Universidade Federal
do Rio de Janeiro
Escola Politécnica

ALGORITMOS PARA FATORAÇÃO DE MATRIZES NÃO-NEGATIVAS COM APLICAÇÃO EM TRANSCRIÇÃO DE INSTRUMENTOS PERCUSSIVOS

Igor Macedo Quintanilha

Projeto de Graduação apresentado ao Curso de Engenharia Eletrônica e de Computação da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientador: Luiz Wagner Pereira Biscainho

Rio de Janeiro
Abril de 2016

ALGORITMOS PARA FATORAÇÃO DE MATRIZES NÃO-NEGATIVAS COM APLICAÇÃO EM TRANSCRIÇÃO DE INSTRUMENTOS PERCUSSIVOS

Igor Macedo Quintanilha

PROJETO DE GRADUAÇÃO SUBMETIDO AO CORPO DOCENTE DO CURSO DE ENGENHARIA ELETRÔNICA E DE COMPUTAÇÃO DA ESCOLA POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO ELETRÔNICO E DE COMPUTAÇÃO.

Examinado por:

Prof. Luiz Wagner Pereira Biscainho, D. Sc.

Prof. Wallace Alves Martins, D. Sc.

Prof. Markus Vinícius Santos Lima, D. Sc.

RIO DE JANEIRO, RJ – BRASIL
ABRIL DE 2016

Macedo Quintanilha, Igor

Algoritmos para fatoração de matrizes não-negativas com aplicação em transcrição de instrumentos percussivos/Igor Macedo Quintanilha. – Rio de Janeiro: UFRJ/ Escola Politécnica, 2016.

XXII, 88 p.: il.; 29,7cm.

Orientador: Luiz Wagner Pereira Biscainho

Projeto de Graduação – UFRJ/ Escola Politécnica/ Curso de Engenharia Eletrônica e de Computação, 2016.

Referências Bibliográficas: p. 77 – 84.

1. Processamento digital de áudio. 2. Fatoração de matrizes não-negativas. 3. Transcrição de instrumentos percusivos. I. Pereira Biscainho, Luiz Wagner. II. Universidade Federal do Rio de Janeiro, Escola Politécnica, Curso de Engenharia Eletrônica e de Computação. III. Título.

À minha família brasileira e à
minha família japonesa.

Agradecimentos

Primeiramente, gostaria de agradecer à minha família, que apesar de todos os percalços sempre me apoiou e nunca duvidou das minhas capacidades, mesmo quando eu duvidava de mim.

Gostaria de agradecer à minha família japonesa, que nos últimos sete anos me adotou como um filho, sempre me ajudando.

Agradeço a todos os amigos que criei nesses últimos anos de faculdade e àqueles que me acompanham desde o colégio. Meu eterno obrigado, sem as amizades esse caminho teria sido muito mais difícil.

Agradeço imensamente ao meu orientador, pela paciência, pelo bom humor e por acreditar sempre em mim.

Por fim, gostaria de agradecer a Karina Yumi Atsumi, por ter tido tanta paciência, aceitado minhas ausências, maus humores e muitas outras coisas que ocorreram nesses anos, e sempre me retornado amor, carinho e zelo.

Resumo do Projeto de Graduação apresentado à Escola Politécnica/ UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro Eletrônico e de Computação.

Algoritmos para fatoração de matrizes não-negativas com aplicação em transcrição de instrumentos percussivos

Igor Macedo Quintanilha

Abril/2016

Orientador: Luiz Wagner Pereira Biscainho

Curso: Engenharia Eletrônica e de Computação

Esse trabalho realiza um estudo dos algoritmos para a fatoração de matrizes não-negativas (NMF), uma ferramenta bastante estudada em álgebra linear que possui aplicações em mineração de texto, biologia, separação de fontes sonoras, *image inpainting* etc. Como aplicação dessa técnica, esse trabalho apresenta o uso da NMF nas áreas de transcrição de instrumentos musicais e separação/reconstrução de fontes sonoras. Enfatiza-se o caso de instrumentos percussivos, menos abordado na literatura. O desempenho da separação foi regular sob o ponto de vista da qualidade dos sinais reconstruídos, entretanto, para efeito de transcrição, os resultados de separação se mostraram suficientes. Uma possível estratégia para melhorar os resultados percebidos é tratar com cuidado a fase dos sinais reconstruídos.

Palavras-chave: processamento digital de áudio, NMF, transcrição.

Abstract of Undergraduate Project presented to POLI/UFRJ as a partial fulfillment of the requirements for the degree of Engineer.

ALGORITHMS FOR NONNEGATIVE MATRIX FACTORIZATION WITH APPLICATIONS IN PERCUSSIVE INSTRUMENTS TRANSCRIPTION

Igor Macedo Quintanilha

April/2016

Advisor: Luiz Wagner Pereira Biscainho

Course: Electronic Engineering

In this work, we present a study of several algorithms applied to nonnegative matrix factorization (NMF), which is a tool widely studied in linear algebra that has also been extensively used in text mining, biology, sound source separation, image inpainting, etc. Also, we present an application of NMF in the areas of transcription of musical instruments and separation/reconstruction of sound sources. The case of percussive instruments, not often studied in the literature, is emphasized. Regarding the quality of reconstructed signals, the overall performance is average; on the other hand, when transcription is the target, separation results are sufficiently good. A possible strategy to improve perceptual results is to give more attention to the phase of reconstructed signals.

Keywords: digital audio processing, NMF, transcription.

Sumário

Lista de Figuras **xi**

Lista de Tabelas **xiii**

Lista de Símbolos **xv**

Lista de Abreviaturas **xxi**

1 Introdução **1**

1.1 Organização 2

1.2 Notação 2

2 Separação Cega **5**

2.1 Separação baseada na independência 6

2.2 Separação baseada na esparsidade 8

2.3 Separação baseada na não-negatividade 8

3 Fatoração de Matrizes Não-Negativas **13**

3.1 Inicialização 14

3.1.1 Múltiplas camadas 14

3.2 Algoritmos 16

3.2.1 MUR 16

3.2.2 Gradiente descendente 19

3.2.3 ALS 22

3.2.4 ANLS 23

3.2.5 Outros algoritmos 28

3.3 Critério de parada 28

3.4 Adicionando restrições 31

3.4.1 Esparsidade 31

3.4.2 Continuidade e suavização 33

3.5 Outros tipos de função-custo 33

3.5.1 β-divergente 34

3.6 Efeitos de regularização . . . 35
3.7 Interpretação . . . 35
3.7.1 Aplicações em áudio . . . 37

4 Experimentos com NMF **41**
4.1 Método de avaliação . . . 41
4.2 Base de dados . . . 41
4.3 Algoritmos utilizados . . . 43
4.4 Resultados . . . 43

5 Aplicação: Transcrição de Instrumentos Percussivos **51**
5.1 Transitório x *onset* x ataque . . . 51
5.2 Esquema geral dos algoritmos de detecção de *onset* . . . 52
5.3 Pré-processamento . . . 52
5.3.1 Múltiplas bandas . . . 53
5.4 Redução . . . 53
5.4.1 Características temporais . . . 53
5.4.2 Características espectrais . . . 54
5.4.3 Características espectrais utilizando a fase . . . 55
5.5 Pós-processamento . . . 56
5.6 Escolha de picos . . . 56
5.7 NMF aplicada à transcrição . . . 57
5.7.1 NMF com bases adaptativas . . . 58
5.7.2 NMF com bases fixas . . . 58
5.7.3 NMF com bases semiadaptativas . . . 59

6 Experimentos com Transcrição **61**
6.1 Método de avaliação . . . 61
6.1.1 Avaliação da separação: Medidas baseadas em SNR . . . 62
6.2 Base de dados . . . 64
6.3 Resultados . . . 65
6.3.1 Avaliação das funções de detecção . . . 65
6.3.2 Separação, transcrição e ressíntese . . . 67

7 Conclusão e Trabalhos Futuros **75**

Referências Bibliográficas **77**

A Cálculo Matricial **85**
A.1 Propriedades básicas do Traço . . . 85
A.2 Derivadas . . . 85

A.2.1 Derivadas de matrizes, vetores e escalares 86
A.2.2 Derivada do Traço 86

B Condições de Karush-Kuhn-Tucker (KKT) 87

Lista de Figuras

2.1 Modelo de BSS. 6
2.2 A natureza não-negativa e a representação por matrizes. 9
2.3 Diferentes representações para a NMF. 11

3.1 Diferentes usos para a NMF. (Imagens adaptadas de Cédric Févotte, Junho de 2015). 36
3.2 Experimento de Lee e Seung 37
3.3 Exemplo ilustrativo de como é um gráfico da NMF para a BSS em áudio 38

4.1 As imagens contidas na matriz $\mathbf{V}$ para a base Stirling. 43
4.2 Tempo acumulado gasto ao longo das iterações utilizando $R = 10$. . . 44
4.3 Tempo acumulado gasto ao longo das iterações utilizando $R = 80$. . . 45
4.4 Tempo acumulado gasto ao longo das iterações utilizando $R = 160$. . 46
4.5 Custo relativo em função do tempo 47
4.6 Redução da função-custo utilizando a base Stirling. 48
4.7 Imagens reconstruídas utilizando a base Stirling 49

5.1 Diferença entre transitório, *onset* e ataque. 51
5.2 Diagrama de blocos para detecção de *onsets*. 52
5.3 Exemplificação da transcrição de instrumentos percussivos utilizando a NMF 58

6.1 Exemplo de detecção de VP, FP e FN. Note que duas notas próximas são mescladas. 62
6.2 Uma bateria composta por (1) chimbau, (2) caixa e (3) bumbo, além de outros tambores e pratos. 65
6.3 Valor médio e desvio padrão de δ para cada função de detecção. . . . 67
6.4 Espectrogramas dos instrumentos da categoria WaveDrum 68
6.5 Espectrogramas dos instrumentos estimados da categoria WaveDrum 70
6.6 Valores de SIR utilizando bases adaptativas. 72
6.7 Valores de SAR utilizando bases adaptativas. 72

6.8 Valores de SDR utilizando bases adaptativas. 73
6.9 Valores de SIR do chimbau para cada tipo de separação 73

Lista de Tabelas

3.1 Comparação entre os diversos métodos para a NMF. 29
3.2 Como escolher a função-custo? (Adaptado de Cédric Févotte). 34

4.1 Informações das bases de dados utilizadas. 42

6.1 Resultados do teste de detecção de *onsets* utilizando a base MULT . 66
6.2 Resultados para a base IDMT maximizando o valor de F1 71
6.3 Resultados para a base IDMT utilizando $\delta = 0{,}0567$ 72

Lista de Símbolos

$\mathbf{A}$	Matriz não-negativa resultante da NMF ($\mathbf{Y} = \mathbf{AX}$), $\mathbf{A} \in \mathbb{R}^{M \times R}$
α_i	Constante não-negativa, $\alpha_i \in \mathbb{R}$, $i \in \{1,2\}$
$\boldsymbol{\alpha}_{\boldsymbol{\chi}}$	Pesos a serem dados nas restrições, $\boldsymbol{\alpha}_{\boldsymbol{\chi}} \in \mathbb{R}^{S \times 1}$
$\mathbf{A} \odot \mathbf{B}$	Produto ponto-a-ponto, $[\mathbf{A} \odot \mathbf{B}]_{i,j} = [\mathbf{A}]_{i,j}[\mathbf{B}]_{i,j}$ para todo i,j
$\mathbf{A} \oslash \mathbf{B}$	Divisão ponto-a-ponto, $[\mathbf{A} \oslash \mathbf{B}]_{i,j} = [\mathbf{A}]_{i,j}/[\mathbf{B}]_{i,j}$ para todo i,j
$\mathbf{A} \otimes \mathbf{B}$	Produto de Kronecker, $[\mathbf{A} \otimes \mathbf{B}]_{i,j} = [\mathbf{A}]_{i,j}\mathbf{B}$ para todo i,j
β	Definição da função-custo no β-divergente
C	Número de camadas para o algoritmo de múltiplas camadas
$\boldsymbol{\chi}$	Matriz arbitrária
$\mathbf{d}$	Direção do gradiente
$d(n)$	Amostras temporais após o processamento pela função de detecção
δ	Limiar para a escolha dos picos
$\hat{\delta}$	Limiar adaptativo
$\mathbf{diag}(\mathbf{x})$	Transformação de um vetor em uma matriz diagonal, $[\mathbf{diag}(\mathbf{x})]_{i,j} = x_i$ se e somente se $i = j$
$\mathbf{D}$	Matriz diagonal inversível, $[\mathbf{D}]_{i,j} \neq 0$ se e somente se $i = j$
$\Delta\phi(n,k)$	Desvio de fase
Δ	Raiz quadrada da soma das normas ao quadrado dos gradientes projetados
Δ_0	Valor de Δ utilizando $\mathbf{W}^{(0)}$ e $\mathbf{H}^{(0)}$

