma sequência de decisões, quando existe uma temporalidade inerente, em outros termos, quando o estado no tempo t é influenciado pelo tempo $t - 1$. Nos HMMs os estados não são diretamente observáveis, e por isto não podemos construir os modelos de estados diretamente das informações de cada estado isoladamente. HMMs são formados por duas componentes: um conjunto de variáveis escondidas que não podem ser observadas diretamente, e uma propriedade de Markov que é usualmente associada a algum comportamento das variáveis escondidas. De outra forma, HMM é um processo duplamente estocástico, sendo um não visível, ou não observável, mas que pode ser observado por outro processo estocástico que produz a sequência de observações. As observações no caso do sinal de áudio são os descritores, ou um modelo de misturas gaussianas que representam o descritor. A definição formal de HMM com densidades de observações contínuas é a seguinte. Sejam

1. N o número de estados em um modelo, onde denotamos cada estado como

$$S = \{s_1, \dots, s_N\},$$

2. Q a sequência de estados fixa de tamanho T , que corresponde a uma sequência de observações X ,

$$X = x_1, \dots, x_T, \quad e$$

$$Q = q_1, \dots, q_t, \dots, q_T,$$

onde q_t é o estado no tempo t .

3. $A = \{a_{ij}\}$ a matriz de distribuição de probabilidade das transições de estado, onde

$$a_{ij} = P(q_{t+1} = S_j | q_t = S_i), \quad 1 \leq i, j \leq N,$$

ou seja, a_{ij} é a probabilidade de que no tempo $t + 1$, o estado seja igual a S_j , dado que no tempo t o estado é igual a S_i . Nos casos em que todos os estados atingem um ao outro, então $a_{ij} > 0, \forall i, j$.

4. $b_j(x)$ a distribuição de probabilidade no estado j ,

$$B = [b_j(x)], \quad b_j(x) = \sum_{m=1}^M c_{jm} \mathcal{N}(x, \mu_{j,m}, \Sigma_{j,m}), \quad 1 \leq j \leq N \quad (4.22)$$

onde a distribuição acima é um modelo de misturas de gaussianas (ver 4.3.2), sendo c_{jm} o coeficiente do m -ésimo componente da mistura no estado S_j , e \mathcal{N} é uma função de densidade normal – embora possa ser qualquer outra distribuição da família exponencial – com média $\mu_{j,m}$ e matriz de covariância $\Sigma_{j,m}$ para o m -ésimo componente da mistura no estado S_j .

5. $\pi = \{\pi_j\}$ a distribuição inicial do estado j , onde

$$\pi_j = P(q_1 = S_j), \quad 1 \leq j \leq N,$$

ou seja, π_j é a probabilidade de uma observação qualquer ser gerada pelo estado S_j no tempo $t = 1$.

De forma simples, HMMs são definidos então por

$$\lambda = (A, B, \pi). \quad (4.23)$$

Existe muito a se dizer a respeito de HMMs, como, por exemplo, como se dá o treinamento de um modelo λ através do algoritmo de re-estimação de Baum-Welch, ou como podemos escolher a sequência de estados $Q = q_1 \dots q_T$ que melhor explica, ou que maximiza a verossimilhança de uma sequência de observações $X = x_1 \dots x_T$ dado um modelo λ . Entretanto, dadas as definições básicas, preferimos simplesmente indicar para como se dá a utilização desta técnica na segmentação musical. O leitor interessado pode consultar o texto de Rabiner (1989) para mais detalhes.

Estes modelos foram utilizados extensivamente em reconhecimento de voz para modelar as expressões orais, que podem ser uma palavra falada, uma parte da palavra, ou até mesmo uma sentença completa. Nestes casos, para cada expressão s é criado um modelo HMM λ_s , e o reconhecimento se dá pela máxima verossimilhança de uma sequência dados os modelos previamente construídos. Para treinar um modelo λ_s são utilizadas várias “versões” X_i da expressão a ser reconhecida, podendo ter inclusive tamanhos diferentes, para que os pesos de A_s , B_s e π_s sejam estimados corretamente. Além disto, cada expressão s é projetada com certa exclusividade, sendo determinado *a priori* o número de estados e a sua topologia (quais são as possíveis transições entre estados).

Este cenário de reconhecimento de voz utilizando HMMs difere de um contexto de segmentação musical nos seguintes pontos:

1. No caso musical, o objeto de reconhecimento não é universal, ou seja, não é algo que ocorre a qualquer instante e em qualquer lugar, como é o caso de uma palavra, mas ocorre em uma única música e, portanto, não faria sentido criar um modelo λ genérico para reconhecer uma seção musical.
2. Ao criar um modelo para cada seção musical, supõe-se que cada uma foi modelada por mais de um estado, e se este é o caso, foi preciso ter um conhecimento quase especialista da seção musical a ser modelada. Desta forma, se conhecemos muito bem uma seção, podemos dizer que conhecemos também onde ocorrem as seções e quais elas são, e, com isto, não faz sentido segmentar e rotular algo que já conhecido.
3. Qualquer que seja a expressão oral, esta nunca pode ser comparada com o que é uma seção musical, pois não existem regras definidas para saber qual foi o raciocínio empregado pelo compositor para a inserção de uma mesma seção em uma música, podendo a mesma ser reduzida, ampliada (omitir ou acrescentar partes musicais) ou variada em termos temporais, harmônicos ou melódicos. Em outros termos, por não existirem regras que definem o que é uma seção, não se pode definir um modelo geral para que a encontremos, mesmo que tenhamos todas as ocorrências de uma seção em uma música, pois estas podem ser tão diferentes, em termos estruturais, que a probabilidade de erro seria muito alta, salvo as músicas que tem repetições rigorosamente iguais.

Com isto, a estratégia adotada para a utilização de HMMs é considerar que cada estado representa uma seção, e que cada estado é modelado com misturas de gaussianas, omitindo assim a temporalidade interna de cada seção, mas, em contrapartida, ressaltamos a característica de encontrar seções com o mesmo timbre musical. Outra vantagem desta abordagem é que, deixando de lado a questão temporal interna de cada seção, ressaltamos a temporalidade do todo musical, onde as transições entre as seções são mais importantes que as transições internas das seções, uma vez que é este o objetivo de nossa pesquisa. Isto vai de encontro com a pesquisa de Au-couturier e Sandler (2001), em que os estados representam as seções. No caso supervisionado, a etapa de treinamento se resume, portanto, a estimar as densidades de cada estado $b_j(x)$ – modelos

de misturas de gaussianas— através do algoritmo de Esperança e Maximização para misturas de gaussianas (ver subseção 4.3.2). A matriz de transição A , e consequentemente sua topologia, é determinada de acordo com as frequências em que cada seção ocorre dentro da música, mas em um caso em que esta informação não está disponível, o valor é distribuído igualitariamente. O mesmo ocorre com a distribuição inicial de cada estado π_j , em que os estados podem ter probabilidades iguais, caso a informação não esteja disponível. Para extrair a informação de qual é a sequência de estados Q que melhor explica uma sequência de observações X , basta executar o algoritmo de Viterbi (Rabiner, 1989).

4.4 Segmentação Musical Não-supervisionada

Em uma segmentação não-supervisionada, supõe-se que os dados de treinamento não estão disponíveis, e é preciso que todas as observações estejam disponíveis na memória para a execução das técnicas aqui descritas. Isto pode ser um problema em termos de espaço de memória computacional, mas dependendo do contexto, ou da aplicação que se deseja construir, os resultados podem ser satisfatórios. Por este motivo é que dizemos que estas técnicas são não-supervisionadas, distinguindo-as das técnicas não-supervisionadas em tempo real, que mesmo não tendo os dados de treinamento disponíveis, não têm a necessidade de ter todos os dados disponíveis.

Uma das facilidades de se ter todas as observações disponíveis é que podemos utilizar técnicas de agrupamento ou calcular matrizes de similaridade entre as observações, o que facilita, por exemplo, o reconhecimento de padrões musicais⁷. No nosso caso, a matriz de similaridade ajuda a encontrar os pontos de mudança de regime, como veremos nas subseções a seguir.

Lembramos ao leitor que as técnicas sinalizadas com um * são propostas novas que não encontramos na literatura.

4.4.1 Segmentação via Matriz de Dissimilaridade*

Uma segmentação por limiar de dissimilaridade é construída em duas etapas: (1) detecção dos pontos de mudança, e (2) rotulação através de uma das técnicas de separabilidade entre classes (seção 4.1). Na primeira etapa, os pontos de mudança são detectados a partir de grandes e rápidas mudanças no conteúdo do sinal, utilizando uma matriz de dissimilaridade (ver seção 3.4.1). Apresentaremos duas opções para a segmentação por matriz de dissimilaridade, uma utilizando um limiar e outra com detecção de picos.

Detecção de segmentos por limiar. Um limiar alto de dissimilaridade, $d \geq \alpha$, é utilizado na segmentação para reduzir o efeito da variação rápida do sinal, supondo que em cada segmento, o sinal supostamente tende a variar lentamente ou muito pouco. O algoritmo 9 mostra uma forma simples de calcular a sequência de dissimilaridade de amostras conjuntas.

Um exemplo de como esta heurística se comporta, considerando a matriz de dissimilaridade da figura 4.5, pode ser visualizado na figura 4.6. Os pontos vermelhos indicam as mudanças reais de um segmento para outro. Um problema que podemos ver imediatamente é que se o limiar fosse muito alto, $\alpha = .99$, somente um ponto seria identificado, e de outra forma, se escolhêssemos um limiar $\alpha = .2$, poderíamos pegar muitos outros pontos errados.

Detecção de segmentos pelos picos de dissimilaridade. De acordo com o algoritmo 9, a dissimilaridade $d_i = M_{i,i-1}$ considera somente uma amostra na vizinhança, e uma maneira um pouco mais elaborada de encontrar os pontos de mudança a partir da matriz de similaridade é calcular a sequência de similaridade considerando uma vizinhança de tamanho l , onde a dissimilaridade

⁷Para mais informações sobre reconhecimentos de padrões musicais a partir da matriz de similaridade, veja em Dannenberg e Goto (2009), p. 313

Algoritmo 9 Calcula a sequência de dissimilaridade

$$P = \text{DISS_SEQ}(M, \alpha)$$

Entradas:

M ; Matriz de dissimilaridade $N \times N$

α ; limiar para identificar os pontos de mudança de regime

Saída: P ; pontos de mudanças entre as seções

- 1: $D \leftarrow (d_1, d_2, \dots, d_N)$; Sequência de dissimilaridade entre as observações conjuntas
 - 2: **para** $i \leftarrow 2$ até N **faz**
 - 3: $d_i \leftarrow M_{i,i-1}$;
 - 4: **fim para**
 - 5: $D \leftarrow D / \max(D)$; normaliza os valores da sequência
 - 6: $P \leftarrow i \in N$ tal que $d_i \geq \alpha$
-

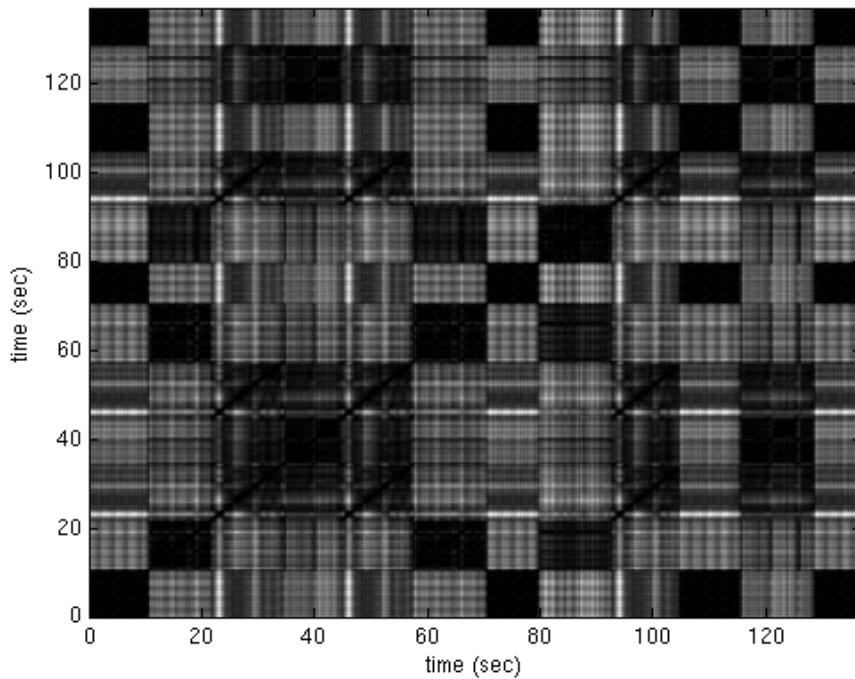


Figura 4.5: Matriz de dissimilaridade de um sinal musical.

d é em função dos vizinhos dos vizinhos, ou seja:

$$d(i) = \langle w, v_i \rangle, \quad (4.24)$$

onde $v_i = (M_{i-1,i+1}, M_{i-2,i+2}, \dots, M_{i-l,i+l})$ e w é uma função de distribuição qualquer. Desta forma, e considerando que existe uma restrição de tempo de processamento, o sistema final poderia calcular somente a “faixa” de vizinhança da matriz de similaridade, e não a matriz inteira, pois nem todas as combinações seriam utilizadas.

Em nossos estudos, w é uma função exponencial da forma $w = e^{-2\pi q}$, onde $q = [0, 1]$. Pela definição de v_i podemos ver que depois de considerar os pesos imediatamente mais próximos de i , os próximos pontos que são considerados são aqueles que vão se distanciando de i . O que justifica a utilização desta heurística é a observação de um fenômeno na matriz de dissimilaridade (observando somente os valores abaixo da diagonal, veja figura 4.8), que para cada ponto de transição entre seções, existe uma forma geométrica retangular cuja aresta superior esquerda se encontra próximo ao ponto onde gostaríamos de segmentar. A função acima serve ao propósito

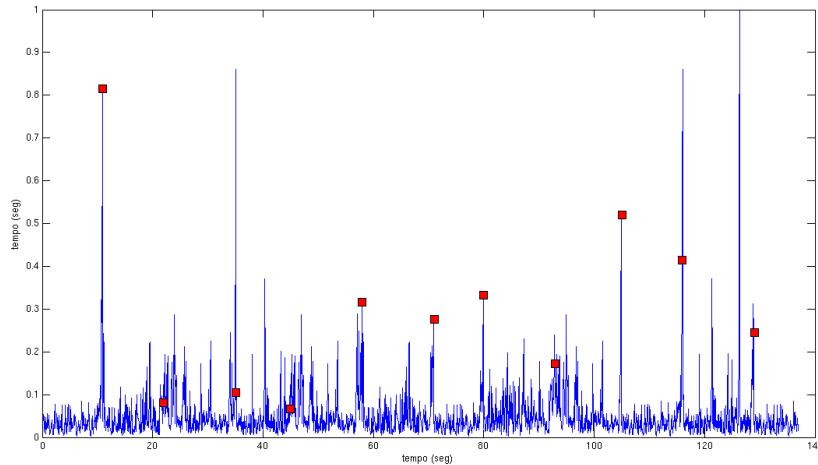


Figura 4.6: Sequência de dissimilaridade utilizando algoritmo 9. Os pontos vermelhos indicam as mudanças reais de seção, adquiridas manualmente.

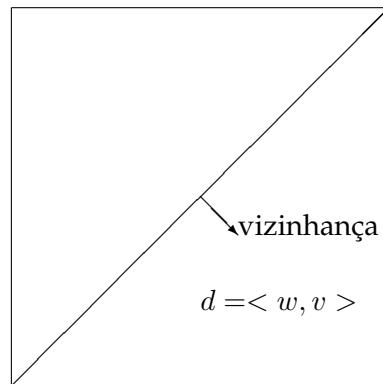


Figura 4.7: Representação da matriz de dissimilaridade e a direção de varredura da vizinhança.

de maximizar a dissimilaridade, ou seja, quando o ponto i estiver “colado” na aresta superior esquerda da forma geométrica, d_i atinge um valor máximo da sua vizinhança. Veja na figura 4.7 uma representação da fórmula acima.

Outro fenômeno observado na matriz de dissimilaridade com descritores dinâmicos (seção 3.4.1) é que, quanto maior for o parâmetro de memória temporal L (em janelas), mais distante a forma geométrica fica da diagonal principal da matriz de dissimilaridade, ou seja, nestes casos, para se ter uma maior eficácia no cálculo da sequência de dissimilaridade, é preciso considerar somente os vetores perpendiculares à diagonal principal e que estão a uma distância de $L/2$ amostras. Além disto, devido à suavização temporal na criação do descritor, o ponto de mudança é antecipado de $L/2$ amostras, o que significa que é preciso ajustar o ponto de mudança para $i + L/2$. Na figura do exemplo 4.8, os pontos ajustados estão representados em amarelo, que dificilmente podemos ver, e isto acontece pois os pontos verdes (gabarito) estão renderizando por cima, o que significa que a segmentação está no caminho correto.

Outra desvantagem do algoritmo anterior (algoritmo 9) é que é preciso determinar um limiar para o corte, e além disto, mesmo determinando um limiar, teríamos que ignorar os pontos vizinhos de alguma maneira. Outra forma de encontrar os pontos de mudança é utilizar um algoritmo para detectar os picos de uma função que seja tolerante a ruídos⁸. Os picos encontrados devem ser ordenados por ordem decrescente (para pegar sempre a maior similaridade) e ignorar aqueles que estão próximos de outros pontos, onde a proximidade ν é um parâmetro que determina qual

⁸Em nossos estudos, utilizamos um código em MATLAB de Nathanael C. Yoder, peakfinder.m

é o tamanho mínimo do segmento que devemos identificar.

Desta forma, considerando estas variáveis, a segmentação baseada na dissimilaridade é descrito no algoritmo 10.

Algoritmo 10 Calcula a sequência de dissimilaridade

 $P = \text{DISS_SEQ}(M, L, l, \nu)$

Entradas:

M ; Matriz de dissimilaridade $N \times N$

L ; parâmetro de memorial temporal na criação do descriptor

l ; vizinhança para o cálculo da dissimilaridade

ν ; tamanho mínimo de uma seção a ser identificada

Saída: P ; pontos de mudanças entre as seções

- 1: $D \leftarrow (d_1, d_2, \dots, d_N)$
 - 2: **para** $i \leftarrow 2$ até N **faca**
 - 3: $v_i \leftarrow (M_{i-1-L/2, i+1+L/2}, M_{i-2-L/2, i+2+L/2}, \dots, M_{i-L/2, i+l+L/2})$
 - 4: w é uma função de distribuição de densidade qualquer, preferencialmente com decaimento exponencial;
 - 5: $d_i \leftarrow \langle w, v \rangle$;
 - 6: **fim para**
 - 7: $R \leftarrow$ picos de D ; encontra os picos de D
 - 8: $V \leftarrow$ é o vetor de valores de R em ordem decrescente
 - 9: $P \leftarrow V(1)$
 - 10: **para** $i \leftarrow 2$ até tamanho de V **faca**
 - 11: **se** $|V(i) - V(j)| > \nu, \forall j \in P$ **então**
 - 12: $P \leftarrow P \cup \{V(i)\}$
 - 13: **fim se**
 - 14: **fim para**
 - 15: $P \leftarrow P + L/2$
-

A última etapa do algoritmo deve rotular os segmentos encontrados. Para isto, consideramos a técnica de separabilidade entre classes abordada na seção 4.1, onde a construção dos modelos gaussianos é imediata através da estimação por máxima verossimilhança sobre os dados circunscritos a cada seção encontrada. Em seguida é realizado um agrupamento hierárquico com um limiar determinado pelos dados para o corte na árvore.

4.4.2 Segmentação via Matriz de Similaridade de Cooper e Foote (2002)

No método proposto por Cooper e Foote (2002), a matriz de similaridade é utilizada para encontrar um segmento de máxima similaridade de tamanho l . Embora o objetivo da pesquisa seja encontrar um único sumário que melhor representa a música, podemos utilizar os princípios para extrair mais de um trecho para representar a música. Uma forma de calcular a máxima similaridade de tamanho l é através da média das colunas da matriz de similaridade no intervalo q, \dots, r , e, portanto, os intervalos com uma grande similaridade interna terá uma média maior. Assim, seja $S(m, n)$ a matriz de similaridade, o valor médio de similaridade para o intervalo q, \dots, r é dado por

$$\bar{S}(q, r) = \frac{1}{N(r-q)} \sum_{m=q}^r \sum_{n=1}^N S(m, n). \quad (4.25)$$

Desta forma pode-se generalizar a equação, e para encontrar o segmento ótimo de tamanho l , deve ser encontrado o trecho com este tamanho que tenha a pontuação máxima definida na

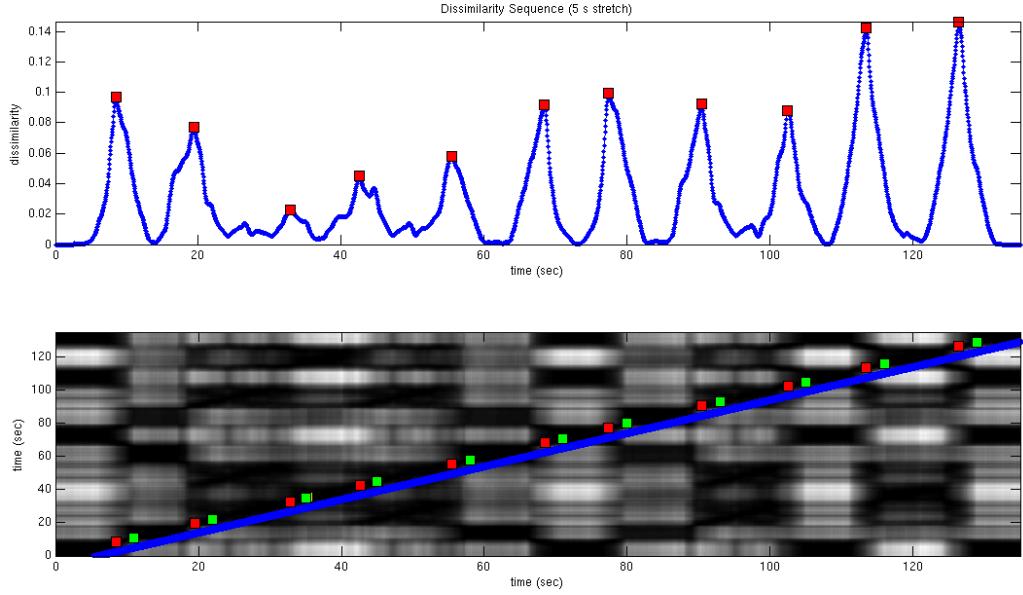


Figura 4.8: Resultado da primeira etapa da segmentação utilizando descritor dinâmico (norma Euclidiana e memória temporal de 5 segundos). Sequência de dissimilaridade (figura superior), sendo os pontos vermelhos os picos encontrados; e matriz de dissimilaridade (figura inferior), onde os pontos vermelhos são os picos encontrados, os pontos verdes são os pontos de mudança reais (gabarito), os pontos amarelos, que coincidem com os pontos verdes, são os pontos depois do ajuste $P + L/2$, e a faixa azul são os pontos da vizinhança.

equação 4.25. A pontuação $Q_l(i)$ é definida como

$$Q_l(i) = \bar{S}(i, i + l) = \frac{1}{NL} \sum_{m=i}^{i+L} \sum_{n=1}^N S(m, n), \quad (4.26)$$

para $i = 1, \dots, N - L$. O melhor ponto para iniciar a segmentação ocorre no instante q_l^* e termina no instante $q_l^* + l$, onde q_l^* maximiza a pontuação

$$q_l^* = \underset{1 \leq i \leq N-l}{\operatorname{argmax}} Q_l(i) \quad (4.27)$$

Os próximos pontos de máxima similaridade são encontrados a partir da localização dos picos da pontuação. Na figura 4.9, a matriz de similaridade (parte superior) gerou a pontuação Q , onde podemos ver os inícios estimados e reais. Notamos que em alguns pontos, o algoritmo acertou, mas outros nem tanto, indicando pontos até mesmo próximos. Notamos também que não somente os picos devessem ser considerados, mas também os vales, o que poderia ser justificado se considerarmos que a mínima pontuação corresponde também a um início de seção.

A última etapa do algoritmo deve rotular os segmentos encontrados. Para isto, consideraremos a técnica de separabilidade entre classes abordada na seção 4.1. Em seguida é realizado um agrupamento por K-Médias, sendo necessário que se tenha o número total de seções.

4.4.3 Segmentação por Delta de Mahalanobis (Tzanetakis e Cook, 1999)

O algoritmo proposto por Tzanetakis e Cook (1999) pode ser dividido em três estágios:

1. Calcula uma distância Δ_i entre observações adjacentes, digamos x_i e x_{i-1} . A distância de Mahalanobis

$$\Delta_i = (x_i - x_{i-1})^T \Sigma^{-1} (x_i - x_{i-1}) \quad (4.28)$$

é utilizada, onde Σ é a matriz de covariância amostral de todas as observações X .

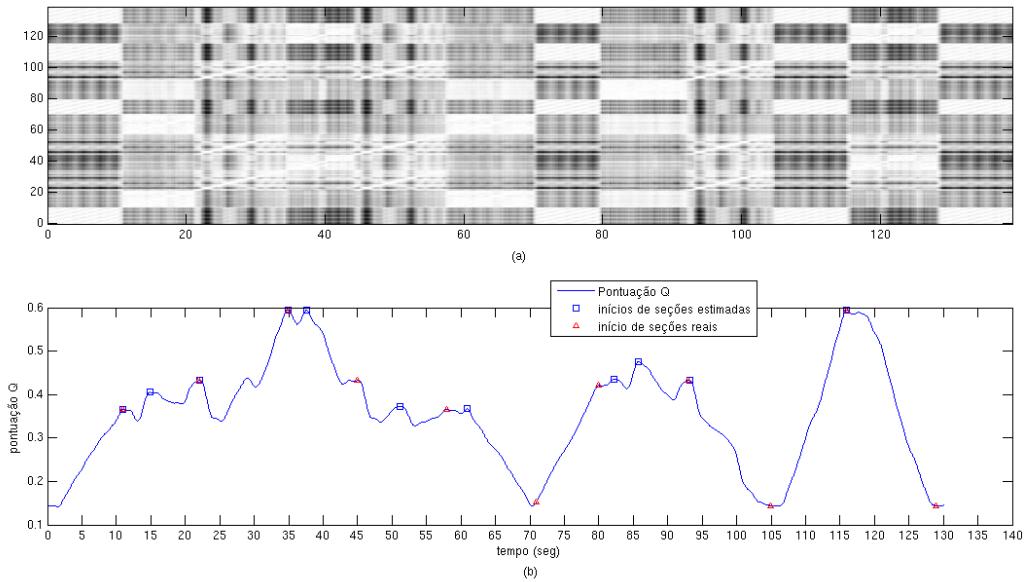


Figura 4.9: (a) Matriz de similaridade. (b) Pontos de segmentação estimados através dos picos da pontuação Q de Cooper e Foote (2002)

2. Calcula a derivada $\frac{d\Delta_t}{dt}$ da distância, que tem valores baixos para variações pequenas de timbres musicais, e valores altos para mudanças repentinas, e os picos corresponderiam a mudanças de seções.
3. Os picos são obtidos utilizando simples heurísticas, como por exemplo, uma parâmetro de duração mínima para as seções mínima pode ser utilizado para evitar seções muito pequenas.

Em nossos experimentos, os rótulos são encontrados utilizando uma das técnicas de separabilidade entre classes (seção 4.1).

4.4.4 Segmentação por Dissimilaridade e Processamento de Imagem*

Depois de calcular a matriz de dissimilaridade da sequência de observações, é possível considerá-la uma imagem em tons de cinza. Para nós humanos, é relativamente fácil olhar para uma imagem da matriz de dissimilaridade e enxergar os pontos onde ocorrem mudanças no sinal, porém, mesmo a olho nu, o ponto exato não é bem definido quando a mesma é calculada com descritores gerados com afrouxamento no tempo. Mais adiante veremos como se comporta a matriz de dissimilaridade após filtrá-la com o limiar de Otsu (1975) e aplicar operadores morfológicos - técnicas conhecidas em Processamento de Imagem - para o problema de segmentação não supervisionada.

Matriz de dissimilaridade binária. Considerando a matriz de dissimilaridade como uma imagem em tons de cinza, pode-se aplicar filtros⁹ específicos para torná-la binária, o que permite identificar mais facilmente os pontos de mudança na matriz. Para exemplificar, veja na figura 4.10 uma matriz de dissimilaridade, que foi calculada com a distância do cosseno, de observações que contém descritores gerados dos momentos estatísticos do MFCC e memória temporal de 3 segundos. Na figura, os pontos pretos indicam dissimilaridade, e os pontos brancos indicam dissimilaridade, e assim, podemos ver que próximo aos 20 segundos ocorre uma mudança de textura, e o mesmo ocorre próximo aos 60 segundos. Uma representação binária desta matriz ajudaria a identificar estas regiões de mudança. Desta forma, foram aplicados os seguintes filtros:

*filtros aqui se refere a qualquer procedimento que altere a imagem

- Limiar de Otsu
- Operador morfológico de Dilatação
- Operador morfológico para Ligar Pontos
- Operador morfológico para Estreitar objetos

A justificativa para utilizar tais técnicas de processamento de imagem estão relacionadas com aquelas levantadas na seção 4.4.1, onde identificamos as formas geométricas na matriz de dissimilaridade, sendo estas as pistas para encontrar as mudanças entre seções.

A figura 4.11 é um exemplo de como fica a matriz depois de ser processada, com os pontos de mudança de seção identificados, que são aqueles em que o algoritmo 11 de segmentação detectou onde ocorrem mudanças de textura.

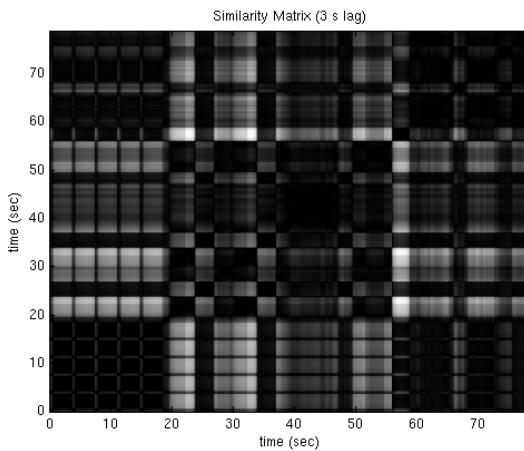


Figura 4.10: Matriz de dissimilaridade com distância de cosseno dos descritores dinâmicos

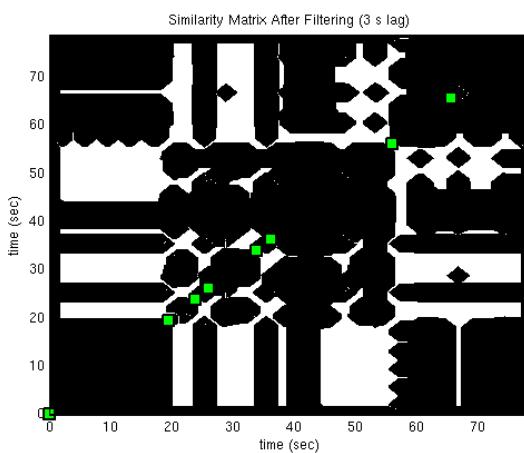


Figura 4.11: Matriz de dissimilaridade binária após técnicas de processamento de imagem com os pontos de mudança de seção.

Neste exemplo, o algoritmo encontrou 3 segmentos a mais do que o desejado (de acordo com o gabarito), mas alguns deles próximos aos pontos reais. A tabela 4.2 mostra os pontos de mudança detectados manualmente e a tabela 4.3 mostra os pontos de mudança estimados.

Recorrer a uma técnica de processamento de imagens para resolver o problema da dissimilaridade parece um caminho natural, mas o problema é que muita informação se perde durante o procedimento, e além disto, os parâmetros dos operadores morfológicos parecem ser específicos para cada caso. Apesar dos resultados iniciais não serem satisfatórios, acreditamos que este

Algoritmo 11 Calcula os pontos de mudança de seção a partir da matriz binária
 $P = \text{SEG_MATBIN}(M, \nu)$

Entradas:

M ; Matriz de dissimilaridade binária $N \times N$
 ν ; tamanho mínimo de uma seção a ser identificada

Saída: P ; pontos de mudanças entre as seções

```

1:  $p \leftarrow 1$ 
2:  $P \leftarrow \{\}$ 
3: enquanto  $p \leq N$  faz
4:   se  $M_{p,p} = 1$  então
5:      $aux \leftarrow p$ 
6:     enquanto  $M_{p,p} = 1$  e  $p \leq N$  faz
7:        $p \leftarrow p + 1$ 
8:     fim enquanto
9:      $P \leftarrow P \cup \{|p - aux)\| \}$ 
10:     $aux \leftarrow \nu$  ; avança ponteiro para evitar segmentos muito próximos
11:    enquanto  $aux > 0$  faz
12:       $p \leftarrow p + 1$ 
13:       $aux \leftarrow aux - 1$ 
14:    fim enquanto
15:  fim se
16: fim enquanto

```

20	41	50	59
----	----	----	----

Tabela 4.2: Pontos de mudança de textura reais, detectados manualmente (em segundos)

é um caminho promissor e que um maior esforço utilizando tais técnicas pode gerar resultados interessantes.

A última etapa do algoritmo deve rotular os segmentos encontrados. Para isto, consideramos a técnica de separabilidade entre classes abordada na seção 4.1. Em seguida é realizado um agrupamento por K-Médias, sendo necessário que se tenha o número total de seções.

4.4.5 Segmentação em Multi-passos - Peeters *et al.* (2002b)

Para Peeters *et al.* (2002b), o sumário musical é baseado na representação da música como uma sucessão de estados, tal que cada estado representa, de alguma maneira, uma informação similar encontrada em diferentes partes da música. A justificativa para utilizar uma técnica de multi-passos é que, assim como em vídeos, os humanos desempenham uma segmentação e agrupamento melhor depois de ouvirem (ou verem) a mídia em uma segunda vez.

- A primeira escuta permite a detecção de variações na música sem um conhecimento prévio se uma parte específica se repetirá mais tarde.
- A segunda escuta permite encontrar as estruturas da música utilizando os modelos mentalmente criados na primeira escuta. No algoritmo proposto, esta etapa se divide em três estágios: (1) os modelos são comparados de forma a reduzir as redundâncias; (2) o conjunto

19.44	23.84	26.12	33.96	36.23	56.09	65.74
-------	-------	-------	-------	-------	-------	-------

Tabela 4.3: Pontos de mudança de textura estimados (em segundos)

de modelos reduzidos é utilizado na inicialização do K-médias (ver subseção 4.2.1) e, portanto, deve-se conhecer o número de estados; e (3) a saída do algoritmo de agrupamento é utilizado no treinamento de um modelo oculto de Markov (HMM). Por fim, uma representação da sequência de estados da música é obtida através da aplicação do algoritmo de Viterbi no modelo HMM.

Veja na figura 4.12 uma representação geral do algoritmo.

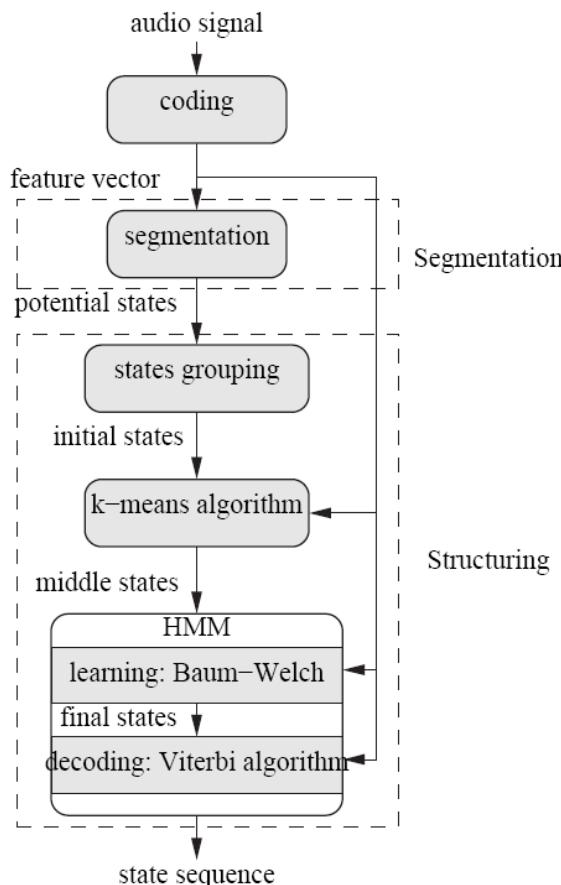


Figura 4.12: Fluxo do algoritmo de segmentação em multi-passos proposto por Peeters et al. (2002b), que se resume em encontrar os estados potenciais através da matriz de similaridade; encontrar os estados iniciais pela redução dos estados encontrando aqueles que são redundantes; encontrar os estados intermediários pelo agrupamento das observações encontradas em cada estado; e por fim, considerar os estados intermediários em um modelo de HMM, onde o tempo é levado em consideração.

O primeiro passo do algoritmo proposto faz uma primeira segmentação baseada na matriz de dissimilaridade (ver subseção anterior 4.4.1), utilizando um limiar fixo $\alpha = 0.99$, para construir os q estados potenciais, onde cada estado potencial $p_i = \mu_i$, $i = 1, \dots, q$, ou seja, p_i é representado pela valor médio de todas as observações circunscritas ao segmento encontrado.

O segundo passo do algoritmo opera em três etapas:

1. **Reducir os estados potenciais.** O objetivo é fornecer estados iniciais que não sejam redundantes, e para isto, é realizada a redução dos estados potenciais pelo agrupamento de estados que são aproximadamente idênticos (similaridade ≥ 0.99). Depois do agrupamento, o número de estados (ou seções) é K , e estes são os estados iniciais da solução. Com esta etapa, é fornecido um número estimado de seções, o que permitirá uma melhor inicialização do agrupamento na próxima etapa.
2. **Agrupamento por K-Médias.** Cada estado inicial é representado por $t_i = \mu_i$, $i = 1, \dots, K$, e estes valores são utilizados na inicialização do agrupamento por K-Médias. Isto equivale

a dizer que a linha 1 do algoritmo 7 seria alterada para receber como entrada pontos médios definidos por cada estado inicial. O resultado são os estados intermediários m_i que podem ser modelados como uma distribuição normal $m_i = \mathcal{N}(\mu_i, \Sigma_i)$, e que servirão como entrada para a próxima etapa.

3. **Introduzir restrições temporais.** Até agora a informação temporal não foi levada em consideração, pois o algoritmo de agrupamento K-Médias somente associa cada observação isoladamente a uma classe. Assim, são utilizados modelos ocultos de Markov (HMM), onde os estados encontrados pelo algoritmo de K-Médias são utilizados no treinamento do modelo. Cada estado é modelado como uma normal multivariada, e tem seus parâmetros iniciais igual a m_i . O modelo oculto de Markov é depois treinado pelo algoritmo de Baum-Welch onde são estimados os parâmetros do modelo $\lambda = (A, B, \pi)$ (ver eq. 4.23). A sequência de estados correspondente à peça musical é obtida através da decodificação utilizando o algoritmo de Viterbi, dado o modelo λ e a sequência de observações X .