$\mathbf{e}_i$	Interferência causada por outras fontes
$\mathbf{e}_a$	Defeitos inseridos devido à separação
$\mathbf{e}_r$	Ruído presente na mistura
ϵ	Constante não-negativa
$\eta_{\boldsymbol{\chi}}$	Magnitude do passo a ser dado no algoritmo gradiente descendente
$\boldsymbol{\eta}_{\boldsymbol{\chi}}$	Matriz com valor do passo para a regra de atualizações multiplicativas
$E_0(n)$	Seguidor de envelope
$E(n)$	Seguidor de energia local
$f(\cdot)$	Função-custo
f_s	Frequência de amostragem
$f(\bar{m})$	Média ou mediana móvel
F1	Média harmônica entre P e S
γ	Controle de esparsidade definido por Hoyer, $\gamma \in [0,1]$
h	Salto entre janelas em amostras
$H(\cdot)$	Retificador
$\mathbf{H}$	Matriz não-negativa resultante da NMF ($\mathbf{V} = \mathbf{WH}$), $\mathbf{H} \in \mathbb{R}^{R \times N}$
$\mathbf{I}$	Matriz identidade
$\mathbf{1}$	Matriz com todos os elementos iguais a um, $[\mathbf{1}]_{i,j} = 1$ para todo i,j
$J^{\mathrm{d}}_{\boldsymbol{\chi}}$	Restrição de continuidade utilizando a primeira ou segunda derivada da matriz $\boldsymbol{\chi}$
$\mathbf{J}_{\boldsymbol{\chi}}$	Restrições, $\mathbf{J}_{\boldsymbol{\chi}} \in \mathbb{R}^{S \times 1}$
$J^{\ell_p}_{\mathbf{x}}$	Restrição utilizando a norma ℓ_p, onde $\mathbf{x} = \mathbf{vec}(\mathbf{X})$
$J^{\mathrm{H}}_{\mathbf{X}}$	Restrição de esparsidade de Hoyer

k	Raias de frequências, $k \in \{1, \ldots, K\}$
κ	Inteiro não-negativo utilizado no algoritmo gradiente projetado
L	Número de amostras em um quadro
λ	Constante para a regularização de Levenberg-Marquardt
$\max(\epsilon, \boldsymbol{\chi})$	Máximo calculado ponto-a-ponto, $[\max(\epsilon, \boldsymbol{\chi})]_{i,j} = \max(\epsilon, [\boldsymbol{\chi}]_{i,j})$ para todo i,j
μ	Constante utilizada no algoritmo gradiente projetado, $\mu \in (0,1)$
$\nabla_{\boldsymbol{\chi}} f$	Primeira derivada (gradiente) de f em relação a $\boldsymbol{\chi}$, $\frac{\partial f}{\partial \boldsymbol{\chi}}$
$\nabla^p_{\boldsymbol{\chi}} f$	Gradiente projetado
$[\nabla_{\boldsymbol{\chi}} f]_+$	Parte positiva do gradiente
$[\nabla_{\boldsymbol{\chi}} f]_-$	Parte negativa do gradiente
$\nabla^2_{\boldsymbol{\chi}} f$	Segunda derivada (Hessiana) de f em relação a $\boldsymbol{\chi}$, $\frac{\partial^2 f}{\partial \boldsymbol{\chi}^2}$
ν	Constante positiva de ponderação para o algoritmo de NMF com bases semiadaptativas
$\mathbf{P}$	Matriz de permutação, $\mathbf{P} \in \mathbb{R}^{R \times R}$
P	Precisão
$\phi(n,k)$	Frequência na n-ésima amostra temporal e na k-ésima raia de frequência
φ	Expoente do algoritmo de NMF com bases semiadaptativas
Q	Número de reinicializações para o algoritmo de inicialização robusta
$\mathbf{Q}$	Matriz ortogonal utilizada na fatoração **QR**
R	Número de componentes/sinais emitidos
$\mathbf{R}$	Matriz triangular superior utilizada na fatoração **QR**
$\mathbb{R}$	Conjunto dos reais
$\mathbf{s}$	Vetor que contém os sinais emitidos pelas fontes, $\mathbf{s} \in \mathbb{R}^{R \times 1}$

$\hat{\mathbf{s}}$	Fonte separada
$\mathbf{spar}(\mathbf{x})$	Medida de esparsidade definida por Hoyer [1]
$\mathbf{s}$	Fonte original
S	Número de restrições
S	Sensibilidade
σ	Constante utilizada no algoritmo gradiente projetado, $\sigma \in (0,1)$
ς	Constante definida por Hoyer
t	Contador da iteração, $t \in \{1, \ldots, T\}$
$\mathbf{T}$	Matriz de transformação arbitrária
T_{init}	Número de iterações para o algoritmo de inicialização robusta
τ	Fator de decaimento no algoritmo de mínimos quadrados não-negativos hierárquicos
$\mathbf{V}$	Matriz não-negativa a ser fatorada. $\mathbf{V} \in \mathbb{R}_+^{M \times N}$
$\hat{\mathbf{V}}$	Aproximação da matriz $\mathbf{V}$
$\mathbf{vec}(\mathbf{X})$	Operador que transforma uma matriz em um vetor, $\mathbf{vec}(\mathbf{X}) = [X_{1,1}\ X_{2,1}\ \cdots X_{M,1}\ X_{1,2}\ X_{2,2}\ \cdots\ X_{M,2}\ \cdots\ X_{M,N}]^T$
$\mathbf{W}$	Matriz não-negativa resultante da NMF ($\mathbf{V} = \mathbf{WH}$), $\mathbf{W} \in \mathbb{R}^{M \times R}$
$\mathbf{W}_p$	Base espectral com valores provenientes de um treinamento
$\mathbf{w}$	Janela arbitrária, $\mathbf{w} \triangleq [w_{-L/2}\ \cdots\ w_{L/2-1}]^T$
W_k	Peso dependente da frequência
x	Escalar, $x \in \mathbb{R}$
$\mathbf{x}$	Vetor coluna, $\mathbf{x} \in \mathbb{R}^{N \times 1} \triangleq [x_1 \cdots x_M]^T$
x_i	i-ésimo elemento do vetor $\mathbf{x}$
$\mathbf{X}$	Matriz não-negativa resultante da NMF ($\mathbf{Y} = \mathbf{AX}$), $\mathbf{X} \in \mathbb{R}^{R \times N}$

$X(n,k)$	Um elemento do espectrograma do sinal $\mathbf{x}$
$[\mathbf{X}]_{i,j}$	Elemento da i-ésima linha e j-ésima coluna da matriz $\mathbf{X}$
$X_{i,j}$	Elemento da i-ésima linha e j-ésima coluna da matriz $\mathbf{X}$
$[\mathbf{X}]_{i,*}$	Vetor linha formado pelos elementos $(X_{i,1} \ \cdots X_{i,N})$ pertencentes a i-ésima linha da matriz de $\mathbf{X}$
$[\mathbf{X}]_{*,j}$	Vetor coluna formado pelos elementos $(X_{1,j} \ \cdots X_{M,j})^T$ pertencentes a j-ésima coluna da matriz de $\mathbf{X}$
$\mathbf{X}^{\cdot p}$	Exponenciação ponto-a-ponto, $[\mathbf{X}^{\cdot p}]_{i,j} = [\mathbf{X}]_{i,j}^p$ para todo i,j
$\mathbf{X}^{(t)}$	Valor da matriz $\mathbf{X}$ na t-ésima iteração
ξ	Constante positiva para a fórmula do limiar adaptativo
$\mathbf{Y}$	Matriz não-negativa a ser fatorada. $\mathbf{Y} \in \mathbb{R}_+^{M \times N}$
$[\,\cdot\,]_+$	Projeção dos valores negativos, $[\cdot]_+ = \max(\epsilon, \cdot)$
'\'	Operador de eliminação Gaussiana
$(\cdot)^\dagger$	Pseudoinversa de Moore-Penrose

Lista de Abreviaturas

ALS	*Alternating Least Squares*, p. 20
ANLS	*Alternating Non-negative Least Squares*, p. 21
AS	*Active Set*, p. 23
BPP	*Block Principal Pivoting*, p. 26
BSS	*Blind Source Separation*, p. 4
CDMA	*Code Division Multiple Access*, p. 6
DFT	*Discrete Fourier Transform*, p. 39
FIR	*Finite Impulse Response*, p. 53
FNMA	*Fast Non-negative Matrix Approximation*, p. 14
FN	Falso Negativo, p. 57
HALS	*Hierarchical Alternating Least Squares*, p. 21
HFC	*High Frequency Content*, p. 50
HH	*High Hat*, p. 63
ICA	*Independent Component Analysis*, p. 5
KD	*Kick Drum*, p. 63
LP	*Linear Programming*, p. 14
MIR	*Music Information Retrieval*, p. 2
MUR	*Multiplicative Update Rule*, p. 14
NMF2D	*Non-Negative Matrix Factor 2-D Deconvolution*, p. 37
NMFD	*Non-negative Matrix Factor Deconvolution*, p. 36

NMF	*Non-negative Matrix Factorization*, p. 1
NNLS	*Non-negative Least Squares*, p. 21
PCA	*Principal Component Analysis*, p. 5
PD	*Phase Deviation*, p. 52
PG	*Projected Gradient*, p. 17
PQNK	*Practical Quasi-Newton* desenvolvido por Kim [2], p. 27
PQNZC	*Practical Quasi-Newton* desenvolvido Zdunek e Cichocki [3], p. 27
PQN	*Practical Quasi-Newton*, p. 19
SAR	*Sources-to-Artifacts Ratio*, p. 58
SDR	*Source-to-Distortion ratio*, p. 58
SD	Snare Drum, p. 63
SD	*Spectral Difference*, p. 50
SF	*spectral flux*, p. 50
SIR	*Sources-to-Inteferences Ratio*, p. 58
SNMF	*Sparse Non-negative Matrix Factorization* [1], p. 27
SNR	*Signal-to-Noise Ratio*, p. 52
STFT	*Short-Time Fourier Transform*, p. 35
SVD	*Singular Value Decomposition*, p. 12
VP	Verdadeiro Positivo, p. 57
WPD	*Weighted Phase Deviation*, p. 52

Capítulo 1

Introdução

A fatoração de matrizes não-negativas (NMF, do inglês *Non-negative Matrix Factorization*) é uma ferramenta de álgebra linear bastante utilizada para redução de dimensionalidade e para o uso em *factor analysis*[1]. Muitas técnicas de redução de dimensão estão intimamente ligadas a técnicas de aproximação de baixo posto, e a NMF é um caso especial, em que as matrizes de baixo posto são restritas a serem não-negativas. A não-negatividade é muitas vezes uma representação natural de muitos problemas que envolvem situações reais e, ao contrário de outros métodos, a NMF resulta em representações que contêm algum significado físico [5].

A NMF foi introduzida por Pateero e Tapper [6] como fatoração de matrizes positivas (do inglês, *positive matrix factorization*) e subsequentemente foi popularizada por Lee e Seung [7]. Desde então, a NMF tem sido usada com sucesso em mineração de texto [8], análise espectral [9], *speech enhancement* [10] e separação cega [11].

A principal meta dessa monografia é descrever dentro de um mesmo arcabouço os principais algoritmos desenvolvidos nos últimos anos para resolver a fatoração. De todos os algoritmos que serão estudados, o mais popular é, sem dúvidas, a regra de atualizações multiplicativas criada por Lee e Seung [12], por ser fácil de implementar e entender. No entanto, como apontado por diversos autores, esse algoritmo apresenta uma baixa taxa de convergência. Felizmente, novos algoritmos com fundamentação teórica mais sólida e com provas de convergência foram criados, tais como os baseados em mínimos quadrados não-negativos alternados [13] e mínimos quadrados alternados hierárquicos [14], que também serão devidamente descritos nos próximos capítulos.

A segunda parte deste projeto final consiste na aplicação da NMF na área de separação de instrumentos musicais, exemplificando como e onde a NMF pode ser utilizada. Será mostrado como se realiza a transcrição de instrumentos dada uma

[1] *Factor analysis* é um método estatístico para descrever a variabilidade de variáveis correlacionadas observáveis através de variáveis não observáveis (fatores) [4].

composição musical disponível em uma única mistura gravada. A transcrição automática de instrumentos musicais possui duas vertentes principais: a transcrição tonal (como as notas de uma flauta) e percussiva (como uma bateria) [15]. Este trabalho focará na tarefa de transcrição de sinais percussivos, que consiste em estimar temporalmente eventos de *onset*, *i.e.*, os momentos em que as notas são acionadas.

Os algoritmos que serão descritos para a transcrição de instrumentos percussivos podem ser facilmente aplicáveis em diferentes áreas de extração de informação musical (MIR, do inglês *Music Information Retrieval*).

1.1 Organização

O Capítulo 2 descreve as principais técnicas que existem na literatura aplicáveis à separação cega de fontes, com o intuito de motivar o leitor ao uso da fatoração de matrizes não-negativas. Em seguida, no Capítulo 3, serão descritos detalhadamente os principais algoritmos para resolver o problema da NMF, focando sempre na multiplicidade de aplicações da fatoração de matrizes não-negativas. Também será descrito como melhorar a acurácia dos algoritmos utilizando restrições e outros tipos de função-custo. Os experimentos demonstrando os defeitos e qualidades de cada algoritmo serão descritos no Capítulo 4. A segunda parte do projeto final, que consiste na transcrição de instrumentos percussivos, será trabalhada no Capítulo 5. No Capítulo 6 será feita a aferição de resultados comparando os diversos algoritmos com algumas bases de dados. Finalmente, o Capítulo 7 irá conduzir o leitor às considerações finais do trabalho e sugerir possíveis caminhos a serem tomados futuramente.

1.2 Notação

Uma matriz sempre será representada em caixa alta e em negrito, $\mathbf{X} \in \mathbb{R}^{M \times N}$. Um elemento da matriz na i-ésima linha e j-ésima coluna é representado como $[\mathbf{X}]_{i,j}$ e o uso do caractere especial '$*$' denota a seleção de uma linha ou coluna inteira da matriz, por exemplo: $[\mathbf{X}]_{i,*}$ denota um vetor linha formado pelos elementos $(x_{i,1} \ \ldots \ x_{i,N})$ pertencentes a i-ésima linha da matriz $\mathbf{X}$. Um vetor é representado em negrito utilizando caixa baixa, como $\mathbf{x} \in \mathbb{R}^{N \times 1} \triangleq [x_1 \ \cdots \ x_N]^T$ e um elemento do vetor $\mathbf{x}$ é denotado como x_i, $i \in \{1, \ldots, N\}$.

A letra $t \in \{1, \cdots, T\}$ indicará a iteração e T a iteração máxima. O número de componentes para realizar a fatoração será indicado por R, que também significa a quantidade de sinais emitidos.

A matriz $\mathbf{P} \in \mathbb{R}^{R \times R}$ é uma matriz de permutação, *i.e.*, $\mathbf{P}\mathbf{P}^T = \mathbf{I}$, sendo $\mathbf{I}$ a matriz identidade. O símbolo $\mathbf{1} \in \mathbb{R}^{M \times N}$ representa uma matriz com M linhas e N

colunas com todos os elementos iguais a um: $[\mathbf{1}]_{i,j} = 1$, $\forall i,j$. A matriz $\mathbf{D}$ é uma diagonal inversível, isto é

$$\mathbf{D} = \begin{bmatrix} D_{1,1} & 0 & 0 & \cdots & 0 \\ 0 & D_{2,2} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & D_{M,M} \end{bmatrix} \tag{1.1}$$

onde $[\mathbf{D}]_{i,i} \neq 0$, $\forall i$.

Dada uma função f, o seu gradiente e sua Hessiana em relação à uma matriz $\boldsymbol{\chi}$ é denotada por $\nabla_{\boldsymbol{\chi}} f$ e $\nabla^2_{\boldsymbol{\chi}} f$, respectivamente.

Uma lista de símbolos completa pode ser encontrada na página xiv.

Capítulo 2

Separação Cega

Grande parte das pesquisas relacionadas à análise de sinais complexos (*e.g.*, sinais de fala) envolvem a realização de alguma transformação para simplificação de sua representação ao mesmo tempo que lhe empresta uma interpretação física evidente/natural.

Considere-se, por exemplo, uma situação em que há um número arbitrário tanto de sinais sendo emitidos quanto de sensores. As fontes emissoras podem ser instrumentos de uma orquestra, os corações de um feto e de sua mãe, celulares transmitindo sinais de rádiofrequência ou até mesmo elementos diversos em uma composição de imagens. A partir de suas misturas capturadas pelos respectivos sensores (microfones, antenas etc.), o problema de separação cega de sinais (BSS, do inglês *Blind Source Separation*) consiste em determinar quem são essas fontes e possivelmente como foram misturadas.

Por sua vez, essas misturas podem ser convolutivas [16], como na transmissão de sinais por um canal linear com memória, ou simplesmente aditivas, quando ocorre uma soma ponderada de diversos sinais, como em um misturador (em inglês, *mixer*) de áudio. Separar os sinais misturados no primeiro caso é um problema mais complicado e que utiliza sistemas mais complexos; as soluções para o segundo caso já foram mais profundamente estudadas e produzem bons resultados.

Matematicamente, o problema a ser resolvido pela BSS no caso de mistura aditiva pode ser descrito como

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \tag{2.1}$$

em que $\mathbf{A} \in \mathbb{R}^{M \times R}$ contém os pesos da mistura e $\mathbf{s} \in \mathbb{R}^{R \times 1}$ contém os sinais emitidos pelas fontes, ambos desconhecidos, conforme ilustrados na Figura 2.1. Considerar os termos do lado direito da equação como sendo indeterminados é natural, pois conhecer os pesos $[\mathbf{A}]_{i,j}$ requer a identificação do sistema físico por trás da mistura, o que é na maioria dos casos inviável, e não há gravações diretas dos sinais s_i

originais.

Em outras palavras, o problema da separação cega de sinais consiste em descobrir as fontes s_i e possivelmente os pesos $[\mathbf{A}]_{i,j}$ tendo somente as observações x_i. Nesse sentido, a denominação "cega" consiste em saber nada ou muito pouco sobre as fontes [17].

Para o caso em que os pesos $[\mathbf{A}]_{i,j}$ apresentam variabilidade suficiente (posto completo por colunas), basta realizar o caminho inverso[1], *i.e.*, $\mathbf{s} \approx \hat{\mathbf{s}} = \mathbf{W}\mathbf{x}$. Neste sentido, estimar $\mathbf{W}$ também reconstrói total ou parcialmente as fontes.

Nas últimas décadas surgiram diferentes métodos aplicáveis às mais diversas situações, cujo intuito é obter uma boa representação dos sinais separados. Porém, para essas soluções serem viáveis, algumas suposições são feitas sobre as fontes, tais como: independência, esparsidade e/ou não-negatividade.

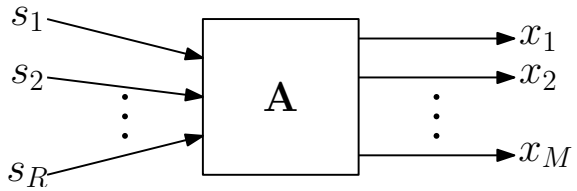


Figura 2.1: Modelo de BSS.

2.1 Separação baseada na independência

Diferentemente da Análise de Componente Principais (PCA, do inglês *Principal Component Analysis*) [18], que se baseia na descorrelação entre sinais, a Análise de Componentes Independentes (ICA, do inglês *Independent Component Analysis*), proposta por Pearson [19] e independentemente por Hotelling [20], se baseia na independência estatística (uma propriedade muito mais forte que a descorrelação) e chega a uma solução surpreendentemente simples para a BSS.

Busca-se estimar a matriz $\mathbf{W}$ de modo que o produto $\mathbf{W}\mathbf{A}$ seja o mais próximo possível da identidade e, assim, o vetor estimado $\hat{\mathbf{s}}$ seja o mais próximo possível do original $\mathbf{s}$, com a restrição de que haja no mínimo tantos sensores quanto fontes ($R \leq M$). Caso a desigualdade seja estrita, geralmente aplica-se a PCA para retornar as R maiores componentes de $\mathbf{x}$, permitindo trabalhar com matrizes quadradas.

A ICA pressupõe que as amostras dos sinais originais antes da mistura constituem variáveis aleatórias mutuamente independentes e não-gaussianas, permitindo, assim, lançar mão do Teorema do Limite Central (que diz que a soma de variáveis aleatórias

[1]Se o número de sensores (linhas) linearmente independentes for igual ao número de fontes (colunas), o sistema é determinado e possui uma única solução dada pela matriz inversa de $\mathbf{A}$ ($\mathbf{W} = \mathbf{A}^{-1}$); se for maior, o sistema é sobredeterminado e admite uma solução de erro quadrático mínimo por pseudoinversa de $\mathbf{A}$ ($\mathbf{W} \approx \mathbf{A}^{\dagger} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$); se for menor, há infinitas soluções. Nesse caso, costuma-se adicionar alguma restrição ao problema a fim de obter uma única solução.

independentes tende a ter uma distribuição tão mais próxima de uma gaussiana quanto maior o número de variáveis) para determinar o próprio critério de separação: busca-se o conjunto de sinais desmisturados minimamente gaussianos. Isso pode ser feito, por exemplo, pela maximização da negentropia [21] ou pela minimização do módulo da curtose [22]. A minimização da informação mútua [23, 24] é outra forma de realizar a separação.

Há dois problemas inerentes ao método de ICA que não podem ser ignorados e que se propagam para outros métodos que serão citados: a indeterminação da ordem das componentes originais e de suas energias.

Observação 1 *Indeterminação da ordem das componentes:*
Existem uma matriz de permutação $\mathbf{P}$ *e sua inversa* $\mathbf{P}^T$ *tais que*

$$\mathbf{x} = \mathbf{A}\mathbf{P}\mathbf{P}^T\mathbf{s};$$

portanto, $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{P}$ *é uma outra possível matriz de mistura se* $\tilde{\mathbf{s}} = \mathbf{P}^T\mathbf{s}$ *são as fontes com linhas trocadas. Então, não há como determinar qual é a ordem original da mistura que foi realizada.*

Observação 2 *Indeterminação da energia das componentes:*
Existe uma matriz $\mathbf{D}$ *diagonal inversível tal que*

$$\mathbf{x} = \mathbf{A}\mathbf{D}\mathbf{D}^{-1}\mathbf{s};$$

assim, temos uma outra possível matriz de mistura $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{D}$ *se* $\tilde{\mathbf{s}} = \mathbf{D}^{-1}\mathbf{s}$*, de forma que cada fonte pode aparecer escalada em relação ao problema original. Então, não há como garantir qual é a energia do sinal.*

No entanto, há técnicas de pré-processamento que mitigam este segundo problema, assumindo, por exemplo, que as fontes tenham média zero e variância unitária (o que é conhecido como branqueamento).