O algoritmo proposto acima abre espaço para várias outras técnicas de segmentação em multi-passos. Entretanto, ao analisar e executar o algoritmo, percebemos que o mesmo poderia ser melhorado em alguns pontos. Primeiramente, em relação ao primeiro passo, como já dissemos anteriormente, os melhores resultados foram obtidos com uma estratégia de encontrar os picos de dissimilaridade, e não através de um limiar fixo (ver subseção 4.4.1).

Outro ponto se refere ao algoritmo de agrupamento de estados potenciais. O texto de Peeters *et al.* (2002b) não deixa claro explícito o modo como o agrupamento foi realizado. A primeira dúvida que se poderia levantar é se foi construída uma matriz de dissimilaridade para distinguir quais são os trechos mais ou menos similares. Outra questão é se somente estados conjuntos (no tempo) foram agrupados ou se a similaridade deve agrupar todos os estados independentemente da questão temporal. Caso a última seja a resposta correta, seria necessário recalcular a similaridade do novo estado em relação a todos os outros estados. Como estes pontos não foram descritos nos textos, propomos implementar diferentes versões do algoritmo de Segmentação em Multi-Passos de Peeters *et al.* (2002b) (MPS-PBR ou *Multi Pass Segmentation by Peeters, Burth and Rødet*), com configurações diferentes, sendo que no primeiro passo, para todas as configurações, utilizamos a localização dos segmentos potenciais utilizando o algoritmo 10.

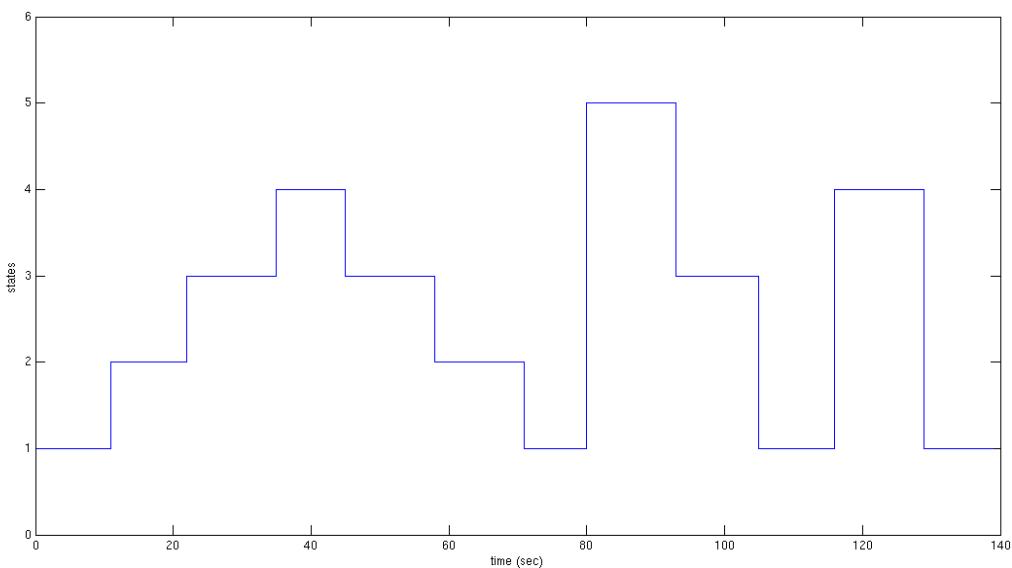


Figura 4.13: Segmentação manual de um sinal musical

- **MPS-PBR-1.** No segundo passo, para efetuar o agrupamento dos estados potenciais, utiliza, além da média μ , a informação da covariância Σ das observações de cada estado. A distância de Mahalanobis (eq. 4.7) é utilizada para calcular a distância entre todos os pares de estados. Após o cálculo das distâncias, o agrupamento é realizado considerando um limiar fixo α para determinar se um estado pode ser agrupado com outro, e isto é feito entre todos os estados, e não somente aqueles que estão justapostos na sequência musical. Neste algoritmo, não é feito um novo cálculo de distâncias para cada novo grupo criado, e para contornar este problema, sempre que o algoritmo for verificar se um estado pode ser agrupado com um grupo de estados já formado, o limiar α é utilizado para cada estado do grupo. Caso o limiar seja menor que a distância entre qualquer um dos estados do grupo, então o estado não é agrupado, e, caso contrário, ele é incluído no grupo. Para maiores detalhes, veja o algoritmo 12.
- **MPS-PBR-2.** No segundo passo, as distâncias entre cada estado são calculadas através da distância do cosseno, e, para cada estado conjunto encontrado no tempo, verifica-se se sua distância é menor que um dado limiar α . Enquanto a distância for menor que α então aqueles estados fazem parte de um mesmo grupo.
- **MPS-PBR-3.** Funciona da mesma maneira que a versão **MPS-PBR-1**, no entanto, ao invés de utilizar a distância de Mahalanobis, utiliza a distância de cosseno (eq. 4.8).
- **MPS-PBR-4.** O agrupamento dos estados potenciais é realizado através do algoritmo de K-Médias, sendo o número final de estados igual ao número real de segmentos identificados manualmente, e cada estado é representado pelo vetor μ das observações do estado. Com esta versão no algoritmo, poderemos avaliar se a hipótese, de que saber de antemão o número de estados reais melhoraria a segmentação, é verdadeira ou não.

Tabela 4.4: Configurações para o segmentador MPS-PBR

Para comparar como estas variações do algoritmo de Peeters *et al.* (2002b), alguns testes foram feitos em um sinal de áudio, cuja segmentação manual é exibida na figura 4.13. Para comparação, o resultado da execução de MPS-PBR-1 pode ser visualizado nas figuras da tabela 4.14. Pelo exemplo podemos ver que a solução de redução de estados não se aproxima do desejável (5 estados), mas é razoavelmente eficiente, reduzindo de 29 para 13 estados. Dentre as outras variações, esta fornece a terceira melhor segmentação. O exemplo da execução da variação MPS-PBR-2 pode ser visualizado nas figuras da tabela 4.15. Dentre as outras variações do algoritmo, esta é a que fornece a pior segmentação, reduzindo de 29 para 24 estados. A terceira variação, MPS-PBR-3, pode ser observada nas figuras da tabela 4.16. Esta variação fornece a segunda melhor segmentação, e foi a que mais conseguiu se aproximar da segmentação manual, com uma redução de 29 para 9 estados. Por último, nas figuras da tabela 4.17, vemos o resultado da execução de MPS-PBR-4, quando temos a informação do número total de segmentos, e esta é a que fornece a melhor segmentação, confirmando a hipótese levantada anteriormente.

4.4.6 Segmentação em Multi-passos com HMM de Misturas de Gaussianas*

Este método de segmentação é baseado no algoritmo multi-passos proposto por Peeters *et al.* (2002b). A principal motivação desta proposta é que gostaríamos de modelar os estados (potenciais, iniciais e finais) com distribuições normais, e, quando possível, com misturas de gaussianas. Por isto, propomos uma segmentação baseada em multi-passos com HMM e misturas de gaussianas (MPS-GHMM) para representar os estados. Depois de encontrar os pontos de mudança de

Algoritmo 12 Agrupa os estados potenciais (MPS-PBR-1)

$$Y = \text{AGRUPA_ESTADOS_POTENCIAIS}(P, \Sigma, \alpha)$$

Entradas:

$P = p_i, \quad p_i = \mu_i \in \mathbb{R}^d, \quad i = 1, \dots, N$; *Estados potenciais*
 $\Sigma = \Sigma_i, \quad i = 1, \dots, N$; *Matriz de covariância dos estados potenciais*
 α ; *Límiar para agrupamento*

Saída: $Y = y_i, \quad y_i = \mu_i, \quad i = 1, \dots, K, \quad K < N$; *Estados iniciais*

- 1: $M_{i,j} \leftarrow (p_j - p_i)^T \Sigma_i^{-1} (p_j - p_i), \quad \forall i, j = 1, \dots, N, \quad i \neq j, \quad i < j$
 - 2: $G \leftarrow \{\}$; *armazena os agrupamentos*
 - 3: $K \leftarrow 0$
 - 4: **para** todo par i, j de M ordenado em ordem decrescente de similaridade e tal que $M_{i,j} \geq \alpha$ **faça**
 - 5: **se** $\exists g_k$ tal que $p_i \in g_k, g_k \in G$ e $M_{r,j} > \alpha, \forall r \in g_k$ **então**
 - 6: $g_k \leftarrow g_k \cup \{p_j\}$
 - 7: **fim se**
 - 8: **se** $\exists g_k$ tal que $p_j \in g_k, g_k \in G$ e $M_{r,i} > \alpha, \forall r \in g_k$ **então**
 - 9: $g_k \leftarrow g_k \cup \{p_i\}$
 - 10: **fim se**
 - 11: **se** $p_i \notin g_k$ e $p_j \notin g_k, \forall g_k \in G$ **então**
 - 12: $K \leftarrow K + 1$
 - 13: $g_K \leftarrow \{p_i, p_j\}$
 - 14: **fim se**
 - 15: $G \leftarrow G \cup \{g_k\}$
 - 16: **fim para**
 - 17: **para** todo estado i que não foi agrupado **faça**
 - 18: $K \leftarrow K + 1$
 - 19: $g_K \leftarrow \{p_i\}$
 - 20: **fim para**
 - 21: $Y = \{\}$
 - 22: **para** $i \in G$ **faça**
 - 23: $y_i = \bar{g}_i$; *Cada estado inicial é a média dos estados agrupados*
 - 24: **fim para**
-

seção, os estados potenciais são modelados como uma distribuição normal, $p_i = \mathcal{N}(\mu, \Sigma)$, através da estimação de máxima verossimilhança dos dados circunscritos a cada seção. A consequência imediada é que o agrupamento dos estados potenciais, para evitar estados redundantes, é feito utilizando uma medida de similaridade que compare modelos normais, como a distância de Bhattacharyya (ver seção 4.1.2). Outra diferença com o algoritmo anterior é que a etapa de encontrar os estados intermediários foi excluída, pois não seria possível utilizar o algoritmo de K-Médias que considerasse uma inicialização com os vetores da média μ e as matrizes de covariância Σ . O algoritmo proposto seria então delineado da seguinte forma. No primeiro passo, são construídos os estados potenciais:

- **Encontrar os estados potenciais.** Como vimos nas seções anteriores, existem alguns métodos para determinar os pontos de mudança do sinal musical e, como ponto de partida, utilizaremos três diferentes métodos para encontrar os pontos de mudança: (1) por dissimilaridade e limiar (seção 4.4.1); (2) por dissimilaridade e processamento de imagens (seção 4.4.4); e (3), por segmentação de Tzanetakis e Cook (1999) (seção 4.4.3). Estes métodos fornecem os pontos de mudança de regime e, para construir os estados potenciais, identificamos as observações que estão dentro dos limites de cada seção, digamos $X_a \subset X$ tal que $X_a = x_{a_1}, \dots, x_{a_n}$, e estimamos o modelo do estado potencial $p_a = \mathcal{N}(\mu_a, \Sigma_a)$ através do estimador de máxima verossimilhança das observações X_a .

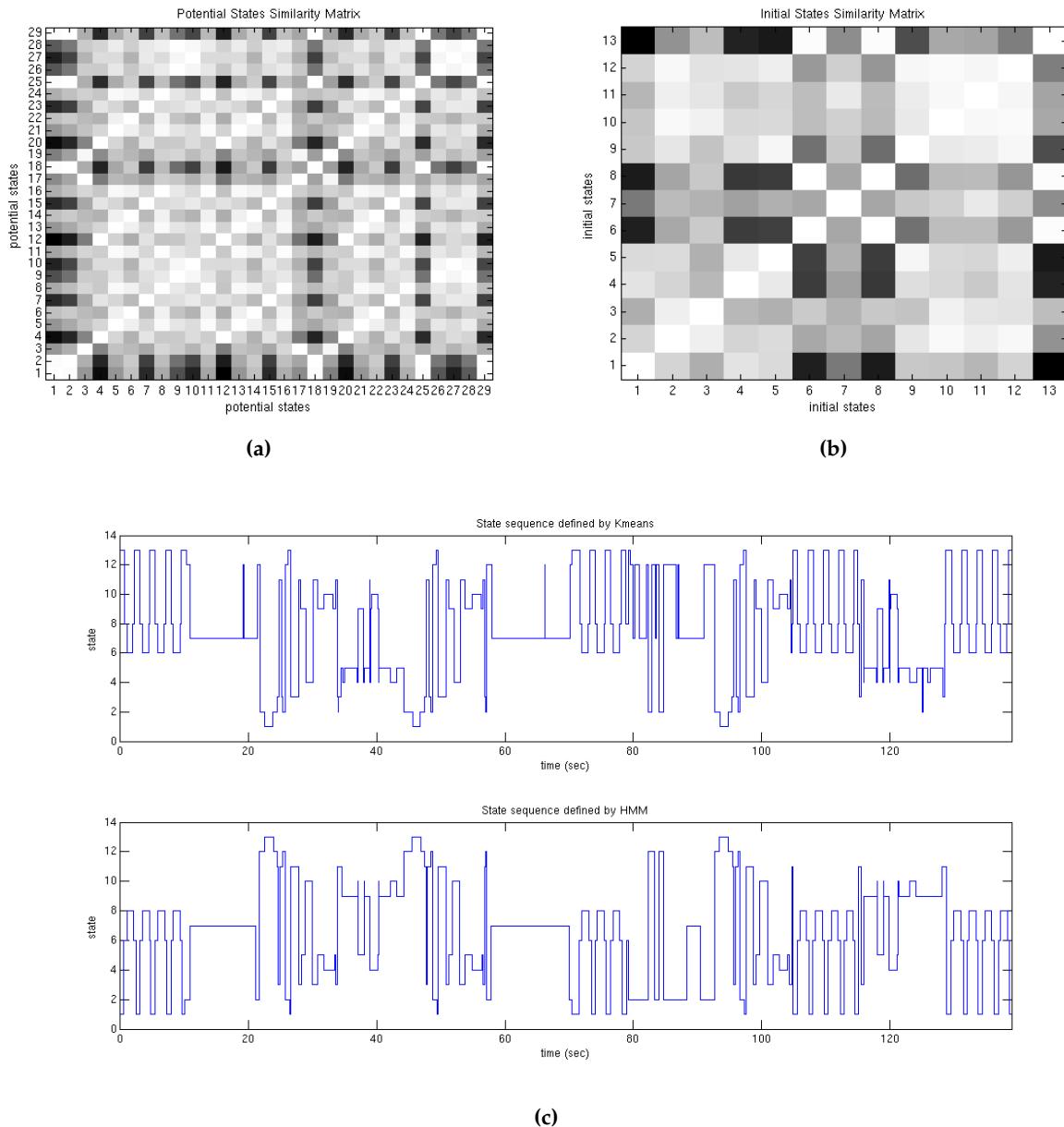


Figura 4.14: Resultado de segmentação com algoritmo **MPS-PBR-1**. Estados potenciais (superior-esquerdo); estados iniciais (superior direito); sequência de estados definida por K-Médias (central); e sequência de estados definida por HMM (inferior)

O segundo passo do algoritmo é dividido em duas etapas:

1. **Reducir os estados potenciais.** Esta etapa é realizada de forma similar à apresentada no algoritmo 12. Primeiramente, as distâncias entre os estados potenciais são dadas pela distância de Bhattacharyya (ver seção 4.1.2)¹⁰. Em segundo lugar, ao invés de calcular os estados iniciais através da média das centroides de cada estado potencial (como era o caso anterior), são construídos modelos de misturas de gaussianas através dos dados de todos os estados potenciais agrupados. Em outras palavras, se existe um grupo $g_i = p_a, p_b$, então as misturas gaussianas serão construídas através dos dados X_a e X_b pelo algoritmo de Maximização da Esperança (EM ou *Expectation Maximization*). Ao final desta etapa, são construídos K estados iniciais modelados como misturas de gaussianas.

¹⁰Uma consequência imediata disto é que o método pressupõe, a partir deste momento, que os dados tenham uma distribuição normal.

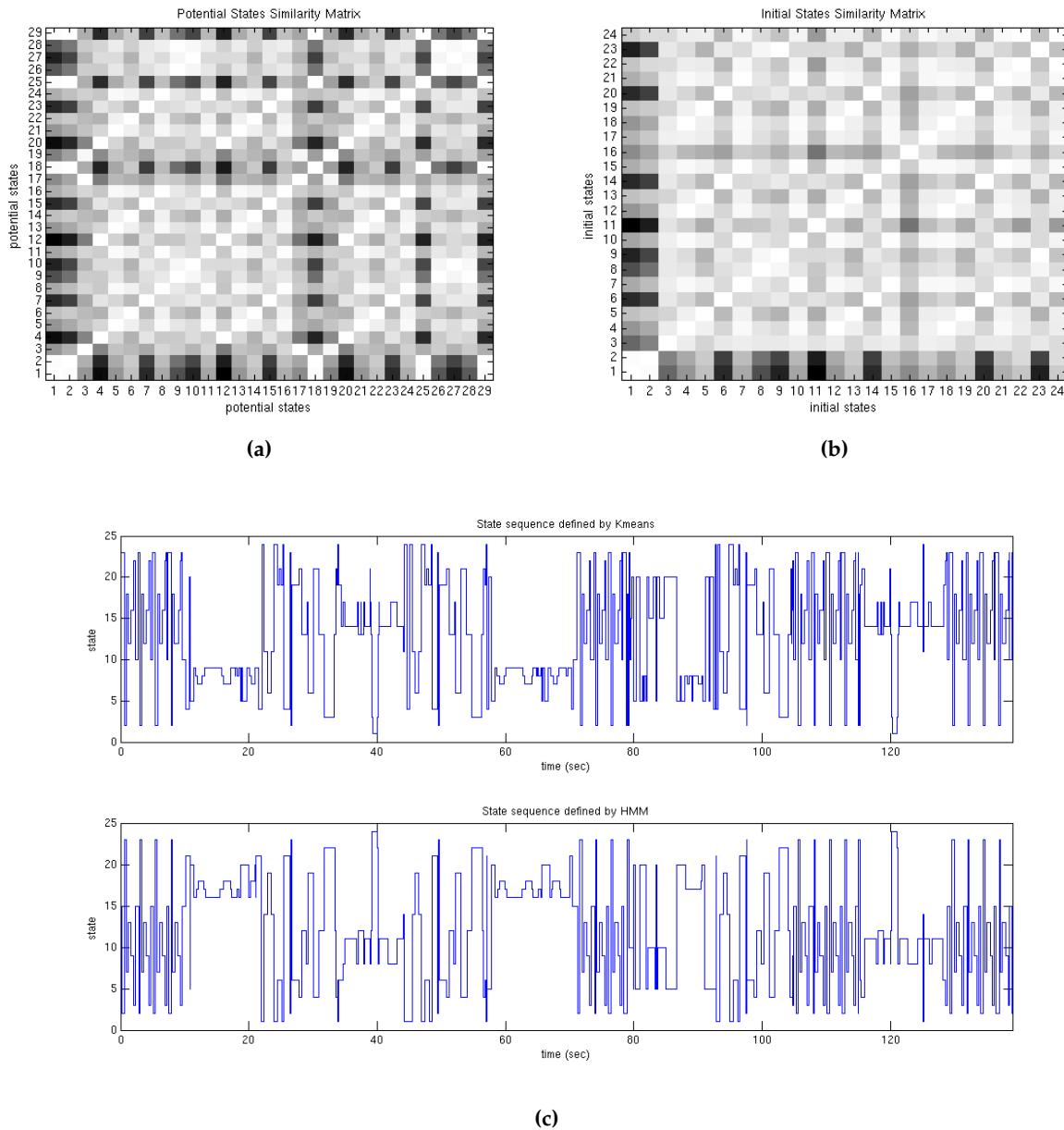


Figura 4.15: Resultado de segmentação com algoritmo **MPS-PBR-2**. Estados potenciais (superior-esquerdo); estados iniciais (superior direito); sequência de estados definida por K-Médias (central); e sequência de estados definida por HMM (inferior)

2. **Introduzir restrições temporais.** De forma similar ao algoritmo de Peeters *et al.* (2002b), os estados finais são construídos através da inicialização de um modelo escondido de Markov com uma topologia ergódica, porém com a diferença que cada estado é modelado como uma mistura de gaussianas. O modelo oculto de Markov é depois treinado pelo algoritmo de Baum-Welch onde são re-estimados os parâmetros do modelo $\lambda = (A, B, \pi)$, e a sequência de estados correspondente à peça musical é obtida através da decodificação utilizando o algoritmo de Viterbi, dado o modelo λ e a sequência de observações X .

Dadas as diferentes configurações do primeiro passo, denotamos o método de três diferentes formas:

- **MPS-GHMM-1.** Passo 1 realizado via Tzanetakis e Cook (1999).
- **MPS-GHMM-2.** Passo 1 realizado via dissimilaridade e processamento de imagens (seção 4.4.4).

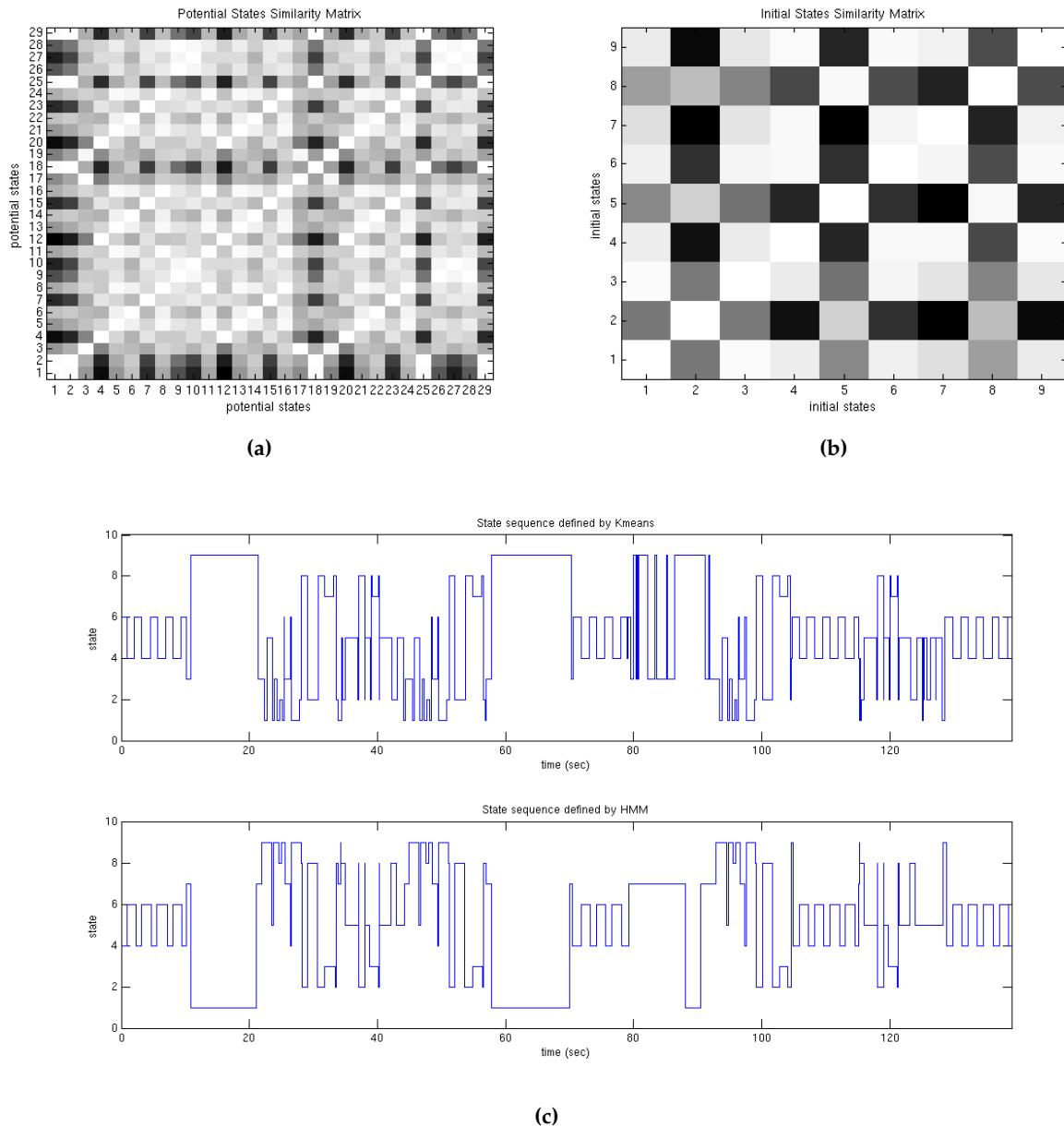


Figura 4.16: Resultado de segmentação com algoritmo **MPS-PBR-3**. Estados potenciais (superior-esquerdo); estados iniciais (superior direito); sequência de estados definida por K-Médias (central); e sequência de estados definida por HMM (inferior)

- **MPS-GHMM-3.** Passo 1 realizado via dissimilaridade e limiar (seção 4.4.1).

Dada a mesma música de referência da figura 4.13, os exemplos da segmentação utilizando os métodos acima podem ser visualizados nas figuras das tabelas 4.18, 4.19 e 4.20.

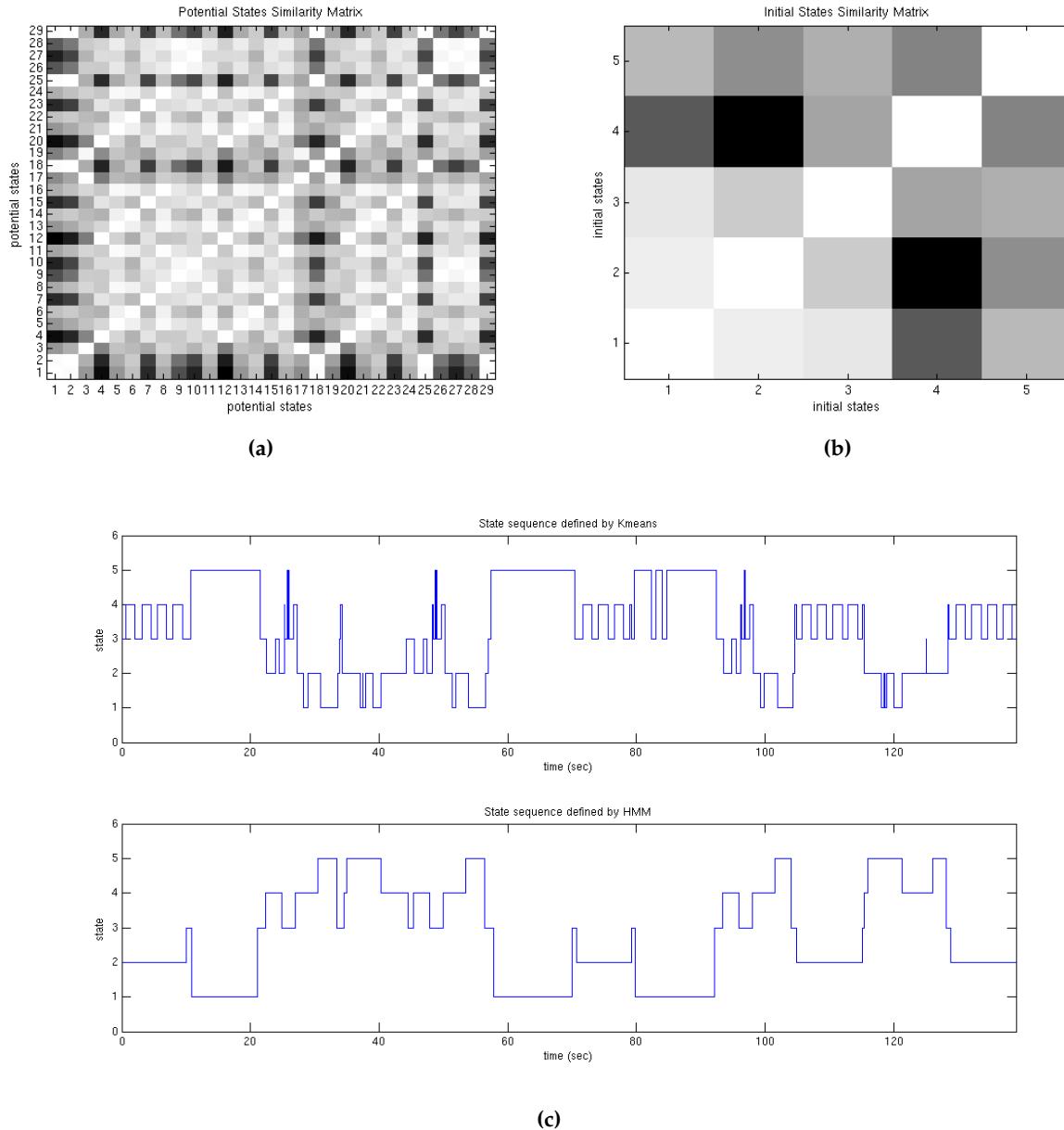


Figura 4.17: Resultado de segmentação com algoritmo **MPS-PBR-4**. Estados potenciais (superior-esquerdo); estados iniciais (superior direito); sequência de estados definida por K-Médias (central); e sequência de estados definida por HMM (inferior)

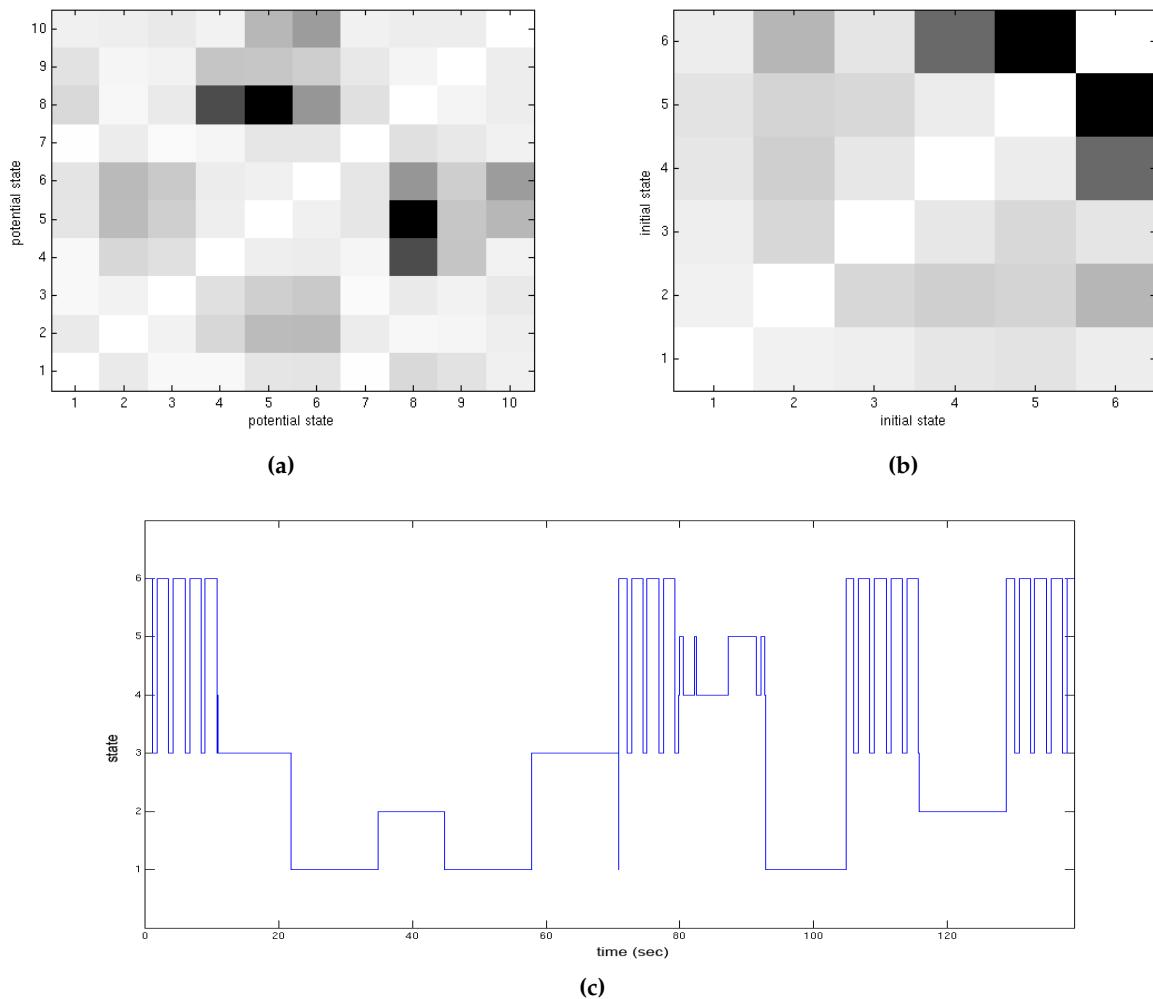


Figura 4.18: Resultado de segmentação com algoritmo **MPS-GHMM-1**. Estados potenciais (superior-esquerdo); estados iniciais (superior direito); e sequência de estados definida por HMM (inferior)

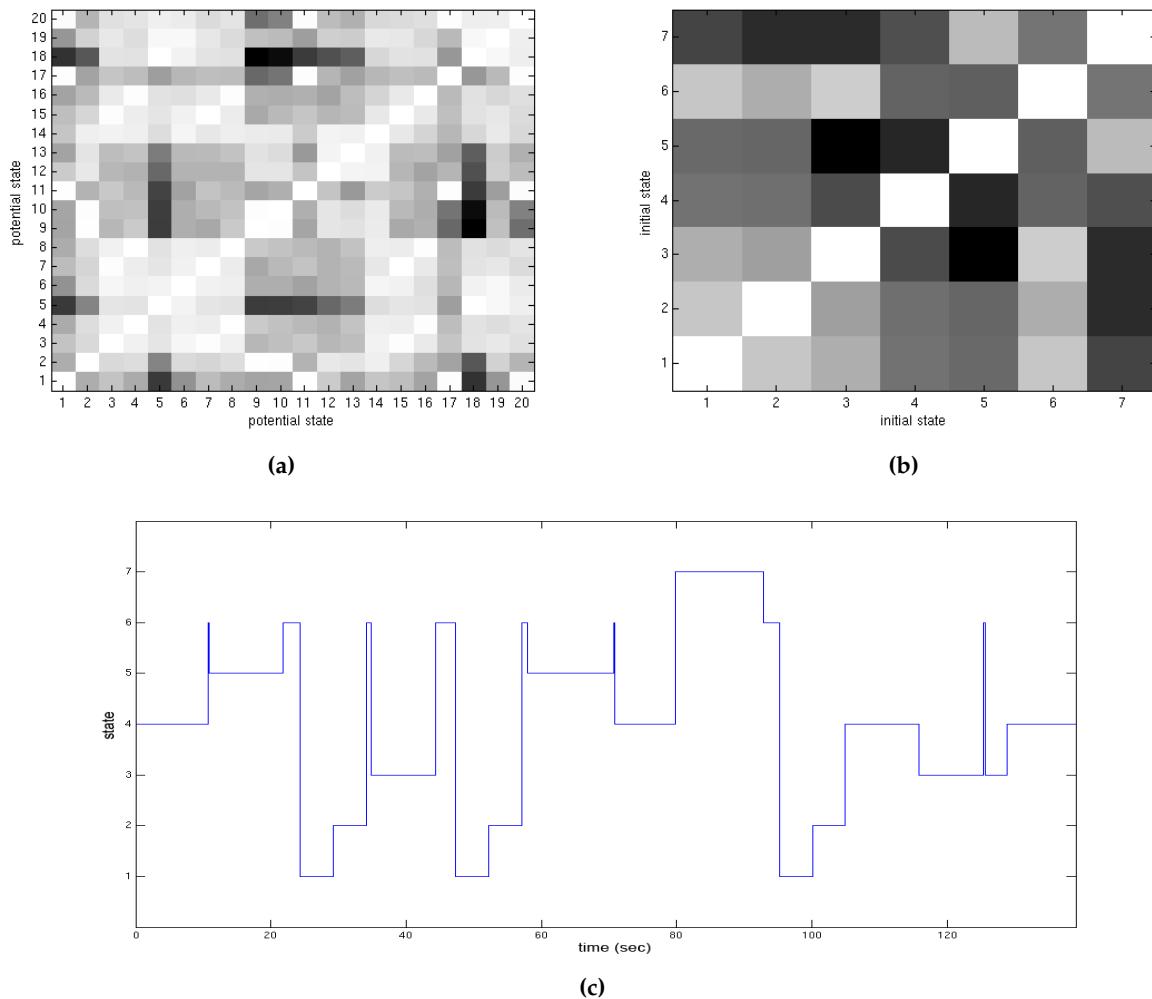


Figura 4.19: Resultado de segmentação com algoritmo **MPS-GHMM-2**. Estados potenciais (superior-esquerdo); estados iniciais (superior direito); e sequência de estados definida por HMM (inferior)

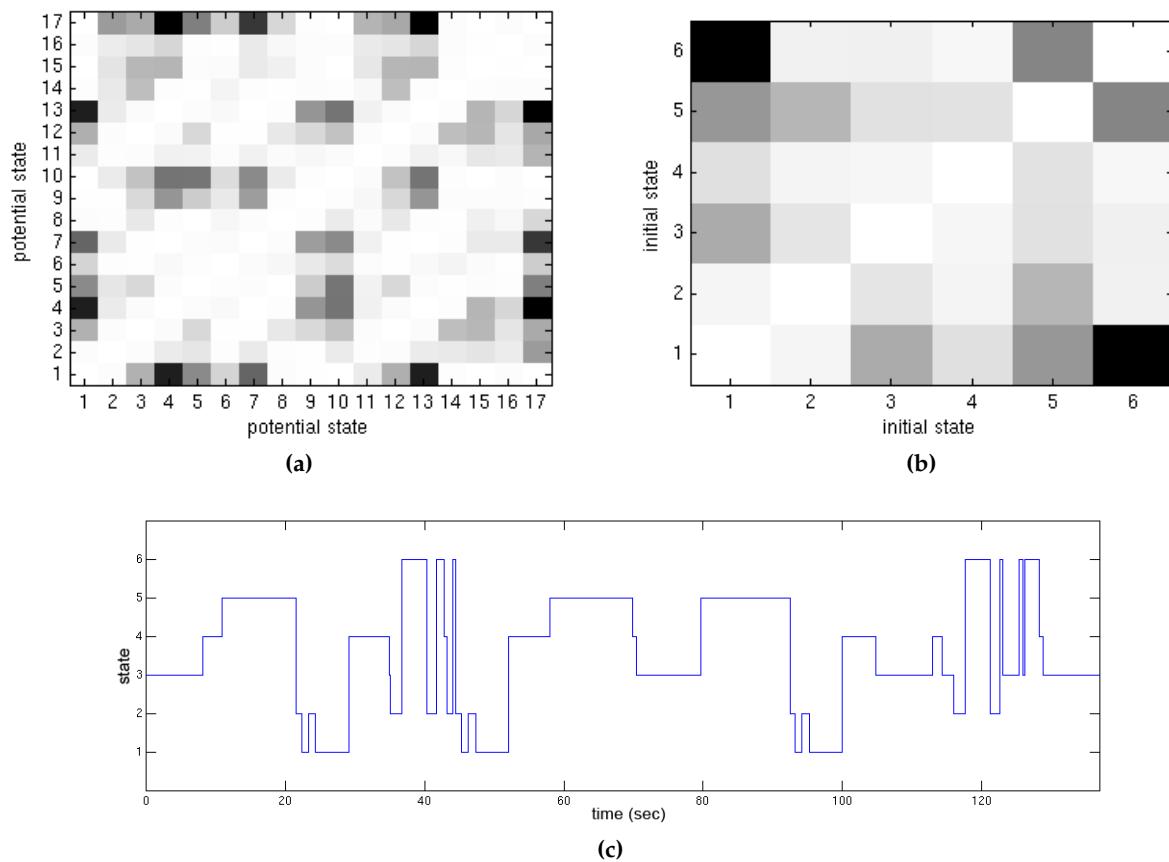


Figura 4.20: Resultado de segmentação com algoritmo **MPS-GHMM-3**. Estados potenciais (superior-esquerdo); estados iniciais (superior direito); e sequência de estados definida por HMM (inferior)

4.5 Segmentação Musical Não-supervisionada em Tempo Real

A segmentação não supervisionada em tempo real deve partir de dois princípios: (1) não existem dados de treinamento disponíveis e (2) em tempo de execução do algoritmo, os dados chegam continuamente e, portanto, não se sabe de antemão o instante em que a música termina, ou mesmo quando ela começou. Neste contexto, podemos trabalhar com ainda outros conceitos além daqueles que tínhamos trabalhado até o momento, como, por exemplo, a memória musical humana e como seu funcionamento poderia auxiliar em um processo de segmentação musical. Entretanto, este seria um estudo que, neste instante, fugiria do escopo inicial de nosso trabalho.