A ICA é utilizada nas mais diversas áreas, tais como em árvores de decisão, mineração de texto, análise financeira, remoção de artefatos de imagens provenientes de telescópios espaciais e até mesmo em comunicações por CDMA (do inglês, *Code Division Multiple Access*) [25].

Como pode ser notado, o ecossistema existente por trás da ICA é muito extenso e caso haja o interesse de se aprofundar sobre o assunto, o leitor interessado pode buscar mais detalhes em [17].

2.2 Separação baseada na esparsidade

Considerar que há sempre disponível um número de sensores maior que ou igual ao número de fontes impossibilita o uso prático da ICA em muitos cenários. Em contrapartida, operar sobre um sistema subdeterminado ($R > M$) obriga que se façam mais hipóteses sobre o sinal de entrada.

Uma dessas conjecturas é o critério de esparsidade, e a classe de algoritmos sem supervisão que lida com este problema é a codificação esparsa (do inglês, *Sparse Coding*) [26]. Originalmente, este algoritmo surgiu na área de estudos do comportamento do cérebro humano, porém vem sendo amplamente utilizado nas mais diversas áreas de processamento de sinais.

Neste projeto de graduação, o termo esparsidade é definido como sendo uma característica das representações que contêm poucos elementos não-nulos; portanto é dito que um vetor $\mathbf{x}$ é esparso se quase todos seus elementos são zero[2]. A codificação esparsa assume que as fontes são ou possuem representações esparsas (*e.g.*, a transformada de Fourier de um cosseno) e, através dos sinais adquiridos pelos sensores, pode-se determinar essas fontes buscando-se a representação maximamente esparsa delas.

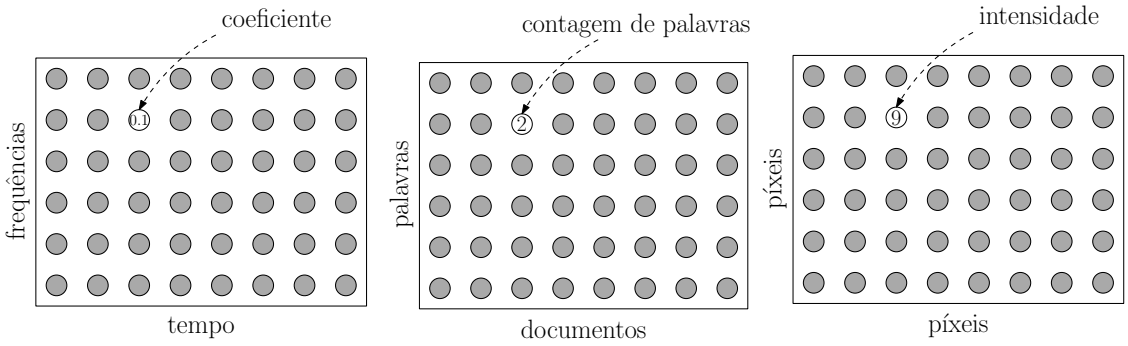
Grande parte dos dados de interesse são esparsos por natureza e por isso, no próximo capítulo, o critério de esparsidade será combinado com outras técnicas de separação cega de fontes.

2.3 Separação baseada na não-negatividade

Um aspecto essencial dos métodos discutidos até agora é que eles exploram alguma característica facilitadora presente em sua representação (independência entre fontes, esparsidade das fontes no tempo e/ou na frequência). Por vezes, os dados de entrada estão disponíveis na forma de uma representação não-negativa, como por exemplo, o espectrograma de magnitude ou os píxeis de uma imagem (vide Figura 2.2). Tal família de representações apresenta uma interessante característica facilitadora para a separação, embora não imediatamente aparente: ela impede a ocorrência de interferência destrutiva entre diferentes sinais da mistura, e assim preserva ao menos a possibilidade teórica de desacoplá-los de volta. Nesse caso, qualquer ferramenta de separação precisa incluir como restrição a preservação da não-negatividade.

Apesar de existirem métodos com restrições de não-negatividade para a ICA [27], eles são mais complicados e não levam a resultados de fácil interpretação.

[2]Na prática este conceito é relaxado, considerando que um vetor é esparso se quase todos os seus elementos possuem valores perto de zero.



(a) Representação tempo-frequencial. (b) Contagem de palavras. (c) Imagem em tons de cinza.

Figura 2.2: A natureza não-negativa e a representação por matrizes.

Neste contexto, a Fatoração de Matrizes Não-Negativas (NMF, do inglês *Non-negatize Matrix Factorization*) [28, 29] firmou-se como a alternativa mais promissora para estimar sinais não-negativos. Como se discutirá adiante, a hipótese subjacente que garante a capacidade de separação da NMF está associada à esparsidade das fontes.

A NMF tem sido objeto de extensa pesquisa nas últimas décadas, sendo utilizada nas mais diversas áreas, como *clustering* [8], *text mining* [30], análise espectral [31], processamento de imagens [32], classificação de câncer [13, 33–35], separação de fontes sonoras [36] e transcrição automática de música [15].

O problema pode ser exposto da seguinte maneira.

Problema 1 *Fatoração de Matrizes Não-negativas:*

Dada uma matriz $\mathbf{V} \in \mathbb{R}_+^{M\times N}$*, o objetivo da NMF é reduzir a dimensão para* $R \ll \min(M, N)$ *fatorando* $\mathbf{V}$ *em duas matrizes não-negativas* $\mathbf{W} \in \mathbb{R}_+^{M\times R}$ *e* $\mathbf{H} \in \mathbb{R}_+^{R\times N}$ *(Figura 2.3a) tais que*

$$\mathbf{V} \approx \hat{\mathbf{V}} = \mathbf{WH}. \tag{2.2}$$

As matrizes $\mathbf{W}$ *e* $\mathbf{H}$ *podem ser encontradas resolvendo-se o problema de otimização*

$$\begin{aligned} &\textit{minimizar} && f(\mathbf{W}, \mathbf{H}) \\ &\textit{sujeito a} && \mathbf{W} \geq 0, \\ & && \mathbf{H} \geq 0, \end{aligned} \tag{2.3}$$

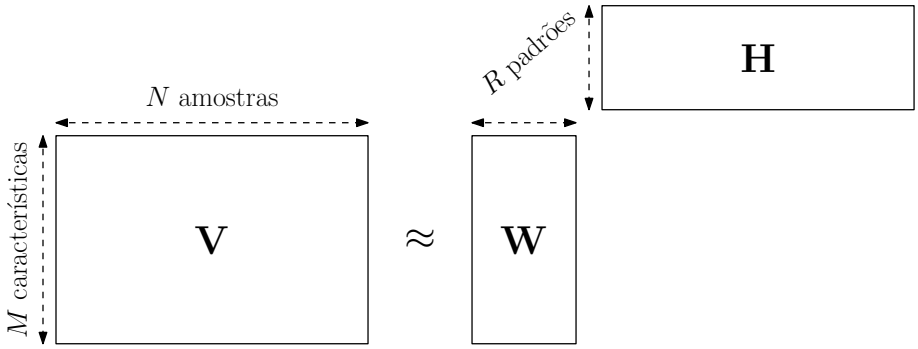
sendo $f(\mathbf{W}, \mathbf{H})$ *uma função-custo que denota uma medida de distorção entre a mistura estimada* $\hat{\mathbf{V}}$ *e a mistura original* $\mathbf{V}$*. Normalmente, a divergência de Kullback-Leibler ou a norma de Frobenius é utilizada. Infelizmente, este problema não é*

convexo para $\mathbf{W}$ *e* $\mathbf{H}$*; no entanto, é convexo em* $\mathbf{W}$ *com* $\mathbf{H}$ *fixo ou em* $\mathbf{H}$ *com* $\mathbf{W}$ *fixo.*

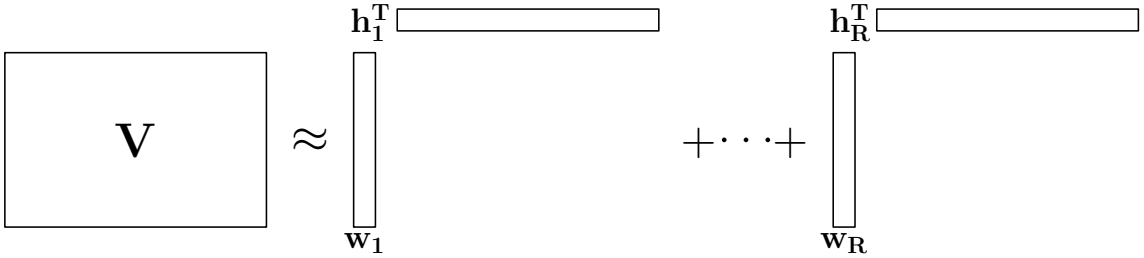
Essa formulação do problema não é única. A NMF pode ser representada como uma forma especial de um modelo bilinear (veja a Figura 2.3b),

$$\mathbf{V} \approx \hat{\mathbf{V}} = \sum_{r=1}^{R} \mathbf{w}_r {\mathbf{h}_r}^T; \tag{2.4}$$

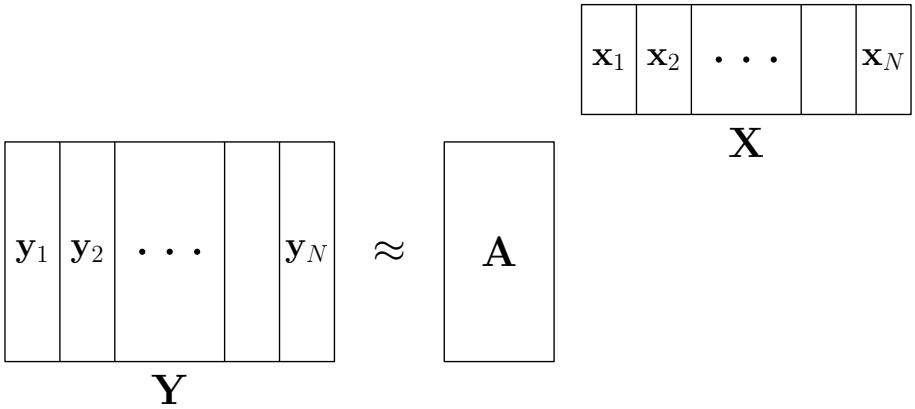
portanto, pode-se aproximar a representação não-negativa da matriz $\mathbf{V}$ por uma soma de matrizes não-negativas de posto um. Note que o problema é simétrico: ao se adotar $\mathbf{V}^T \approx \mathbf{H}^T\mathbf{W}^T$ a fatoração é a mesma, portanto dizer que $\mathbf{H}$ mistura $\mathbf{W}$ gerando $\hat{\mathbf{V}}$ ou $\mathbf{W}$ mistura $\mathbf{H}$ gerando $\hat{\mathbf{V}}$ na NMF é arbitrário. Isso será utilizado mais adiante no desenvolvimento dos algoritmos. Como consequência, a NMF também pode ser vista como uma fatoração do tipo $\mathbf{y}_i = \mathbf{A}\mathbf{x}_i$, $i \in \{1, \ldots, N\}$, sendo $\mathbf{Y} = [\mathbf{y}_1 \ \cdots \ \mathbf{y}_N]$, $\mathbf{X} = [\mathbf{x}_1 \ \cdots \ \mathbf{x}_N]$ e $\mathbf{A} \in \mathbb{R}_+^{M \times N}$ (Figura 2.3c). Neste caso, $\mathbf{Y} = \mathbf{A}\mathbf{X} = \mathbf{W}\mathbf{H}$, e a transposta também é uma fatoração de matrizes não-negativas.



(a) Representação matricial ilustrando a fatoração da matriz não-negativa $\mathbf{V}$ em $\mathbf{W}$ e $\mathbf{H}$.



(b) Modelo bilinear para a NMF. A matriz $\mathbf{V}$ é aproximada por uma combinação linear de matrizes não-negativas de posto um.



(c) A NMF como uma união de representações vetoriais. O vetor $\mathbf{y}_i$ é aproximado por uma combinação linear de uma matriz não-negativa $\mathbf{A}$ e um vetor não-negativo $\mathbf{x}_i$.

Figura 2.3: Diferentes representações para a NMF.

Capítulo 3

Fatoração de Matrizes Não-Negativas

Como visto no capítulo anterior, a fatoração de matrizes não-negativas constitui uma vasta área de pesquisa, sendo empregada em áreas que vão da biologia à engenharia, não se limitando à solução de problemas de separação cega de fontes.

A NMF tornou-se famosa a partir de 1999, com os esforços de Daniel D. Lee e H. Sebastian Seung [7, 12] no sentido de obterem um método fácil para sua implementação, apesar de, anos antes, um método baseado em mínimos quadrados [6] ter sido proposto. Desde então, diversos outros métodos foram surgindo, visando a melhorar convergência, velocidade e robustez do sistema.

Os métodos mais utilizados para realizar a fatoração se baseiam em atualizações alternadas, devido ao fato de o problema não ser estritamente convexo em ambas as variáveis; a alternativa de otimizar o problema original requereria uma busca exaustiva em um espaço que cresce exponencialmente com o tamanho da matriz $\mathbf{V}$, caracterizando o problema como NP-completo [37]. Os algoritmos podem ser descritos pelos seguintes passos:

1) $t \leftarrow 0$. Inicialize $\mathbf{W}^{(0)}$ e $\mathbf{H}^{(0)}$;

2) Fixe $\mathbf{W}^{(t)}$ e encontre $\mathbf{H}^{(t+1)}$ tal que

$$f(\mathbf{W}^{(t)}, \mathbf{H}^{(t+1)}) \leq f(\mathbf{W}^{(t)}, \mathbf{H}^{(t)});$$

3) Fixe $\mathbf{H}^{(t+1)}$ e encontre $\mathbf{W}^{(t+1)}$ tal que

$$f(\mathbf{W}^{(t+1)}, \mathbf{H}^{(t+1)}) \leq f(\mathbf{W}^{(t)}, \mathbf{H}^{(t+1)});$$

4) (Opcional) Normalize as colunas de $\mathbf{W}^{(t+1)}$ e as linhas de $\mathbf{H}^{(t+1)}$;

5) $t \leftarrow t + 1$. Repita os passos 2, 3 e 4 até que o critério de parada seja satisfeito.

Portanto, há três etapas cruciais: inicialização, execução e parada. Todos os passos serão elucidados neste capítulo, que tem o intuito de apresentar ao leitor as ferramentas para realizar a fatoração de matrizes não-negativas.

3.1 Inicialização

O fato de a fatoração não ser convexa deixa o problema extremamente sensível à inicialização; portanto, dependendo de suas condições iniciais, a NMF irá convergir para pontos distintos. Então, uma boa escolha de $\mathbf{W}^{(0)}$ e $\mathbf{H}^{(0)}$ produz mínimos locais melhores. É importante ter métodos robustos para a inicialização dos parâmetros, algo negligenciado por vários autores, em especial na área de BSS de sinais musicais.

O modo mais simples, e bem eficaz, de se realizar o processo é utilizar uma simulação de Monte Carlo para determinar a melhor inicialização, como mostrado no Pseudocódigo 1. A inicialização robusta [38] é obtida gerando-se Q matrizes[1] distintas $\mathbf{W}^{(0)}$ e $\mathbf{H}^{(0)}$ com base em inicialização aleatória ou em outro método arbitrário. Após isso, é necessário fatorar as matrizes utilizando um dos algoritmos que serão explicados na próxima seção, tendo como critério de parada um número pequeno de iterações (tipicamente entre 10 e 20). Finalmente, as melhores candidatas são as matrizes $\mathbf{W}_q$ e $\mathbf{H}_q$ que retornam o menor custo. Após o processo, essas matrizes candidatas serão utilizadas para a inicialização da fatoração final.

Portanto, a inicialização robusta retorna a estimação que possui as melhores matrizes no sentido de diminuição da função-objetivo $f(\mathbf{W}, \mathbf{H})$. Geralmente, a divergência generalizada de Kullback-Leibler (KL) é utilizada como critério de comparação, em conjunto com um $T_{\text{init}} \geq 10$.

Vários pesquisadores propuseram diferentes métodos de inicialização ao longo dos anos com o intuito de acelerar a convergência ou para obter mínimos melhores. Wild *et al.* [39] utilizaram uma clusterização *k-means* esférica para inicializar $\mathbf{W}$, enquanto que em [40] usaram uma inicialização baseada na decomposição de valores singulares (SVD, do inglês *Singular Value Decomposition*). No entanto, esse tema ainda continua sendo um problema em aberto.

3.1.1 Múltiplas camadas

Outro modo de contornar o problema de o algoritmo estagnar em mínimos locais ruins foi proposto por Cichocki *et al.* [41]. A ideia é trocar a fatoração $\mathbf{W}$ e $\mathbf{H}$ por uma cascata de matrizes $\mathbf{W}_c$,

$$\mathbf{V} = \mathbf{W}_1 \mathbf{W}_2 \cdots \mathbf{W}_C \mathbf{H}. \tag{3.1}$$

[1] O Q tipicamente utilizado fica entre 20 e 30.

Pseudocódigo 1 Inicialização Robusta

Entrada:

$\mathbf{V} \in \mathbb{R}_+^{M \times N}$

R: número de componentes

Q: número de reinicializações

T_{init}: número de passos

Saída:

$\mathbf{W} \in \mathbb{R}_+^{M \times R}$ e $\mathbf{H} \in \mathbb{R}_+^{R \times N}$

1: **para** $q = 1$ até Q **faça**
2: Inicialize $\mathbf{W}^{(0)}$ e $\mathbf{H}^{(0)}$
3: $[\mathbf{W}_q, \mathbf{H}_q] \leftarrow \text{NMF}(\mathbf{V}, \mathbf{W}^{(0)}, \mathbf{H}^{(0)}, T_{\text{init}})$
4: $f_q \leftarrow f(\mathbf{W}_q, \mathbf{H}_q)$
5: **fim para**
6: $q_{\min} \leftarrow \text{argmin}\, f_q$
7: **retorna** $\mathbf{W}_{q_{\min}}, \mathbf{H}_{q_{\min}}$

Após o processo, todas as matrizes $\mathbf{W}_c$, $c \in \{1, \ldots, C\}$ podem ser multiplicadas, resultando em uma única matriz $\mathbf{W}$. Como mostrado pelos autores, isto reduz o risco de convergência para mínimos locais ruins, principalmente para matrizes mal condicionadas.

Como pode ser observado no Pseudocódigo 2, a matriz $\mathbf{V}$ é fatorada em $\mathbf{V} \approx \mathbf{W}_1\mathbf{H}_1$ usando um algoritmo NMF arbitrário. Após a primeira iteração, a matriz $\mathbf{V}$ será substituída pelo $\mathbf{H}_1$, gerando uma nova fatoração. Esse processo pode ser repetido até que se satisfaça algum critério de parada. Note que não há restrições quanto ao uso de diferentes regras de atualização para cada camada.

Pseudocódigo 2 Múltiplas camadas

Entrada:

$\mathbf{V} \in \mathbb{R}_+^{M \times N}$

R: número de componentes

C: número de camadas

Saída: $\mathbf{W} \in \mathbb{R}_+^{M \times R}$ e $\mathbf{H}_C \in \mathbb{R}_+^{R \times N}$

1: $\mathbf{H}_0 \leftarrow \mathbf{V}$
2: **para** $c = 1$ até C **faça**
3: $[\mathbf{W}_c, \mathbf{H}_c] \leftarrow \text{NMF}_c(\mathbf{H}_{c-1}, R)$
4: **fim para**
5: **retorna** $\mathbf{W} = \mathbf{W}_1 \cdots \mathbf{W}_C$, $\mathbf{H}_C$

3.2 Algoritmos

Após o artigo publicado na Nature em 2001 por Lee e Seung, portas foram abertas para a comunidade científica. Naquele ano foi demonstrado que a NMF pode representar partes de rostos [12] e, paralelamente, ela foi utilizada para extrair informações de textos. Anos mais tarde, em 2003, Bergman, Ihmels e Barkai começaram a utilizar a fatoração em microarranjos de DNA [42] e no mesmo ano, Smaragdis e Brown estavam extraindo notas de uma música polifônica [43].

Enquanto a NMF ia sendo aplicada nas mais diversas áreas, múltiplos algoritmos foram criados e adaptados. Em 2004, Hoyer propôs a fatoração utilizando uma função-custo adaptada para esparsidade [1]. Surgiram métodos baseados em mínimos quadrados, assim como algoritmos que lidavam diretamente com restrições de não-negatividade. Quase-Newton [3], FNMA (do inglês, *Fast Non-negative Matrix Approximation*) [2], gradiente projetado [44], conjunto ativo [13], pontos interiores [45, 46], mínimos quadrados hierárquicos [14] e *block principal pivoting* [47, 48] são alguns dentre muitos que podem ser citados. Esses dois últimos, após sofrerem modificações ao longo dos anos, têm sido considerados como o estado da arte em questões de desempenho [49].

Os algoritmos descritos no resto dessa seção foram selecionados por trazerem conteúdos históricos, como as regras de atualizações multiplicativas de Lee e Seung, ou para explicar um pouco dos algoritmos do estado da arte, como o conjunto ativo, que serve como base para o entendimento do algoritmo *block principal pivoting*.

3.2.1 MUR

Lee e Seung mostraram como realizar um processo iterativo para realizar a minimização do problema da fatoração utilizando a norma de Frobenius ou a distância de Kullback-Leibler como função-custo.

Ambas as funções são convexas somente em $\mathbf{W}$ ou $\mathbf{H}$, mas não são convexas em ambas as variáveis, portanto o algoritmo desenvolvido resulta em ótimos locais. No caso, Lee e Seung utilizam o método de gradiente descendente, que possui uma convergência lenta para problemas que não são de programação linear (LP, do inglês *Linear Programming*) e é altamente sensível ao passo escolhido. De fato, a escolha de um passo inteligente garante o que eles chamaram de regra de atualização multiplicativa (MUR, do inglês *Multiplicative Update Rule*), proporcionando que, uma vez tendo-se matrizes não-negativas, a atualização garanta a preservação da não-negatividade.