Uma questão importante na segmentação em tempo real é a dificuldade em determinar o momento correto de rotular as seções segmentadas. Inicialmente podemos imaginar que o momento correto é quando a música acaba. Entretanto, esta não é uma informação válida neste contexto. Outra ideia é realizar a rotulação sempre que houver uma ação externa de um suposto usuário, que queira entender como a estrutura musical evoluiu até um certo instante, e esta parece ser a estratégia mais adequada para um sistema desenhado com o fim de auxiliar o usuário na tarefa de segmentação em tempo real. O maior problema é que os dados chegam continuamente, e estes alteram a todo o momento o contorno geral da segmentação musical, e até que a música “acabe”, não é possível determinar todas as seções da música – mesmo que exista um sistema que encontre todas as seções musicas sem erro algum –, e “acabar”, neste contexto, é totalmente subjetivo. Para eliminar este problema e poder realizar todas as avaliações do algoritmo, consideraremos que toda música, segmentada em tempo real, terá pelo menos uma simulação da ação do usuário, que, por definição, ocorre após a entrada da última observação gerada.

O ponto de partida para iniciar a exposição dos algoritmos de segmentação não supervisionada em tempo real, é apresentar a casca do algoritmo, que poderia ser utilizado por qualquer uma das técnicas. Suponha que I seja um canal de entrada de descritores $x \in \mathbb{R}^d$, ou seja, existe outro processo anterior que transforma o sinal de áudio em um vetor de descritores previamente selecionados.

O algoritmo 13 serve para ilustrar como um algoritmo que lê continuamente um sinal deve operar, e quais são os principais parâmetros desta solução: o tamanho mínimo da janela a ser analisada, w_{\min} , que serve para evitar encontrar segmentos muito pequenos; o tamanho máximo da janela a ser analisada, w_{\max} , que serve para não esperar que se chegue a uma janela muito grande do sinal a considerar observações muito distantes das atuais, o que pode aumentar a diferença entre as seções, mas diminuiria o número de seções potenciais; e o tamanho mínimo de um segmento a ser identificado, s_{\min} , tal que $s_{\min} < w_{\min}$, que também evita que encontramos seções muito pequenas, mas neste momento, depois que encontramos o ponto de mudança na sequência de dados sendo analisada. A diferença entre w_{\min} e s_{\min} é que a primeira delimita o tamanho mínimo da janela a ser analisada, e a segunda delimita o tamanho mínimo de um segmento a ser que pode ser encontrado dentro de uma janela de análise.

Nos algoritmos apresentados nesta seção, quando nos referirmos a uma janela de análise de segmentação, estamos nos referindo à janela construída ao longo da leitura das observações, que depende dos parâmetros de tamanho mínimo e máximo. De modo geral, o algoritmo funciona da seguinte forma:

- Lê continuamente o vetor de descritores x ;
- acumula os valores de x até que atinja o mínimo da janela de análise;
- encontra os pontos de mudança mais significativos até que não exista mais entradas; e
- como último passo do algoritmo (linha 23), executa a rotulação, onde rotula(S) é responsável pela rotulação de cada segmento encontrado. Como veremos mais adiante, podemos resolver isto de algumas formas, incluindo a informação de separabilidade entre as classes (solução utilizada em algumas técnicas não-supervisionadas da seção anterior). Em outras técnicas aqui apresentadas (4.5.3), veremos que esta etapa pode ser inserida durante a localização do ponto de mudança de regimes.

Algoritmo 13 Algoritmo genérico para segmentação não-supervisionada em tempo real

$[S, C] = \text{SEGMENTACAO_TEMPOREAL}(I, w_{\min}, w_{\max}, s_{\min})$

Entradas:

I ; Canal de entrada para o sinal de áudio
 w_{\min} ; tamanho mínimo de uma janela de
 w_{\max} ; limiar para identificar os pontos de mudança de regime
 s_{\min} ; limiar para identificar os pontos de mudança de regime

Saída: S ; pontos de mudanças entre as seções

C ; sequência de estados dos segmentos já rotulados

```

1:  $S \leftarrow \{\}$ 
2:  $p \leftarrow 1$  ; contador para auxiliar na localização do ponto da mudança
3:  $T \leftarrow \{\}$  ; inicializa o vetor que acumula as observações  $x$ 
4:  $x \leftarrow \text{proxima\_observacao}(I)$ 
5: enquanto ( $x \neq \text{NULL}$  faça)
6:    $T \leftarrow T \cup \{x\}$  ; concatena  $x$  como a última posição de  $T$ 
7:   se  $N + 1 \geq w_{\min}$  então
8:      $r \leftarrow \text{ponto\_mudanca}(T)$  ; encontra o ponto relativo de mudança
9:     se  $r = \text{NULL}$  então
10:    se  $N + 1 > w_{\max}$  então
11:       $T \leftarrow \{t_i : t_i \in T, i \geq w_{\min}\}$  ; descarta as primeiras  $w_{\min}$  observações de  $T$ 
12:    fim se
13:  senão
14:     $a \leftarrow (p - 1) - N + r$  ; calcula o ponto de mudança absoluto
15:     $T \leftarrow \{t_{r+1}, \dots, t_N, x\}$  ; descarta as  $r$  primeiras observações de  $T$ 
16:  fim se
17:   $s_{N_s} \leftarrow \text{recupera o último ponto de mudança de } S$ 
18:  se  $a - s_{N_s} \geq s_{\min}$  então
19:     $S \leftarrow \{s_1, \dots, s_{N_s}, a\}$  ; adiciona o novo ponto de mudança ao vetor
20:  fim se
21:   $p \leftarrow p + 1$ 
22: fim se
23:  $C \leftarrow \text{rotula}(S)$  ; rotula as seções dados os pontos de mudança  $S$ 
24:  $x \leftarrow \text{proxima\_observacao}(I)$ 
25: fim enquanto

```

4.5.1 Critério de Informação Bayesiano e Soma Cumulativa

O critério de informação Bayesiana (BIC ou *Bayesian Information Criterion*) é um método mais conhecido na literatura estatística, tendo pouca ou quase nenhuma utilização em sistemas voltados para a recuperação de informação musical (pelo menos não é de nosso conhecimento que esta técnica tenha sido utilizada em dados puramente musicais). No entanto, encontramos trabalhos (Chen e Gopalakrishnan, 1998; Omar *et al.*, 2005) em que o ponto de mudança a ser detectado era a mudança de um locutor para outro, a entrada ou saída de um locutor para um ambiente (música ou propaganda ou fundo ruidoso), enfim, um problema de *detecção de mudança acústica*.

BIC é baseada na razão do log da verossimilhança entre dois modelos, e estes modelos representam duas hipóteses: ter duas classes ou somente uma classe na sequência de observação. BIC adiciona também uma penalidade para considerar a diferença entre o número de parâmetros de cada modelo. Os parâmetros de cada modelo são estimados utilizando o critério de máxima verossimilhança. Assim, seja $X = x_1, \dots, x_N$ os dados em que desejamos realizar o teste de hipóteses. Desejamos então testar a hipótese do sinal ter sido gerado por uma distribuição $\mathcal{N}(\mu, \Sigma)$

$$H_0 : x_1, \dots, x_N \sim \mathcal{N}(\mu, \Sigma) \quad (4.29)$$

contra a hipótese de existir uma mudança no tempo i , e o sinal ter sido gerado por duas distribuições normais $\mathcal{N}(\mu_1, \Sigma_1)$ e $\mathcal{N}(\mu_2, \Sigma_2)$

$$H_1 : x_1, \dots, x_i \sim \mathcal{N}(\mu_1, \Sigma_1); \quad x_{i+1}, \dots, x_N \sim \mathcal{N}(\mu_2, \Sigma_2). \quad (4.30)$$

A razão de máxima verossimilhança estatística¹¹ é dada por

$$R(i) = N \log |\Sigma| - N_1 \log |\Sigma_1| - N_2 \log |\Sigma_2| \quad (4.31)$$

onde $N_1 = i$ e $N_2 = N - i + 1$ são o número de observações utilizadas para estimar os modelos $\mathcal{N}(\mu_1, \Sigma_1)$ e $\mathcal{N}(\mu_2, \Sigma_2)$, respectivamente e Σ , Σ_1 e Σ_2 são as matrizes de covariância amostral de x_1, \dots, x_N , x_1, \dots, x_i e x_{i+1}, \dots, x_N , respectivamente.

O objetivo desta técnica é comparar dois modelos, um que considera uma modelagem dos dados com duas gaussianas, e outro que considera somente uma gaussiana. Os valores BIC para cada posição i pode ser calculado então como a razão de verossimilhança menos um fator de correção ou penalidade P , dado pelo número de parâmetros do modelo,

$$BIC(i) = R(i) - \lambda P \quad (4.32)$$

e

$$P = \frac{1}{2}(d + \frac{1}{2}d(d + 1)) \log N. \quad (4.33)$$

O modelo de duas gaussianas é favorecido caso a equação 4.32 seja positiva, e, assim, o ponto de mudança de regimes é dado por

$$i = \operatorname{argmax}_i \{BIC(i) | BIC(i) > 0\}.$$

Para a segmentação musical encontramos problemas ao estimar um valor para λ adequado para todos os sinais musicais gerados. Nos nossos experimentos adotamos $\lambda = 6.25$ através de um ajuste empírico. Além disto, é necessário adicionar ainda as restrições relativas ao tamanho mínimo de cada seção s_{\min} , ou seja,

$$i = \operatorname{argmax}_i \{BIC(i) | BIC(i) > 0, 1 + s_{\min} < i < N - s_{\min}\}, \quad (4.34)$$

No algoritmo de Soma Cumulativa (CuSum ou *Cumulative Sum*), as hipóteses H_0 e H_1 são formuladas da seguinte maneira:

$$H_0 = x_1, \dots, x_r \sim \mathcal{N}(\mu_1, \Sigma_1)$$

e

$$H_1 = x_{N-r}, \dots, x_N \sim \mathcal{N}(\mu_2, \Sigma_2),$$

onde os modelos normais são estimados com algumas poucas observações do começo e do fim de X , digamos $x_1, \dots, x_r \rightarrow f_1(x) \sim \mathcal{N}(\mu_1, \Sigma_1)$ e $x_{N-r}, \dots, x_N \rightarrow f_2(x) \sim \mathcal{N}(\mu_2, \Sigma_2)$. A razão do log da verossimilhança é dada por $R_k = l_0^k - l_1^k$, onde

$$l_0 = \sum_{x_i \in f_1} \log f_1(x_i) \quad (4.35)$$

e

$$l_1 = \sum_{x_i \in f_2} \log f_2(x_i). \quad (4.36)$$

A soma cumulativa é dada pela somatória de todas as razões de verossimilhança $R_k, k =$

¹¹Para mais detalhes sobre a razão da máxima verossimilhança, ver apêndice C.1

$1, \dots, N$, e o máximo valor da soma cumulativa é então comparada com o limiar α . O ponto de mudança i é dado por

$$i = \operatorname{argmax}_i \left\{ \sum_{k=1}^N R_k \mid \sum_{k=1}^N R_k > \alpha, 1 \leq k \leq N \right\}. \quad (4.37)$$

A vantagem deste método em comparação com BIC é que não é necessário criar sempre um modelo a cada posição do vetor. Entretanto, existe ainda a dificuldade de encontrar um número ideal de observações r para a construção dos modelos iniciais, e determinar um valor para o limiar α adequado.

A última etapa do algoritmo deve rotular os segmentos encontrados, lembrando que a rotulação pode ser executada a qualquer momento da execução. Para isto, consideramos a técnica de separabilidade entre classes abordada na seção 4.1. Em seguida é realizado um agrupamento por K-Médias, sendo que o número de seções pode ser fornecido ou podemos reduzir o número de estados por um limiar de similaridade, a exemplo do que foi realizado em Peeters *et al.* (2002b).

4.5.2 Segmentação com Hiper-elipses*

É comum encontrar referências na literatura de análise multivariada (Mardia *et al.*, 1979) a ideias geométricas associadas à normal multivariada. Lembremos da equação da normal multivariada:

$$f(x) = \frac{1}{2\pi^{d/2} |\Sigma|^{1/2}} e^{-1/2(x-\mu)^T \Sigma^{-1}(x-\mu)}. \quad (4.38)$$

Notemos que esta função só depende de x através do quadrado da distância de Mahalanobis

$$(x - \mu)^T \Sigma^{-1}(x - \mu).$$

Podemos notar também que ao passo que $(x - \mu)^T \Sigma^{-1}(x - \mu)$ aumenta, a função $f(x)$ diminui, e $(x - \mu)^T \Sigma^{-1}(x - \mu)$ aumenta quanto maior for a distância entre x e μ .

Portanto, a densidade $f(x)$ é constante para todos os valores de X se a distância de Mahalanobis for uma constante, digamos c^2 ,

$$c^2 = (x - \mu)^T \Sigma^{-1}(x - \mu), \quad (4.39)$$

e esta é a equação de uma hiper-elipse com centro em μ .

Sabe-se que se $X \in \mathbb{R}^d$ é um vetor normal de observações aleatórias $X \sim \mathcal{N}(\mu, \Sigma)$, então o quadrado da distância de Mahalanobis tem uma distribuição Chi-quadrado com d graus de liberdade:

$$(x - \mu)^T \Sigma^{-1}(x - \mu) \sim \chi_d^2 \quad (4.40)$$

Assim, se calcularmos o quadrado da distância de Mahalanobis e igualarmos com uma distribuição χ_d^2 com d graus de liberdade, e avaliar com um nível α , então a probabilidade da observação X cair dentro da hiper-elipse é

$$P((x - \mu)^T \Sigma^{-1}(x - \mu) \leq \chi_d^2) = 1 - \alpha. \quad (4.41)$$

Para o objetivo de encontrar pontos de segmentação, podemos nos perguntar qual é a probabilidade de uma observação aleatória estar dentro desta hiper-elipse, e caso seja uma probabilidade menor que $1 - \alpha$, gostaríamos de marcá-la como uma mudança de regime. O método pode ser descrito nas seguintes etapas. Seja $X = x_1, \dots, x_i, \dots, x_N : x_i \in \mathbb{R}^d$ a sequência de observações de uma janela de análise de segmentação, e $\alpha = [0, 1]$ o nível para avaliação da *fdp* χ^2 .

1. Estima μ e Σ através das r primeiras observações $\{x_1, \dots, x_r\}$.

2. Para todo $x_i \in X$, calcula o quadrado da distância de Mahalanobis c_i^2 .
3. $t = \operatorname{argmax}_i(c_i^2)$, $i = 1, \dots, N$
4. Calcula a probabilidade p_t para c_t^2 dada a distribuição \mathcal{X}_d^2 com d graus de liberdade.
5. Se $p_t < 1 - \alpha$, então a posição no tempo t é candidata a ponto de mudança.
6. O ponto t é um ponto de mudança se (1) x_t está a uma distância maior que um certo limite, que serve para garantir que o segmentador não pegue pontos muito próximos à hiper-elipse, e (2) se o ponto encontrado não é um ponto fora da curva (*outlier*).

Por último, a rotulação dos segmentos é realizada da mesma forma como apresentada na seção 4.5.1.

4.5.3 Modelos escondidos de Markov Adaptativos*

Os métodos anteriores se preocuparam, primeiramente, em encontrar todos os pontos de mudança (a segmentação de fato), e somente no último passo, de posse dos pontos de mudança é que a rotulação pode ser realizada. Apesar desta rotulação poder ser executada a qualquer momento (em uma execução em tempo real), o conhecimento dos modelos de seção previamente construídos não tinham nenhuma utilidade durante o processo. Neste método consideramos a etapa de rotulação *durante a segmentação*, ou seja, sempre que encontramos um ponto de mudança, compararmos o modelo da nova seção com todos os modelos de seção pré-existentes. Caso encontremos um modelo “mais similar”, então atualizamos o modelo encontrado com as observações da nova seção, e, caso contrário, é adotado um novo modelo de seção, construído com as observações da nova seção encontrada. Os modelos de seção são modelados como estados da cadeia de Markov, e assim, além de atualizar o modelo do estado, podemos também atualizar os outros parâmetros da HMM contínua, como a matriz de transição entre estados A . Neste modelo, cada estado é modelado como uma mistura de gaussianas, e o mesmo representa um modelo de seção. O método pode ser descrito nas seguintes etapas.

1. Inicializa o modelo HMM $\lambda(A, B, \pi)$ com 0 estados.
2. Encontra uma seção $T = t_1, \dots, t_N$ em um instante qualquer do tempo, como descrito no algoritmo 13.
3. Encontra o estado b_j da HMM mais verossímil à seção T , que é aquele que tem a máxima média da verossimilhança, maior que um limiar fixo α .

$$j = \operatorname{argmax}_j \left\{ \frac{1}{N} \sum_{i=1}^N \log b_j(t_i) \mid \frac{1}{N} \sum_{i=1}^N \log b_j(t_i) > \alpha \right\} \quad (4.42)$$

4. Caso nenhum estado atenda às restrições dadas pelo passo anterior, então é criado um novo estado. Como cada estado é modelado como uma mistura de gaussianas (equação 4.22), os parâmetros são estimados através do algoritmo de Maximização da Esperança.
5. Caso um estado b_j atenda às restrições dadas pelo passo anterior (equação 4.42), então os parâmetros do estado b_j são re-estimados. No caso da re-estimação dos parâmetros de misturas de gaussianas, temos dois cenários.
 - O primeiro é quando não armazenamos as observações de entrada de cada estado. Neste caso, é possível utilizar o algoritmo de Maximização da Esperança para re-estimar os parâmetros das m gaussianas de b_j , dado que os parâmetros iniciais já estão estimados, e o algoritmo EM converge para a máxima verossimilhança mesmo na ausência

de todos os dados. Outra alternativa mais simples seria adotar $m = 1$, ou seja, somente uma gaussiana $\mathcal{N}(\mu_j, \Sigma_j)$ para cada estado b_j , e a re-estimação dos parâmetros μ_j^k, Σ_j^k poderia ser feita de forma iterativa, da seguinte maneira:

$$\begin{cases} \mu_j^0 = \mu_j \\ \mu_j^{k+1} = \mu_j^k + \frac{t_{k+1} - \mu_k}{k+1}, & 0 \leq k \leq N-1 \end{cases} \quad (4.43)$$

e

$$\begin{cases} \Sigma_j^0 = \Sigma_j \\ \Sigma_j^{k+1} = \left(1 - \frac{1}{j}\right) \Sigma_j^k + (k+1)(\mu_j^{k+1} - \mu_j^k)^2, & 0 \leq k \leq N-1. \end{cases} \quad (4.44)$$

- O segundo cenário é quando podemos armazenar observações para cada estado. Neste caso, podemos executar a re-estimação sempre com todas as observações que fazem parte de um determinado estado b_j .

6. Atualiza as transições entre estados. Esta etapa simplesmente reforça a transição entre o estado anterior – e isto força o algoritmo a sempre armazenar o estado anterior – e o estado b_j encontrado, independentemente de ser um estado novo ou não.
7. Construção do modelo oculto de Markov final. O procedimento compreende as seguintes etapas:

- Agrupamento hierárquico. Agrupa os estados encontrados até o momento. Isto é realizado através de um agrupamento hierárquico, utilizando a distância de Bhattacharyya entre os modelos normais da mistura de cada estado.
- Atualização das transições entre estados, uma vez que agora os mesmos estão agrupados.
- Com os estados B agrupados e a matriz de transição A atualizada, construímos um HMM $\lambda(A, B, \pi)$, que é depois treinado pelo algoritmo de Baum-Welch onde são re-estimados os parâmetros do modelo $\hat{\lambda}$. A sequência de estados correspondente à peça musical é obtida através da decodificação utilizando o algoritmo de Viterbi, dado o modelo $\hat{\lambda}$ e a sequência de observações armazenada até este instante.

Alguns dos passos do método acima podem ser executados de diferentes maneiras, como, por exemplo, o passo 2 e o agrupamento em 7. Para isto, foram construídas diferentes configurações deste modelo, como descritas a seguir. Em todas as configurações, os estados do modelo de Markov foram modelados como uma mistura de duas gaussianas.

- **AHMM-1** No passo 2 foi utilizado o algoritmo de Tzanetakis e Cook (1999) (seção 4.4.3).
- **AHMM-2** No passo 2, os pontos de mudança forma encontrados por matrizes de similaridade (seção 4.4.1)
- **AHMM-3** O passo 2 foi executado com matriz de similaridade e processamento de imagens (seção 4.4.4).
- **AHMM-4** No passo 2 foi utilizado o critério de informação bayesiana (BIC) para segmentar as janelas.

Tabela 4.5: Configurações para o segmentador AHMM

As figuras 4.21 e 4.22 mostram a sequência de estados/seções de uma segmentação utilizando este modelo. A primeira mostra a evolução (no tempo) da construção dos modelos de seção, e podemos ver que o método indica muito mais seções do que realmente existem, se a compararmos

com a sequência do gabarito (imagem 4.13). A segunda imagem mostra a sequência de estados após um agrupamento hierárquico, de onde vemos uma redução de pouco mais de vinte e cinco para oito estados.

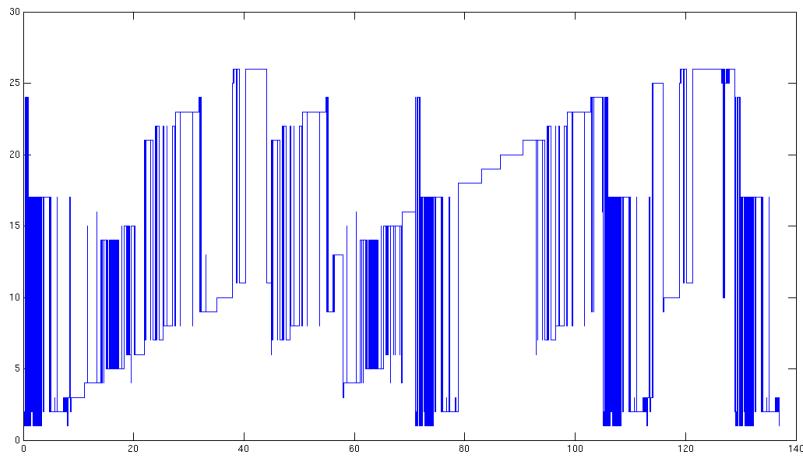


Figura 4.21: Sequência de estados sem o agrupamento das seções encontradas.

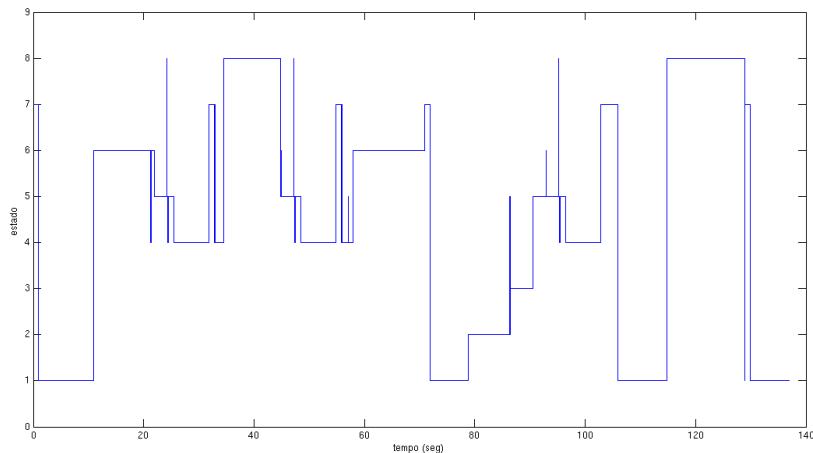


Figura 4.22: Sequência de estados com agrupamento hierárquico das seções encontradas.

4.6 Pós-processamento

O pós-processamento é a última etapa antes de avaliar a segmentação. Recordemos que a saída dos métodos de segmentação apresentados até aqui é basicamente uma sequência $T = \omega_i : \omega_i \in 1, \dots, p$ e $i = 1, \dots, N$, onde p é o número de seções, ou classes, estimados pelo método executado, e N é o número de observações do sinal musical. Assim, o objetivo do pós-processamento é avaliar a saída e corrigir falhas muito evidentes, como, por exemplo, quando temos uma rápida variação na sequência de estados, o que tornaria a segmentação incoerente, uma vez que estamos interessados em captar seções com uma duração maior que uma janela de análise de 50 ms.

4.6.1 Suavização por moda dos vizinhos

Uma forma de evitar rápidas alternâncias entre seções é suavizar a saída através da moda dos vizinhos. As figuras 4.23 e 4.24 ilustram uma segmentação antes e depois deste processo, respectivamente, onde utilizamos a moda de 21 vizinhos para a obtenção destes resultados. Depois do pós-processamento, os pontos de mudança entre as seções estão melhor definidas e com poucos ruídos, mantendo, ao mesmo tempo, a forma original.

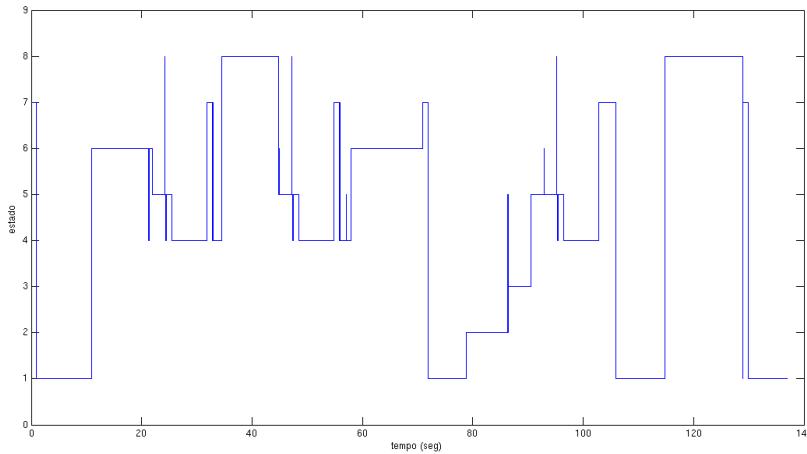


Figura 4.23: Saída de uma segmentação antes do pós-processamento

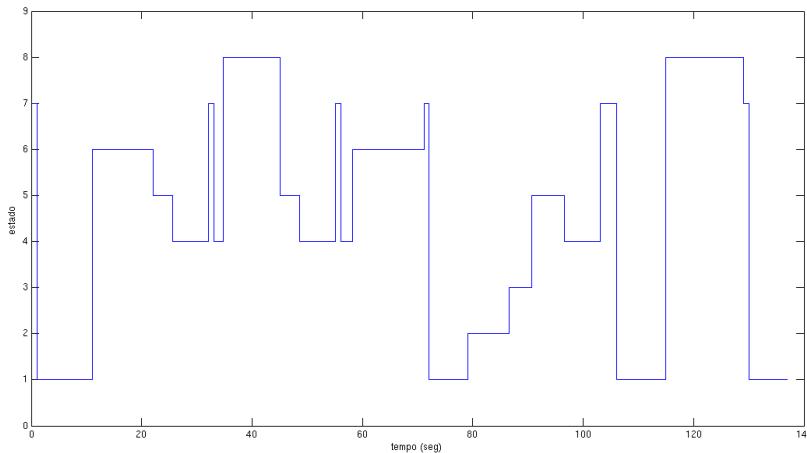


Figura 4.24: Saída de uma segmentação depois do pós-processamento

4.7 Avaliação dos Segmentadores

A avaliação da segmentação depende de dois fatores: a probabilidade de erro dos rótulos (se os rótulos da sequência sendo testada são consistentes com os rótulos da sequência de referência), e a probabilidade de erro da segmentação (se as localizações temporais das transições da sequência de teste estão consistentes com as da sequência de referência). A maior dificuldade de propor um método de avaliação é que quando estamos avaliando os rótulos, não conseguimos avaliar a segmentação, e vice-versa. Isto significa que não temos uma medida única para avaliar o segmentador. Outros problemas aparecem quando vamos avaliar os rótulos, pois a quantidade de rótulos encontrados pelo segmentador pode ser diferente do real. Além disto, quando vamos

avaliar as transições entre seções, tanto o número de transições das duas sequências (referência e teste) podem ser diferentes, quanto o podem ser suas posições temporais. Assim, uma solução simples, como a contagem de pontos estimados corretamente sobre a quantidade de pontos reais totais, não é uma boa medida, pois uma segmentação que fornecesse um ponto de transição pra cada amostra teria um ótimo rendimento, o que na realidade seria uma afirmação falsa.

Chai (2006) sugere que o desempenho da rotulação é simplesmente a proporção das amostras que foram rotuladas corretamente.

$$\text{Desempenho da rotulação} = \frac{\#\text{Amostras rotuladas corretamente}}{\text{número total de amostras}}$$

Esta afirmação por si só não nos ajuda a avaliar a segmentação, uma vez que os segmentadores não devolvem, necessariamente, a mesma quantidade de rótulos e tampouco os mesmos símbolos nos rótulos para as duas sequências sendo analisadas. Então, apesar de concordarmos que a fórmula acima é útil, existem alguns pontos a serem considerados. Sobre a avaliação da segmentação, Chai (2006) sugere que deve ser adotado um parâmetro de distância δ tal que se a distância de um ponto da sequência de teste t_1 para um ponto da sequência de referência t_2 for menor que δ , então t_1 é uma transição relevante. Assim, o autor propõe duas medidas para avaliar o desempenho da segmentação. Precisão¹², que é definida como a proporção de transições detectadas que são relevantes; e Recordação¹³, que é definida como a proporção das transições detectadas. Portanto, destas definições temos que B = transições relevantes, C = transições detectadas, e $A = B \cap C$, e então Precisão = $\frac{A}{C}$ e Revocação = $\frac{A}{B}$. Outros métodos de avaliação são apresentados em Paulus *et al.* (2010).

Mapeamento de rotulação

O primeiro problema a ser resolvido é encontrar o mapeamento correto que minimiza as discrepâncias entre a sequência de referência e a sequência de teste. Isto ocorre devido ao fato que o segmentador não devolve os mesmos símbolos adotados pela sequência de referência, que é o nosso gabarito. A forma como criamos este mapeamento gera duas medidas diferentes, onde a primeira penaliza tanto o segmentador que estima erroneamente o número de rótulos quanto o segmentador que é temporalmente impreciso, enquanto a segunda releva o erro da estimação do número de rótulos, mas penaliza a imprecisão temporal.

Definições Seja N o número de observações, p o número de rótulos manualmente definidos da sequência de referência $f(t)$, tal que $f(t) : \mathbb{N} \rightarrow \{1, \dots, p\}$, $t = 1, \dots, N$, e seja q o número de rótulos encontrados na sequência de teste $g(t)$, tal que $g(t) : \mathbb{N} \rightarrow \{1, \dots, q\}$, $t = 1, \dots, N$.

Desejamos encontrar uma função de mapeamento $m : \{1, \dots, q\} \rightarrow \{1, \dots, p\}$ correspondente a uma translação de rótulos produzidos algorítmicamente para os rótulos definidos manualmente, de tal forma que a sequência remarcada $h(t) = m(g(t))$ seja mais próxima possível da sequência de referência no sentido de minimizar a taxa de erro

$$\epsilon(m) = \frac{1}{N} \sum_{t=1}^N \delta(f(t), m(g(t))) = \frac{1}{N} \sum_{t=1}^N \delta(f(t), h(t)),$$

onde δ é o delta de Kronecker. Isto é necessário devido ao fato de que o método de rotulação não produz necessariamente os mesmos símbolos adotados pela sequência de referência.

A diferença entre as duas medidas está na forma como é definida a função de mapeamento, e qual é seu espaço de busca para encontrar a mínima discrepância.

¹²Precisão, do inglês *Precision*

¹³Recordação, do inglês *Recall*.

Medida que penaliza a estimativa do número de rótulos

Nesta medida, busca-se penalizar segmentadores que estimaram erroneamente o número de rótulos. Assim, o espaço de busca de m , que representa os rótulos da sequência de teste, é dado por $q!$, onde q é o número de símbolos diferentes dos rótulos da sequência de teste. A propriedade desta medida é que dois rótulos de teste distintos sempre serão mapeados em dois rótulos distintos antes da comparação com a sequência de referência, mesmo que isso significasse “jogar rótulos fora” (associando-os a valores fora da faixa $1 \dots p$).

Com esta propriedade, sabemos de antemão que mesmo que $p \neq q$, a função h sempre terá q rótulos, e, portanto, caso o segmentador tiver estimado um número muito alto de estados/rótulos, ou um número muito baixo, o erro aumenta. O caso ideal é aquele em que o número de rótulos estimados é o mesmo que o número de rótulos do gabarito.

Podemos ver um exemplo de como o mapeamento foi construído nas imagens 4.25, 4.26 e 4.27, onde todos os cenários estão cobertos. Veja que o contorno geral da função h é mesmo que da função g em todos os casos.

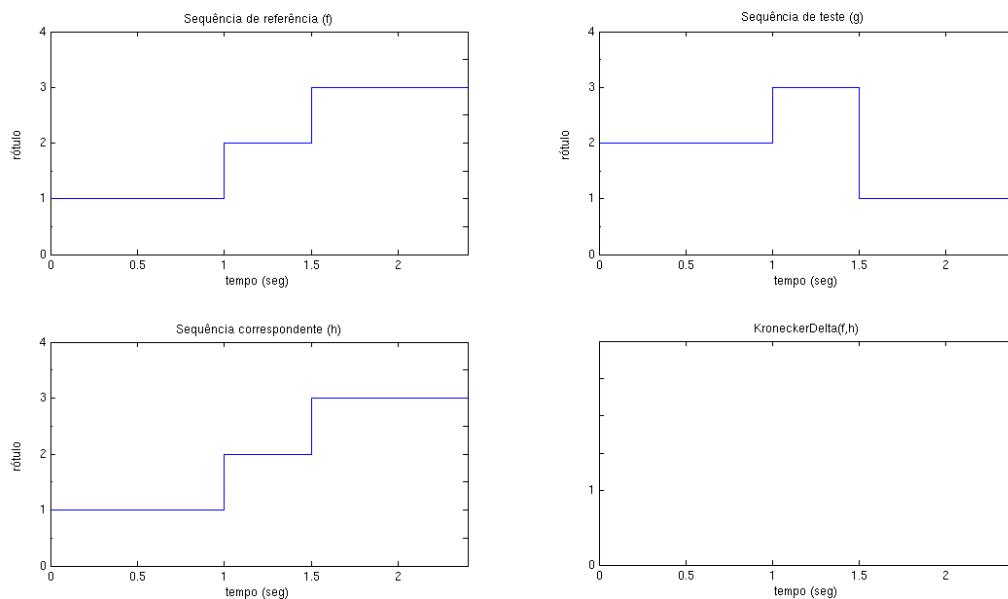


Figura 4.25: Exemplo de mapeamento utilizando função bijetora, com $q = p$. Sequência de referência $f(t)$, ou gabarito (canto superior esquerdo); sequência de teste $g(t)$ (canto superior direito); sequência de mapeamento correspondente $h(t)$ (canto inferior esquerdo); e sequência de discrepâncias entre as funções f e h (canto inferior direito)

Medida que penaliza a precisão temporal das fronteiras das seções

Nesta medida não existem restrições para a função de mapeamento, e por este motivo, ela pode ser tanto uma função injetora quanto sobrejetora, caso o número q de símbolos distintos de g seja menor ou maior que o número p de símbolos distintos de f .

As funções de mapeamento m são todas as sequências de tamanho q formadas pelos elementos $\{1, \dots, p\}$. Existem, portanto, p^q sequências no espaço de busca.

Com esta medida, consideramos que a segmentação manual pode não ter exatamente a mesma característica timbrística durante toda sua execução, visto que os trechos musicais selecionados contém variações dentro de si, e que quem as selecionou foi um humano, que certamente “ouve” características diferentes das da máquina. Com isto, aqueles estados que variam dentro de uma mesma seção seriam mapeados para uma única seção, e o erro final seria somente aquele das localizações temporais erradas.

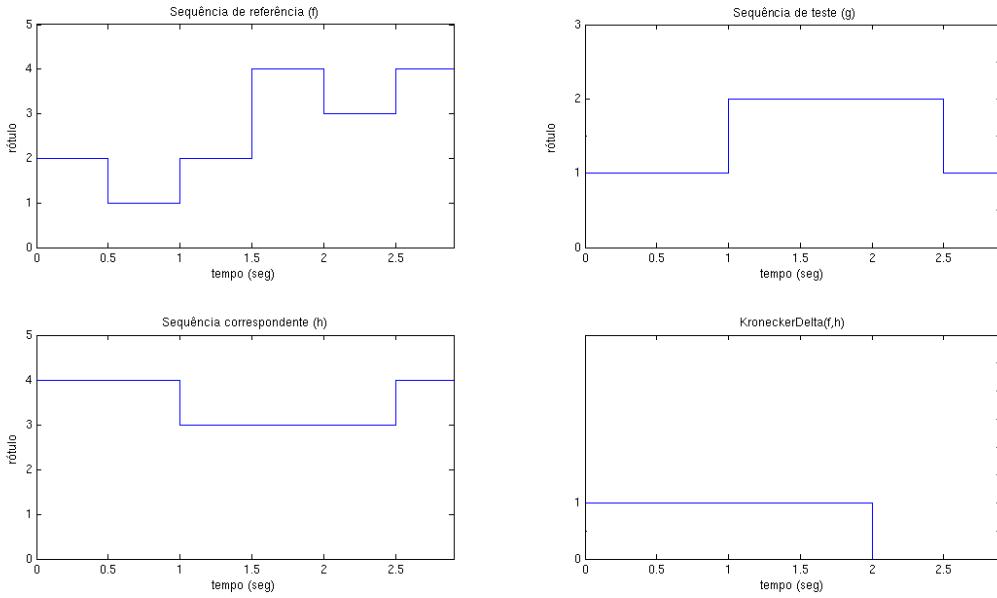


Figura 4.26: Exemplo de mapeamento utilizando função bijetora, com $q < p$. Sequência de referência $f(t)$, ou gábarito (canto superior esquerdo); sequência de teste $g(t)$ (canto superior direito); sequência de mapeamento correspondente $h(t)$ (canto inferior esquerdo); e sequência de discrepâncias entre as funções f e h (canto inferior direito)

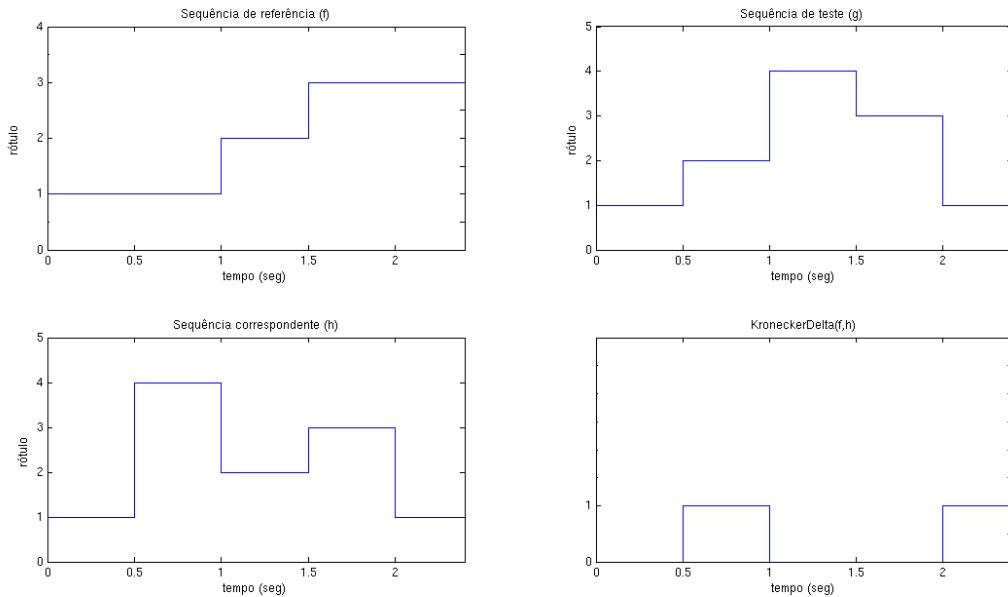


Figura 4.27: Exemplo de mapeamento utilizando função bijetora, com $q > p$. Sequência de referência $f(t)$, ou gábarito (canto superior esquerdo); sequência de teste $g(t)$ (canto superior direito); sequência de mapeamento correspondente $h(t)$ (canto inferior esquerdo); e sequência de discrepâncias entre as funções f e h (canto inferior direito)

Podemos ver um exemplo de como o mapeamento foi construído nas imagens 4.28, 4.29 e 4.30, onde todos os cenários estão cobertos. Veja que quando o número de rótulos de g é maior que o de f , o erro final é menor que aquele apresentado pela função anterior.

Um aspecto interessante desta medida é o fato dela não penalizar demasiadamente uma estratégia de segmentação que tenha subdividido seções de referência por pequenas variações de timbre, mas penaliza as fronteiras das seções detectadas incorretamente. Em um caso extremo, se o rotulador gerasse um rótulo novo para cada observação, esta medida de distância daria um

erro zero, pois cada rótulo (distinto) da sequência de teste seria mapeado perfeitamente no rótulo correto do gabarito, fornecendo discrepância zero. Na prática este tipo de rotulação degenerada não acontece devido à característica dos modelos construídos.

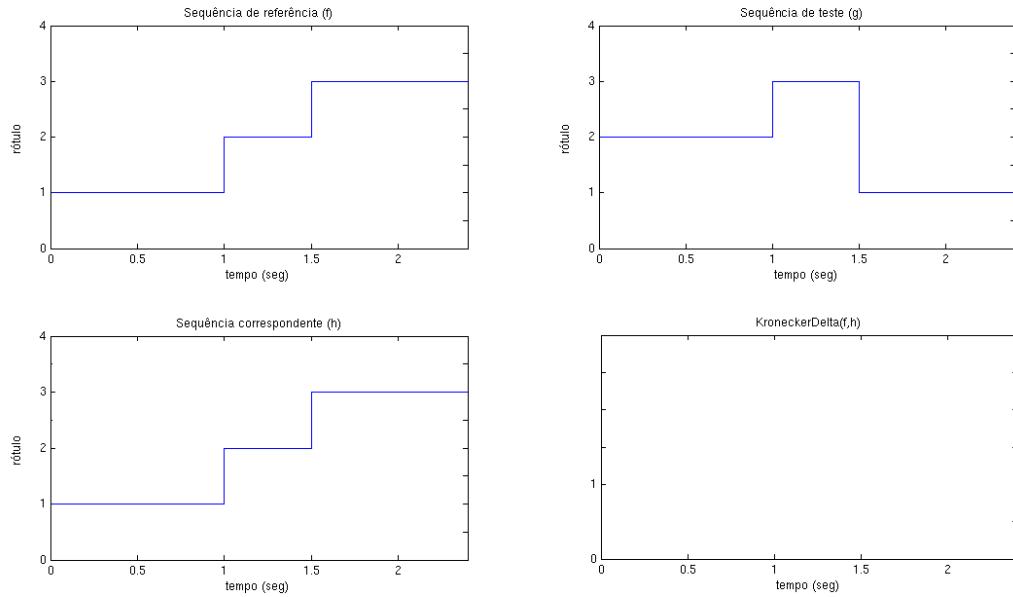


Figura 4.28: Exemplo de mapeamento ótimo (função injetora), com $q = p$. Sequência de referência $f(t)$, ou gabarito (canto superior esquerdo); sequência de teste $g(t)$ (canto superior direito); sequência de mapeamento correspondente $h(t)$ (canto inferior esquerdo); e sequência de discrepâncias entre as funções f e h (canto inferior direito)

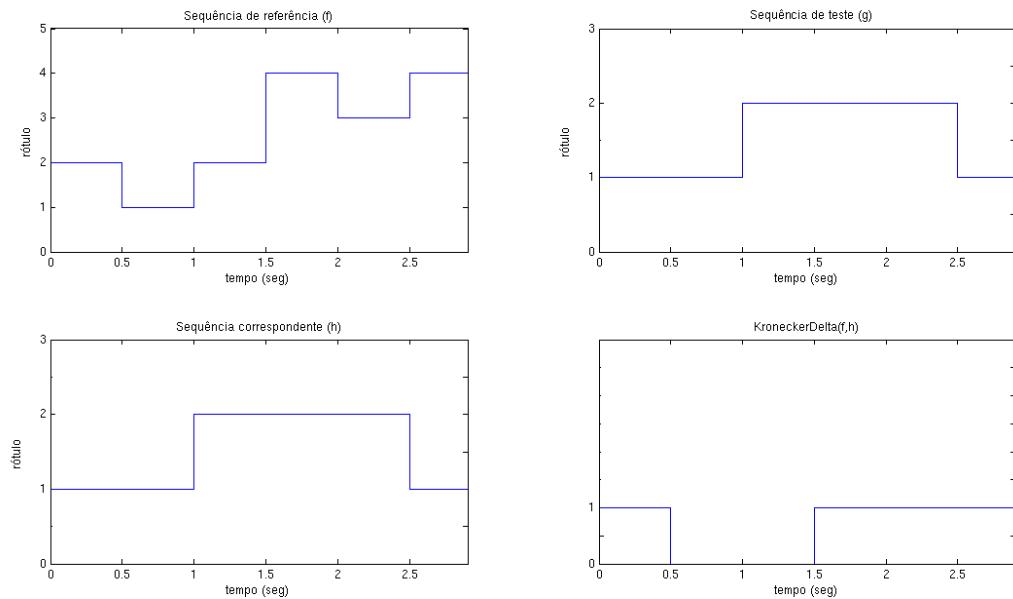


Figura 4.29: Exemplo de mapeamento ótimo (função injetora), com $q < p$. Sequência de referência $f(t)$, ou gabarito (canto superior esquerdo); sequência de teste $g(t)$ (canto superior direito); sequência de mapeamento correspondente $h(t)$ (canto inferior esquerdo); e sequência de discrepâncias entre as funções f e h (canto inferior direito)

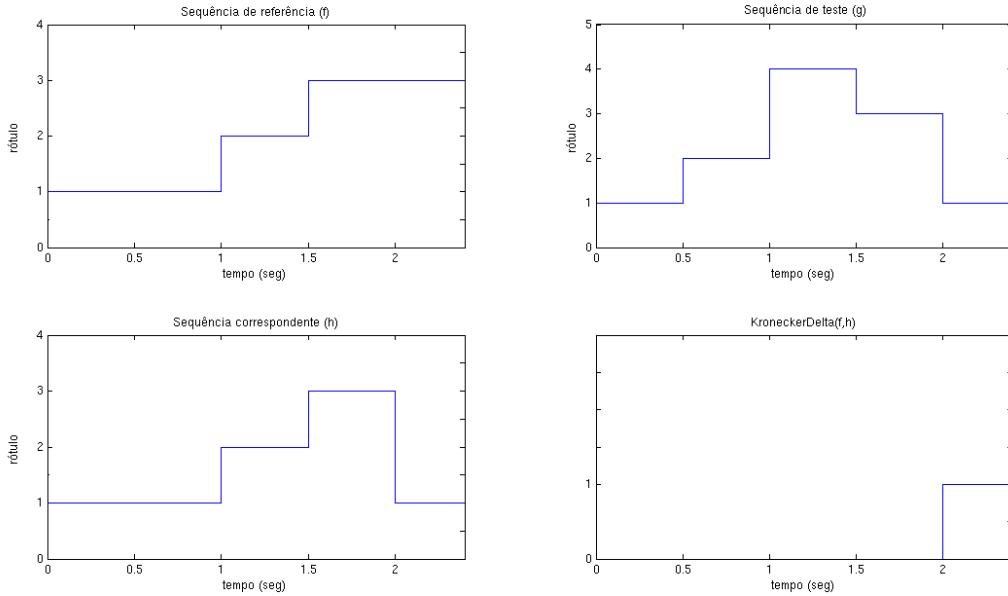


Figura 4.30: Exemplo de mapeamento ótimo (função sobrejetora), com $q > p$. Sequência de referência $f(t)$, ou gabarito (canto superior esquerdo); sequência de teste $g(t)$ (canto superior direito); sequência de mapeamento correspondente $h(t)$ (canto inferior esquerdo); e sequência de discrepâncias entre as funções f e h (canto inferior direito)

Redução da complexidade para as funções de mapeamento

Como vimos anteriormente, para encontrar a função que minimiza a discrepância entre f e h , temos uma complexidade computacional de $O(q!)$ e $O(p^q)$, no caso das funções bijetoras e não-bijetoras, respectivamente. O problema pode ser reduzido se definirmos uma matriz de associação $S_{p \times q}$

$$S = \begin{bmatrix} s_{1,1} & \cdots & s_{1,q} \\ \vdots & \ddots & \vdots \\ s_{p,1} & \cdots & s_{p,q} \end{bmatrix} \quad (4.45)$$

onde $s_{i,j}$ é definido como a soma das ocorrências de i e j no mesmo instante de tempo. Veja o algoritmo 14 para mais detalhes da geração da matriz.

Algoritmo 14 Cálculo da matriz de associação

$S = \text{GERA-MATRIZ-ASSOCIAÇÃO}(f, g)$

Entradas:

f # função da sequência de referência
 g # função da sequência de teste

Saída:

$S_{p \times q}$ # matriz de associação

- 1: inicializa S com $s_{i,j} = 0$
 - 2: **para** $t \leftarrow 1$ até N **faça**
 - 3: $s_{f(t),g(t)} = s_{f(t),g(t)} + 1$
 - 4: **fim para**
-

Com isto podemos temos que

$$\sum_{i=1}^q s_{m(i),i} = \text{número de observações rotuladas corretamente segundo a associação } m(i). \quad (4.46)$$

Solução para medida que penaliza a precisão temporal das fronteiras das seções Para a segunda medida apresentada, (seção 4.7), quando a solução permite que dois rótulos da resposta sejam associadas ao mesmo rótulo do gabarito, a solução é direta.

$$\theta(i) = \operatorname{argmax}_j s_{j,i} \quad (4.47)$$

Solução para função bijetora O número de observações rotuladas corretamente por este método é dada por

$$\max \sum_{i=1}^q s_{\theta(i),i}, \quad (4.48)$$

e o problema deve ser dividido em três cenários:

1. $q = p$

Onde θ é bijetora

$$\theta : \{1, \dots, q\} \rightarrow \{1, \dots, p\}$$

2. $q > p$

Onde $P \subseteq \{1, \dots, q\}$, $|P| = p$, e θ é bijetora,

$$\theta : \{1, \dots, q\} \rightarrow P.$$

3. $q < p$

Onde $Q \subseteq \{1, \dots, p\}$, $|Q| = q$, e θ é bijetora,

$$\theta : \{1, \dots, q\} \rightarrow Q.$$

Podemos ver este problema como o problema de *atribuição de tarefas*: Existe um número de pessoas e um número de tarefas. Qualquer pessoa pode ser designada a executar qualquer tarefa, estando sujeito a um custo que pode variar dependendo da atribuição da pessoa com a tarefa. É necessário executar todas as tarefas atribuindo exatamente uma pessoa para cada tarefa de forma que o custo total das atribuições seja minimizado.

O *método húngaro*, algoritmo inicialmente publicado por Kuhn (1955), resolve este problema com um tempo de complexidade polinomial. No nosso caso, a diferença é que devemos maximizar o custo, que são as rotulações corretas, e não minimizar, o que pode ser traduzido invertendo os sinais dos custos.

Capítulo 5

Resultados em Seleção de Descritores e Segmentação

Pareceu-nos que (...) as ligações temporais não eram nem tão sólidas, nem tão uniformes, nem tão gerais como se gosta de afirmar. O fio do tempo é cheio de nós. (...) O real não para de tremer em torno de nossos pontos de referência abstratos.

Gaston Bachelard, "A dialética da duração"

Os resultados desta pesquisa podem ser divididos em resultados das seleções de descritores e resultados das execuções dos algoritmos de segmentação. Na seleção de descritores mostraremos primeiramente alguns exemplos de dados gerados, o que servirá como um primeiro contato com o arquivo musical segmentado manualmente e os gabaritos dos seus rótulos e, em seguida, uma análise dos descritores selecionados.

Os resultados da segmentação musical serão analisados de acordo com as três visões: visão de tipo de descritor, visão de tipo de descritor dinâmico cumulativo e visão de memória temporal. Na visão de tipo de descritor, vamos analisar o desempenho dos algoritmos de acordo com suas famílias (MFCC, ALLDs e DDBM); na visão de tipo de descritor dinâmico, vamos comparar os resultados de acordo com o tipo de descritor dinâmico cumulativo (*moments*, *Euclidean*, *weighted* e *fft*); por último, na visão de memória temporal, vamos analisar os resultados de acordo com o parâmetro de tempo utilizado para a construção dos descritores dinâmicos (0, 0.5, 1, 5 e 10 segundos).

Abreviações

Neste capítulo trabalharemos com muitas abreviações e, portanto, julgamos importante definirlas ou relembrá-las aqui, mesmo tendo-as definido anteriormente.

Abreviações de Tipo de Erro

- **EER** : Erro de Estimação do número de Rótulos.
- **EPT** : Erro de Precisão Temporal.

Abreviações de Tipos de Descritores

- **MFCC** : Coeficientes Mel Cepstrais (*Mel-Frequency Cepstrum Coefficients*).

- **ALLDs** : Todos os Descritores de Baixo Nível (*All Low Level Descriptors*).
- **DDBM** : Descritores Dinâmicos por Bandas Mel.

Abreviações de Tipos de Descritores Dinâmicos Cumulativos

- **DDC**, Descritores Dinâmicos Cumulativos. Onde os tipos estão limitados aos seguintes valores:
 - *none*, ou seja, os descritores são usados em sua forma primitiva, sem modelagem temporal.
 - *moments*, quando nos referimos aos momentos estatísticos.
 - *Euclidean*, quando nos referimos à norma Euclidiana.
 - *weighted*, quando nos referimos à média ponderada.
 - *fft*, quando nos referimos aos coeficientes da transformada de Fourier.

Lembramos ao leitor que esta geração de descritores dinâmicos só se aplica aos descritores de tipo MFCC e ALLDs. Para mais detalhes, ver seção 3.4.1.

Abreviações de Técnicas de Segmentação

- **HMM** : Modelos Ocultos de Markov (*Hidden Markov Models*).
- **GMM** : Modelos de Misturas de Gaussianas (*Gaussian Mixture Models*).
- **MLP** : Perceptron de Multi Camadas (*Multi Layer Perceptron*).
- **K-NN** : K-Vizinhos mais próximos (*K-Nearest Neighbors*).
- **NBC** : Classificador Ingênuo de Bayes (*Naive Bayes Classifier*).
- **MDISS** : Segmentação via Matriz de Dissimilaridade (ver seção 4.4.1).
- **MDISS-IPROC** : Segmentação por Dissimilaridade e Processamento de Imagem (ver seção 4.4.4).
- **COOPER/FOOTE**: Segmentação via Matriz de Similaridade de Cooper e Foote (2002) (ver seção 4.4.2).
- **TZ/COOK**: Segmentação por Delta de Mahalanobis (Tzanetakis e Cook, 1999) (ver seção 4.4.3).
- **MPS-GHMM**: Segmentação em Multi-passos com HMM de Misturas de Gaussianas (ver seção 4.4.6 para mais detalhes).
- **MPS-PBR**: Segmentação em Multi-passos - Peeters *et al.* (2002b) (ver seção 4.4.5 para mais detalhes).
- **AHMM**: Modelos escondidos de Markov Adaptativos (ver seção 4.5.3 para mais detalhes).
- **HIPER-ELIPSE**: Segmentação com Hiper-elipses (ver seção 4.5.2 para mais detalhes).
- **CUSUM**: Segmentação por Soma Cumulativa (ver seção 4.5.1 para mais detalhes).
- **BIC**: Segmentação por Critério de Informação Bayesiano (ver seção 4.5.1 para mais detalhes).

5.1 Geração de Dados e Seleção de Descritores

A estratégia adotada para a seleção de descritores depende unicamente de se ter os dados de treinamento. Esta restrição éposta somente para conseguirmos executar métodos de seleção que necessitam das classes dos dados (como é o caso do LDA). Uma forma de contornar esta restrição é gerar automaticamente os dados musicais. Nesta modelagem de dados foram geradas 81 músicas, sendo que o número de rótulos das músicas geradas variam de 2 a 7, e estes caracterizam as classes das seções. Isto foi feito para tentar mimetizar músicas reais nos testes de segmentação, uma vez que nossa intuição seja a de que quanto maior for o número de rótulos, maior a dificuldade em identificá-los.

Desta forma, a seleção de descritores é uma etapa que se repete para todos as músicas geradas e ainda para todas as gerações de descritores dinâmicos. Excluímos então a etapa de seleção de descritores quando estamos nos referindo aos descritores MFCC, deixando esta etapa somente para o ALLDs e para o DDBM. De modo geral, forçamos nosso algoritmo a selecionar um conjunto de descritores com dimensão próxima ao do MFCC (\mathbb{R}^{13}), pois um dos nossos objetivos é avaliar se somente o conjunto de descritores MFCC extraído do sinal de áudio é robusto o suficiente para as tarefas de segmentação, evitando assim a incorporação de outra etapa de seleção de descritores.

Nesta seção vamos apresentar uma breve análise dos dados, partindo de um exemplo musical gerado e alguns resultados de seleção de descritores a partir do ALLDs. Deixamos de lado aqui os descritores DDBM, pois estes não fazem parte de nenhuma taxonomia de descritores, mas sim de um método de extração, e não teríamos como mostrar de forma coerente os eixos selecionados utilizando este método de extração de descritores.

Guia de Leitura dos Descritores

Para simplificar a exposição dos descritores, adotamos um padrão para a nomenclatura dos mesmos. Procuramos abranger dois cenários nesta padronização: o número de componentes para cada descritor e o possível aumento da dimensão a partir de uma geração de descritores dinâmicos (como é o caso da geração com momentos estatísticos e da geração com os coeficientes da FFT). Assim, para cada descritor d , uma componente i e uma nova variável aleatória adicionada de índice j , o descritor fica representado por d_i^j .

Por exemplo, quando estamos nos referindo ao segundo coeficiente do descritor MFCC, sem geração de descritores dinâmicos ou com geração de descritores dinâmicos do tipo *norma Euclidiana* e *média ponderada*, podemos nos referir a ele como $MFCC_2$. Se escrevermos $MFCC_{2,3}$ significa que estamos nos referindo a dois coeficientes do MFCC, o segundo e o terceiro. Por outro lado, se estivermos nos referindo ao segundo coeficiente do MFCC com geração de descritores dinâmicos do tipo *momentos estatísticos*, podemos, por exemplo, escrever $MFCC_2^2$, o que significa que estamos nos referindo ao segundo momento estatístico do segundo coeficiente do MFCC. Por último, se gerarmos descritores dinâmicos com *coeficientes da FFT*, podemos nos referir ao primeiro coeficiente da FFT do segundo coeficiente do MFCC como $MFCC_2^1$. Para distinguir entre os descritores do tipo *momentos estatísticos* e do tipo *coeficientes da FFT*, vamos sempre fazer referências ao tipo de descritor dinâmico utilizado, se for o caso de existir um. Veja a tabela 5.1 para mais exemplos desta nomenclatura.

$MFCC_5$	Quinto coeficiente do MFCC
<i>Momentos estatísticos</i> - $LPC_{3,4,5}^1$	Primeiros momentos estatísticos do terceiro, quarto e quinto coeficientes do LPC.
<i>Coeficientes da FFT</i> - $MFCC_{1,2,3,4}^2$	Segundos coeficientes da FFT do primeiro, segundo, terceiro e quarto coeficientes do MFCC.
<i>Norma Euclidiana</i> - $Momentos_4$	Norma Euclidiana do quarto momento estatístico (curtose) do sinal.

Tabela 5.1: Exemplos de leitura dos descritores segundo o padrão adotado.

5.1.1 Análise das amostras

Como foi dito anteriormente, foram geradas 81 músicas para a última bateria de testes, sendo que em metade destas músicas geradas executamos a seleção de descriptores. Considerando todas as combinações possíveis de descriptores dinâmicos e memórias temporais, realizamos um total de 680 seleções de descriptores, e ficaria muito exaustivo apresentar uma análise de todas estas seleções. No entanto, com um único exemplo deste conjunto podemos demonstrar os dados selecionados. No exemplo que se segue, realizamos uma seleção de descriptores com DDC *weighted* e memória temporal de 0.5 segundos em um arquivo de áudio com 6 seções distintas. As figuras 5.1 e 5.2 mostram o gráfico das amostras de cada eixo de descriptor selecionado. Para que o leitor possa identificar na imagem os pontos de mudança de seção, geramos dois gráficos com os rótulos das seções, que se encontram na parte superior das figuras.

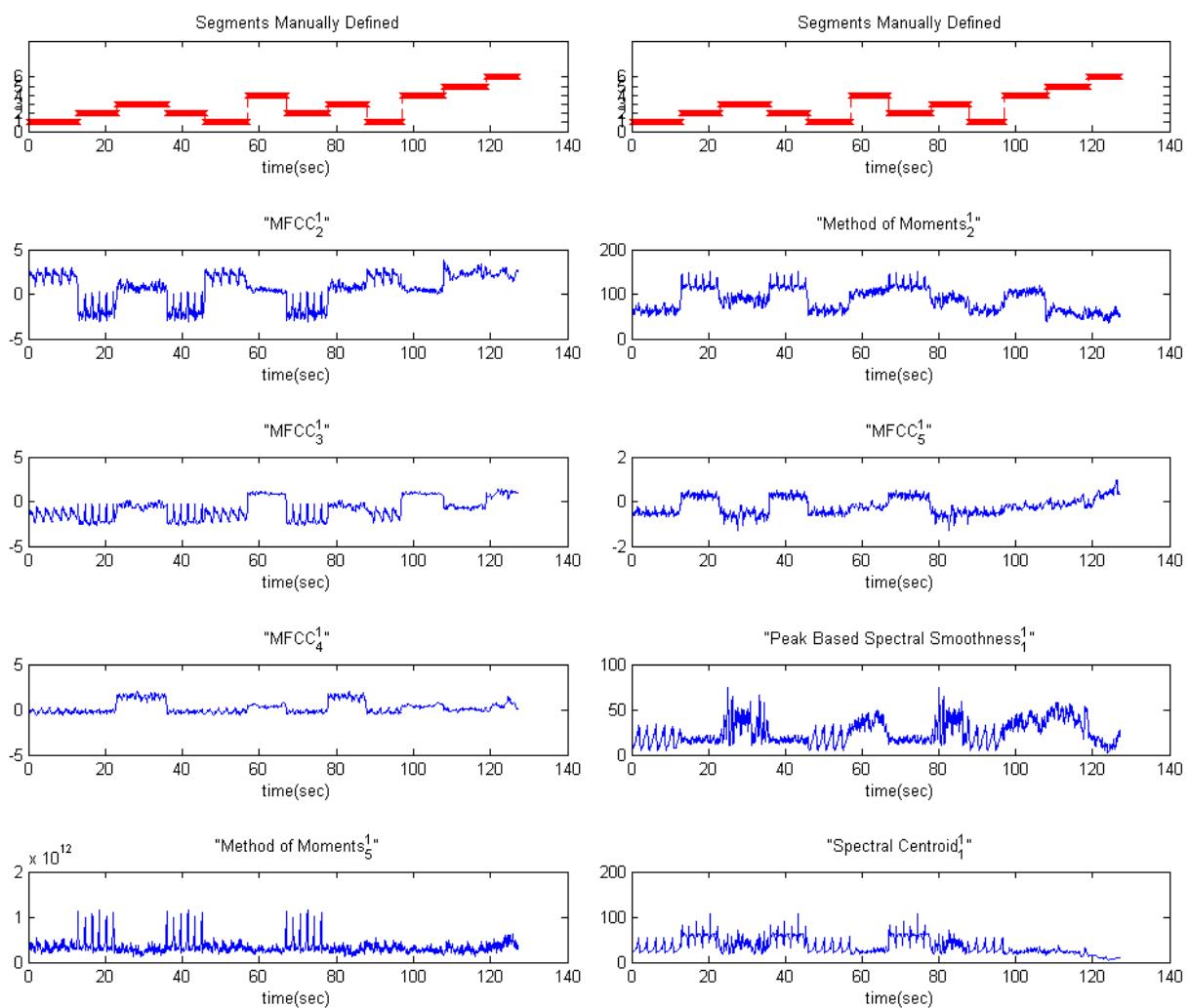


Figura 5.1: Descriptores selecionados com DDC *weighted* e memória temporal de 0.5 segundos. Na parte superior esquerda e direita estão os rótulos das seções para cada amostra no tempo. Os gráficos restantes são as amostras para os oito primeiros descriptores selecionados.

Uma das preocupações que tivemos durante a geração dos dados musicais foi o *fade-in* e *fade-out* entre cada seção, como podemos ver no arquivo de orquestra do Csound (ver apêndice D.1) onde o tempo de ataque e de decaimento para cada seção é de 1 ms. Esta estratégia permitiu simular as transições contínuas de uma música real, embora se trate de uma música perceptivelmente plástica, artificial.

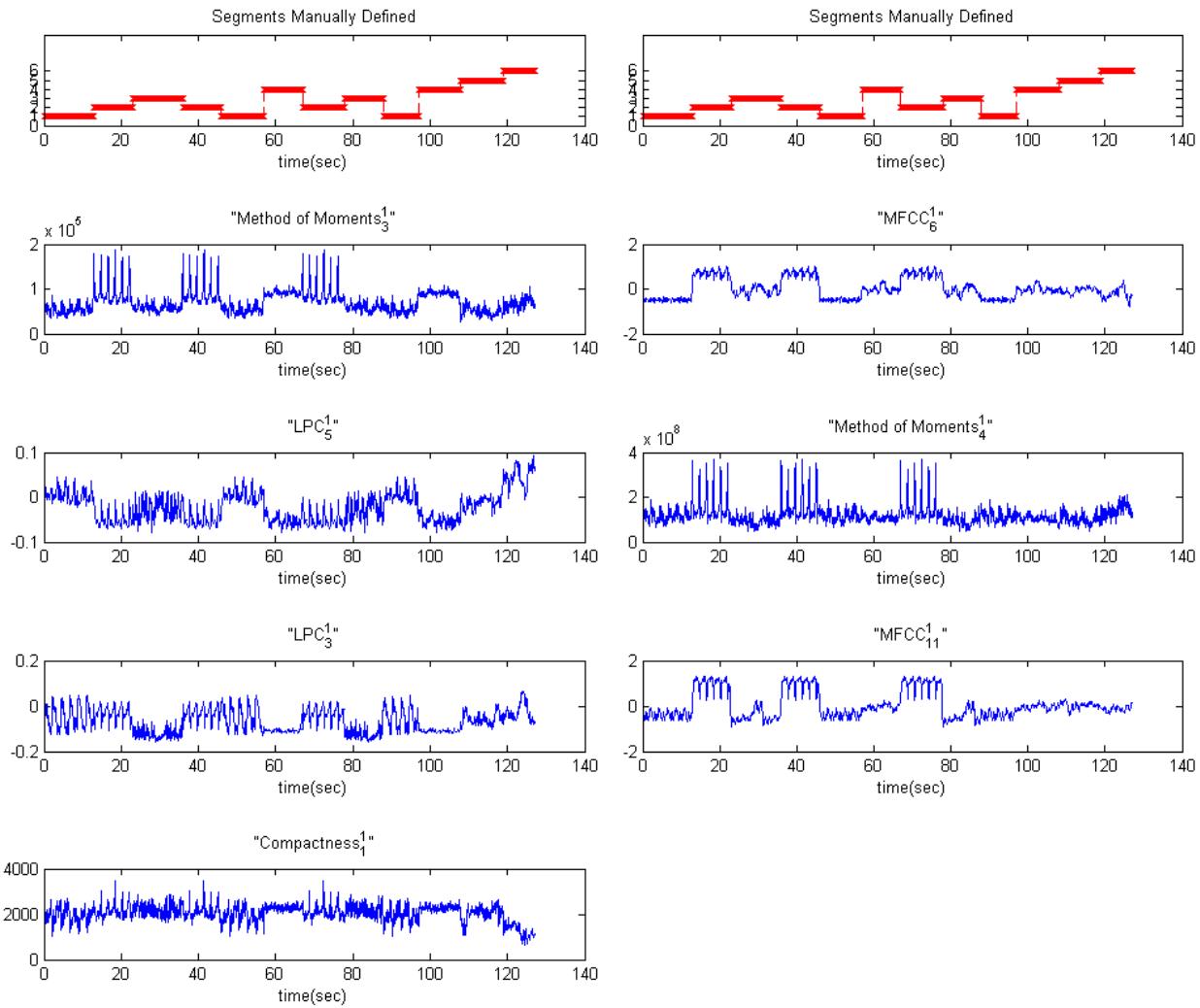


Figura 5.2: Descritores selecionados com DDC weighted e memória temporal de 0.5 segundos. Na parte superior esquerda e direita estão os rótulos das seções para cada amostra no tempo. Os gráficos restantes são as amostras para os sete últimos descritores selecionados.

Os gráficos de dispersão para as combinações entre os eixos podem ser vistos na figura 5.3. A ordem dos descritores contidos nesta matriz de gráficos de dispersão é a mesma dos descritores informados nas figuras 5.1 e 5.2, contando da esquerda para a direita, e de cima para baixo. Através desta matriz podemos ver que alguns pares de descritores fornecem uma boa separabilidade entre as classes, como, por exemplo, MFCC₃ e Momentos₂.

Lembremos que na maior parte dos métodos de segmentação, supomos que as variáveis aleatórias são homocedásticas com distribuições normais, e ao analisarmos a matriz da figura 5.3, vemos que isto não ocorre para todos os vetores de variáveis aleatórias, como é o caso da Centróide do Espectro e do LPC₃. Os histogramas para os vetores da mesma variável aleatória estão apresentados na diagonal principal. Apesar de estarmos trabalhando com um conjunto finito de amostras, estes gráficos apontam uma tendência a uma distribuição normal dos dados.

5.1.2 Descritores selecionados

A motivação inicial de realizar uma seleção de descritores é comparar os resultados de segmentação utilizando descritores MFCC com outros descritores. MFCC tem sido utilizada com bastante frequência em estudos de recuperação de informação musical, mas não é de nosso conhecimento que exista um estudo que comprove que o MFCC seja robusto em termos de desempenho de classificação/segmentação. De modo geral é uma tarefa difícil determinar qual descritor

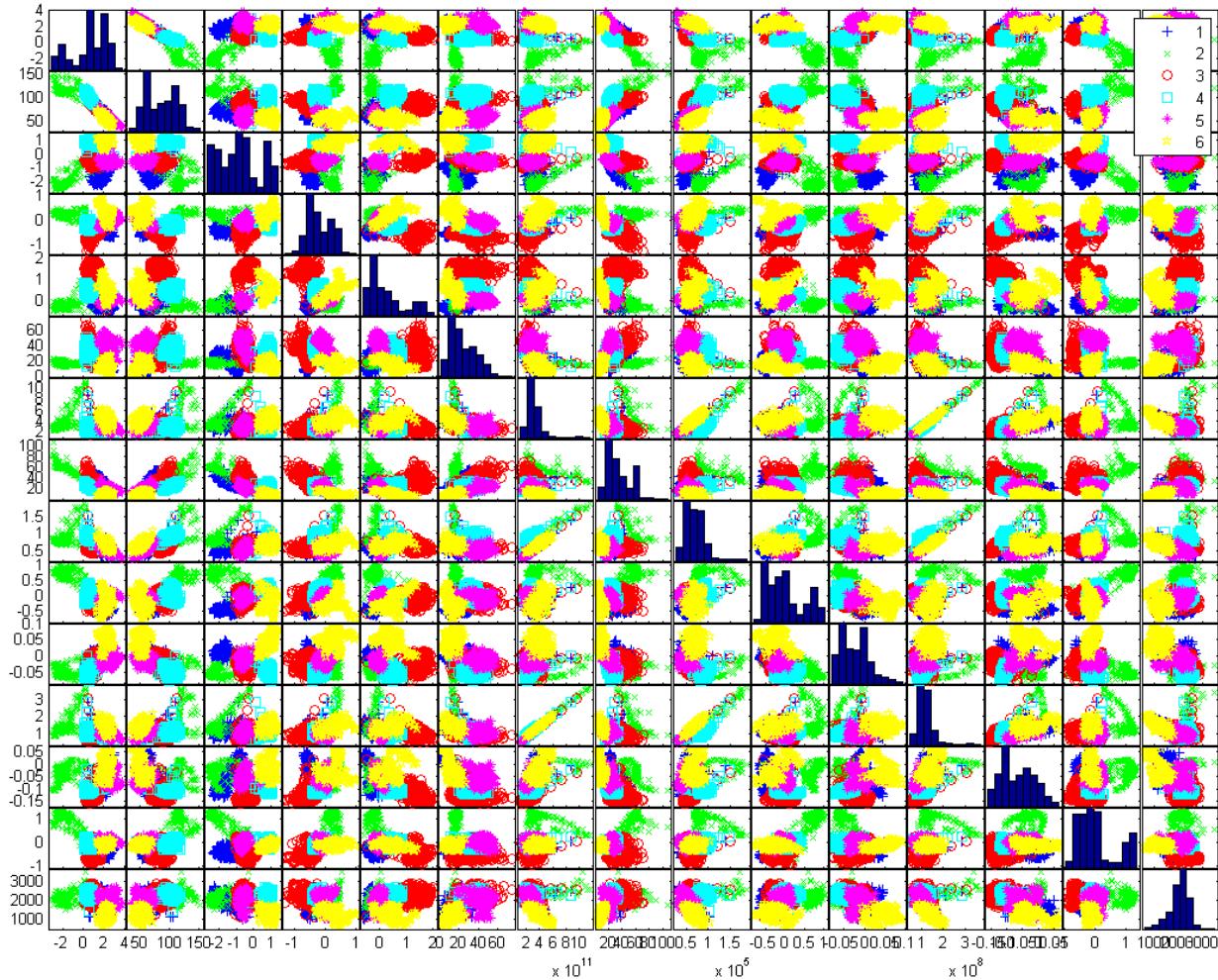


Figura 5.3: Gráfico de dispersão dos descritores selecionados, considerando que existem 6 rótulos distintos para representar as seções. Na diagonal se encontram os histogramas dos eixos. Os eixos desta matriz estão ordenados da seguinte forma: MFCC₂, Momentos₂, MFCC₃, MFCC₅, MFCC₄, Spectral Smoothness₃, Momentos₅, Centróide do Espectro, Momentos₃, MFCC₆, LPC₅, Momentos₄, LPC₃, MFCC₁₁, Compacidade.

ou conjunto de descritores teria um melhor desempenho sem ter conhecimento prévio dos dados. Todavia, em nosso contexto, obtivemos dados que possibilitam comparar os desempenhos dos segmentadores utilizando cada tipo de descritor. A questão do custo computacional para a seleção de descritores é importante para sistemas de segmentação em tempo real e não é menos importante em nossa avaliação qualitativa. O principal questionamento é que se o desempenho utilizando descritores MFCC não for tão diferente daquele obtido ao realizar uma prévia seleção, porque não utilizá-los, uma vez que a seleção de descritores é uma operação computacionalmente custosa?

O ponto de partida em uma seleção de descritores é ter um conjunto grande de descritores. No nosso caso, trabalhamos com um conjunto de 69 descritores e gostaríamos de selecionar um número de descritores próximo a 13. Um dos motivos é que estamos trabalhando com diferentes técnicas de segmentação, e qualquer mudança de variáveis pode alterar o resultado da comparação final. Desta forma, pretendemos eliminar possíveis erros devido a uma mudança no número de dimensões do vetor de descritores selecionados.

O primeiro cenário ocorre quando não geramos nenhum descritor dinâmico e gostaríamos de saber quais foram os descritores selecionados para ALLDs. A tabela 5.2 mostra aqueles que tiveram uma frequência de seleção maior ou igual a 2 dentre as seleções com todas as músicas. O que nos chama a atenção neste resultado é que, dentre os descritores selecionados, vemos que a maior parte dos coeficientes do MFCC estão no conjunto mais significativo e que a Compacidade

e os descritores de Momentos Estatísticos também estão presentes.