Para a norma de Frobenius, temos o problema definido como

$$\text{minimizar} \quad f(\mathbf{W},\mathbf{H}) = \frac{1}{2}\|\mathbf{V}-\mathbf{WH}\|_{\mathrm{F}}^2 = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[|\mathbf{V}-\mathbf{WH}|\right]_{i,j}^2, \tag{3.2}$$

onde as restrições são ignoradas a princípio. O processo iterativo consiste em alternadamente realizar o método de gradiente descendente [50] para a matriz $\mathbf{W}$ e $\mathbf{H}$:

$$\mathbf{W} \leftarrow \mathbf{W} - \boldsymbol{\eta}_{\mathbf{W}}\nabla_{\mathbf{W}} f, \quad \mathbf{H} \leftarrow \mathbf{H} - \boldsymbol{\eta}_{\mathbf{H}}\nabla_{\mathbf{H}} f, \tag{3.3}$$

sendo[2] $\boldsymbol{\eta}_{\boldsymbol{\chi}}$ uma matriz contendo os passos a serem escolhidos e $\nabla_{\boldsymbol{\chi}} f$ o gradiente. Podemos reescrever a norma de Frobenius (Apêndice A)

$$\begin{aligned}\frac{1}{2}\|\mathbf{V}-\mathbf{WH}\|_{\mathrm{F}}^2 &= \frac{1}{2}\operatorname{Tr}[(\mathbf{V}-\mathbf{WH})^T(\mathbf{V}-\mathbf{WH})] \\ &= \frac{1}{2}\left\{\operatorname{Tr}(\mathbf{V}^T\mathbf{V}) + \operatorname{Tr}(\mathbf{H}^T\mathbf{W}^T\mathbf{WH}) - \operatorname{Tr}(\mathbf{H}^T\mathbf{W}^T\mathbf{V}) - \operatorname{Tr}(\mathbf{V}^T\mathbf{WH})\right\} \\ &= \frac{1}{2}\left\{\operatorname{Tr}(\mathbf{V}^T\mathbf{V}) - 2\operatorname{Tr}(\mathbf{V}^T\mathbf{WH}) + \operatorname{Tr}(\mathbf{H}^T\mathbf{W}^T\mathbf{WH})\right\},\end{aligned}$$

e assim, calcular seu gradiente em relação a $\mathbf{W}$ torna-se mais fácil:

$$\nabla_{\mathbf{W}} f = \mathbf{WHH}^T - \mathbf{VH}^T. \tag{3.4}$$

Analogamente, em relação a $\mathbf{H}$

$$\nabla_{\mathbf{H}} f = \mathbf{W}^T\mathbf{WH} - \mathbf{W}^T\mathbf{V}, \tag{3.5}$$

e substituindo as Equações (3.4) e (3.5) na Equação (3.3), as respectivas regras de atualização são dadas por

$$\begin{aligned}\mathbf{W} &\leftarrow \mathbf{W} - \boldsymbol{\eta}_{\mathbf{W}}\left(\mathbf{WHH}^T - \mathbf{VH}^T\right), \\ \mathbf{H} &\leftarrow \mathbf{H} - \boldsymbol{\eta}_{\mathbf{H}}\left(\mathbf{W}^T\mathbf{WH} - \mathbf{W}^T\mathbf{V}\right).\end{aligned} \tag{3.6}$$

A ideia proposta por Lee e Seung consiste em escolher um passo $\boldsymbol{\eta}_{\boldsymbol{\chi}}$ para que a atualização seja sempre multiplicativa, resultando em fatores positivos caso as matrizes sejam não-negativas. Por isso, ao adotar[3] $\boldsymbol{\eta}_{\mathbf{W}} = \mathbf{W} \oslash \left(\mathbf{WHH}^T\right)$ e $\boldsymbol{\eta}_{\mathbf{H}} =$

[2] $\boldsymbol{\chi}$ é utilizado por conveniência para não ter que reescrever a mesma equação para $\mathbf{W}$ e $\mathbf{H}$.
[3] O símbolo '$\oslash$' representa a divisão ponto-a-ponto: $[\mathbf{A} \oslash \mathbf{B}]_{i,j} = [\mathbf{A}]_{i,j}/[\mathbf{B}]_{i,j}$, $\forall i,j$.

$\mathbf{H} \oslash \left(\mathbf{W}^T\mathbf{W}\mathbf{H}\right)$, têm-se[4]

$$\mathbf{W} \leftarrow \mathbf{W} \odot \left[\mathbf{V}\mathbf{H}^T \oslash \left(\mathbf{W}\mathbf{H}\mathbf{H}^T\right)\right] \tag{3.7}$$

$$\mathbf{H} \leftarrow \mathbf{H} \odot \left[\mathbf{W}^T\mathbf{V} \oslash \left(\mathbf{W}^T\mathbf{W}\mathbf{H}\right)\right], \tag{3.8}$$

que são as MUR propostas por aqueles autores para a norma de Frobenius. Realizando o mesmo procedimento, mas desta vez adotando a divergência de Kullback-Leibler

$$f(\mathbf{W},\mathbf{H}) = \sum_{i,j}\left([\mathbf{V}]_{i,j}\log\frac{[\mathbf{V}]_{i,j}}{[\hat{\mathbf{V}}]_{i,j}} - [\mathbf{V}]_{i,j} + [\hat{\mathbf{V}}]_{i,j}\right), \tag{3.9}$$

podemos reescrever como[5]

$$\begin{aligned} f(\mathbf{W},\mathbf{H}) &= \mathrm{Tr}\left[\mathbf{V}^T\log\mathbf{V} - \mathbf{V}^T\log\hat{\mathbf{V}} - \mathbf{1}^{N\times M}\mathbf{V} + \mathbf{1}^{N\times M}\hat{\mathbf{V}}\right] \\ &= \mathrm{Tr}\left(\mathbf{V}^T\log\mathbf{V}\right) - \mathrm{Tr}(\mathbf{V}^T\log\hat{\mathbf{V}}) + \\ &\quad \mathrm{Tr}\left(\mathbf{1}^{N\times M}\mathbf{V}\right) + \mathrm{Tr}(\mathbf{1}^{N\times M}\hat{\mathbf{V}}), \end{aligned} \tag{3.10}$$

sendo a operação $[\log(\mathbf{A})]_{i,j} = \log([\mathbf{A}]_{i,j})$ para todo i,j. Calculando o gradiente em relação a $\mathbf{W}$

$$\nabla_{\mathbf{W}} f = -(\mathbf{V} \oslash \hat{\mathbf{V}})\mathbf{H}^T + \mathbf{1}^{M\times N}\mathbf{H}^T,$$

e em relação a $\mathbf{H}$

$$\nabla_{\mathbf{H}} f = -\mathbf{W}^T(\mathbf{V} \oslash \hat{\mathbf{V}}) + \mathbf{W}^T\mathbf{1}^{M\times N},$$

e substituindo na Equação (3.3), adotando $\boldsymbol{\eta}_{\mathbf{W}} = \mathbf{W} \oslash \left(\mathbf{1}^{M\times N}\mathbf{H}^T\right)$ e $\boldsymbol{\eta}_{\mathbf{H}} = \mathbf{H} \oslash \left(\mathbf{W}^T\mathbf{1}^{M\times N}\right)$, finalmente têm-se

$$\mathbf{W} \leftarrow \mathbf{W} \odot \left\{\left[(\mathbf{V} \oslash \hat{\mathbf{V}})\mathbf{H}^T\right] \oslash \left(\mathbf{1}^{M\times N}\mathbf{H}^T\right)\right\} \tag{3.11}$$

$$\mathbf{H} \leftarrow \mathbf{H} \odot \left\{\left[\mathbf{W}^T(\mathbf{V} \oslash \hat{\mathbf{V}})\right] \oslash \left(\ \mathbf{W}^T\mathbf{1}^{M\times N}\right)\right\}. \tag{3.12}$$

Generalizando, as regras de atualizações multiplicativas são construídas da seguinte forma:

$$\boldsymbol{\chi} \leftarrow \boldsymbol{\chi} \odot \frac{[\nabla_{\boldsymbol{\chi}} f]_-}{[\nabla_{\boldsymbol{\chi}} f]_+}, \tag{3.13}$$

sendo $[\nabla_{\boldsymbol{\chi}} f]_-$ e $[\nabla_{\boldsymbol{\chi}} f]_+$ as partes negativa e positiva do gradiente $\nabla_{\boldsymbol{\chi}} f$, respectivamente.

Pode-se observar que em nenhum momento as restrições são tratadas diretamente; no entanto, ao se adotar $\mathbf{W}^{(0)}, \mathbf{H}^{(0)}, \mathbf{V}$ não-negativos, as atualizações mul-

[4] O símbolo '$\odot$' representa a multiplicação ponto-a-ponto: $[\mathbf{A} \odot \mathbf{B}] = [\mathbf{A}]_{i,j}[\mathbf{B}]_{i,j}$, $\forall i,j$.

[5] O símbolo '$\mathbf{1}^{N\times M}$' representa uma matriz com N linhas e M colunas com todos os valores iguais a um. Isto é, $[\mathbf{1}^{N\times M}]_{i,j} = 1$, $\forall i,j$.

tiplicativas sempre garantirão a não-negatividade de $\mathbf{W}^{(t)}, \mathbf{H}^{(t)}$. Além disso, Lee e Seung demonstraram que a função é não-crescente a cada atualização, *i.e.*,

$$f(\mathbf{W}^{(t+1)}, \mathbf{H}^{(t)}) \leq f(\mathbf{W}^{(t)}\mathbf{H}^{(t)}) \tag{3.14}$$

$$f(\mathbf{W}^{(t+1)}\mathbf{H}^{(t+1)}) \leq f(\mathbf{W}^{(t+1)}\mathbf{H}^{(t)}). \tag{3.15}$$

O algoritmo proposto por Lee e Seung é, sem dúvida, o mais utilizado nos mais diversos campos de pesquisa com NMF, tendo passado por diversas ramificações para permitir melhor adaptação às restrições do sistema de estudo.

Porém, esse algoritmo possui algumas desvantagens. Gonzales e Zhang [51] mostraram que a equação acima não implica necessariamente as condições da KKT (Apêndice B). Além disso é reportado por alguns autores que o método não possui rápida convergência e até mesmo falha em convergir [44, 51]. Outra desvantagem é utilizar uma regra fixa no passo para ter as atualizações multiplicativas: uma vez que um elemento é zerado, *i.e.*, $[\boldsymbol{\chi}]_{i,j} = 0$ para algum par i,j, ele continuará como zero, fazendo com que o algoritmo continue sempre em direção a um mínimo que pode ser ruim.

3.2.2 Gradiente descendente

Nesta subseção será descrita a classe de algoritmos que são baseados nos métodos de gradiente descendente, caracterizados por buscarem o mínimo da função-custo indo no sentido oposto ao do seu gradiente (que aponta na direção de maior variação). Como mencionado, o algoritmo de atualização multiplicativa é um método de gradiente descendente com uma regra fixa para o cálculo do passo. De modo geral, o método é definido por

$$\boldsymbol{\chi}^{(t+1)} = \boldsymbol{\chi}^{(t)} - \eta_{\boldsymbol{\chi}^{(t)}} \boldsymbol{d}_{\boldsymbol{\chi}^{(t)}}, \tag{3.16}$$

onde $\mathbf{d}_{\boldsymbol{\chi}^{(t)}}$ é a direção e $\eta_{\boldsymbol{\chi}^{(t)}}$ é o tamanho do passo a ser dado na t-ésima iteração, que variam de acordo com o algoritmo. Pode-se notar que, na forma que foi definida, essa atualização não impede a matriz de adquirir valores negativos, fato que será discutido logo adiante.

3.2.2.1 Gradiente descendente projetado

O método mais simples de lidar com a não-negatividade é utilizar a projeção dos resultados, resultando no que é chamado de método de gradiente descendente projetado (PG, do inglês *Projected Gradient*). As fórmulas de atualização são dadas

por

$$\begin{aligned} \mathbf{W}^{(t+1)} &= \left[\mathbf{W}^{(t)} - \eta_{\mathbf{W}^{(t)}} \nabla_{\mathbf{W}^{(t)}} f\right]_+, \\ \mathbf{H}^{(t+1)} &= \left[\mathbf{H}^{(t)} - \eta_{\mathbf{H}^{(t)}} \nabla_{\mathbf{H}^{(t)}} f\right]_+, \end{aligned} \tag{3.17}$$

onde $[\cdot]_+$ denota a projeção de "$\cdot$" no conjunto viável $\{[\cdot]_{i,j} \in \mathbb{R} \ : \ [\cdot]_{i,j} \geq 0\}$, $\nabla_{\mathbf{W}^{(t)}} f$ e $\nabla_{\mathbf{H}^{(t)}} f$ são os gradientes da função objetivo em relação a $\mathbf{W}^{(t)}$ e a $\mathbf{H}^{(t)}$, respectivamente, e $\eta_{\mathbf{W}^{(t)}}$ e $\eta_{\mathbf{H}^{(t)}}$ são os tamanhos dos passos a serem dados na t-ésima iteração.