ALLDs	
	MFCC ₂
	Compacidade
	MFCC ₆
	Momentos ₅
	Momentos ₃
	MFCC ₃
	Momentos ₄
	Momentos ₂
	MFCC ₄
	MFCC ₇
	ZCR
	MFCC ₁
	MFCC ₅
	MFCC ₈
	MFCC ₉

Tabela 5.2: Lista ordenada por frequência de uso dos descritores ALLDs após processo de seleção.

Os outros cenários são aqueles em que estamos gerando Descritores Dinâmicos Cumulativos e temos ainda que considerar a variável de *Memória Temporal* em cada geração, que em nossos experimentos assumiu os valores apresentados na tabela 5.3.

Memória Temporal			
0.5 seg.	1 seg.	5 seg.	10 seg.

Tabela 5.3: Valores de Memória Temporal utilizados para a geração de descritores dinâmicos.

Os descritores selecionados em cada combinação de DDC e memória temporal estão listados na tabela 5.4.

O primeiro ponto a ser observado é a constância com que o MFCC aparece em todas as seleções de descritores, não importando o DDC e a memória temporal utilizados. Na geração por *média ponderada* o MFCC ocorre com grande frequência mesmo com uma memória temporal de 10 segundos. O segundo ponto observado é que à medida que aumentamos o tempo na geração dos descritores, mais importância adquirem os descritores formados por sua derivada, com exceção da geração com *coeficientes da FFT*. Outro descritor que está presente na maioria das seleções é a Compacidade, mesmo quando aumentamos a memória temporal.

Memória Temporal				
	.5 seg	1 seg	5 seg	10 seg
Norma Euclidiana	MFCC _{2,3,4,5,7}	Compacidade	MFCC _{1,2,3,4,5,7}	Derivada da Centroide do Espectro
	Compacidade	MFCC _{3,4,5,7}	Derivada do Momento _{2,4,5}	Derivada do Momento _{2,3,4,5}
	Momentos _{3,4,5}	Derivada do Momento _{1,5}	Derivada do MFCC ₅	ZCR
	Derivada do Momento _{1,4,5}	Derivada da Centroide do Espectro	Momentos _{3,4,5}	Momentos _{1,3,4,5}
	Derivada do MFCC ₂	Derivada do MFCC ₂	Compacidade	Derivada do MFCC _{2,3}
		Derivada do Momento ₁	ZCR	Derivada do LPC _{2,4}
		Derivada da Compacidade	Derivada da Compacidade	MFCC ₅
Momentos estatísticos	MFCC _{2,3,4,5,6,7,8,9,10,11,12,13}	Compacidade ¹	Compacidade ¹	Derivada do LPC ₃ ¹
	LPC _{3,4,7}	Derivada do MFCC ₅ ²	Derivada do Momento ₄ ²	Derivada do Momento ₅ ²
		LPC _{3,4} ¹	Derivada do MFCC ₃ ¹	Derivada do Momento ₄ ²
		MFCC _{2,3,4,5,6,7,8,9,11}	Derivada do MFCC ₄ ²	Derivada do MFCC _{3,6} ¹
			LPC _{4,9} ¹	LPC ₃ ¹
			Momentos ₄ ²	Momentos ₃ ²
			MFCC _{5,2,10,11} ¹	MFCC ₆ ¹
Média Ponderada	Compacidade	Compacidade	Compacidade	Derivada de ZCR
	LPC ₃	LPC ₄	Derivada do Momento _{2,5}	MFCC _{1,3,5,6,7,12}
	Momentos _{2,3,4,5}	Momentos _{2,4,5}	Derivada de ZCR	Centroide do Espectro
	MFCC _{1,2,3,4,5,6,7,8,9,13}	MFCC _{1,2,3,4,5,6,7,8,10}	LPC ₃	Derivada do Momento ₅
	Centroide do Espectro	Centroide do Espectro	Momentos _{2,3}	Derivada do MFCC ₃
		ZCR	MFCC _{1,2,3,4,5,6,7}	LPC _{6,7}
				Momentos _{1,3,5}
Coeficientes da FFT	Compacidade ¹	Compacidade ¹	Compacidade ^{1,3}	Compacidade ¹
	Derivada de ZCR ¹	LPC _{3,4} ¹	Momentos _{3,5} ¹	Momentos _{3,5} ¹
	Momentos _{1,2,3,4,5} ¹	Momentos _{2,3,4,5} ¹	Momentos _{4,5} ²	Momentos ₃ ²
	MFCC _{1,2,3,4} ¹	MFCC _{1,2,3,4,8} ¹	MFCC _{1,3,4,11} ¹	MFCC _{1,4,6} ¹
	Centroide do Espectro ¹	ZCR ¹	MFCC ₅ ²	MFCC _{3,4} ²
				MFCC ₈ ³
				ZCR ¹

Tabela 5.4: Lista de descritores que foram selecionados duas ou mais vezes durante o processo de seleção de descritores, considerando o tipo de descritor dinâmico (ver seção 3.4.1) e a memória temporal em sua geração.

Estas informações nos mostram quais foram, na média, os descritores mais selecionados nesta etapa de seleção de descritores. O desempenho de cada descriptor gerado será tema de discussão em seguida (ver seção 5.2.1), e só então teremos informações suficientes para tirar conclusões sobre a seleção de descritores.

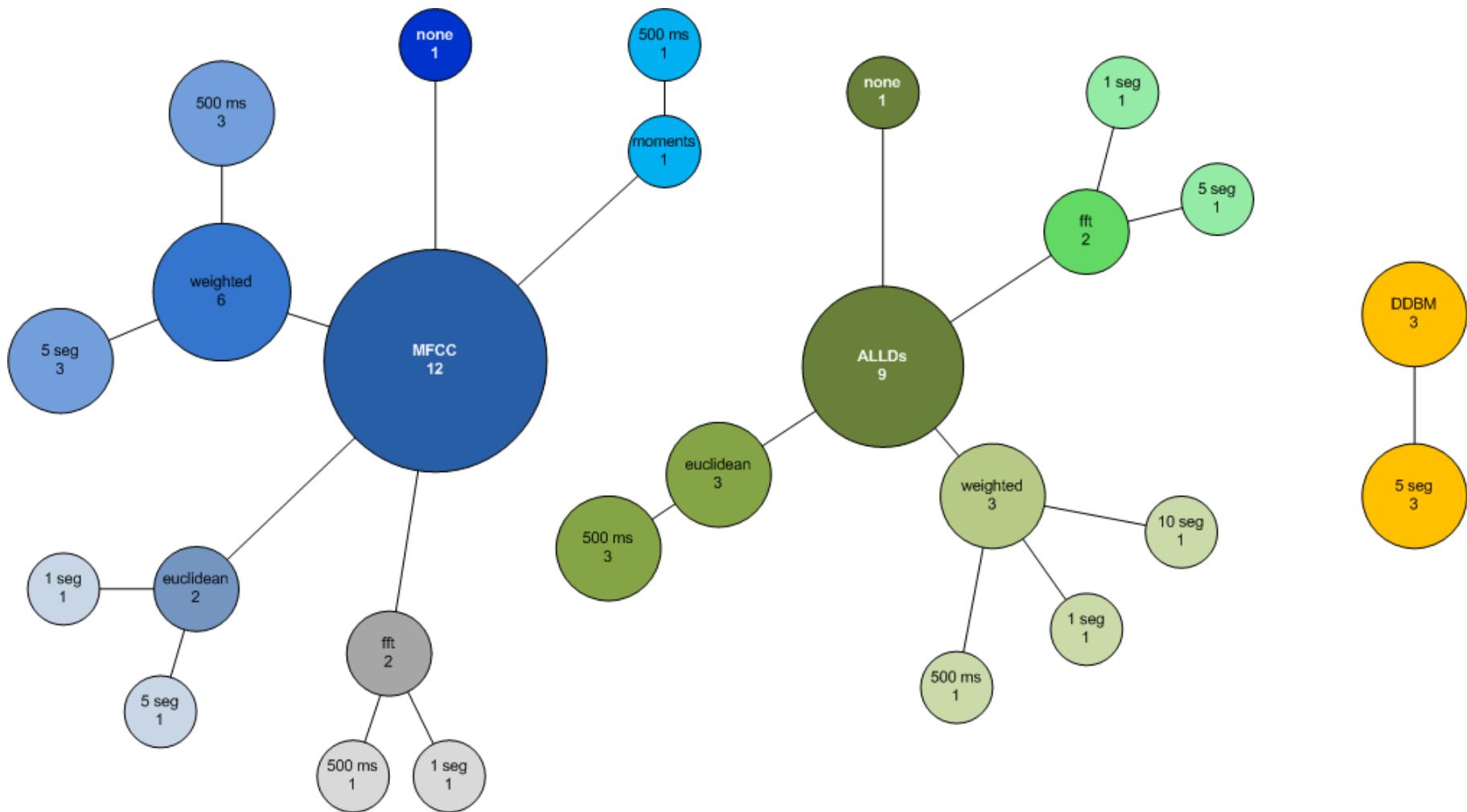


Figura 5.4: Árvore de descritores utilizados nos segmentadores para atingirem, na média, a taxa de erro mínima.

A figura 5.4 exibe a árvore de descritores utilizados pelos segmentadores que forneceram os melhores resultados. Nesta representação, o tamanho das bolas indicam quantos segmentadores estão utilizando cada uma das configurações, e o número dentro de cada uma indica quantos segmentadores estão utilizando cada configuração em cada nível. Na raiz da árvore temos o Tipo de Descritor, o segundo nível é o Tipo de DDC, e o terceiro nível é a Memória Temporal (no caso do DDBM, não existe Tipo de DDC e, portanto, o segundo nível neste caso é a Memória Temporal). Por esta representação e para este conjunto de dados, fica claro que a geração de descritores dinâmicos cumulativos podem gerar melhores resultados, considerando que os mesmos não super-enfatizam as observações passadas.

5.2 Segmentação Musical

Para os algoritmos de segmentação, consideramos as seguintes variáveis nas avaliações: tipo de descritor (MFCC, ALLDs ou DDBM), tipo de descritor dinâmico (DDC), e o parâmetro de memória temporal para a construção do descritor dinâmico. Além disso, estamos trabalhando com dois tipos de medidas de erro: EER e EPT.

O arcabouço de avaliação de segmentação foi construído em MATLAB® e alguns detalhes de implementação dos métodos de segmentação são apresentados abaixo:

- **HMM**

Nesta técnica utilizamos HMM contínua, com duas Gaussianas por estado. Os centróides e as matrizes de covariância foram inicializados com o algoritmo GMM. A biblioteca HMM utilizada para este experimento foi codificada por Kevin Murphy (2005)¹.

- **GMM**

Utilizamos a biblioteca Netlab² (Nabney, 2002), com duas Gaussianas por classe, que representa a seção a ser classificada. Na configuração da biblioteca, utilizamos matrizes de covariância total e um número máximo de 10 iterações para o algoritmo EM.

- **MLP, K-NN e NBC**

Para estes métodos, foi utilizada a biblioteca Weka³ com a configuração padrão de cada classificador.

- **MDISS**

Neste método, além de ser necessário informar a duração da memória temporal, temos que informar também o tempo em segundos que corresponde à vizinhança da observação sendo analisada, e para esta variável utilizamos um valor fixo de 1 segundo, estimado empiricamente através dos dados de treinamento.

- **MDISS-IPROC**

Neste método utilizamos as funções de processamento de imagem disponibilizadas no MATLAB®. Para o limiar de Otsu utilizamos a função *graythresh* e para os operadores morfológicos utilizamos a função *bwmorph* com 35 repetições nos seguintes operadores: *dilate*, *bridge* e *thin*. Esta técnica se mostrou bastante sensível a estes valores pré-fixados, o que torna o método pouco flexível.

¹Biblioteca HMM de Kevin Murphy – <http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html>. Existe uma versão mais recente que não utilizamos, pois quando construímos os modelos, a nova versão não havia sido lançada. O leitor interessado pode encontrá-la em <http://code.google.com/p/pmtk3/>

²O código original se encontra em <http://www1.aston.ac.uk/eas/research/groups/nrcg/resources/netlab/>. O código teve de ser alterado em dois pontos: o primeiro para resolver um problema de conflito na função *kmeans* com o MATLAB®, e outra modificação no arquivo *gmmem.m*, onde inserimos mais uma verificação na matriz de covariância, para garantir que ela seja positiva definida.

³Weka é uma biblioteca em Java desenvolvida pelo grupo de aprendizado de Máquina da Universidade de Waikato – <http://www.cs.waikato.ac.nz/ml/weka/>

- **COOPER/FOOTE**

Neste método utilizamos um valor fixo de L (ver equação 4.26) correspondente a 9 segundos de sinal de áudio, como aquele utilizado pelos autores (Cooper e Foote, 2002).

- **TZ/COOK** Neste método é necessário informar o tamanho mínimo de uma seção que desejamos encontrar e, desta forma, encontrar os picos que correspondem ao início de cada seção. Utilizamos um tamanho mínimo de seção correspondente a 9 segundos.

- **MPS-PBR-1, MPS-PBR-2 e MPS-PBR-3**

Nestes métodos, o limiar α para o segundo passo foi fixado em 0.99.

- **MPS-PBR-4**

Neste método, o segundo passo agrupa os estados potenciais através do método *kmeans* disponibilizado pelo MATLAB®.

Taxa de erro. A taxa de erro exibida nas figuras e tabelas são calculadas através da média dos erros gerados pela segmentação do conjunto de teste, cujas observações são geralmente agrupadas de acordo com a configuração de tipo de descritor. Nesta apresentação dos resultados vamos mostrar somente o erro EPT para simplificar a exposição e, ao expor os resultados, exibiremos os resultados obtidos por todas as técnicas de segmentação, com a finalidade de comparação.

A tabela 5.5 exibe os erros gerados para cada método de segmentação. O erro médio da melhor configuração foi de 1.89% para o segmentador supervisionado *K*-NN, com a seguinte configuração: família de descritor ALLDs, tipo de descritor dinâmico *weighted* e memória temporal de 1 segundo. Veja figura 5.6.

<i>f</i>	EPT	Desvio Padrão
BIC	11.28%	11.41
CUSUM	55.49%	10.59
HIPER-ELIPSE	19.12%	7.65
AHMM-1	14.06%	12.64
AHMM-2	21.90%	18.08
AHMM-3	20.48%	14.67
AHMM-4	27.14%	15.47
MPS-PBR-1	26.36%	18.99
MPS-PBR-2	44.69%	33.96
MPS-PBR-3	437.06%	25.12
MPS-PBR-4	29.44%	10.99
MPS-GHMM-1	22.12%	14.49
MPS-GHMM-2	37.23%	25.10
MPS-GHMM-3	16.55%	9.85
TZ/COOK	36.26%	19.81
COOPER/FOOTE	33.16%	15.65
MDISS	42.52%	14.76
MDISS-IPROC	54.71%	9.67
NBC	3.65%	2.23
KNN	1.89%	1.66
J48	3.35%	2.31
MLP	2.66%	1.70
GMM	5.10%	3.58
HMM	28.87%	24.64

Tabela 5.5: Erros EPT para os métodos de segmentação configurados com família de descritor ALLDs, tipo de descritor dinâmico *weighted* e memória temporal de 1 segundo.

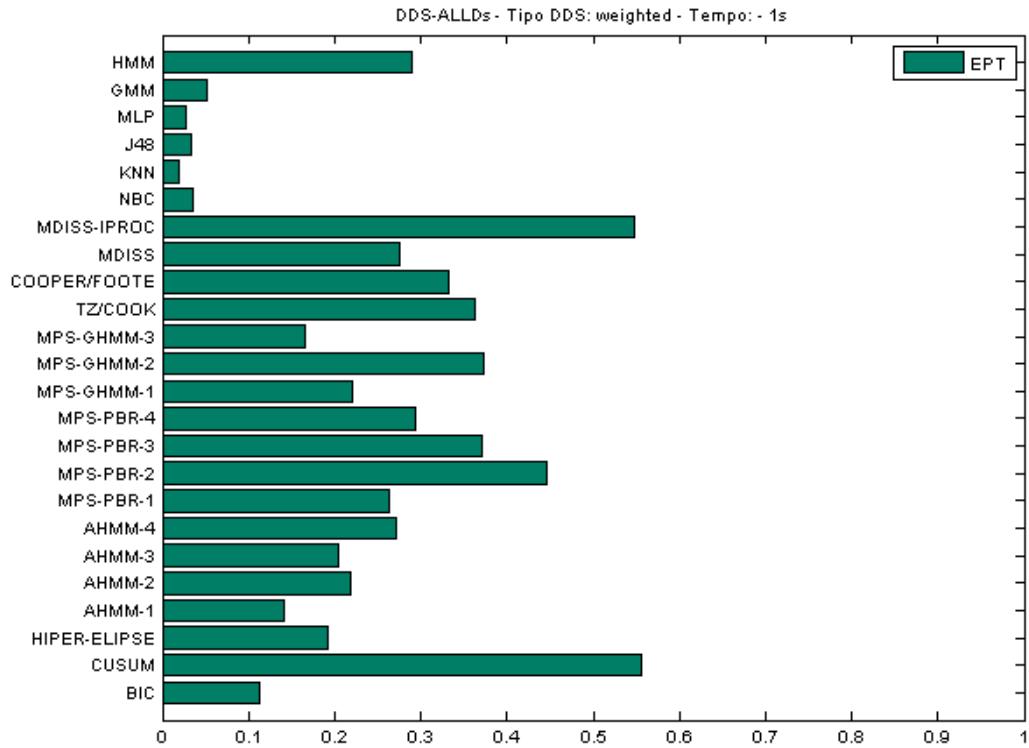


Figura 5.5: Gráfico de erro EPT versus técnicas de segmentação, com ALLDs, DDC weighted e memória temporal de 1 segundo.

Dos erros médios de nossa avaliação, podemos também extrair quais foram os erros médios para as técnicas não-supervisionadas. A figura 5.7 exibe o melhor resultado da segmentação para as técnicas não-supervisionadas com descritores MFCC, DDC *weighted* e memória temporal de 5 segundos. Com esta configuração, o segmentador MDISS gerou um erro de 5.23%. Entre as técnicas de segmentação não-supervisionadas em tempo real, as taxas de erro podem ser vistas na figura 5.8, onde podemos ver que o melhor resultado foi de 8.88% para o segmentador BIC com a seguinte configuração: descritor DDBM e memória temporal de 5 segundos.

Por último exibiremos uma tabela para futuras comparações entre os métodos. Os resultados da tabela 5.6 foram gerados por segmentadores utilizando os descritores MFCC sem qualquer geração de DDC. Desta tabela destacamos os desempenhos de MLP, com 2.9% de erro.

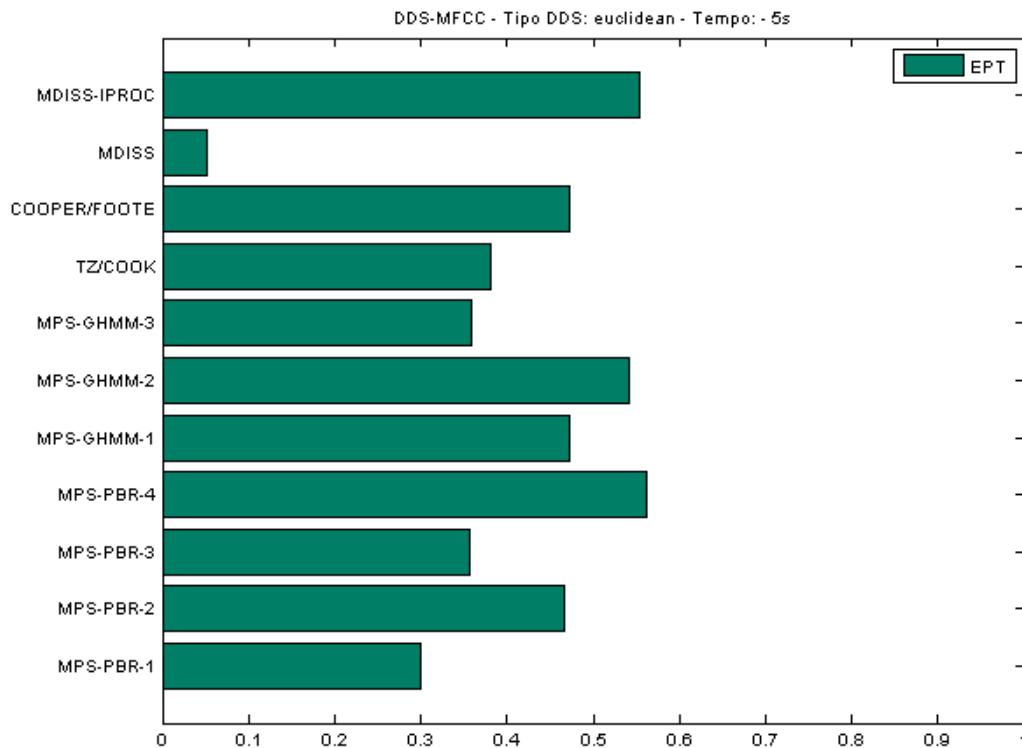


Figura 5.6: Gráfico de erro EPT versus técnicas de segmentação não-supervisionadas, com MFCC, DDC Euclidean e memória temporal de 5 segundos.

<i>f</i>	EPT	Desvio Padrão
BIC	18.4%	15.19
CUSUM	67.0%	3.88
HIPER-ELIPSE	37.1%	20.80
AHMM-1	25.7%	8.83
AHMM-2	25.3%	10.74
AHMM-3	40.8%	35.77
AHMM-4	21.4%	14.47
MPS-PBR-1	73.8%	15.12
MPS-PBR-2	73.8%	15.12
MPS-PBR-3	73.8%	15.12
MPS-PBR-4	73.8%	15.12
MPS-GHMM-1	57.6%	21.97
MPS-GHMM-2	15.1%	9.43
MPS-GHMM-3	44.1%	34.76
TZ/COOK	60.8%	14.98
COOPER/FOOTE	41.63%	6.54
MDISS	20.6%	21.01
MDISS-IPROC	40.1%	12.11
NBC	3.86%	1.59
KNN	2.90%	1.16
J48	3.77%	2.05
MLP	2.90%	1.47
GMM	6.23%	4.01
HMM	67.0%	18.76

Tabela 5.6: Erros EPT para os métodos de segmentação configurados com MFCC sem geração de descriptor dinâmico.

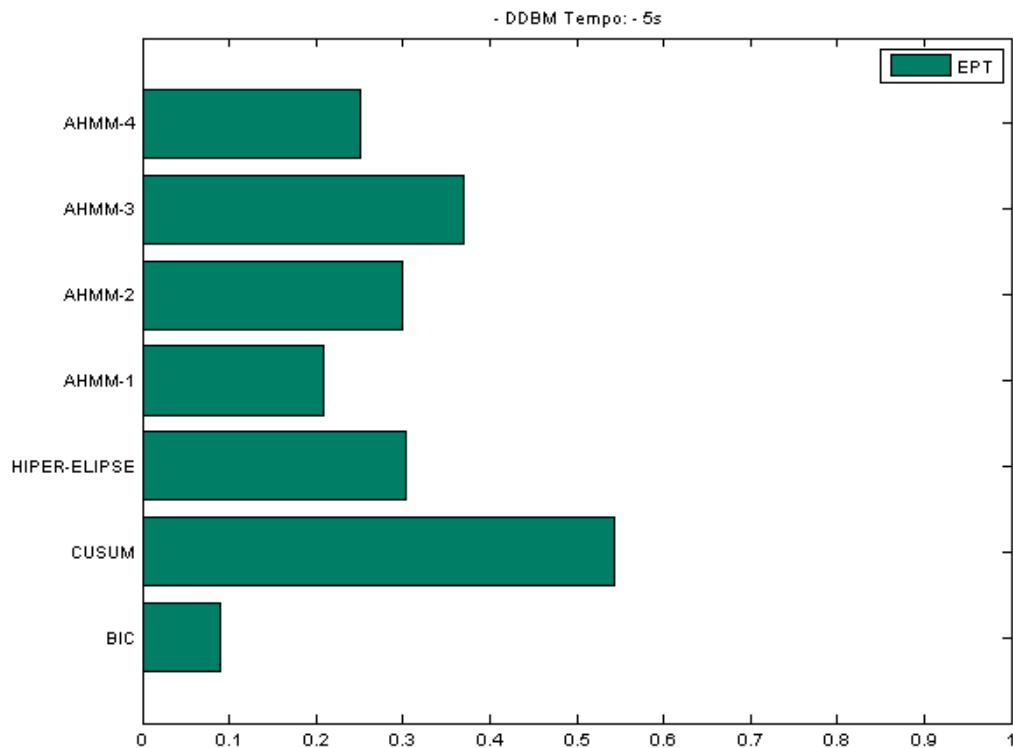


Figura 5.7: Gráfico de erro EPT versus técnicas de segmentação não-supervisionadas, com DDBM e memória temporal de 5 segundos.

Considerando cada técnica individualmente, encontramos as taxas médias de erro mínimas para cada uma delas. A tabela 5.7 contém os erros e as configurações dos descritores utilizadas. Por esta tabela podemos extrair as informações de quais foram os descritores que tiveram uma maior aderência aos segmentadores e a figura 5.4 exibe a árvore de descritores utilizados nesta tabela.

Desta forma, vemos que o descritor DDBM teve uma baixa aderência entre os segmentadores, mas quando utilizado com memória temporal de 5 segundos, conferiu ao segmentador BIC uma taxa de erro relativamente baixa, considerando que se trata de uma técnica não-supervisionada em tempo real. O descritor ALLDs teve a segunda maior aderência entre os segmentadores, tanto com tipos de DDC *weighted* quanto *Euclidean* e, com este último, o segmentador MPS-GHMM-3 obteve uma taxa de erro de 6.8%, que é o segundo melhor resultado entre os segmentadores não-supervisionados. Vemos que a maior parte dos segmentadores obteve um melhor desempenho utilizando descritores da família MFCC, sendo que metade deles com tipo de DDC *weighted*. O tipo de descritor MFCC com tipo de DDC *Euclidean* e *fft* estão presentes com o mesmo peso entre os segmentadores, mas dentre estes a menor taxa de erro é de 5.23% do segmentador MDIS, utilizando MFCC com tipo de DDC *Euclidean* e memória temporal de 5 segundos.

Nas próximas subseções vamos apresentar os resultados de acordo com outras visões, e não somente pela visão do menor erro. Separamos os resultados em três grupos de informações: Visão de Tipo de Descritor, Visão de Tipo de Descritor Dinâmico e Visão de Memória Temporal.

5.2.1 Visão de Tipo de Descritor

Nesta visão queremos analisar o desempenho dos segmentadores em termos de conjunto de descritores, ou seja, para uma combinação específica de tipo de descritor dinâmico cumulativo (e.g. norma Euclidiana) e um valor de tempo para sua geração, queremos comparar o desempenho

<i>f</i>	EPT médio	Desvio Padrão	Descriptor	DDC	Memória Temporal (seg.)
BIC	8.87%	5.49	DDBM		5
CUSUM	51.21%	10.60	MFCC	Euclidean	1
HIPER-ELIPSE	13%	10.82	ALLDs	Euclidean	0.5
AHMM-1	12.26%	8.04	ALLDs	weighted	10
AHMM-2	13.81	8.00	MFCC	fft	1
AHMM-3	12.18%	15.21	ALLDs	fft	0.5
AHMM-4	11.90%	11.99	ALLDs	Euclidean	0.5
MPS-PBR-1	17.22%	9.55	MFCC	weighted	5
MPS-PBR-2	19.18%	10.75	DDBM		5
MPS-PBR-3	16.29%	8.66	MFCC	weighted	5
MPS-PBR-4	15.25%	7.79	MFCC	weighted	5
MPS-GHMM-1	19.55%	22.79	MFCC	weighted	0.5
MPS-GHMM-2	15.16%	9.43	MFCC		
MPS-GHMM-3	6.8%	4.04	ALLDs	Euclidean	0.5
TZ/COOK	23.01%	13.21	MFCC	fft	0.5
COOPER/FOOTE	32.96%	11.20	ALLDs	fft	5
MDISS	5.23%	4.06	MFCC	Euclidean	5
MDISS-IPROC	22.49%	24.98	MFCC	moments	0.5
NBC	3.42%	2.08	MFCC	weighted	0.5
KNN	1.89%	1.66	ALLDs	weighted	1
J48	3.03%	1.60	ALLDs	weighted	0.5
MLP	2.14%	0.94	MFCC	weighted	0.5
GMM	4.34%	2.94	ALLDs		
HMM	13.47%	8.15	DDBM		5

Tabela 5.7: Erros médios e as configurações de descritores para cada técnica de segmentação.

dos segmentadores com os diferentes conjuntos de descritores: MFCC, ALLDs e DDBM. Existe também o caso em que não são gerados descritores dinâmicos e, neste caso, podemos comparar o desempenho dos segmentadores utilizando os descritores MFCC ou ALLDs da forma como foram extraídos do sinal de áudio.

Uma maneira pouco eficaz de comparar os descritores é calcular o erro médio agrupado pelo tipo de DDC e tipo de descritor, como exibido na figura 5.9. O problema com esta visualização é que os erros de todas as técnicas estão agrupados, e podemos não ter um entendimento de qual é a memória temporal que tem o melhor ou pior desempenho. Outro exemplo pode ser visto na figura 5.10, onde exibimos o erro médio agrupado por memória temporal e tipo de descritor. Neste exemplo perdemos a informação de qual tipo de DDC proporciona o pior ou melhor desempenho.

Apesar destes erros médios nos fornecerem uma compreensão fraca dos resultados, eles exibem algumas pistas de como os segmentadores se comportaram para cada tipo de descritor. Vemos, por exemplo na figura 5.9, que quando não são gerados DDC (*none*), os segmentadores apresentam na média quase o mesmo desempenho para ambos os descritores. Entretanto, para os descritores gerados com DDC, somente com *momentos estatísticos* é que o desempenho dos segmentadores com MFCC supera ALLDs.

A figura 5.11 exibe o desempenho dos segmentadores ao utilizar somente os descritores MFCC e ALLDs, sem geração de descritores dinâmicos. Estes resultados mostram que a maioria dos segmentadores obtiveram um melhor desempenho utilizando os descritores MFCC. Ao verificar, por exemplo, BIC com MFCC, o erro é 18.4% e com ALLDs o erro é de 16.3%. Por outro lado, o segmentador MPS-GHMM-2 obteve um melhor desempenho utilizando MFCC, com 15.1% contra 80.1% utilizando ALLDs.

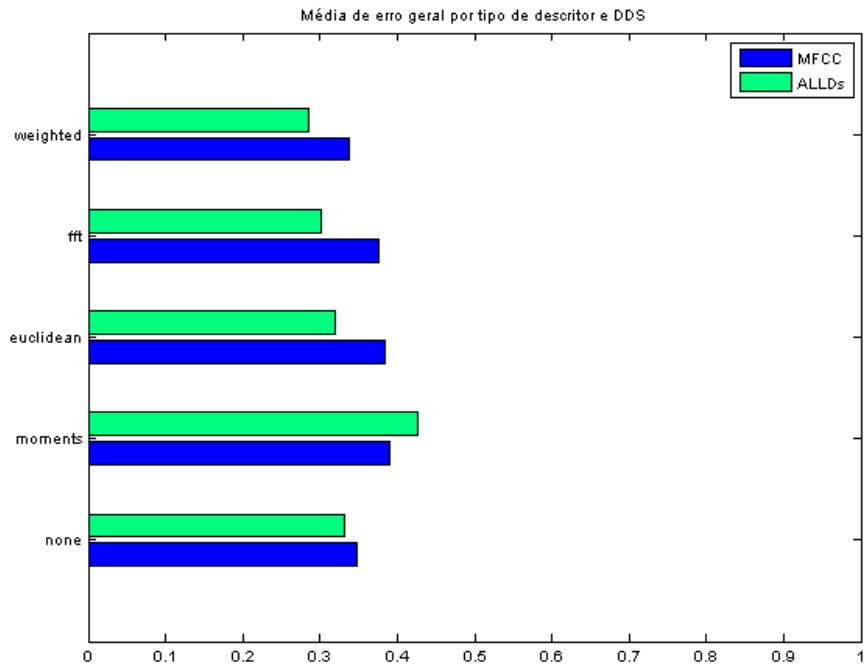


Figura 5.8: Erro médio agrupado por tipo DDC e tipo de descritor (MFCC e ALLDs).

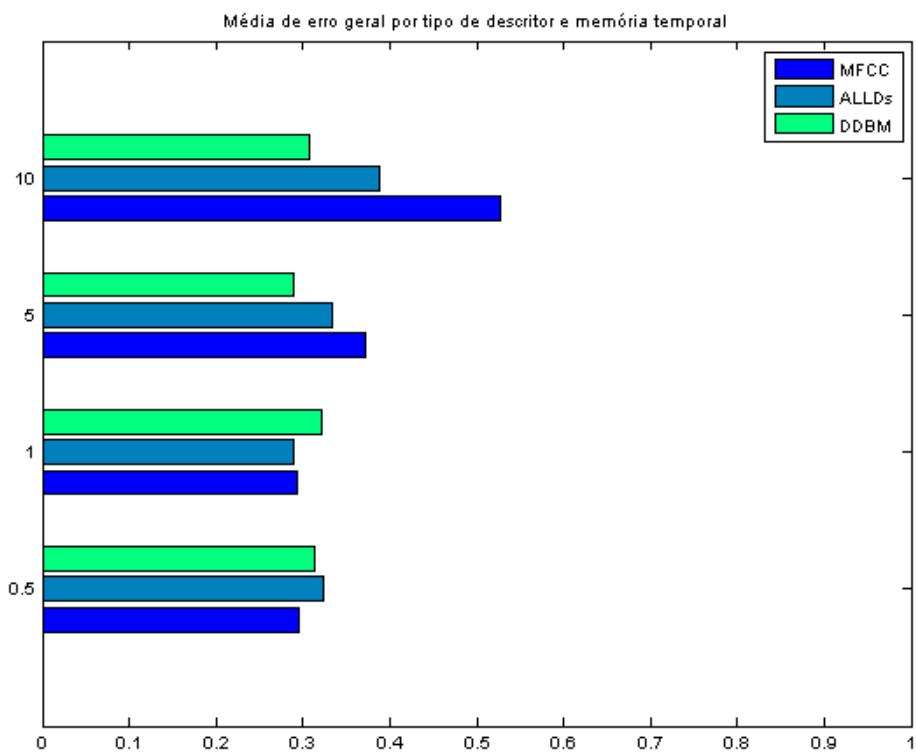


Figura 5.9: Erro médio agrupado por tipo DDC e tipo de descritor (MFCC e ALLDs).

No momento em que começamos a gerar descritores dinâmicos, podemos inserir a variável de tempo e de tipo de descritor dinâmico. As figuras 5.12, 5.13, 5.14 e 5.15 mostram os resultados onde ocorrem os erros médios das melhores configurações para cada descritor dinâmico gerado:

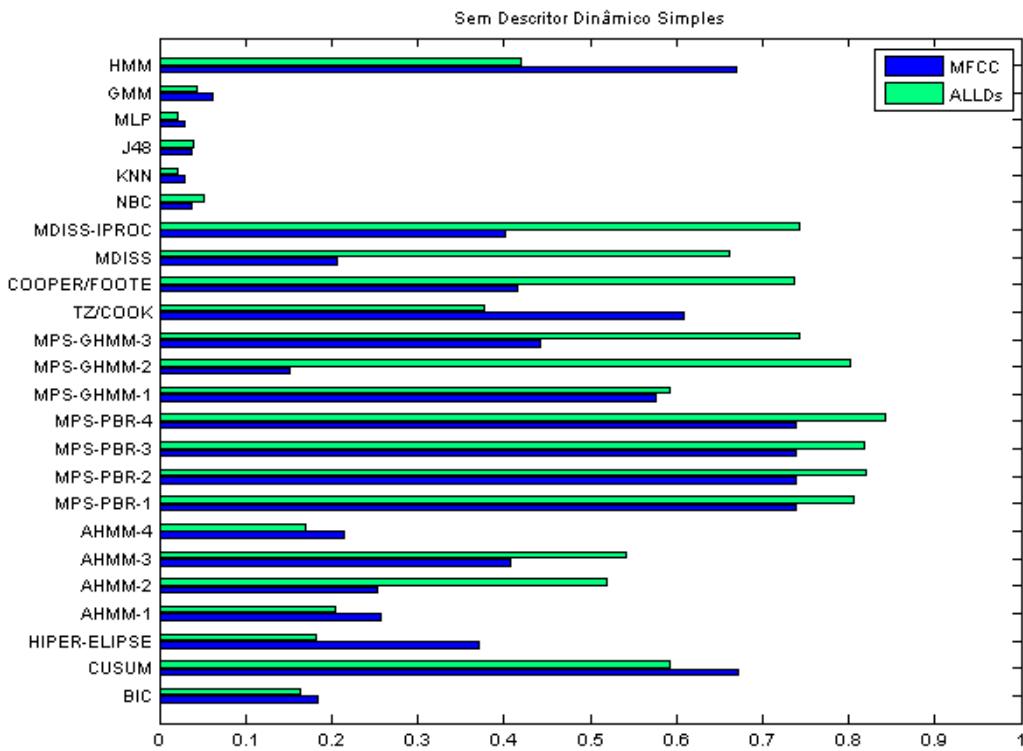


Figura 5.10: Gráfico de erro EPT com descritores MFCC e ALLDs.

norma Euclidiana, momentos estatísticos, média ponderada e coeficientes da FFT, respectivamente. Apesar de exibirmos aqui a configuração onde foi gerado o erro médio da melhor configuração para cada segmentador, não é exatamente isto o que nos interessa, mas sim a comparação com os diferentes tipos de descritores. Escolhemos este ponto da configuração em particular, pois ficaria um exercício muito exaustivo mostrar todas as combinações de configurações.

Ao avaliarmos o DDC *Euclidean* (fig. 5.12), verificamos que o parâmetro de memória temporal que gerou o menor erro foi de 0.5 segundos e, de modo geral, os segmentadores tiveram um melhor desempenho com os descritores ALLDs nos três grupos de segmentadores. O mesmo não ocorre quando comparamos os DDC *moments* e DDBM com memória temporal de 0.5 segundos (fig. 5.13) onde, neste caso, os descritores MFCC forneceram os melhores dados para os segmentadores supervisionados e não-supervisionados em tempo real (e.g. BIC, que teve um erro de 16.8%), enquanto os DDBM forneceram os melhores dados para os segmentadores não-supervisionados (e.g. MPS-PBR-3 que teve um erro de 21.2%).

Quando comparamos os descritores DDC *weighted* e DDBM com memória temporal de 1 segundo (fig. 5.14), vemos que os segmentadores com descritores DDC *weighted* têm um desempenho superior ao DDBM. Nos segmentadores supervisionados, o menor erro encontrado foi de 1.89% com o método K-NN e ALLDs. Nos segmentadores não-supervisionados, o menor erro encontrado foi de 7.47% com MPS-GHMM-3 e MFCC, e nos segmentadores não-supervisionados em tempo real, o menor erro foi de 11.2% com BIC e ALLDs, ambos com memória temporal de 1 segundo.