A projeção no conjunto viável pode ser realizada de diversas formas. No entanto, por motivos práticos, as entradas negativas de "$\cdot$" são alteradas para um valor $\epsilon \in \mathbb{R}_+$, de acordo com

$$[\cdot]_+ = \max\{\epsilon, \cdot\}, \tag{3.18}$$

sendo $\max\{\epsilon,\mathbf{Y}\}$ realizado ponto-a-ponto.

Existem diversos métodos na literatura que definem a escolha do passo no Problema (3.17). Lin [44] propôs dois métodos diferentes para calcular o passo a ser dado — regra de Armijo sobre o arco projetado pelo algoritmo de Bertsekas e regra de Armijo modificada.

No primeiro caso, a cada iteração t, o passo a ser dado é

$$\eta_{\mathbf{H}^{(t)}} = \mu^{\kappa^{(t)}}, \tag{3.19}$$

onde $\kappa^{(t)}$ é o primeiro inteiro não-negativo tal que

$$f(\mathbf{W}^{(t)}, \mathbf{H}^{(t+1)}) - f(\mathbf{W}^{(t)}, \mathbf{H}^{(t)}) \leq \sigma \operatorname{Tr}\left\{(\nabla_{\mathbf{H}^{(t)}} f)^T \left[\mathbf{H}^{(t+1)} - \mathbf{H}^{(t)}\right]\right\}, \tag{3.20}$$

com $\mu \in (0,1)$ e $\sigma \in (0,1)$. A taxa de aprendizado $\eta_{\mathbf{W}}$ é calculada de maneira similar.

No entanto, realizar a busca para encontrar o passo a ser dado é uma tarefa bastante exaustiva, consumindo tempo demasiado de execução do algoritmo; portanto, é vantajoso encontrar $\eta_{\boldsymbol{\chi}^{(t)}}$ no menor número possível de passos. Nesse caso, considera-se que $\eta_{\boldsymbol{\chi}^{(t-1)}}$ e $\eta_{\boldsymbol{\chi}^{(t)}}$ sejam parecidos. Por isso, ao realizar um sistema com memória onde $\eta_{\boldsymbol{\chi}^{(t-1)}}$ é a estimativa inicial do passo a ser dado na iteração seguinte, o tempo gasto para encontrar a nova taxa de aprendizado é reduzido — processo denominado pelo autor de regra de Armijo modificada. Em [44], foram utilizados $\sigma = 0{,}01$ e $\mu = 0{,}1$.

3.2.2.2 Quase-Newton

Até agora, os métodos têm se resumido a encontrar pontos estacionários da função-objetivo com base em direções obtidas por aproximações de primeira ordem, utilizando seu gradiente. Uma subsequente evolução para isso é realizar uma aproximação mais fiel utilizando a informação de segunda ordem da função-custo, originando as atualizações do método de Newton

$$\boldsymbol{\chi}^{(t+1)} = \boldsymbol{\chi}^{(t)} - (\nabla^2_{\boldsymbol{\chi}^{(t)}} f)^{-1} \nabla_{\chi^{(t)}} f, \tag{3.21}$$

onde $\nabla^2_{\boldsymbol{\chi}} f$ é a segunda derivada da função-custo calculada em relação a matriz $\boldsymbol{\chi}$. No entanto, calcular diretamente a inversa da Hessiana pode ser um problema inviável, ou por ser esta desconhecida ou por ser mal condicionada ou por ser muito grande. A classe de métodos com aproximação de segunda ordem que resolvem o problema de otimização sem lidar com a segunda derivada exata são denominados de métodos quase-Newton [50].

O princípio dos métodos quase-Newton é calcular a direção de busca utilizando uma matriz que assume de forma aproximada o papel da inversa da Hessiana.

No contexto da NMF, as regras de atualização para o método de Newton seriam dadas por

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta_{\mathbf{W}^{(t)}} (\nabla^2_{\mathbf{W}^{(t)}} f)^{-1} \nabla_{\mathbf{W}^{(t)}} f, \tag{3.22}$$

$$\mathbf{H}^{(t+1)} = \mathbf{H}^{(t)} - \eta_{\mathbf{H}^{(t)}} (\nabla^2_{\mathbf{H}^{(t)}} f)^{-1} \nabla_{\mathbf{H}^{(t)}} f, \tag{3.23}$$

onde $\eta_{\boldsymbol{\chi}}$ é um escalar real não-negativo que define o tamanho do passo a ser dado. Então, o problema consiste em encontrar $(\nabla^2_{\boldsymbol{\chi}} f)^{-1}$. Como muitas vezes o problema é esparso (*e.g.*, representações de imagens ou espectograma de áudio), para se encontrar a inversa da Hessiana, alguma regularização é necessária, originando assim um método quase-Newton[6]. Zdunek e Cichocki [3] utilizam a regularização de Levenberg-Marquardt com um parâmetro de regularização λ e, além disso, reduzem o tempo computacional utilizando uma fatoração QR [52] para o cálculo da inversa, chamando esse algoritmo de PQN (do inglês, *Practical Quasi-Newton*). Logo,

$$\mathbf{QR} = \nabla^2_{\boldsymbol{\chi}} f + \lambda \mathbf{I} \tag{3.24}$$

$$\mathbf{R}^{-1}\mathbf{Q}^T = (\nabla^2_{\boldsymbol{\chi}} f + \lambda \mathbf{I})^{-1} \tag{3.25}$$

$$\mathbf{R}^{-1}\mathbf{Q}^T \nabla_{\boldsymbol{\chi}} f = (\nabla^2_{\boldsymbol{\chi}} f + \lambda \mathbf{I})^{-1} \nabla_{\boldsymbol{\chi}} f \tag{3.26}$$

[6]Note que o conceito de quase-Newton difere do encontrado em problemas de otimização, onde a Hessiana é estimada em vez de calculada [50].

e assim, a forma final do método de quase-Newton é dada por

$$\boldsymbol{\chi}^{(t+1)} = [\boldsymbol{\chi}^{(t)} - \eta_{\boldsymbol{\chi}^{(t)}} \mathbf{R}_{\boldsymbol{\chi}^{(t)}} \setminus \mathbf{W}_{\boldsymbol{\chi}^{(t)}}]_+ \tag{3.27}$$

$$\mathbf{W}_{\boldsymbol{\chi}^{(t)}} = \mathbf{Q}^T_{\boldsymbol{\chi}^{(t)}} \nabla_{\boldsymbol{\chi}^{(t)}} f \tag{3.28}$$

$$\mathbf{Q}_{\boldsymbol{\chi}^{(t)}} \mathbf{R}_{\boldsymbol{\chi}^{(t)}} = \nabla^2_{\boldsymbol{\chi}^{(t)}} f + \lambda \mathbf{I}, \tag{3.29}$$

sendo $\mathbf{I}$ a matriz identidade e o operador '\', a eliminação gaussiana [52]. Em [3] adotaram $\eta_{\boldsymbol{\chi}} = 0{,}9$. Como essa otimização quase-Newton não garante a não-negatividade, é usada a projeção para o conjunto viável do mesmo jeito que é feito nos métodos anteriores.

Devido a esse passo de projeção, a NMF utilizando o método de quase-Newton projetado pode levar a um aumento da função-custo, fato apontado e corrigido por [2].

3.2.3 ALS

Outra classe de algoritmos para a NMF é a de mínimos quadrados alternados (ALS, do inglês *alternating least squares*). Nesse tipo de algoritmo, a cada iteração é resolvido um problema de mínimos quadrados de forma alternada, como mostrado no Pseudocódigo 3.

Esta classe explora o fato de que o problema original não é convexo, mas é convexo para $\mathbf{W}$ e para $\mathbf{H}$ isoladamente. Portanto, dada uma matriz fixa, a outra pode ser encontrada por um simples passo de mínimos quadrados sem restrição,

$$\mathbf{V} = \mathbf{WH} \tag{3.30}$$

$$\mathbf{W}^T\mathbf{V} = \mathbf{W}^T\mathbf{WH} \tag{3.31}$$

$$\mathbf{H} = (\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\mathbf{V} = \mathbf{W}^\dagger\mathbf{V} \tag{3.32}$$

e alternadamente,

$$\mathbf{V}^T = (\mathbf{WH})^T = \mathbf{H}^T\mathbf{W}^T \tag{3.33}$$

$$\mathbf{HH}^T\mathbf{W}^T = \mathbf{HV}^T \tag{3.34}$$

$$\mathbf{W}^T = (\mathbf{HH}^T)^{-1}\mathbf{HV}^T = \mathbf{H}^{T\dagger}\mathbf{V}^T, \tag{3.35}$$

sendo $(\cdot)^\dagger$ a pseudoinversa de Moore-Penrose. O método para garantir a não-negatividade é simplório, pois consiste em projetar todos os elementos negativos provenientes dos mínimos quadrados em zero. Realizar o processo dessa forma possui algumas vantagens, pois além de adicionar esparsidade (projetando valores ne-

gativos em zero), o algoritmo não é fixo em relação a um elemento zero como na classe de algoritmos MUR, evitando, assim, mínimos ruins.

Pseudocódigo 3 ALS-NMF

1: **repita**
2: (LS) Resolva $\mathbf{H}$ na equação $\mathbf{W}^T\mathbf{W}\mathbf{H} = \mathbf{W}^T\mathbf{V}$
3: (NONEG) Projete todos os elementos negativos de $\mathbf{H}$ em 0
4: (LS) Resolva $\mathbf{W}$ na equação $\mathbf{H}\mathbf{H}^T\mathbf{W}^T = \mathbf{H}\mathbf{V}^T$
5: (NONEG) Projete todos os elementos negativos de $\mathbf{W}$ em 0
6: **até** critério de parada ser satisfeito

3.2.4 ANLS

O método imposto pelo ALS é muito simples e não tem garantias de convergência, pois não é fácil realizar sua análise devido à projeção. No entanto, é de consenso acadêmico que esse método não é poderoso o suficiente para a utilização da NMF em suas mais diversas aplicações.

Por isso, gerou-se uma generalização da classe ALS, dada no Pseudocódigo 4, denominada de ANLS (do inglês *alternating non-negative least squares*), onde são realizadas minimizações com restrições do tipo NNLS (do inglês *non-negative least squares*) [13, 35]. De certo modo, os algoritmos propostos anteriormente também podem ser vistos como uma subclasse da ANLS.

Pseudocódigo 4 ANLS-NMF

1: **repita**
2: (NNLS) Encontre $\mathbf{H}^{(t+1)}$ tal que: $\min\limits_{\mathbf{H}^{(t+1)}\geq 0} \left\| \mathbf{H}^{(t)^T}\mathbf{W}^{(t)^T} - \mathbf{V}^T \right\|_{\mathrm{F}}^2$
3: (NNLS) Encontre $\mathbf{W}^{(t+1)}$ tal que: $\min\limits_{\mathbf{W}^{(t+1)}\geq 0} \left\| \mathbf{W}^{(t)}\mathbf{H}^{(t+1)} - \mathbf{V} \right\|_{\mathrm{F}}^2$
4: **até** critério de parada ser satisfeito

3.2.4.1 Mínimos quadrados não-negativos hierárquicos

Cichocki *et al.* [14] foram dos primeiros a propor utilizar funções-custo de maneira local cuja minimização é realizada hierarquicamente (daí o nome do método) através de simples atualizações ALS.