Por último, na comparação dos descritores DDC *fft* e DDBM com memória temporal de 0.5 segundos (fig. 5.15), vemos que os segmentadores com descritores ALLDs têm um desempenho superior. Nos segmentadores supervisionados, o erro médio foi de 3.56% com K-NN utilizando ALLDs. Destacamos somente o desempenho de NBC com MFCC, onde o erro foi de 5.65%. Nos segmentadores supervisionados os segmentadores tiveram um melhor desempenho com MFCC e DDBM, onde o erro médio foi de 8.83% com MPS-GHMM-3 e DDBM, enquanto que nos segmen-

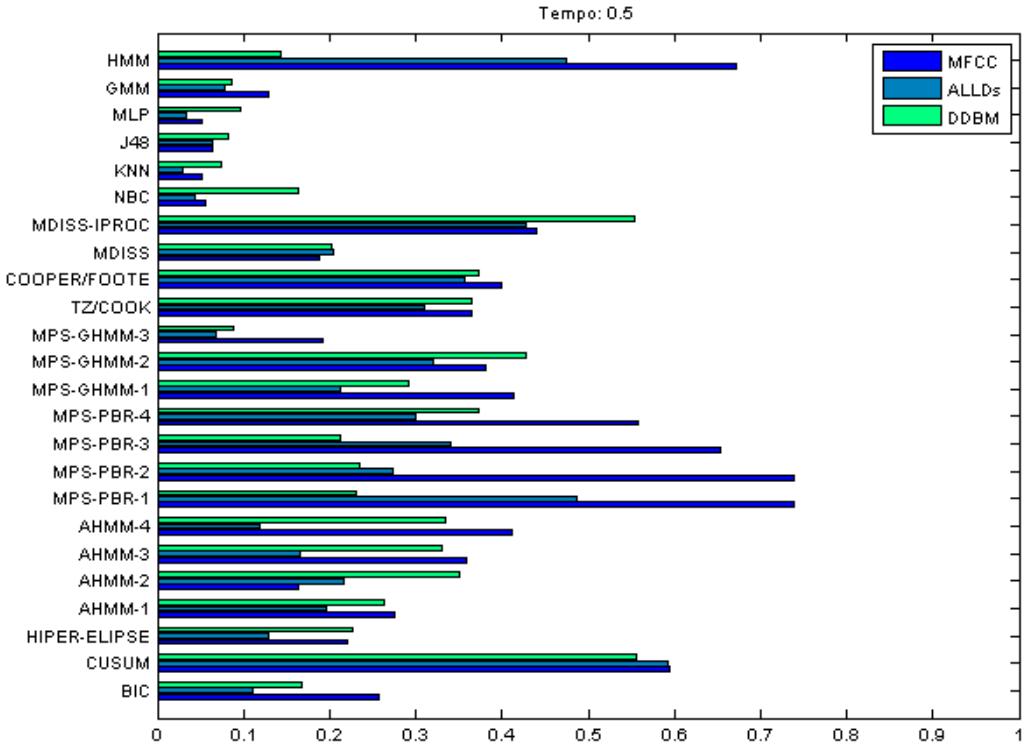


Figura 5.11: Gráfico de erro EPT com descritores MFCC e ALLDs, gerados com DDC Euclidean e DDBM, ambos gerados com memória temporal de 0.5 segundos.

tadores não-supervisionados em tempo real, os segmentadores tiveram um melhor desempenho com ALLDs, onde a taxa de erro foi de 12.1% com AHMM-3.

5.2.2 Visão de Tipo de Descritor Dinâmico

Nesta visão queremos comparar o desempenho dos diferentes tipos de descritores dinâmicos cumulativos. Neste caso, podemos compará-los se agruparmos os resultados por conjunto de descritores (MFCC ou ALLDs) e tipo de descritor dinâmico cumulativo (e.g. norma Euclidiana).

Do grupo MFCC, a configuração com a menor taxa de erro foi com memória temporal de 0.5 segundos. Neste caso, o erro médio encontrado para os segmentadores supervisionados foi de 2.14% com MLP e tipo de DDC *weighted* (ver figura 5.16). Este é um erro mais baixo que ao utilizar MFCC sem DDC, onde foi gerado um erro de 2.90% com o mesmo segmentador, como podemos ver na tabela 5.6.

Quando nos voltamos para os segmentadores não-supervisionados, incluindo em tempo real (ver figuras 5.17 e 5.18), o descritor MFCC com tipo de DDC *weighted* forneceu os melhores dados para a segmentação. Por exemplo, as taxas de erro de 19.4% e 7.39%, com os segmentadores AHMM-1 e MPS-GHMM-3, respectivamente.

Dos resultados do grupo ALLDs, a configuração com a menor taxa de erro foi aquela com memória temporal de 1 segundo. Os resultados dos segmentadores supervisionados podem ser visualizados na figura 5.19, onde vemos que os melhores resultados são obtidos quando utilizamos tipo de DDC *weighted*, tendo um resultado menor ou igual a 5.09% de erro (exceto para HMM).

Os segmentadores não-supervisionados tiveram bons resultados utilizando tanto DDC *weighted* quanto DDC *Euclidean* (ver figura 5.20), no entanto, a menor taxa de erro foi de 13.2% com o segmentador MPS-GHMM-3 e DDC *Euclidean*.

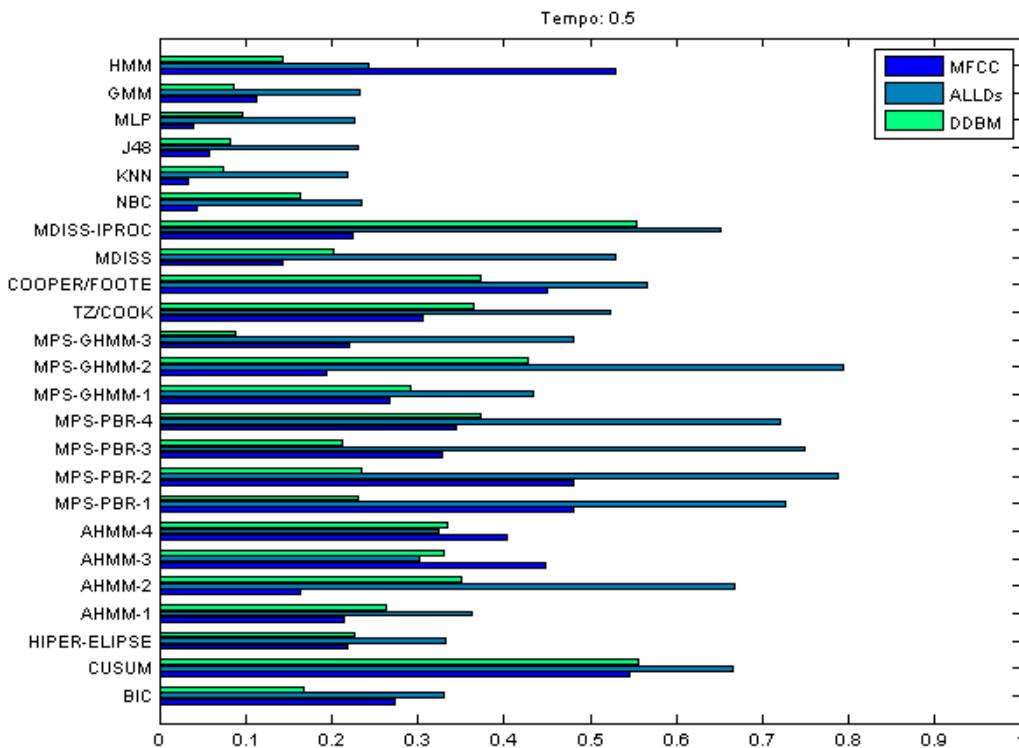


Figura 5.12: Gráfico de erro EPT com descritores MFCC e ALLDs, gerados com DDC moments e DDBM, ambos gerados com memória temporal de 0.5 segundos.

Por fim, para os segmentadores não-supervisionados em tempo real (ver figura 5.21), o melhor resultado foi de 11.2% com BIC e DDC *weighted*, mas o tipo de DDC *fft* gerou, na média, a menor taxa de erros (e.g. a taxa de erro de 13.1% com AHMM-3).

Ao comparar os resultados acima com a tabela base 5.6, vemos que em todos os casos a geração de DDC trouxe melhores resultados para os segmentadores. Por exemplo, houve uma diminuição de 30.9% da taxa de erro com o segmentador MPS-GHMM-3 com tipo de DDC *Euclidean*, uma melhora de 27.7% com o segmentador AHMM-3 com tipo de DDC *weighted*, e outras mais sutis, como, por exemplo, a melhora de 7.2% com o segmentador BIC com tipo DDC *weighted*.

5.2.3 Visão de Memória Temporal

Esta visão permite analisar o desempenho dos segmentadores pelo parâmetro de tempo do descritor dinâmico utilizado. Como ponto de partida para esta visão escolhemos os três pontos onde encontramos a menor taxa de erro para cada categoria de segmentadores.

No primeiro grupo temos os resultados para os segmentadores supervisionados e, como já foi visto, o menor erro foi de 1.89% gerado pelo segmentador K-NN com descritor ALLDs, tipo de DDC *weighted* com memória temporal de 1 segundo. Um ponto interessante desta visão para este grupo é o perfil dos erros gerados pelos segmentadores à medida que aumentamos a memória temporal. Veja na figura 5.22 que todos os outros métodos têm um desempenho melhor quando foram gerados com memórias temporais de 0.5 e 1 segundos, e quando não foi gerado descritor dinâmico, ou quando a duração deste parâmetro aumenta, o desempenho dos segmentadores diminui.

Para os segmentadores não-supervisionados (fig. 5.23), a menor taxa de erro foi de 5.23% com MDIIS, descritores MFCC, tipo de DDC *Euclidean* e memória temporal de 5 segundos. Destacamos aqui o perfil dos erros obtidos por MPS-GHMM-3, que obteve o segundo menor erro desta

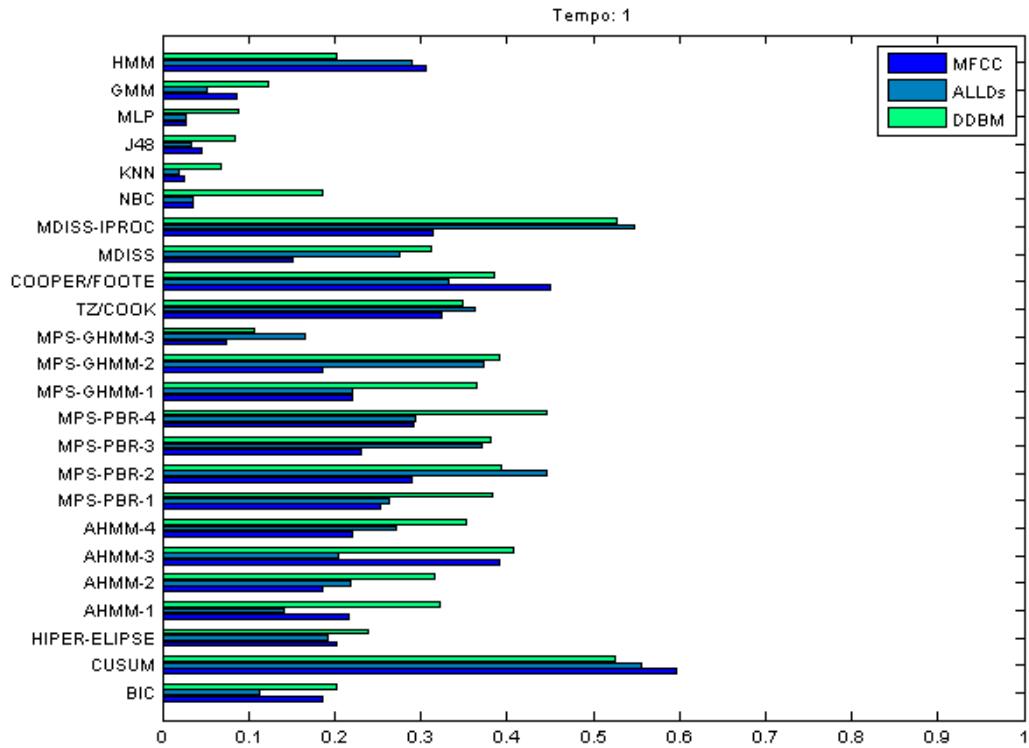


Figura 5.13: Gráfico de erro EPT com descritores MFCC e ALLDs, gerados com DDC weighted, e DDBM, ambos gerados com memória temporal de 1 segundo.

categoria, com 14.39% de erros com o descritor MFCC sem DDC, e à medida que aumentamos a memória temporal, este descritor passou para 19.24%, 30.01%, 35.79% e 60.14% de erro utilizando 0.5, 1, 5 e 10 segundos de memória temporal, respectivamente.

Na figura 5.24 vemos os erros dos métodos não-supervisionados em tempo real, gerados por DDBM. Esta visão não nos oferece muito mais do que já foi visto nos resultados anteriores, como o caso de BIC ter tido o melhor desempenho, com 8.87% de erros.

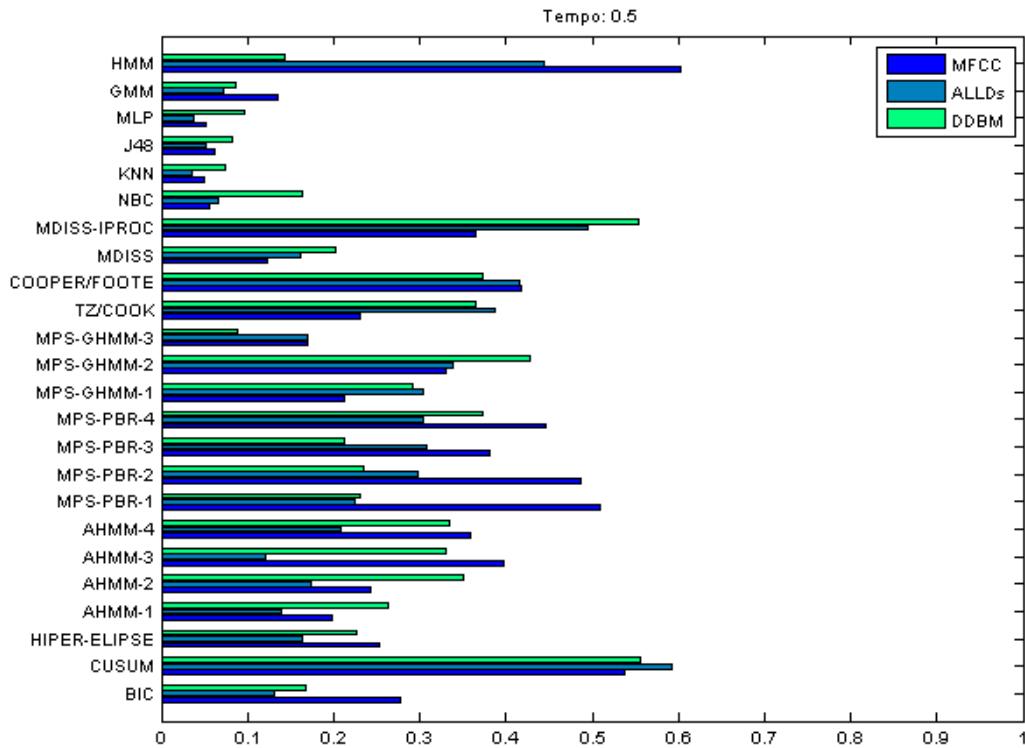


Figura 5.14: Gráfico de erro EPT com descritores MFCC e ALLDs, gerados com DDC fft, e DDBM, ambos gerados com memória temporal de 0.5 segundos.

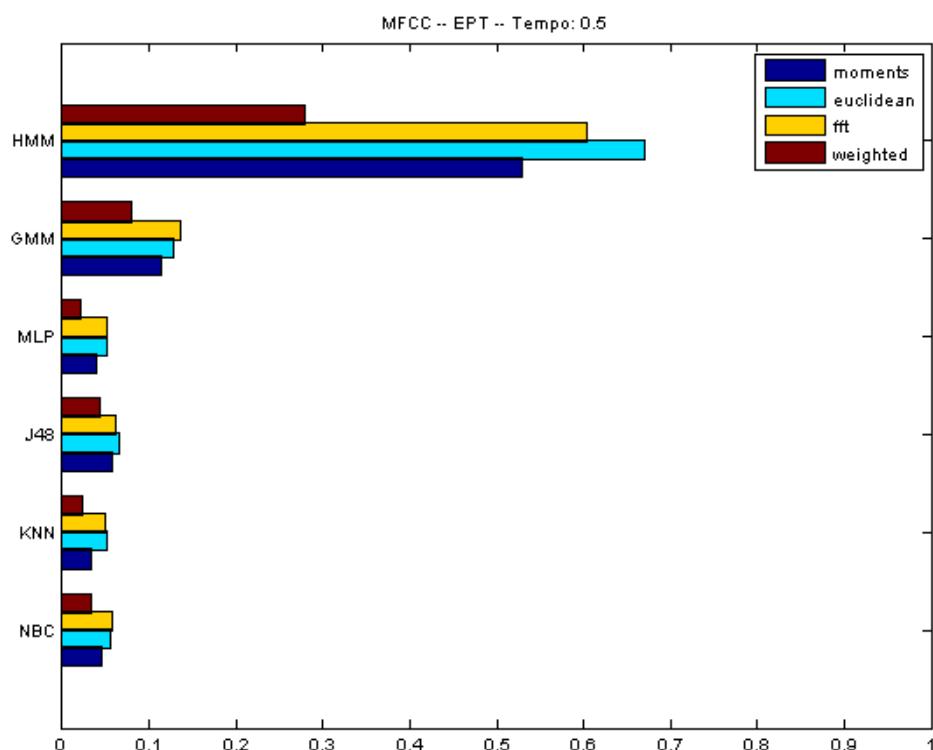


Figura 5.15: Gráfico de erro EPT para os segmentadores supervisionados, com descritores MFCC e memória temporal de 0.5 segundos, agrupados por tipo de DDC.

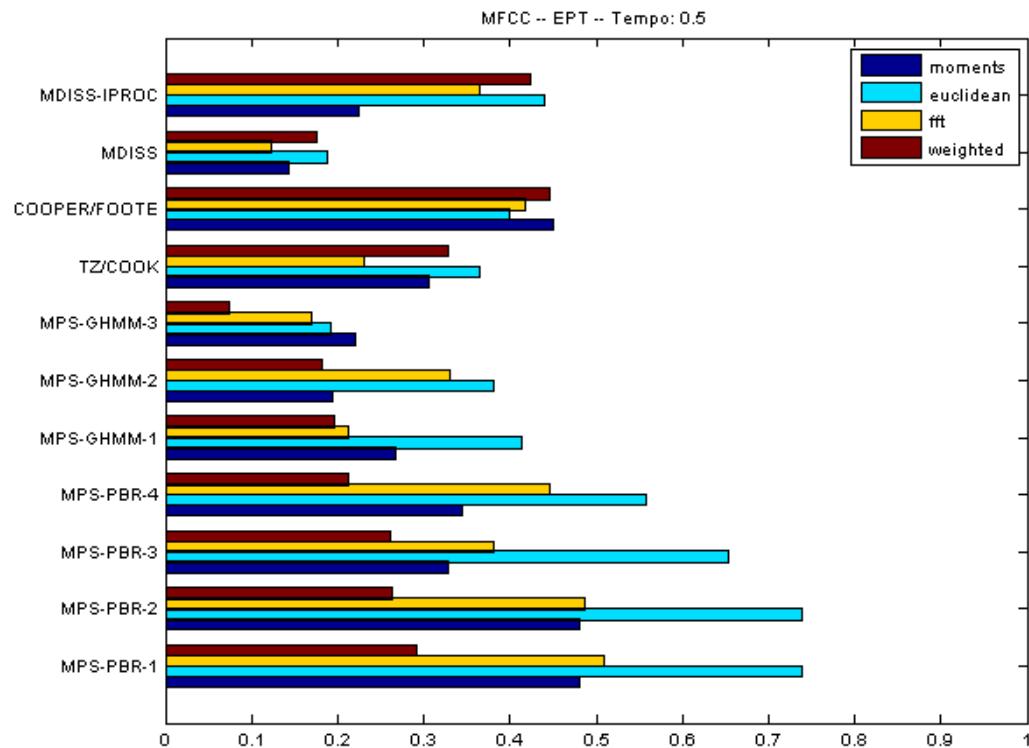


Figura 5.16: Gráfico de erro EPT para os segmentadores não-supervisionados, com descritores MFCC e memória temporal de 0.5 segundos, agrupados por tipo de DDC.

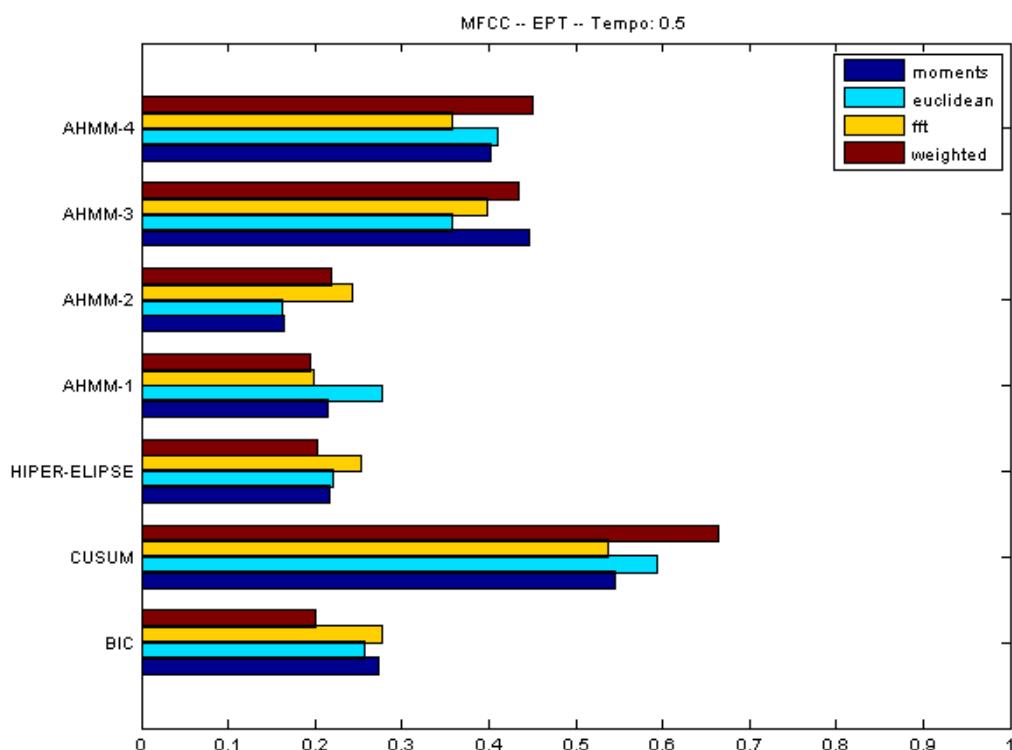


Figura 5.17: Gráfico de erro EPT para os segmentadores não-supervisionados em tempo real, com descritores MFCC e memória temporal de 0.5 segundos, agrupados por tipo de DDC.

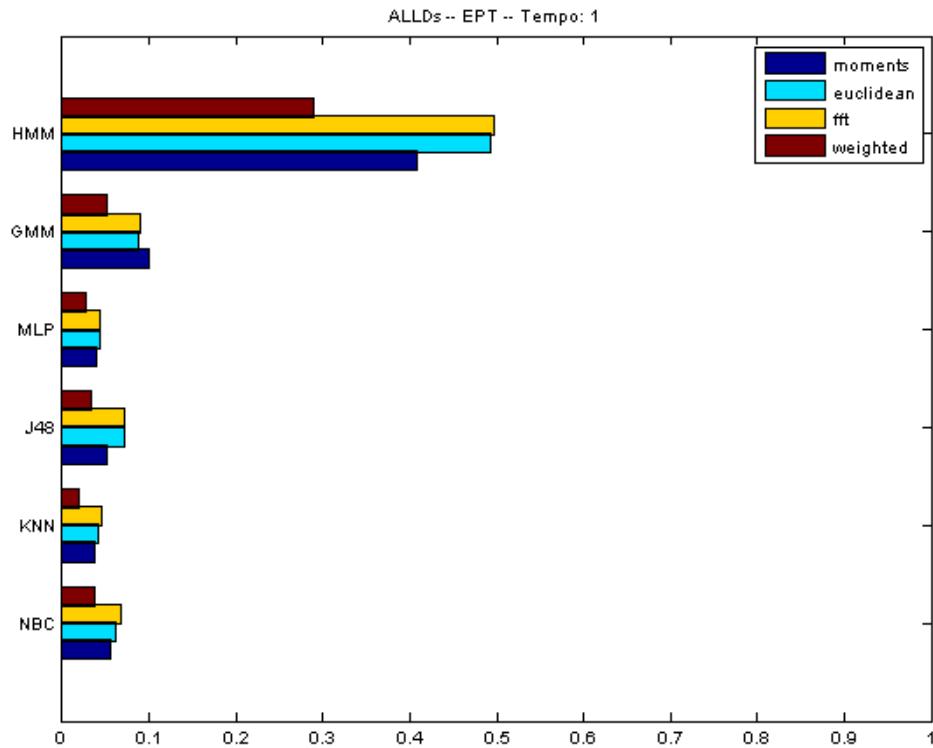


Figura 5.18: Gráfico de erro EPT para os segmentadores supervisionados, com descritores ALLDs e memória temporal de 1 segundo, agrupados por tipo de DDC.

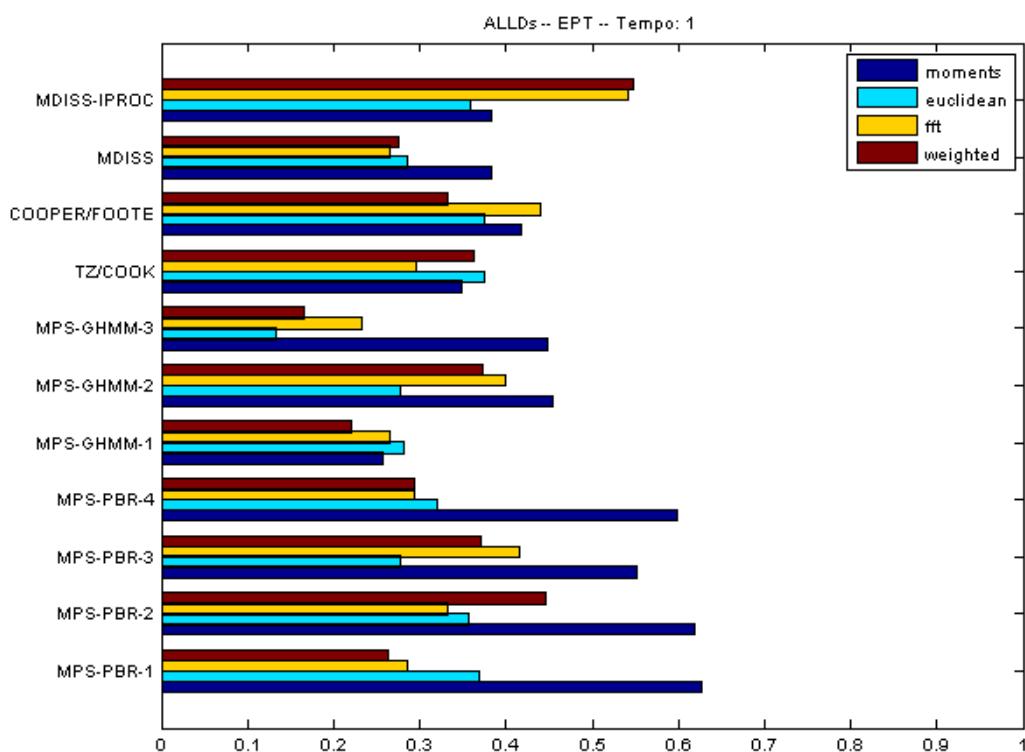


Figura 5.19: Gráfico de erro EPT para os segmentadores não-supervisionados, com descritores ALLDs e memória temporal de 1 segundo, agrupados por tipo de DDC.

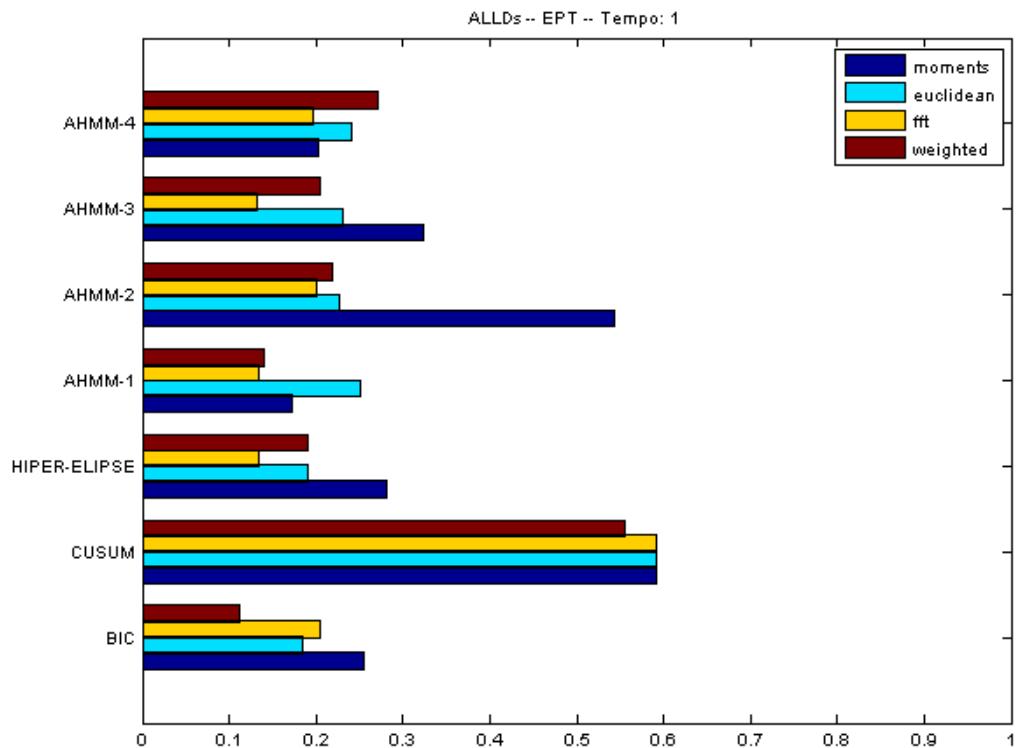


Figura 5.20: Gráfico de erro EPT para os segmentadores não-supervisionados em tempo real, com descritores ALLDs e memória temporal de 1 segundo, agrupados por tipo de DDC.

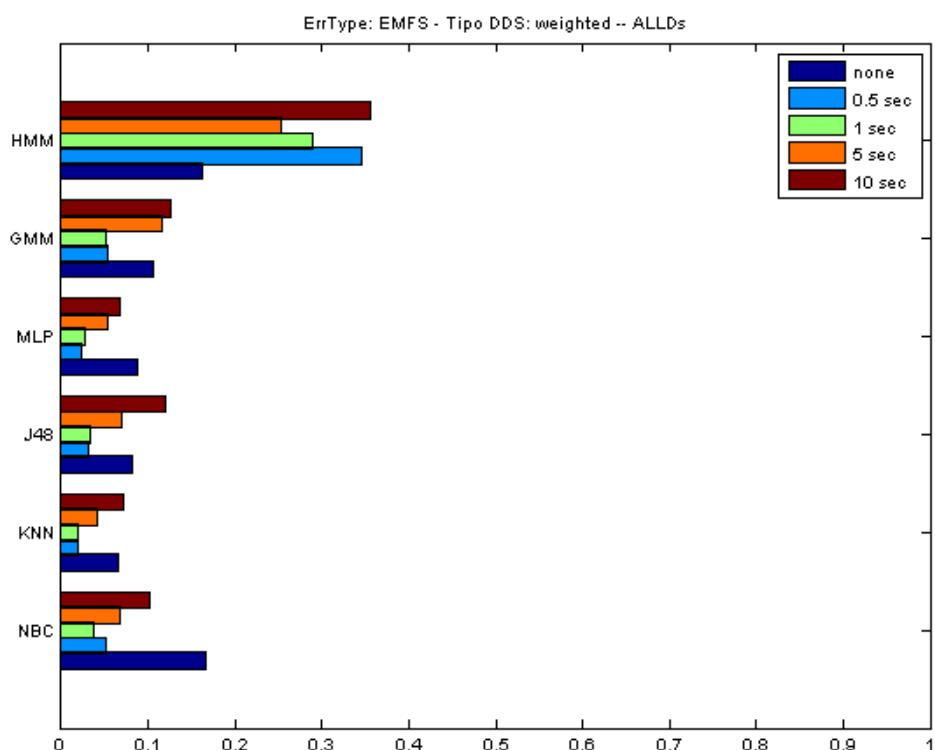


Figura 5.21: Gráfico de erro EPT para os segmentadores supervisionados, com descritores ALLDs e tipo de DDC weighted, agrupados por memória temporal.

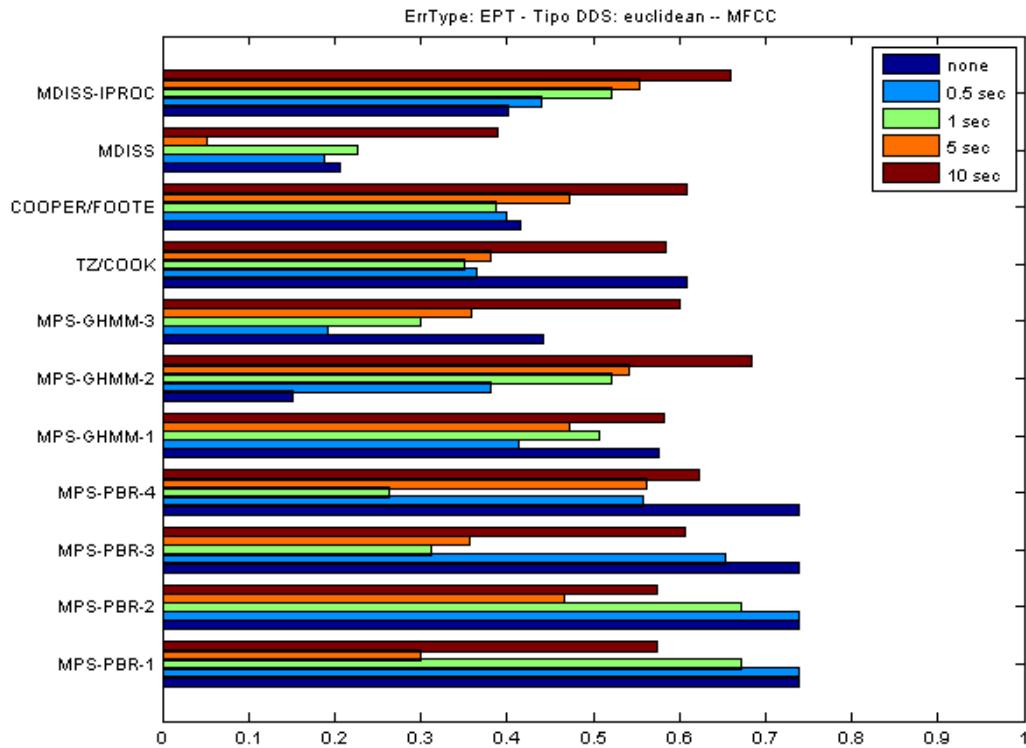


Figura 5.22: Gráfico de erro EPT para os segmentadores não-supervisionados, com descritores MFCC e tipo de DDC Euclidean, agrupados por memória temporal.

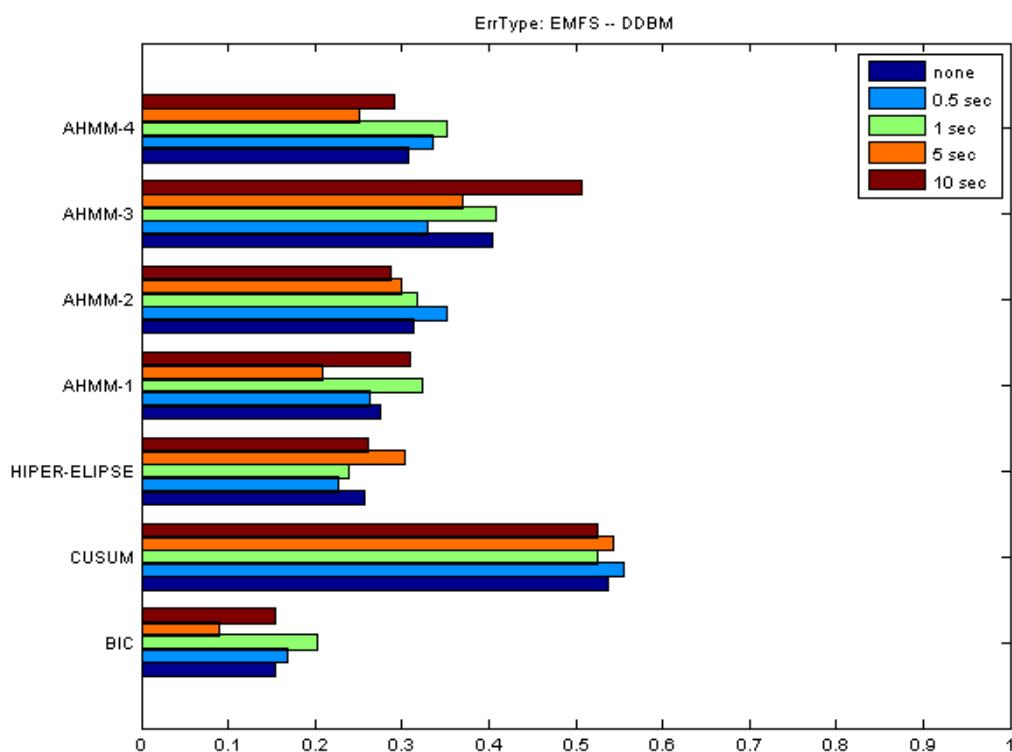


Figura 5.23: Gráfico de erro EPT para os segmentadores não-supervisionados em tempo real, com descritores DDBM, agrupados por memória temporal.

Capítulo 6

Conclusões

Neste capítulo vamos apresentar, de forma resumida, os assuntos abordados nesta dissertação e nossas conclusões acerca deste trabalho de pesquisa. Recordemos que o objetivo traçado no início desta dissertação é comparar diferentes técnicas de segmentação de sinais de áudio musicais, incluindo métodos de segmentação não-supervisionada em tempo real. Para tanto, utilizamos um método (capítulo 2) que compreende os seguintes passos:

- geração automática de dados para o problema de segmentação e geração de arquivos utilitários contendo informações a respeito das seções musicais do material sonoro,
- extração de descritores sonoros e geração de descritores dinâmicos cumulativos e por bandas mel,
- seleção automática de descritores baseada em um método de multi-critérios,
- execução da segmentação e rotulação, a fim de extrair a segmentação estrutural e
- avaliação dos resultados obtidos com base na segmentação de referência gerada no começo do processo.

Com respeito aos descritores, apresentamos técnicas de seleção de descritores para reduzir o espaço dimensional e métodos de geração de descritores dinâmicos com o objetivo de comparar as técnicas de segmentação e rotulação utilizando diferentes tipos de descritores, com diferentes valores de memória temporal. Ao executar nossos experimentos, verificamos que a modelagem temporal dos descritores estáticos (que corresponde aos descritores dinâmicos cumulativos) fornecem, na média, os melhores resultados para as técnicas de segmentação e que os descritores que fornecem as menores taxas de erro são aqueles que se recordam das informações passadas, porém sem enfatizar demasiadamente a informação temporal, e.g. MFCC com DDC *weighted* e memória temporal de 500ms. Identificamos também que os descritores MFCC foram os mais selecionados em nossos experimentos, e isto significa que, para os dados gerados de nossos experimentos, poderíamos utilizar diretamente os descritores MFCC ao invés de realizar uma etapa de seleção de descritores, sem grandes perdas de desempenho, além de economizar tempo computacional para a execução de tais tarefas.

Com relação à execução de segmentação e rotulação, expomos três grupos de métodos: supervisionadas, não-supervisionadas e não-supervisionadas em tempo real, totalizando 16 métodos distintos, sem considerar pequenas variações nas etapas intermediárias dos algoritmos. Dentre estes métodos, 5 deles são novas propostas ou variações de propostas conhecidas: AHMM, MPS-GHMM, MDIIS, MDIIS-IPROC e ECLIPSE. A partir dos resultados verificamos que as técnicas supervisionadas obtiveram os melhores resultados com 1,89% de erro para K-NN, e somente o método HMM sozinho teve um desempenho ruim, se comparado com as outras técnicas supervisionadas, atingindo uma taxa de erro de 13,47%, mas se utilizado em conjunto com outras técnicas, forneceu um bom resultado, como é o caso de MPS-GHMM. Dentre as técnicas não-supervisionadas, a MPS-GHMM e MDIIS foram as que obtiveram melhores resultados, com 6,8%

e 5, 23%, respectivamente. Das técnicas não-supervisionadas em tempo real, BIC e AHMM tiveram os melhores resultados, com 8, 87% e 11, 9% de erro, respectivamente.

6.1 Considerações Finais

Durante o projeto de pesquisa verificamos que os estudos relacionados ao objeto desta pesquisa são ainda recentes, no entanto vemos que o assunto é de interesse da comunidade científica. Algumas referências que encontramos durante a fase de construção do material teórico, consolidam o conhecimento da área até então, como é o caso de Dannenberg e Goto (2009), e enquanto direcionamos os esforços para a construção de nosso arcabouço de testes, outros trabalhos na área foram produzidos, e.g. Paulus *et al.* (2010), e é provável que tenhamos deixado passar algum outro material importante. Apesar desta consideração, acreditamos que esta dissertação cubra as principais características da segmentação estrutural musical, ainda que não seja possível fechar a discussão sobre este assunto.

Reconhecemos que os resultados da segmentação em tempo real não se equiparam aos das técnicas supervisionadas, e nem tínhamos a pretensão de ultrapassá-las. Vemos, porém, que os resultados são promissores e a taxa de erro medida com algumas de nossas propostas é relativamente baixa, considerando que se tratam de métodos não-supervisionados.

Os códigos gerados para a geração automática de dados musicais, métodos de seleção de descritores, segmentação e avaliação desenvolvidos durante este projeto, estão disponíveis em <http://compmus.ime.usp.br/music-segmentation>, juntamente com todos os gráficos com os resultados obtidos durante a fase de experimentos. Além disto, parte dos resultados gerados por este estudo está publicado em Pires e Queiroz (2011).

6.2 Sugestões para Pesquisas Futuras

Como ponto de partida para futuros trabalhos, sugerimos a leitura de Dannenberg e Goto (2009) que fornecem informações suficientes para uma visão geral dos problemas relacionados à segmentação musical e trabalhos realizados em análise de estrutura musical.

Outras medidas de avaliação de segmentação também são apontadas em Paulus *et al.* (2010) e pelo ISMIR¹, e mesmo depois de termos criado nossa própria estrutura, julgamos ser importante utilizar estas métricas padronizadas para comparar com resultados de outros trabalhos. Isto indica um esforço da comunidade científica para padronizar os resultados, e com isto aumentar a qualidade dos trabalhos desenvolvidos. Com isto, em um futuro trabalho, pode-se utilizar esta outra base de dados criada pela comunidade científica. Mais informações sobre as medidas padronizadas e a base de dados estão disponíveis em http://www.music-ir.org/mirex/wiki/-2010:Structural_Segmentation.

Sugerimos também que um estudo mais focado em sistemas em tempo real deva ser realizado para validar a viabilidade da segmentação em tempo real. Julgamos que este estudo possa responder questões sobre as complexidades computacionais de cada etapa da segmentação e sugerir as configurações necessárias para sua execução, como tamanho de janela de análise adequada e possíveis descritores sonoros e dinâmicos.

Dentre as finalidades práticas que podemos sugerir ao atingir os objetivos desta pesquisa, poder-se-ia, por exemplo, construir um produto de segmentação musical em tempo real. Este componente poderia ser desenvolvido, por exemplo, em uma ferramenta de programação visual, e.g. PureData², que possui a vantagem de ser bastante usado pela comunidade de computação musical em geral, incluindo músicos e artistas, que poderiam assim utilizar estas técnicas livremente e fornecer informações para sua melhoria. Neste sentido, podemos citar que já existem

¹<http://www.ismir.net/>

²<http://puredata.info/>

iniciativas de produtos para a extração de descritores sonoros em tempo real, e.g. Jaudio³ e LibXtract⁴.

Paulus *et al.* (2010) citam diversos trabalhos que não foram incluídos nesta pesquisa, e cremos que as novas perspectivas apresentadas devem ser estudadas e adicionadas ao arcabouço de avaliação de segmentadores construído; outra possibilidade é investigar a utilização de esquemas mais complexos de análise de estrutura musical, como florestas de HMM (Bicego *et al.*, 2006) e HMM auto-adaptativo (Liu e Chen, 2003).

³<http://jaudio.sourceforge.net/>

⁴<http://libxtract.sourceforge.net/>

Apêndice A

Descritores de Áudio

Neste apêndice estão contidas as informações sobre os descritores sonoros e as taxonomias de dois projetos de extração de descritores: o projeto CUIDADO (*Content-based Unified Interfaces and Descriptors for Audio/music Databases available Online*), coordenado pelo IRCAM, e o projeto JAudio da Universidade de McGill.

A.1 Taxonomia do Projeto CUIDADO

A taxonomia do projeto CUIDADO é dividida, no primeiro nível, de acordo com a extensão temporal, e no segundo nível, com o tipo de representação do sinal para sua extração. São, portanto, 5 subclasses discriminadas, de acordo com o tipo de representação do sinal, da seguinte forma:

Descritores Temporais (DT)

Descritores Temporais são descritores globais ou instantâneos calculados diretamente do sinal no domínio do tempo ou da envoltória da energia do sinal.

Descritores de Energia (DE)

Descritores de energia são descritores instantâneos que se referem às várias componentes de energia do sinal, seja através da envoltória de energia do sinal, do espectro ou do espectro harmônico do sinal.

Descritores Espectrais (DS)

Descritores Espectrais são descritores instantâneos que são calculados diretamente da Transformada de Fourier (STFT) do sinal.

Descritores Harmônicos (DH)

Descritores Harmônicos são descritores instantâneos e globais calculados a partir do modelo sinusoidal do sinal.

Descritores Perceptuais (DP)

Descritores Perceptuais são descritores instantâneos calculados através de um modelo do processo de escuta humana.

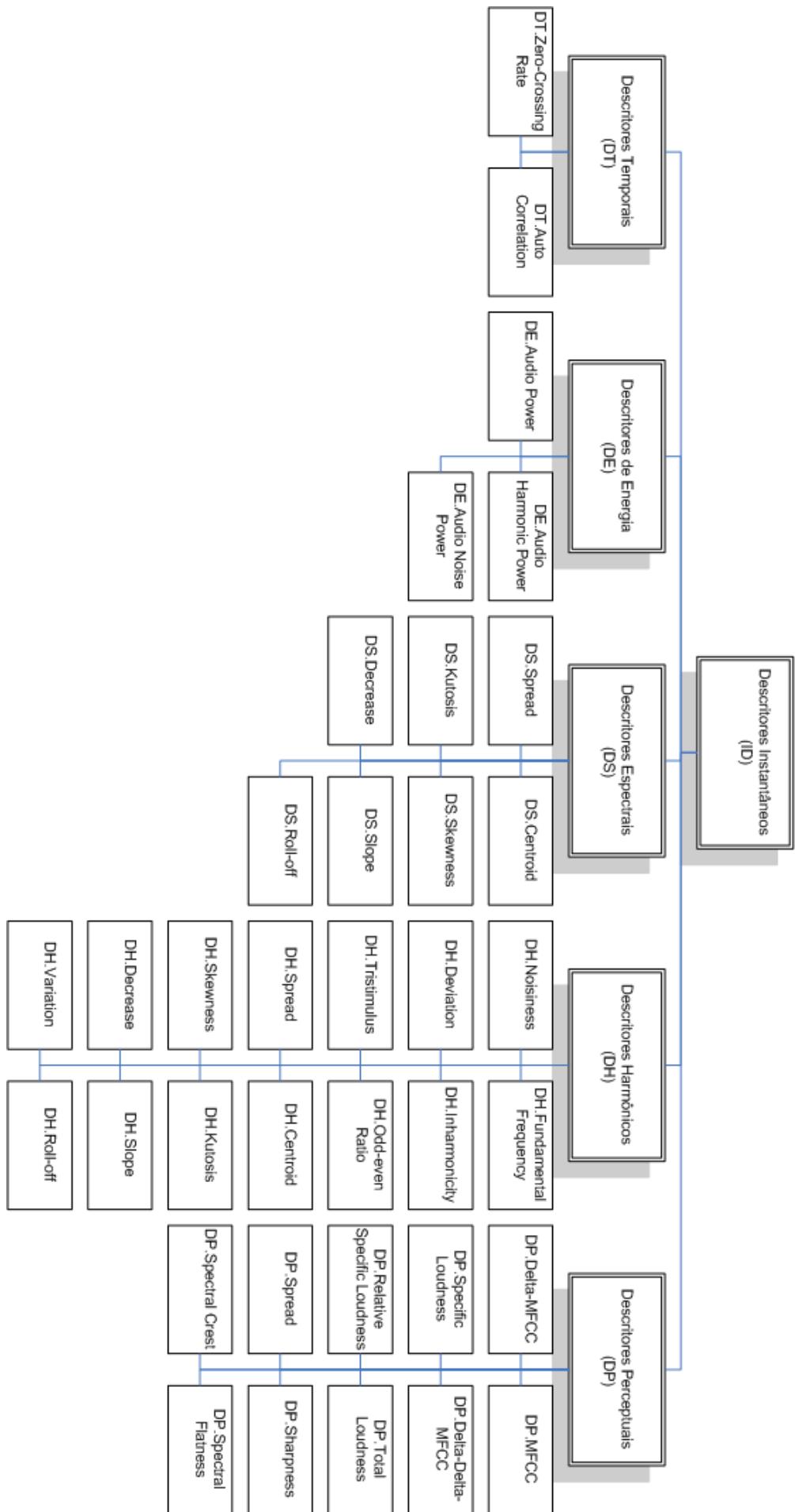


Figura A.1: Organização de Descritores Instantâneos do projeto CUIDADO

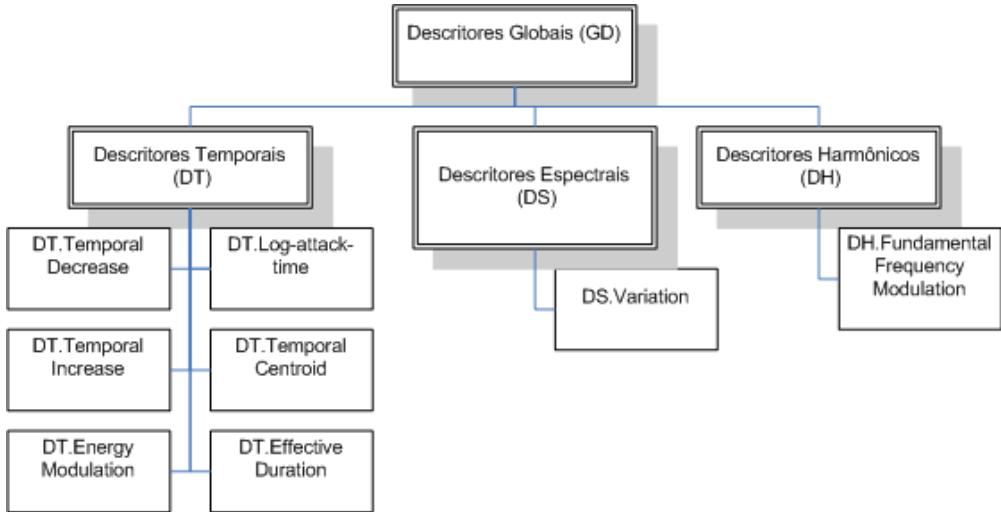


Figura A.2: Organização de Descritores Globais do projeto CUIDADO

Nesta seção vamos apresentar alguns dos descritores sonoros mais utilizados em aplicações de extração de informação musical. Para uma informação mais detalhada dos descritores, o leitor poderá consultar Peeters (2004). Para os descritores aqui apresentados, N é o tamanho da janela de análise, $s(n)$ é o n -ésimo valor (amostra) do sinal de áudio, $e(n)$ é a energia instantânea do sinal em sua n -ésima posição, e fr (frame rate) é a taxa de amostragem do sinal de áudio.

A.1.1 Descritores Temporais (Globais e Instantâneos)

Os descritores temporais são calculados diretamente do sinal no domínio do tempo, fornecendo descritores temporais instantâneos, ou calculados através da energia do sinal, fornecendo descritores temporais globais. A amplitude RMS do sinal pode ser calculada através da expressão:

$$e(n) = \sqrt{\frac{1}{N} \sum_{i=n-N+1}^n s(i)^2} \quad (\text{A.1})$$

DT.auto-correlation (Instantâneo)

O descritor de auto-correlação representa uma distribuição do espectro no domínio do tempo através de c coeficientes. Pode ser calculado da seguinte forma:

$$r(l, m) = x(0)^{-2} \sum_{n=m-N+1}^m x(n)x(n-l), \quad 0 < l < c \quad (\text{A.2})$$

onde $r(l, m)$ representa a similaridade do frame $x(n)$ com o frame deslocado de l amostras.

DT.zero-crossing-rate (Instantâneo)

O valor deste descritor informa o número de vezes que o sinal cruza o eixo zero.

$$zcr = \frac{1}{N} \sum_{n=1}^N T\{x(n)x(n-1) < 0\} \quad (\text{A.3})$$

onde $T\{X\}$ é 1 se o argumento X é verdadeiro e 0 caso contrário.

DT.log-attack-time (Global)

Este descriptor determina o logaritmo de tempo de ataque. É calculado pela fórmula:

$$lat = \log_{10} (ia - fa) \quad (\text{A.4})$$

onde ia e fa correspondem ao início e fim de um trecho de ataque no domínio do tempo, respectivamente. Existem duas maneiras de calcular o inicio e o fim do ataque:

- Método de Limiar Fixo

O início e o fim do ataque são estimados através do envelope da energia, onde são determinados limites para o início e o fim do ataque. No projeto CUIDADO, por exemplo, foi determinado que os limites são de 20% e 90% para o início e fim do ataque, respectivamente. Na figura abaixo, os valores da energia RMS do sinal foram calculados com um valor de janela $N = fr * .15$, o que corresponde a 150ms de áudio.

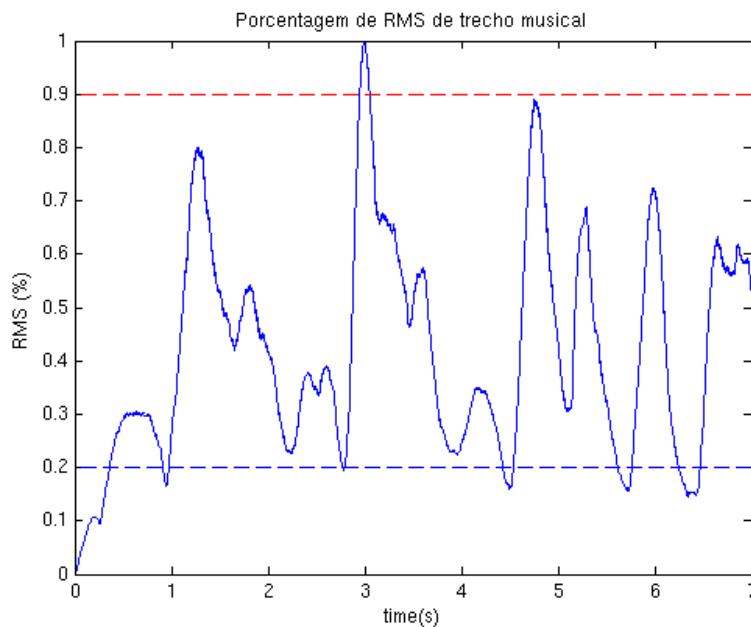


Figura A.3: Porcentagem de RMS para trecho musical e limiares de 20% e 90% para início e fim de ataque.

Pela figura podemos ver que adotar um limiar fixo para qualquer tipo de sinal de áudio pode acabar por não encontrar todos os inícios ou fim dos ataques desejados.

- Método de Limiar Adaptativo

Este método consiste em encontrar o início e o fim do ataque através de uma medida do acente do envelope da energia. O esforço da n -ésima posição do sinal de áudio é dado por $W(n) = e(n+1) - e(n)$. O objetivo então é encontrar o primeiro e o último valor deste acente de energia cujos esforços são maiores que $\bar{w}M$, onde \bar{w} é o valor médio de W e M é uma medida empírica. Em outras palavras, o objetivo é encontrar a posição i e j tal que $W(k) > \bar{w}M$, $i \leq k \leq j$, $W(i-1) < \bar{w}M$, e $W(j+1) < \bar{w}M$.

Depois de encontrados os valores dos inícios e fins dos ataques, é necessário encontrar os mínimos e máximos locais para um melhor refinamento da medida. Na figura abaixo, podemos ver o mesmo envelope RMS apresentado acima com os inícios e fins de ataques encontrados com um valor de $M = 200$. A desvantagem deste método é que o valor de M tem uma dependência do tamanho da janela N na construção do envelope de energia, e além disto, se o envelope de energia tem muitas oscilações, o algoritmo encontra mais inícios e fins de ataques do que realmente existem.

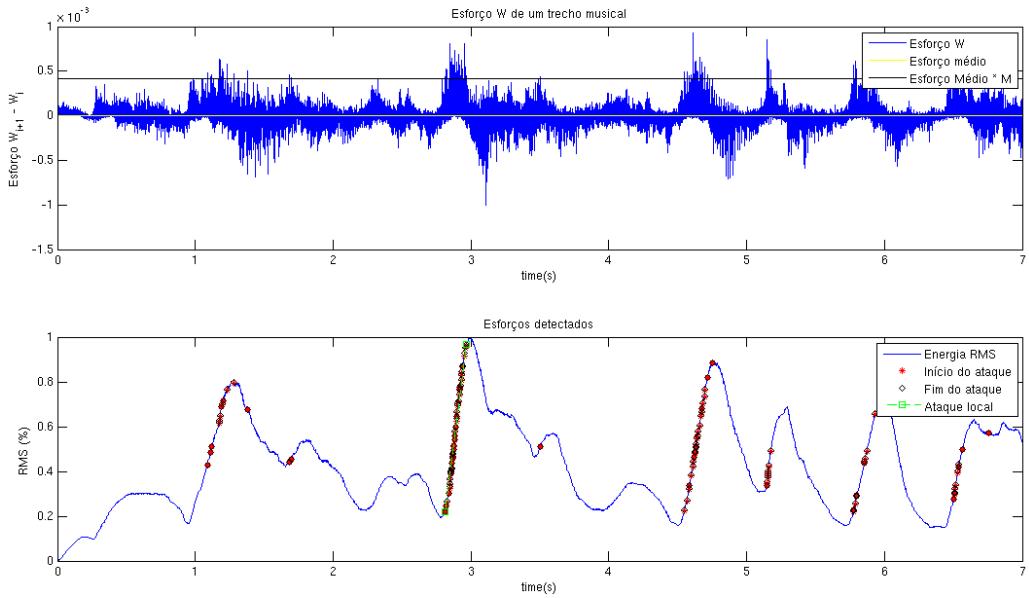


Figura A.4: A figura superior contém os esforços medidos para cada instante do sinal de áudio, o esforço médio \bar{w} , e o limiar $\bar{w}M$; e a figura inferior contém o gráfico da energia do sinal com os pontos do início e o fim dos ataques detectados.

DT.temporal-increase (Global)

Este descritor é uma medida da média do acíope da energia durante o tempo de ataque. É calculada através da média ponderada do esforço w (calculada na subseção anterior), com pesos escolhidos de acordo com uma função gaussiana centralizada no ponto médio do ataque e com desvio padrão de 0.5.

DT.temporal-decrease (Global)

Este descritor mede a taxa de diminuição de energia de um sinal, e permite distinguir sons percussivos de não-percussivos. É calculado seguindo um modelo temporal do envelope da energia do sinal a partir do instante do pico máximo do envelope.

$$S(t) = Ae^{-\alpha(t-t_{max})} \quad , t > t_{max} \quad (\text{A.5})$$

onde t_{max} é o instante da máxima energia do envelope local, e α é estimado por regressão linear no logaritmo do envelope da energia.

$$\alpha = \frac{n \sum_{x=1}^n xe(x) - \sum_{x=1}^n x \sum_{y=1}^n e(y)}{n \sum_{x=1}^n x^2 - \left(\sum_{x=1}^n x \right)^2} \quad (\text{A.6})$$

DT.energy-frequency-modulation (Global)

Este descritor representa as variações lentas de energia do sinal de áudio, cujas frequências estão entre 1Hz e 10Hz. Ele nos permite distinguir sons como o *tremolo*¹, utilizados na música como

¹“O tremolo em instrumentos de corda, é a rápida reiteração da mesma nota, produzido por um rápido movimento do arco.” (Apel, 1969)

forma de expressividade. Cada modulação é representada por uma frequência e uma amplitude. O método para calcular a modulação deve seguir os seguintes passos:

1. Localiza a parte sustentada da energia do sinal.
2. Corrige o envelope da energia, subtraindo a tendência logarítmica da parte sustentada - o que nivelá os valores de energia.
3. Calcula o espectro desta função de energia.
4. Localiza o pico máximo do espectro entre $1Hz$ e $10Hz$.

DT.temporal-centroid (Global)

Este descriptor permite distinguir sons percussivos de não-percussivos. É calculado através da energia ponderada sobre o tempo:

$$tc = \frac{\sum_{t=t_1}^{t_2} te(t)}{\sum_{t_1}^{t_2} e(t)} \quad (\text{A.7})$$

onde t_1 e t_2 são, respectivamente, os instantes iniciais e finais do trecho sendo investigado.

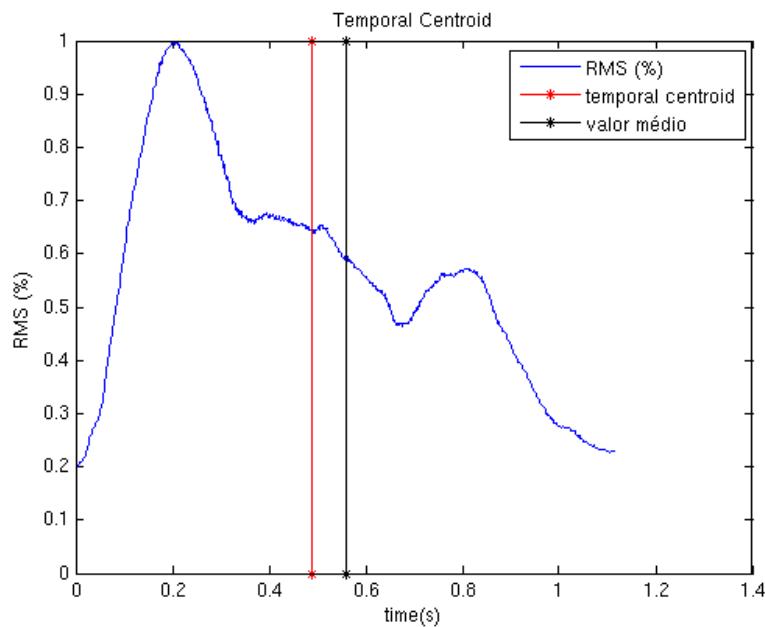


Figura A.5: Centroide temporal de uma envoltória RMS de instrumentos de corda.

DT.effective-duration (Global)

Este descriptor mede o tempo em que o sinal é perceptualmente significativo em termos de energia. É calculado através da soma da energia RMS do sinal que ultrapassa um determinado limiar. No projeto CUIDADO, os autores utilizaram um limiar igual a 40% da energia máxima do sinal.

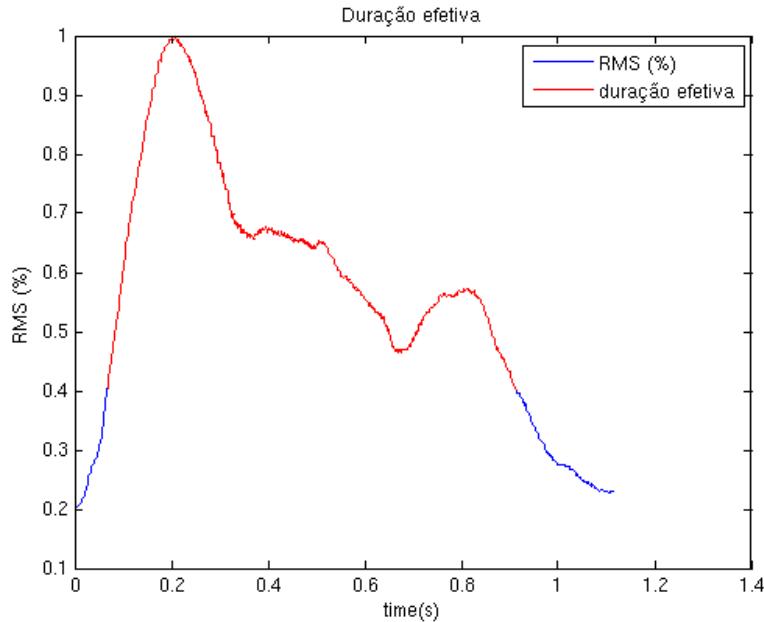


Figura A.6: Duração efetiva com limiar de 40% de uma envoltória de instrumentos de corda.

A.1.2 Descritores de Energia (Instantâneos)

Descritores de energia são descritores instantâneos que se referem às várias componentes de energia do sinal, seja através da energia do envelope do sinal, do espectro ou até mesmo do espectro harmônico do sinal.

DE.audio-power

Este descritor é uma medida da energia total de um sinal em torno de um instante no tempo. É calculado diretamente do sinal de áudio de acordo com a seguinte fórmula:

$$E_k = \frac{\sum_{n=k-N+1}^k x(n)^2}{\sum_{n=k-N+1}^k (w(n)x(n))^2} \quad (\text{A.8})$$

onde N é o tamanho da janela de análise, $x(n)$ é o sinal de áudio no instante n e $w(n)$ é a função da janela de Blackman calculada previamente.

A.1.3 Descritores Espectrais (Instantâneos e Globais)

Os descritores espetrais são os descritores instantâneos calculados diretamente da Transformada de Fourier de Tempo Reduzido(STFT). Para ajudar a entender as fórmulas a seguir, quando nos referirmos ao espectro do sinal, utilizaremos a letra E para indicar o conjunto de todas as componentes do espectro, $f(i)$ para indicar a i -ésima frequência, e $a(i)$ para indicar a i -ésima amplitude.

DS.centroid (Instantâneo)

Este descritor representa o baricentro do espectro, e seu valor é calculado considerando o espectro como uma distribuição, cujos valores são as frequências, e as probabilidades que os ob-

servam são as amplitudes normalizadas.

$$\mu = \frac{\sum_{i=1}^N f(i)a(i)}{\sum_{i=1}^N a(i)} \quad (\text{A.9})$$

onde N é o número de componentes do espectro E .

DS.spread (Instantâneo)

Este descritor representa o espalhamento do espectro em torno de seu valor médio, ou seja, a variância da distribuição definida no centroide normalizado pela equação A.1.3.

$$\delta = \frac{\sigma}{\mu} \quad (\text{A.10})$$

$$\text{onde } \sigma = \sqrt{\frac{\sum_{i=1}^N [f(i) - \mu]^2 a(i)}{\sum_{i=1}^N a(i)}}, \text{ e } N \text{ é o número de componentes do espectro } E.$$

DS.skewness (Instantâneo)

Este descritor representa a medida estatística de assimetria, o terceiro momento, de uma distribuição sobre seu valor médio. Se o valor for negativo, a massa de distribuição está concentrada mais para a direita. Caso contrário, se o valor for positivo, a massa de distribuição está concentrada na esquerda. É calculada da seguinte maneira:

$$\gamma_3 = \frac{m_3}{\sigma^3} \quad (\text{A.11})$$

$$\text{onde } m_3 = \frac{\sum_{i=1}^N [f(i) - \mu]^3 a(i)}{\sum_{i=1}^N a(i)}, \text{ e } N \text{ é o número de componentes do espectro } E.$$

DS.kurtosis (Instantâneo)

Este descritor é o 4º momento da estatística, e representa a medida de nivelamento de uma distribuição sobre seu valor médio. Valores positivos de *kurtosis* indicam uma distribuição com picos, e valores negativos indicam uma distribuição plana. É calculada da seguinte maneira:

$$\gamma_4 = \frac{m_4}{\sigma^4} \quad (\text{A.12})$$

$$\text{onde } m_4 = \frac{\sum_{i=1}^N [f(i) - \mu]^4 a(i)}{\sum_{i=1}^N a(i)}, \text{ e } N \text{ é o número de componentes do espectro } E.$$

DS.slope (Instantâneo)

Este descritor representa a quantidade de decaimento da amplitude do espectro, e é calculado pela sua regressão linear. A nova amplitude \hat{a} encontrada é então uma função linear da seguinte forma: $\hat{a}(i) = \alpha f(i) + \beta$, onde

$$\alpha = \frac{1}{\left(\sum_{i=1}^N a(i)\right)} \frac{N \sum_{i=1}^N f(i)a(i) - \sum_{i=1}^N f(i)a(i)}{N \sum_{i=1}^N f(i)^2 - \left(\sum_{i=1}^N f(i)\right)^2} \quad (\text{A.13})$$

β é uma constante e N é o número de componentes do espectro E .

DS.decrease (Instantâneo)

Este descritor representa a quantidade de decaimento da amplitude do espectro. Sua formulação vem de estudos de percepção acústica, e está supostamente mais relacionado à percepção humana.

$$\text{decrease} = \frac{\sum_{i=2}^N \frac{a(i)-a(1)}{i-1}}{\sum_{i=2}^N a(i)} \quad (\text{A.14})$$

onde N é o número de componentes do espectro E .

DS.roll-off (Instantâneo)

Este descritor representa a frequência de corte, tal que 95% da energia do espectro do sinal estão contidas abaixo desta frequência. Assim, $f_c = f(k)$ é a frequência de corte tal que:

$$\sum_{i=1}^k a(i)^2 = 0.95 \sum_{i=1}^N a(i)^2 \quad (\text{A.15})$$

onde N é o número de componentes do espectro E .

DS.variation (Global)

Este descritor representa a quantidade de variação do espectro ao longo do tempo. É calculado a partir da correlação cruzada entre amplitudes sucessivas da mesma componente do espectro $a_t(i)$ e $a_{t-1}(i)$, onde t é o índice da janela de análise

$$\text{variation} = 1 - \frac{\sum_{i=1}^N a_{t-1}(i)a_t(i)}{\sqrt{\sum_{i=1}^N a_{t-1}(i)^2} \sqrt{\sum_{i=1}^N a_t(i)^2}}. \quad (\text{A.16})$$

Valores próximos de 0 indicam que espectros sucessivos são similares, e valores próximos de 1 indicam que os espectros sucessivos são dissimilares.

DS.spectral-power

Este descritor é uma medida da energia total de um espectro do sinal em uma janela de análise. É calculado diretamente do espectro do sinal de acordo com a seguinte fórmula:

$$E_k = \frac{\sum_{i=1}^N a(i)^2}{\sum_{n=1}^N w(n)} \quad (\text{A.17})$$

onde N é o número de componentes do espectro, e $w(n)$ é a função da janela de Blackman calculada previamente.

DS.spectral-harmonic-power

Este descritor mede a energia da parte harmônica do sinal em uma dada janela de análise k .

$$E_k = \frac{\sum_{i=1}^N |harm(i)|^2}{2} \quad (\text{A.18})$$

onde N é o tamanho da janela de análise, $harm(n)$ é o n -ésimo harmônico definido na janela de análise k . Para mais detalhes sobre o modelo harmônico, veja Kim *et al.* (2005).

DS.spectral-noise-power

Este descritor determina a energia do ruído do espectro do sinal em uma janela de análise k .

$$E_k = \frac{\sum_{i=1}^N y(i)^2}{\sum_{i=1}^N w(i)^2} \quad (\text{A.19})$$

onde $y(i)$ é obtido subtraindo a parte harmônica do espectro original do sinal.

A.1.4 Descritores Harmônicos (Instantâneos e Globais)

Descritores Harmônicos são aqueles calculados do modelo sinusoidal do sinal.

DH.fundamental-frequency (Instantâneo)

A frequência fundamental f_0 é aquela que melhor explica o conteúdo harmônico do espectro do sinal. Para mais informações sobre as estratégias para extrair este descritor, veja Kim *et al.* (2005).

DH.noisiness (Instantâneo)

Este descritor mede a quantidade de ruído presente no espectro. É calculada através da razão entre a energia da parte não-harmônica, ou seja, o ruído, e o total de energia do espectro:

$$\text{noisiness} = \frac{\text{DS.spectral-noise-power}}{\text{DS.spectral-power}}. \quad (\text{A.20})$$

Valores próximos de 1 indicam um espectro com muito ruído, e valores próximos de 0 são sinais puramente harmônicos.

DH.inharmonicity (Instantâneo)

Este descritor representa a divergência entre os componentes do espectro do sinal e um sinal puramente harmônico. Pode ser calculado da seguinte maneira:

$$\text{inharmonicity} = \frac{2}{f_0} \frac{\sum_{i=1}^N |f(i) - if_0| a(i)^2}{\sum_{i=1}^N a(i)^2} \quad (\text{A.21})$$

onde f_0 é a frequência fundamental estimada (A.1.4), N é o número de componentes do espectro. Valores próximos de 1 indicam um espectro inharmônico, e valores próximos de 0 são sinais puramente harmônicos.

DH.deviation (Instantâneo)

Este descritor representa o desvio dos picos de amplitude harmônicos de um envelope global do espectro. Permite, por exemplo, diferenciar entre sons de clarinete e trompete, onde no primeiro, somente os harmônicos ímpares estão presentes, e no segundo, todos os harmônicos estão presentes. Mais detalhes da implementação, veja Kim *et al.* (2005).

DH.odd-even-ratio (Instantâneo)

Este descritor permite distinguir sons em que predominam harmônicos ímpares, como, por exemplo, o clarinete, de outros em que a energia está igualmente distribuída, como, por exemplo, o trompete. Pode ser calculado da seguinte forma:

$$\text{oeratio} = \frac{\sum_{h=1,3,5...}^{H-1} a(h)}{\sum_{h=2,4,6...}^H a(h)} \quad (\text{A.22})$$

onde H é o número de harmônicos do espectro do sinal.

DH.tristimulus (Instantâneo)

Tristímulos são três tipos diferentes de taxa de energia que permitem uma descrição mais fina dos primeiros harmônicos de um espectro, que são aqueles que são perceptivamente mais salientes.

$$T_1 = \frac{a(1)}{\sum_{h=1}^H a(h)} \quad (\text{A.23})$$

$$T_2 = \frac{a(2) + a(3) + a(4)}{\sum_{h=1}^H a(h)} \quad (\text{A.24})$$

$$T_3 = \frac{\sum_{h=5}^H a(h)}{\sum_{h=1}^H a(h)} \quad (\text{A.25})$$

onde H é o número de harmônicos do espectro do sinal.

DH.fundamental-frequency-modulation (Global)

Este descritor representa os sons cujas frequências fundamentais oscilam com períodos lentos, entre 1Hz e 10Hz . Ele nos permite distinguir os sons com *tremolo*, utilizados na música como forma de expressividade. Cada modulação é representada por uma frequência e uma amplitude. A fórmula para calcular a modulação é a seguinte:

1. Localiza a parte sustentada do som, ou seja, a parte que, na evolução do tempo, se mantém com as mesmas características de energia do sinal.
2. Calcula o espectro de todas as janelas.
3. Localiza o pico máximo do espectro entre 1Hz e 10Hz .

Para calcular o *vibrato*², podemos calcular o espectro do sinal $f_0(t)$ (frequência instantânea), e aqui a componente DC corresponde à $f_0(t)$ média e o pico entre 1Hz e 10Hz será o vibrato.

Outros descritores

Além dos descritores citados acima, pode-se calcular também os seguintes descritores:

- DH.centroid
- DH.spread
- DH.skewness
- DH.kurtosis
- DH.slope
- DH.decrease
- DH.roll-off
- DH.variation

Eles são calculados de forma semelhante aos descritores espectrais, porém utilizando as frequências e amplitudes harmônicas ao invés das componentes de frequência e amplitudes do espectro original.

A.1.5 Descritores Perceptuais (Instantâneos)

Descritores perceptuais são aqueles calculados através de um modelo do processo de escuta humana.

²Vibrato - Em instrumentos de corda, o *vibrato* é uma leve flutuação na altura da nota musical produzida em notas suspensas, por um movimento oscilatório com a mão esquerda. (Apel, 1969)

DP.mfcc

Mel-Frequency Cepstrum Coefficients (MFCC) é um vetor de coeficientes que forma um descriptor utilizado com sucesso em estudos como reconhecimento de voz (Rabiner e Juang, 1993). O *cepstrum* é calculado através da Transformada de Fourier Inversa do logaritmo do espectro. Usualmente, a transformada inversa é substituída pela Transformada Discreta de Cosseno (DCT), por utilizar somente valores reais (módulo do espectro). Os coeficientes MFCC são calculados através do logaritmo da magnitude do espectro após um mapeamento não-linear inspirado na escala Mel, e compactado através do uso do DCT. Tipicamente são armazenados de 10 a 30 coeficientes (Rabiner e Juang, 1993). No projeto CUIDADO foram armazenados os 12 primeiros coeficientes calculados pela DCT, excluindo a componente DC.

MFCC's fornecem uma representação do envelope do espectro e são provavelmente mais significativos musicalmente que qualquer outra representação do espectro, como *Linear Predictive Coding* (LPC). Apesar desta força de representação, MFCCs só podem carregar informações sobre comportamentos estáticos na janela de análise, e portanto, a dinâmica temporal não pode ser considerada na janela de análise em que está circunscrita. Outra característica é que MFCCs não tem uma interpretação direta, sendo, portanto, mais abstrato que um descriptor como a centroíde do espectro. Os coeficientes c_n podem ser calculados da seguinte forma:

$$c_n(t) = \frac{1}{2\pi} \int_{\omega=-\pi}^{\pi} \log(S(e^{i\omega})) e^{i\omega} d\omega \quad (\text{A.26})$$

onde $c_n(t)$ é o n-ésimo coeficiente extraído da t-ésima janela de análise.

DP.delta-mfcc

Este descriptor é a derivada de DP.mfcc (ver subseção anterior) de primeira ordem no tempo, e representa a velocidade de variação da forma do espectro no tempo, o que é de grande importância na percepção humana. Pode ser calculado por regressão linear de algumas janelas consecutivas:

$$\Delta c_n(t) = -c_n(t-2) - \frac{1}{2}c_n(t-1) + \frac{1}{2}c_n(t+1) + c_n(t+2) \quad (\text{A.27})$$

onde $\Delta c_n(t)$ é o n-ésimo coeficiente da derivada de MFCC das janelas de índice t .

DP.delta-delta-mfcc

Este descriptor é a derivada de DP.mfcc de segunda ordem no tempo, e representa a aceleração da variação da forma do espectro no tempo. Pode ser calculado da seguinte maneira:

$$\Delta\Delta c_n(t) = c_n(t-2) - \frac{1}{2}c_n(t-1) - c_n(t) - \frac{1}{2}c_n(t+1) + c_n(t+2) \quad (\text{A.28})$$

onde $\Delta\Delta c_n(t)$ é o n-ésimo coeficiente da derivada de segunda ordem de MFCC da janela de índice t .

Outros descritores perceptuais

Existem muitos outros descritores perceptuais, e o leitor interessado pode encontrar mais detalhes das implementações em Kim *et al.* (2005) e Peeters (2004). Segue abaixo uma lista de alguns deles.

- DP.specific-loudness
- DP.total-loudness
- DP.relative-specific-loudness

- DP.sharpness

Este descritor representa o brilho do espectro, e é o equivalente perceptual do centroide do espectro, mas calculado utilizando o volume específico (DP.specific-loudness) das bandas Bark.

- DP.spread

- DP.spectral-flatness

Este descritor é uma medida do quanto um sinal é similar a um ruído branco.

- DP.spectral-crest

- *Linear Predictive Coding* (LPC)

- *Linear Spectral Pairs* (LSPs)

- *Perceptual Linear Prediction* (PLP)

A.1.6 Modelagem Temporal

Os descritores instantâneos podem ser utilizados para calcular descritores globais de tempo longo, como a média e a variância globais.

Média Temporal

É um descritor global do valor médio de um descritor, usualmente ponderado pelo volume (DP.total-loudness) w .

$$\mu = \frac{\sum_{t=1}^T w(t)y(t)}{\sum_{t=1}^T w(t)} \quad (\text{A.29})$$

onde $w(t)$ é o t -ésimo volume total calculado para a janela de instante t , $y(t)$ é o valor do descritor instantâneo para a mesma janela t , e T é o número total de instantes calculados.

Variância Temporal

Descriptor global da variância dos descritores ponderada pelo volume (DP.total-loudness) w .

$$\sigma^2 = \frac{\sum_{t=1}^T w(t)(y(t) - \mu)^2}{\sum_{t=1}^T w(t)} \quad (\text{A.30})$$

onde μ é a média temporal (veja subseção anterior), $w(t)$ é o t -ésimo volume total calculado para a janela de instante t , $y(t)$ é o valor do descritor instantâneo para a mesma janela t , e T é o número total de instantes calculados.

Derivada Temporal

A derivada temporal dos descritores instantâneos representam a velocidade da variação deste descritor no tempo. Usualmente seu valor é ponderado pelo volume total (DP.total-loudness) w .

$$\Delta_y = \frac{\sum_{t=1}^{T-1} [w(t+1) - w(t)][y(t+1) - y(t)]}{\sum_{t=1}^{T-1} w(t+1) - w(t)} \quad (\text{A.31})$$

onde Δ_y é a derivada de um descritor y , $w(t)$ é o t -ésimo volume total calculado para a janela de instante t , $y(t)$ é o valor do descritor instantâneo para a mesma janela t , e T é o número total de instantes calculados.

A.2 Projeto JAudio

O projeto JAudio (<http://jaudio.sourceforge.net/>) é uma ferramenta desenvolvida em Java para extração de descritores com alta qualidade (McEnnis *et al.*, 2005). Este projeto ainda está ativo e um dado interessante é que seu ROADMAP contempla a extração de descritores em tempo real. Segue abaixo alguns descritores sonoros que esta ferramenta disponibiliza, além daqueles citados no projeto CUIDADO.

Histograma de Ritmos (Beat Histogram)

Este descritor representa o histograma de ritmos de um sinal. Este histograma mostra a força de cada ritmos periódicos que ocorrem no sinal. Isto é calculado através do RMS do sinal e então calculando a FFT do resultado.

Soma de Ritmos (Beat Sum)

Este descritor é a soma das batidas de um sinal. Esta é uma boa medida da importância do ritmo em um trecho musical. O descritor é calculado ao somar todos os valores de um histograma de ritmos.

Compacidade (Compactness)

Este é um descritor que representa a quantidade de ruído do sinal de áudio. Ele é calculado ao comparar o valor da magnitude de um espectro com os valores ao redor dele.

$$C = \sum_{i=2}^{N-1} |20 * \log M_i - 20 * \log ((M_{i-1} + M_i + M_{i+1})/3)|, \quad (\text{A.32})$$

onde N é a quantidade de componentes do espectro e M é a magnitude do espectro.

Apêndice B

Trechos Musicais agrupados por Cluster Hierárquico

#	<i>Composer/Artista</i>	<i>Nome da Música</i>
1	A. Vivaldi	Concerto No. 2 Summer
2	ACDC	Highway To Hell - Highway to hell
3	Alanis Morissetti	21 Things I Want In A Lover
4	Alanis Morissetti	All I Really Want
5	Alanis Morissetti	Jagged Little Pill - Head Over Feet
6	Alanis Morissetti	Jagged Little Pill - Hand In My Pocket
7	Alanis Morissetti	Narcissus
8	Alanis Mourissetti	All I Really Want
9	Alanis Mourissetti	Head Over Feet
10	Alanis Mourissetti	You Oughta Know
11	Almeida Prado	Sinfonia dos Orixás Ogum-Obá, a dança da espada de fogo
12	Almeida Prado	Sinfonia dos Orixás Ogum-Obá - Chamado aos Orixás
13	Almeida Prado	Sinfonia dos Orixás Ogum-Obá, a dança da espada de fogo
14	Anton Webern	Passacaglia for Orcherstra
15	Anton Webern	2 Arrangements - F.Schubert - German Dances - No.4
16	Anton Webern	Arrangements - J.S. Bach - Fuga (Ricercata)
17	Anton Webern	Heftig bewegt
18	Anton webern	Op.5 - Heftig Bewegt
19	Apocalyptica	Path
20	Apocalyptica	Beyond Time
21	Apocalyptica	Fight Fire With Fire
22	Apocalyptica	Hall Of Mountain King
23	Apocalyptica	Struggle
24	Baby Einstein	Classical Animals - Tsar Saltan, Tsar's Farewell
25	Baby Einstein	Classical Animals Capriccio Espagnol
26	Beethoven	Symphonie No. 9 - Allegro ma non troppo, un poco maestoso
27	Beethoven	Op 18 No.1
28	Beethoven	Op 18 No.1 I-Allegro Con Brio
29	Beethoven	Op 59 No. 2 IV-Finale (Presto)
30	Beethoven	Symphonie No. 9 - Allegro ma non troppo, un poco maestoso
31	Beethoven	Symphonie No. 9 - Adagio molto e cantabile
32	Beethoven	Symphonie No. 9 - Allegro ma non troppo, un poco maestoso
33	Beethoven	Symphonie No. 9 - Molto vivace
34	Beethoven	Symphonie No. 9 - Presto
35	Brian Ferneyhough	La chute d'icare
36	Brian Ferneyhough	mnemosyne
37	Chico Buarque	Construção - Construção
38	Chico Buarque	Construção - Cotidiano
39	Chico Buarque	Construção - Desalento
40	Chico Buarque	Construção - Deus Lhe Pague

41	Chico Buarque	Construção - Olha Maria
42	Chico Buarque (1971)	Construção - Acalanto
43	Cult	Hall Of Mountain King
44	Dalbavie	Seuils I
45	Dalbavie	Seuils II
46	Dalbavie	09 - Diadèmes II
47	Dalbavie	Seuils I
48	Dalbavie	Seuils II
49	Dalbavie	Seuils III
50	Dalbavie	Seuils V
51	Dalbavie	Seuils VII
52	Downland	(A Piece Without Tittle)
53	Emerson, Lake and Palmer	Pictures at an Exhibition - The Gnome
54	Emerson, Lake and Palmer	A fist full of music - Il Ritorno Di Ringo
55	Ennio Morricone	A fist full of music - Navajo Joe
56	Ennio Morricone	A fist full of music - The Good, The Bad And the Ugly
57	Ennio Morricone	A fist full of music - Una Pistola Per Ringo
58	Ennio Morricone	Ad Ogni Costo
59	Ennio Morricone	Moses Theme
60	Ennio Morricone	The Ballad Of Sacco And Vanzett
61	Ennio Morricone	The Big Gundown
62	Ennio Morricone	Odeon
63	Ernesto Nazaré /Nara Leão	Odeon
64	Ernesto Nazaré/Nara Leão	Book 1 Page 1
65	Fantômas	Book 1 Page 12
66	Fantômas	Book 1 Page 2
67	Fantômas	Book 1 Page 20
68	Fantômas	Book 1 Page 29
69	Fantômas	Book 1 Page 5
70	Fantômas	Delirium Cordia
71	Fantômas	Suspended animation - 01 04 01 05 Friday
72	Fantômas	Suspended animation - 02 04 02 05 Saturday
73	Fantômas	Suspended animation - 04 01 05 Friday
74	Fantômas	Suspended animation - 04 02 05 Saturday
75	Fantômas	Suspended animation - 04 08 05 Friday
76	Fantômas	Suspended animation - 08 04 08 05 Friday
77	Fantômas	Suspended animation - 26 04 26 05 Friday
78	Fantômas	Suspended animation - 26 04 26 05 Tuesday
79	Fantômas	Apostrophe - Dont you eat the yellow snow
80	Frank Zappa	APOSTROPHE - DUKE OF PRUNES
81	Frank Zappa	Apostrophe - Naval Aviation In Art
82	Frank Zappa	Apostrophe- Pedros Drowry
83	Frank Zappa	Freak Out! - How Could I Be Such a Fool
84	Frank Zappa	Freak Out! - Hungry Freaks, Daddy
85	Frank Zappa	Uncle Meat - King Kong IV
86	Frank Zappa	Uncle Meat - Main Title Theme
87	Frank Zappa	Uncle Meat - The Air
88	Frank Zappa	Uncle Meat - We Can Shoot You
89	Frank Zappa	Uncle Meat The Dog Breath Variations
90	Frank Zappa	Cai Dentro
91	Hermeto Pascoal	Slaves Mass - Aquela valsa
92	Hermeto Pascoal	Slaves Mass Chorinho pra ele
93	Hermeto Pascoal	The Number Of The Beast - Hallowed Be Thy Name
94	Iron Maiden	The Number Of The Beast - The Prisioner
95	Iron Maiden	The Number Of The Beast - The Prisioner
96	Iron Maiden	Johannes-Passion Aria
97	J.S. Bach	Johannes-Passion - Aria
98	J.S. Bach	Johannes-Passion - Chorus
99	J.S. Bach	

100	J.S. Bach	Johannes-Passion - Und die Kriegsknechte flochten ...
101	Jaco Pastorius	Bright Size Life
102	Jaco Pastorius	Jam
103	Jaco Pastorius	Liberty City
104	Jaco Pastorius	Opus Pocus
105	Jaco Pastorius	The Chicken
106	John Zorn	The big Gundown - Once Upon a time in the west
107	John Zorn	The big Gundown - Once Upon a time in the west
108	John Zorn	The big Gundown - Poverty
109	John Zorn	Masada string Trio
110	John Zorn	masada string trio - bethor
111	John Zorn	masada string trio - gurid
112	John Zorn	masada string trio - tabaet
113	John Zorn	Path of most resistance - ciocarlia
114	John Zorn	Path of most resistance - combat for the angel
115	John Zorn	Path of most resistance - exodus
116	John Zorn	Path of most resistance - heroes and villians
117	John Zorn	Path of most resistance - mega pyar shalimar
118	John Zorn	Path of most resistance - Renunciation
119	John Zorn	Path of most resistance - ship of fools
120	John Zorn	Path of most resistance - the 4
121	John Zorn	Path of most resistance - the end times
122	John Zorn	The big Gundown
123	John Zorn	The big Gundown - macchie solari
124	John Zorn	The big Gundown - Poverty
125	John Zorn	The big Gundown - The ballad of hank mccain
126	John Zorn	Theatrum os suprasensory - ignition of the art
127	John Zorn	Theatrum os suprasensory - lighting effect
128	John Zorn	Theatrum os suprasensory - palace of putrefaction
129	John Zorn	Theatrum os suprasensory - sn. 1 palace of putrefaction
130	Mahler	Symphony No.5 Trauermarsch
131	Mahler	Symphony No.5 - Rondo Finale
132	Mahler	Symphony No.5 - Scherzo
133	Mahler	Symphony No.5 - Trauermarsch
134	Mr Bungle	Disco Volante - Retrovertigo
135	Mr. Bungle	Carousel
136	Mr. Bungle	Carousel
137	Mr. Bungle	Cottage Cheese
138	Mr. Bungle	Pink Cigarette
139	Mr. Bungle	Pink Panther
140	Mr. Bungle	Quote Unquote
141	Mr. Bungle	Retrovertigo
142	Mr. Bungle	Tetris Theme
143	Pink Floyd	Time
144	Pixinguinha	Chorinho pra ele
145	Pixinguinha	Naquele Tempo
146	Radiohead	Hail to the Thief - The Gloaming
147	Radiohead	Hail to the Thief - 2+2=5
148	Radiohead	Hail to the Thief - Backdrifts
149	Radiohead	Hail to the Thief - I Will
150	Radiohead	Hail to the Thief - Myxomatosis
151	Radiohead	Hail to the Thief - Where I End and You Begin
152	Radiohead	OK Computer - Airbag
153	Radiohead	OK Computer - Karma Police
154	Radiohead	OK Computer - Paranoid Android
155	Radiohead	OK Computer - The Tourist
156	Rage Against The Machine	Testify
157	Rage Against the Machine	Battle of LA - Calm Like A Bomb
158	Rage Against The Machine	Bombtrack

159	Rage Against The Machine	Evil empire - Bulls On Parade
160	Rage Against The Machine	Evil empire - People Of The Sun
161	Rage Against The Machine	Evil empire - Year Of Tha Boomerang
162	Rage Against The Machine	Fistful Of Steel
163	Rage Against The Machine	Mic Check
164	Stevie Ray Vaughan	Pride And Joy
165	Stevie Ray Vaughan	Texas Flood
166	Stevie Ray Vaughan	Texas Flood - Dirty Pool
167	Stevie Ray Vaughan	Texas Flood - Lenny
168	Stockhausen	Helikopter-Streichquartett
169	Stockhausen	kontake - Teil 2
170	Stockhausen	kontake - Teil 1
171	Waldir Azevedo	Pedacinho do Céu

Tabela B.1: Base musical utilizada para recortar os trechos dos dados de treinamento e geração musical.

B.1 Tabelas dos grupos de Trechos Musicais

Composer/Artista	Nome da Música	Nro. do Trecho
Almeida Prado	Sinfonia dos Orixás Ogum-Obá, a dança da espada de fogo	t02
Beethoven	Op 59 No. 2 IV-Finale (Presto)	t01
Beethoven	Symphonie No. 9 - Adagio molto e cantabile	t03
Beethoven	Symphonie No. 9 - Allegro ma non troppo, un poco maestoso	t04
Brian Ferneyhough	La chute d'icare	t08
Dalbavie	Seuils I	t01
Dalbavie	Seuils I	t14
Dalbavie	Seuils I	t15
Dalbavie	Seuils II	t02
J.S. Bach	Johannes-Passion Aria	t02
John Zorn	The big Gundown - Once Upon a time in the west	t02
John Zorn	The big Gundown - Poverty	t01
Mr. Bungle	Carousel	t01

Tabela B.2: Trechos musicais do Grupo 1.

Composer/Artista	Nome da Música	Nro. do Trecho
Almeida Prado	Sinfonia dos Orixás Ogum-Obá, a dança da espada de fogo	t02
Almeida Prado	Sinfonia dos Orixás Ogum-Obá, a dança da espada de fogo	t03
Anton Webern	Passacaglia for Orcherstra	t01
Anton Webern	Passacaglia for Orcherstra	t03
Anton Webern	Passacaglia for Orcherstra	t04
Anton Webern	2 Arrangements - F.Schubert - German Dances - No.4	t02
Beethoven	Op 18 No.1 I-Allegro Con Brio	t04
Beethoven	Symphonie No. 9 - Allegro ma non troppo, un poco maestoso	t05
Beethoven	Symphonie No. 9 - Allegro ma non troppo, un poco maestoso	t02
Beethoven	Symphonie No. 9 - Molto vivace	t06
Beethoven	Symphonie No. 9 - Molto vivace	t08
Beethoven	Symphonie No. 9 - Molto vivace	t07
Beethoven	Symphonie No. 9 - Presto	t03
Beethoven	Symphonie No. 9 - Presto	t04
Brian Ferneyhough	La chute d'icare	t03
Brian Ferneyhough	La chute d'icare	t04
Brian Ferneyhough	La chute d'icare	t07

Chico Buarque (1971)	Construção - Acalanto	t01
Cult	Hall Of Mountain King	t03
Dalbavie	Seuils I	t13
Frank Zappa	Uncle Meat - The Air	t02
J.S. Bach	Johannes-Passion - Chorus	t05
John Zorn	Masada string Trio	t01
John Zorn	The big gundown	t03
Mahler	Symphony No.5 Trauermarsch	t03
Mr. Bungle	Retrovertigo	t03
Ernesto Nazaré/Nara Leão	Odeon	t02
Pink Floyd	Time	t03
Stockhausen	kontake - Teil 2	t02
Stockhausen	kontake - Teil 2	t03
Stockhausen	kontake - Teil 1	t02

Tabela B.3: *Trechos musicais do Grupo 2.*

Composer/Artista	Nome da Música	Nro. do Trecho
Fantômas	Book 1 Page 29	t01

Tabela B.4: *Trechos musicais do Grupo 9.*

Composer/Artista	Nome da Música	Nro. do Trecho
Almeida Prado	Sinfonia dos Orixás Ogum-Obá - Chamado aos Orixás	t06
Almeida Prado	Sinfonia dos Orixás Ogum-Obá - Chamado aos Orixás	t07
Anton Webern	Arrangements - J.S. Bach - Fuga (Ricercata)	t02
Baby Einstein	Classical Animals Capriccio Espagnol	t02
Dalbavie	Seuils I	t02
Dalbavie	Seuils II	t01
Dalbavie	Seuils V	t01
Radiohead	Hail to the Thief - Backdrifts	t01

Tabela B.5: *Trechos musicais do Grupo 12.*

Apêndice C

Estatística

C.1 Razão da máxima verossimilhança estatística

Seja $X = x_1, \dots, x_n$, desejamos testar a hipótese

$$H_0 : x_1, \dots, x_n \sim \mathcal{N}(\mu, \Sigma) \quad (\text{C.1})$$

versus

$$H_1 : x_1, \dots, x_i \sim \mathcal{N}(\mu_1, \Sigma_1); \quad x_{i+1}, \dots, x_n \sim \mathcal{N}(\mu_2, \Sigma_2). \quad (\text{C.2})$$

A função da distribuição normal multivariada é dada por

$$f(x) = \frac{1}{2\pi^{d/2}|\Sigma|^{1/2}} e^{-1/2(x-\mu)^T \Sigma^{-1}(x-\mu)}, \quad (\text{C.3})$$

e a verossimilhança de $f(x)$ é

$$L = \prod_{i=1}^n f(x_i), \quad \log L = \sum_{i=1}^n \log f(x_i) = l. \quad (\text{C.4})$$

A razão de verossimilhança é dada por

$$R = L_0/L_1, \quad (\text{C.5})$$

e, a razão do log da verossimilhança é igual a

$$\log R = \log L_0 - \log L_1 = l_0 - l_1. \quad (\text{C.6})$$

Temos então que

$$\log f(x_i) = \frac{d}{2} \log 2\pi - \frac{1}{2} \log |\Sigma| - \frac{1}{2} (x_i - \mu)^T \Sigma^{-1} (x_i - \mu). \quad (\text{C.7})$$

Assim, o log da verossimilhança de H_0 é dado por

$$l_0 = \sum_{i=1}^n \log f(x_i) = \frac{dn}{2} \log 2\pi - \frac{n}{2} \log |\Sigma| - \frac{1}{2} \sum_{i=1}^n (x_i - \mu)^T \Sigma^{-1} (x_i - \mu), \quad (\text{C.8})$$

sendo que o último termo da equação pode ser reduzido a $-\frac{n}{2}$. Prova:

$$\begin{aligned}
 -\frac{1}{2} \sum_{i=1}^n (x_i - \mu)^T \Sigma^{-1} (x_i - \mu) \\
 &= -\frac{1}{2} \sum_{i=1}^n \text{Tr}[(x_i - \mu)^T \Sigma^{-1} (x_i - \mu)] \\
 &= -\frac{1}{2} \sum_{i=1}^n \text{Tr}[(x_i - \mu)(x_i - \mu)^T \Sigma^{-1}] \\
 &= -\frac{1}{2} n S \Sigma^{-1}
 \end{aligned} \tag{C.9}$$

e se o estimador de $\hat{\Sigma} = S$, então

$$-\frac{1}{2} \sum_{i=1}^n (x_i - \mu)^T \Sigma^{-1} (x_i - \mu) = -\frac{1}{2} \text{Tr}[n S S^{-1}] = -\frac{1}{2} n \text{Tr}[S S^{-1}] = -\frac{nd}{2}, \tag{C.10}$$

pois $\text{Tr}[S S^{-1}] = \text{Tr}[I] = d$.

Então,

$$l_0 = \frac{dn}{2} \log 2\pi - \frac{n}{2} \log |\Sigma| - \frac{nd}{2}. \tag{C.11}$$

Da mesma forma, podemos calcular a verossimilhança de L_1

$$L_1 = \left[\prod_{i=1}^{N_1} f_1(x_i) \right] \left[\prod_{i=1}^{N_2} f_2(x_i) \right], \tag{C.12}$$

e o log da verossimilhança de L_1 é

$$\begin{aligned}
 l_1 &= \left[\sum_{i=1}^{N_1} \log f_1(x_i) \right] \left[\sum_{i=1}^{N_2} \log f_2(x_i) \right] \\
 &= \frac{dN_1}{2} \log 2\pi - \frac{N_1}{2} |\Sigma_1| - \frac{dN_1}{2} + \frac{dN_2}{2} \log 2\pi - \frac{N_2}{2} |\Sigma_2| - \frac{dN_2}{2} \\
 &= \frac{d(N_1 + N_2)}{2} \log 2\pi - \frac{N_1}{2} |\Sigma_1| - \frac{N_2}{2} |\Sigma_2| - \frac{d(N_1 + N_2)}{2} \\
 &= \frac{d(N)}{2} \log 2\pi - \frac{N_1}{2} |\Sigma_1| - \frac{N_2}{2} |\Sigma_2| - \frac{d(N)}{2}.
 \end{aligned} \tag{C.13}$$

Substituindo l_0 e l_1 na equação C.6, temos que

$$\begin{aligned}
 \log R &= \frac{dN}{2} \log 2\pi - \frac{N}{2} \log |\Sigma| - \frac{dN}{2} - \frac{dN}{2} \log 2\pi + \frac{N_1}{2} \log |\Sigma_1| + \frac{N_2}{2} \log |\Sigma_2| + \frac{dN}{2} \\
 &= -\frac{N}{2} \log |\Sigma| + \frac{N_1}{2} \log |\Sigma_1| + \frac{N_2}{2} \log |\Sigma_2|. \quad \blacksquare
 \end{aligned} \tag{C.14}$$

Apêndice D

Códigos e Scripts

D.1 Arquivo .orc Csound

O arquivo de orquestra abaixo foi utilizado para a geração de dados musicais.

```
1 sr = 44100
2 kr = 4410
3 ksmmps = 10
4 nchnls = 1
5
6
7 instr 1
8   kamp = 25000
9   ilen = p3
10  kcps = 1
11  ifn = p4
12  ibas = 1 ; base frequency
13  imod = 1 ; normal loop (forward)
14
15 ; envoltoria
16 aenvr adsr .001, 0, 1, .001
17
18 k1 linenr kamp, .05, .05, .05
19
20 a1 loscil k1, kcps, ifn, ibas, imod
21
22 aenvl = aenvr*a1
23
24 out aenvl
25 endin
```

Listing D.1: Arquivo .orc do Csound utilizado para a geração dos dados musicais.

D.2 Exemplo de arquivo .sco Csound

O arquivo abaixo é um exemplo de partitura (*score*) para o Csound na geração de dados musicais.

```

1 f 102 0 0 1 ".../08_04_08_05_Friday#05.wav" 0 0 0
2 f 101 0 0 1 ".../04_Construção#07.wav" 0 0 0
3 f 103 0 0 1 ".../14_-_heroes_and_villians#01.wav" 0 0 0
4 f 100 0 0 1 ".../2-06_Moses_Theme_(Main_Title)#01.wav" 0 0 0
5 f 104 0 0 1 ".../01_-_Seuils_I#05.wav" 0 0 0
6
7 i 1 0 10 101
8 i 1 10 12 100
9 i 1 22 11 103
10 i 1 33 9 100
11 i 1 42 12 103
12 i 1 54 12 100
13 i 1 66 11 102
14 i 1 77 9 103
15 i 1 86 10 100
16 i 1 96 12 103
17 i 1 108 12 104

```

D.3 Exemplo de arquivo de transição

O arquivo abaixo é um exemplo de arquivo de transição gerado durante a geração de dados musicais.

```

1 "NOME", "NRO_LABEL", "Transicao_Seg", "Transicao_Samples"
2 "01_-_la_chute_d'icare_(1988)#03.wav", 1, 0, 0
3 "01_-_Seuils_I#11.wav", 2, 13, 260
4 "Mr._Bungle_-_Carousel#01.wav", 3, 23, 460
5 "01_-_Seuils_I#11.wav", 2, 36, 720
6 "01_-_la_chute_d'icare_(1988)#03.wav", 1, 46, 920
7 "1-01_Uncle_Meat__Main_Title_Theme#01.wav", 4, 57, 1140
8 "01_-_Seuils_I#11.wav", 2, 67, 1340
9 "Mr._Bungle_-_Carousel#01.wav", 3, 78, 1560
10 "01_-_la_chute_d'icare_(1988)#03.wav", 1, 88, 1760
11 "1-01_Uncle_Meat__Main_Title_Theme#01.wav", 4, 97, 1940
12 "04_Presto#03.wav", 5, 108, 2160
13 "10_I_Will_.(No_Man's_Land.)#01.wav", 6, 119, 2380

```

Referências Bibliográficas

Adorno(1994) T.W. Adorno. Sobre Música Popular. IN: COHN, G. *Theodor Adorno*. São Paulo: Ática. Citado na pág. 42, 43

Agostini et al.(2003) G. Agostini, M. Longari, e E. Pollastri. Musical instrument timbres classification with spectral features. *EURASIP Journal on Applied Signal Processing*, 2003:14. Citado na pág. 22

Apel(1969) W. Apel. *Harvard dictionary of music*. Belknap Press. Citado na pág. 127, 134

Aucouturier e Pachet(2004) J.J. Aucouturier e F. Pachet. Improving timbre similarity: How high is the sky. *Journal of Negative Results in Speech and Audio Sciences*, 1(1):1–13. Citado na pág. 21

Aucouturier e Sandler(2001) J.J. Aucouturier e M. Sandler. Segmentation of musical signals using hidden Markov models. *Preprints-Audio Engineering Society*. Citado na pág. 4, 41, 44, 57

Aucouturier et al.(2005) J.J. Aucouturier, F. Pachet, e M. Sandler. “The way it Sounds”: timbre models for analysis and retrieval of music signals. *IEEE Transactions on Multimedia*, 7(6):1028–1035. Citado na pág. 21, 42

Berenzweig e Ellis(2002) A.L. Berenzweig e D.P.W. Ellis. Locating singing voice segments within music signals. Em *Applications of Signal Processing to Audio and Acoustics, 2001 IEEE Workshop on the*, páginas 119–122. ISBN 0780371267. Citado na pág. 45

Bergstra et al.(2006) J. Bergstra, N. Casagrande, D. Erhan, D. Eck, e B. Kégl. Aggregate features and ADABOOST for music classification. *Machine Learning*, 65(2):473–484. Citado na pág. 21

Bicego et al.(2006) M. Bicego, M. Cristani, e V. Murino. Unsupervised scene analysis: A hidden Markov model approach. *Computer vision and image understanding*, 102(1):22–41. ISSN 1077-3142. Citado na pág. 121

Blum e Rivest(1993) A. Blum e R. Rivest. Training a 3-node neural network is np-complete. *Machine Learning: From Theory to Applications*, páginas 9–28. Citado na pág. 24

Boulez(1985) P. Boulez. Le timbre et l'écriture, le timbre et le langage. *J. Barrière (Comp.), Le timbre, métaphore pour la composition*, páginas 541–549. Citado na pág. 42

Bullock(2007) J. Bullock. Libxtract: A lightweight library for audio feature extraction. Em *Proceedings of the International Computer Music Conference*. Citado na pág. 9, 15

Cabral et al.(0) G. Cabral, J.P. Briot, S. Krakowski, L. Velho, F. Pachet, e P. Roy. Analytical features to extract harmonic or rhythmic information. Em *Proceedings of the 11th Brazilian Symposium on Computer Music (SBCM 2007)*, páginas 153–165. Citado na pág. 4

Catanzaro(2003) T.O. Catanzaro. *Transformações na Linguagem Musical Contemporânea Instrumental e Vocal sob a Influência da Música Eletroacústica entre as Décadas de 1950-70*. Tese de Doutorado, Dissertação (Mestrado) - ECA/USP, São Paulo, 2003. Sob a orientação de Fernando Iazzetta. Citado na pág. 42

- Chai(2006)** W. Chai. Semantic segmentation and summarization of music: methods based on tonality and recurrent structure. *Signal Processing Magazine, IEEE*, 23(2):124–132. ISSN 1053-5888. Citado na pág. 85
- Chen e Gopalakrishnan(1998)** S.S. Chen e P.S. Gopalakrishnan. Speaker, environment and channel change detection and clustering via the bayesian information criterion. Citado na pág. 4, 45, 78
- Cooper e Foote(2002)** M. Cooper e J. Foote. Automatic music summarization via similarity analysis. Em *Proc. IRCAM*, páginas 81–85. Citado na pág. x, xvi, 61, 63, 92, 103
- DALBAVIE(1991)** M.A. DALBAVIE. Pour sortir de l'avant-garde. *J. Barrière (Comp.), Le timbre, métaphore pour la composition*. Citado na pág. 42
- Dannenberg e Goto(2009)** R.B. Dannenberg e M. Goto. Music structure analysis from acoustic signals. *Handbook of Signal Processing in Acoustics*, páginas 305–331. Citado na pág. 4, 41, 45, 58, 120
- Duda et al.(2001)** R.O. Duda, P.E. Hart, e D.G. Stork. *Pattern classification*. Citado na pág. 4, 32, 47, 50, 55
- Fiebrink e Fujinaga(2006)** R. Fiebrink e I. Fujinaga. Feature selection pitfalls and music classification. Em *Proceedings of the 7th International Conference on Music Information Retrieval*, páginas 340–341. Citado na pág. 17
- Herrera-Boyer et al.(2003)** P. Herrera-Boyer, G. Peeters, e S. Dubnov. Automatic classification of musical instrument sounds. *Journal of New Music Research*, 32(1):3–21. Citado na pág. 16, 22
- Kim et al.(2005)** H.G. Kim, N. Moreau, e T. Sikora. *MPEG-7 audio and beyond*. John Wiley. Citado na pág. 4, 16, 132, 133, 135
- Kuhn(1955)** H.W. Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97. ISSN 1931-9193. Citado na pág. 90
- Liu e Yu(2005)** H. Liu e L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on knowledge and data engineering*, páginas 491–502. Citado na pág. 17, 24
- Liu e Chen(2003)** X. Liu e T. Chen. Video-based face recognition using adaptive hidden markov models. *Proc. IEEE IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. ISSN 1063-6919. Citado na pág. 121
- Malt e Jourdan(2009)** M. Malt e E. Jourdan. Real-Time Uses Of Low Level Sound Descriptors As Event Detection Functions Using The Max/MSP Zsa. Descriptors Library. Citado na pág. 4
- Mardia et al.(1979)** K.V. Mardia, J.T. Kent, e J.M. Bibby. *Multivariate analysis*. Citado na pág. 33, 80
- Mart Rocamora(2009)** Luis Jure Mart Rocamora, Ernesto Lopez. Wind instruments synthesis toolbox for generation of music audio signals with labeled partials . Citado na pág. 4
- McEnnis et al.(2005)** D. McEnnis, C. McKay, e I. Fujinaga. jAudio: A feature extraction library. Em *International Conference on Music Information Retrieval*. Citado na pág. 4, 9, 17, 137
- Moore(1990)** F.R. Moore. *Elements of computer music*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA. Citado na pág. 2, 4, 18
- Nabney(2002)** I. Nabney. *NETLAB: algorithms for pattern recognition*. ISBN 1852334401. Citado na pág. 102

- Omar et al.(2005)** M.K. Omar, U. Chaudhari, e G. Ramaswamy. Blind change detection for audio segmentation. Em *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, Philadelphia, USA*. Citado na pág. 4, 45, 78
- Oppenheim et al.(1989)** A.V. Oppenheim, R.W. Schafer, J.R. Buck, et al. *Discrete-time signal processing*, volume 1999. Prentice hall Englewood Cliffs, NJ:. Citado na pág. 17
- Otsu(1975)** N. Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11: 285–296. Citado na pág. 63
- Paulus et al.(2010)** J. Paulus, M. Müller, e A. Klapuri. Audio-based music structure analysis. Em *Proc. of the International Society for Music Information Retrieval Conference, ISMIR - 2010*. Citado na pág. 4, 85, 120, 121
- Peeters(2003a)** G. Peeters. Automatic classification of large musical instrument databases using hierarchical classifiers with inertia ratio maximization. *Preprints-Audio Engineering Society*. Citado na pág. 4, 25, 33
- Peeters(2004)** G. Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. *CUIDADO IST Project Report*, páginas 1–25. Citado na pág. xv, 4, 15, 16, 20, 125, 135
- Peeters e Rodet(2002a)** G. Peeters e X. Rodet. Automatically selecting signal descriptors for sound classification. Em *Proceedings of the International Computer Music Conference*. Citado na pág. 4, 21
- Peeters et al.(2002b)** G. Peeters, A. La Burthe, e X. Rodet. Toward automatic music audio summary generation from signal analysis. Em *Proc. International Conference on Music Information Retrieval*, páginas 94–100. Citado na pág. x, xiv, xv, xvi, xxi, 4, 35, 39, 40, 41, 44, 65, 66, 67, 68, 71, 80, 92
- Pires e Queiroz(2011)** A. Pires e M. Queiroz. Real-time unsupervised music structural segmentation using dynamic descriptors. *SMC Conference 2011*. Citado na pág. 120
- Pratyush e Serra(2010)** Umbert M. Pratyush, A. e X. Serra. A look into the past: Analysis of trends and topics in proceedings of Sound and Music Computing Conference. Citado na pág. 4
- Rabiner e Juang(1993)** L. Rabiner e B.H. Juang. Fundamentals of speech recognition. *Englewood Cliffs, NJ*. Citado na pág. 4, 135
- Rabiner(1989)** L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286. ISSN 0018-9219. Citado na pág. 4, 57, 58
- Roads(1996)** C. Roads. *The Computer Music Tutorial*. The MIT Press. Citado na pág. 42
- Roads(2004)** C. Roads. *Microsound*. The MIT Press. Citado na pág. 15
- Rocamora e Herrera(2007)** M. Rocamora e P. Herrera. Comparing audio descriptors for singing voice detection in music audio files. Em *Brazilian Symposium on Computer Music, 11th. San Pablo, Brazil*, volume 26, página 27. Citado na pág. 4
- Sadie e Tyrrell(2001)** S. Sadie e J. Tyrrell. The new Grove dictionary of music and musicians. Citado na pág. 42
- Sainath et al.(2007)** T.N. Sainath, D. Kanevsky, e G. Iyengar. Unsupervised audio segmentation using extended baum-welch transformations. Em *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 1. ISBN 1424407273. Citado na pág. 4, 45
- Schoenberg(1993)** A. Schoenberg. *Fundamentos da composição musical*. Edusp. Citado na pág. 41
- Sethares(2005)** W.A. Sethares. *Tuning, timbre, spectrum, scale*. Springer Verlag. Citado na pág. 42

- Somol et al.(2006)** P. Somol, J. Novovičová, e P. Pudil. Flexible-hybrid sequential floating search in statistical feature selection. *Structural, Syntactic, and Statistical Pattern Recognition*, páginas 632–639. Citado na pág. 4, 21
- Theodoridis e Koutroumbas(2008)** Sergios Theodoridis e Konstantinos Koutroumbas. *Pattern Recognition, Fourth Edition*. Academic Press. ISBN 1597492728. Citado na pág. 4, 17, 21, 22, 23, 25, 27, 30, 32, 33, 47, 48, 50, 55
- Tzanetakis e Cook(1999)** G. Tzanetakis e P. Cook. Multifeature audio segmentation for browsing and annotation. Em *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. Citado na pág. x, 4, 62, 69, 71, 82, 92
- Velivelli et al.(2003)** A. Velivelli, C. Zhai, e T.S. Huang. Audio segment retrieval using a synthesized HMM. Em *Proceedings of the ACM SIGIR workshop on multimedia information retrieval, Toronto, Canada*. Citado na pág. 21
- von Helmholtz(1865)** H. von Helmholtz. *The Sensations of Tone as a physiological basis for the Theory of Music*. Citado na pág. 16
- Witten e Frank(2005)** I.H. Witten e E. Frank. *Data Mining: Practical machine learning tools and techniques*. ISBN 0120884070. Citado na pág. 53