Nesse caso o problema pode ser fatorado como um conjunto de mínimos quadrados do tipo

$$\min_{\mathbf{a}_i\geq 0,\mathbf{x}_i\geq 0} \frac{1}{2} \left\| \tilde{\mathbf{Y}}_i - \mathbf{a}_i{\mathbf{x}_i}^T \right\|_{\mathrm{F}}^2, \tag{3.36}$$

para $i \in \{1, \ldots, R\}$ (ver Seção 2.3), onde

$$\tilde{\mathbf{Y}}_i = \mathbf{Y} - \sum_{r \neq i} \mathbf{a}_r \mathbf{x}_r^T, \tag{3.37}$$

sendo $\mathbf{a}_i \in \mathbb{R}^{M \times 1}$ as colunas da matriz $\mathbf{A}$ e $\mathbf{x}_i^T \in \mathbb{R}^{1 \times N}$ as linhas de $\mathbf{X}$. A construção do conjunto dessas funções-custo segue do princípio que $\mathbf{Y} = \sum_{r=1}^{R} \mathbf{a}_r {\mathbf{x}_r}^T$, ou seja, é uma soma de matrizes de posto um. Com isso, o algoritmo a cada iteração reduz o posto da matriz a ser fatorada, realizando a minimização hierarquicamente. As derivadas em relação aos vetores $\mathbf{a}_i, \mathbf{x}_i$ são dadas por

$$\frac{\partial f}{\partial \mathbf{a}_i} = \mathbf{a}_i \mathbf{x}_i^T \mathbf{x}_i - \tilde{\mathbf{Y}}_i \mathbf{x}_i, \tag{3.38}$$

$$\frac{\partial f}{\partial {\mathbf{x}_i}^T} = \mathbf{a}_i^T \mathbf{a}_i \mathbf{x}_i^T - \mathbf{a}_i^T \tilde{\mathbf{Y}}_i. \tag{3.39}$$

Igualando-se o gradiente a zero e assumindo que as restrições serão satisfeitas forçando a não-negatividade projetando os valores negativos para quase zero, as regras de atualização hierárquicas são dadas por:

$$\mathbf{x}_i^T \leftarrow \left[\frac{1}{\mathbf{a}_i^T \mathbf{a}_i} \mathbf{a}_i^T \tilde{\mathbf{Y}}_i \right]_+ \tag{3.40}$$

$$\mathbf{a}_i \leftarrow \left[\frac{1}{\mathbf{x}_i^T \mathbf{x}_i} \hat{\mathbf{Y}}_i^T \mathbf{x} \right]_+, \quad i \in \{1, \ldots, R\}, \tag{3.41}$$

onde $[\cdot]_+ = \max\{\epsilon, \cdot\}$, e ϵ é uma constante para evitar instabilidade numérica. Essa simples atualização possui uma desvantagem: pode ser extremamente lenta [14]. No entanto, as colunas de $\mathbf{A}$ podem ser estimadas simultaneamente em vez de uma por uma; e além disso, adotando-se a normalização das colunas de $\mathbf{a}_i$ pela norma ℓ_2, obtém-se um algoritmo mais rápido, onde as linhas de $\mathbf{X}$ são atualizadas localmente e as colunas de $\mathbf{A}$ são atualizadas globalmente [14]:

$$\mathbf{x}_i^T \leftarrow \left[\mathbf{a}_i^T \tilde{\mathbf{Y}}_i \right]_+, \tag{3.42}$$

$$\mathbf{A} \leftarrow \left[\mathbf{Y}\mathbf{X}^T (\mathbf{X}\mathbf{X}^T)^{-1} \right]_+. \tag{3.43}$$

Ademais, pode-se aplicar o método quase-Newton para estimar a matriz $\mathbf{A}$, utilizando informações de segunda ordem. A Hessiana é dada por[7] $\nabla^2_{\mathbf{A}} f = \mathbf{I} \otimes \mathbf{X}\mathbf{X}^T \in \mathbb{R}^{MR \times MR}$ e tem uma estrutura blocodiagonal. Como a Hessiana pode ser mal condicionada, especialmente se $\mathbf{X}$ for esparso, pode-se aplicar o método de Levenberg-Marquardt para controlar o condicionamento da matriz. Com isso, teremos o se-

[7] O símbolo '$\otimes$' representa o produto de Kronecker: $[\mathbf{A} \otimes \mathbf{B}]_{i,j} = [\mathbf{A}]_{i,j}\mathbf{B}$, $\forall i,j$.

guinte algoritmo:

$$\mathbf{x}_i^T \leftarrow \left[\mathbf{a}_i^T \tilde{\mathbf{Y}}_i\right]_+ \tag{3.44}$$

$$\lambda \leftarrow \lambda_0 \mathrm{e}^{-\tau \mathrm{t}} \tag{3.45}$$

$$\mathbf{A} \leftarrow \left[\mathbf{A} - (\mathbf{AX} - \mathbf{Y})\mathbf{X}^T(\mathbf{XX}^T + \lambda \mathbf{I})^{-1}\right]_+, \tag{3.46}$$

onde t é a iteração. O uso de um λ exponencial vem de uma analogia com o *simulated annealing* [53], e evita que o algoritmo fique preso em mínimos locais. O autor em [14] utilizou, heuristicamente, os parâmetros $\lambda_0 = 100$ e $\tau = 0{,}02$ e em [54], os mesmos autores apresentaram uma versão ainda mais eficiente e rápida do algoritmo.

3.2.4.2 Conjunto ativo

Os outros algoritmos mostrados até aqui não lidam especificamente com o problema da restrição de não-negatividade, e o método de conjunto ativo (do inglês *active set*) visa a contornar esse problema. A NMF com função-custo dada pela norma de Frobenius pode ser reformulada como a minimização de múltiplos problemas de mínimos quadrados

$$\min_{\mathbf{X}\geq 0} \frac{1}{2} \|\mathbf{AX} - \mathbf{Y}\|_{\mathrm{F}}^2 \rightarrow \min_{\mathbf{x}_1 \geq 0} \frac{1}{2} \|\mathbf{Ax}_1 - \mathbf{y}_1\|_2^2, \cdots, \min_{\mathbf{x}_n \geq 0} \frac{1}{2} \|\mathbf{Ax}_N - \mathbf{y}_N\|_2^2, \tag{3.47}$$

onde $\mathbf{X} = [\mathbf{x}_1 \ \cdots \ \mathbf{x}_N] \in \mathbb{R}^{R\times N}$ e $\mathbf{Y} = [\mathbf{y}_1 \ \cdots \ \mathbf{y}_N] \in \mathbb{R}^{M\times N}$. A função-custo é estritamente convexa, e portanto garante solução única se e somente se $\mathbf{A}$ tiver posto completo por colunas. No contexto da NMF, assumiremos que as matrizes $\mathbf{H}^T$ e $\mathbf{W}$ possuem posto completo por colunas, já que formam a matriz $\mathbf{A}$ e $R \ll \min(M,N)$. Então, cada NNLS

$$\min_{\mathbf{x}\geq 0} \frac{1}{2} \|\mathbf{Ax} - \mathbf{y}\|_2^2$$

pode ser resolvido utilizando-se o método de conjunto ativo [13]. O método de conjunto ativo faz uma busca do conjunto ótimo de restrições ativas e inativas. No caso, se $x_i = 0$, $i \in \{1, \ldots, R\}$, a restrição é considerada ativa; no caso contrário, $x_i > 0$, é considerada inativa. Esse método é motivado pelo fato de que, dadas as

condições da KKT (Apêndice B),

$$\mathbf{A}^T(\mathbf{Ax}-\mathbf{y})-\boldsymbol{\lambda}=0 \tag{3.48}$$

$$\mathbf{x}\geq 0 \tag{3.49}$$

$$\boldsymbol{\lambda}\geq 0 \tag{3.50}$$

$$\lambda_i x_i = 0, \quad i=\{1,\ldots,R\}, \tag{3.51}$$

e caso esses conjuntos sejam conhecidos a priori, a solução é direta, já que para o conjunto inativo $\lambda_i = 0$; então,

$$\mathbf{A}^T\mathbf{Ax}=\mathbf{A}^T\mathbf{y}. \tag{3.52}$$

Sejam $\mathbf{x}^{(t)}$ uma solução viável obtida na t-ésima iteração e $\mathcal{A}^{(t)}$ o conjunto que contenha os índices das restrições ativas, denominado de conjunto ativo. Na próxima iteração teremos

$$\mathbf{x}^{(t+1)}=\mathbf{x}^{(t)}+\eta^{(t)}\mathbf{d}^{(t)}, \tag{3.53}$$

onde o passo $\eta^{(t)}$ é não-negativo. As restrições que estão ativas em $\mathbf{x}^{(t)}$ continuarão ativas se

$$x_j^{(t+1)}=0, \quad j\in\mathcal{A}^{(t)}, \tag{3.54}$$

o que leva a

$$d_j^{(t)}=0, \quad j\in\mathcal{A}^{(t)}. \tag{3.55}$$

Portanto, a direção pode ser encontrada resolvendo-se o seguinte problema:

$$\min\ f(\mathbf{x}+\mathbf{d}), \quad \text{sujeito a } d_j=0,\ j\in\mathcal{A}, \tag{3.56}$$

onde a restrição é escolhida para que os elementos na fronteira não possam ser alterados. A Equação (3.56) pode ser reescrita como

$$\min\ \mathbf{d}^T\mathbf{A}^T\mathbf{A}\mathbf{d}^T+\mathbf{p}^T\mathbf{d}, \quad \text{sujeito a } d_j=0,\ j\in\mathcal{A}, \tag{3.57}$$

onde $\mathbf{p}=-\mathbf{A}^T\mathbf{y}+\mathbf{A}^T\mathbf{Ax}$ é o gradiente da função-custo original. Eliminando o conjunto ativo $\mathcal{A}$ do problema, teremos

$$\min_{\mathbf{d}_\mathcal{W}} \mathbf{d}_\mathcal{W}{}^T\mathbf{A}_\mathcal{W}{}^T\mathbf{A}_\mathcal{W}d_\mathcal{W}{}^T+\mathbf{p}_\mathcal{W}{}^T\mathbf{d}_\mathcal{W}, \tag{3.58}$$

sendo $\mathbf{A}_{\mathcal{W}}$ o subconjunto das colunas de $\mathbf{A}$ que representa a variável no conjunto de restrições inativas $\mathcal{W}$[8], bastando, assim, a simples resolução de um problema de mínimos quadrados sem restrições.

Após se determinar a direção a ser dada, o passo η é escolhido de modo a minimizar a função-objetivo no ponto $\mathbf{x}^{(t)} + \eta\mathbf{d}^{(t)}$ sem violar nenhuma restrição. Podemos, matematicamente, escrever o problema como

$$\min_{\alpha} \phi(\mathbf{x} + \eta\mathbf{d}), \text{ sujeito a } \mathbf{x} + \eta\mathbf{d} \geq 0,\ \eta \geq 0, \tag{3.59}$$

onde α é um escalar e $\mathbf{x}$, $\mathbf{d}$ são dados. A derivada da função-custo é dada por

$$\frac{\mathrm{d}}{\mathrm{d}\eta}\phi(\mathbf{x} + \eta\mathbf{d}) = \mathbf{p}^T\mathbf{d} + \eta\mathbf{d}^T\mathbf{A}^T\mathbf{A}\mathbf{d}, \tag{3.60}$$

$$\frac{\mathrm{d}^2}{\mathrm{d}\eta^2}\phi(\mathbf{x} + \eta\mathbf{d}) = \mathbf{d}^T\mathbf{A}^T\mathbf{A}\mathbf{d}. \tag{3.61}$$

Nesse caso, como $\mathbf{d}$ é uma direção decrescente, *i.e.*, $\mathbf{p}^T\mathbf{d} < 0$, temos um ponto de mínimo único para o problema sem restrição dado por $\eta_{\max} = -\mathbf{p}^T\mathbf{d}/\mathbf{d}^T\mathbf{A}^T\mathbf{A}\mathbf{d}$. Portanto, temos que a função é decrescente no intervalo $[0, \eta_{\max})$.

Quando $\mathbf{d} = (\mathbf{d}_{\mathcal{W}}, \mathbf{d}_{\mathcal{A}})$ com $\mathbf{d}_{\mathcal{W}}$ e $\mathbf{d}_{\mathcal{A}}$ dado pela solução da Equação (3.58), temos que $\mathbf{d}^T\mathbf{A}^T\mathbf{A}\mathbf{d}^T = {\mathbf{d}_{\mathcal{W}}}^T{\mathbf{A}_{\mathcal{W}}}^T\mathbf{A}_{\mathcal{W}}{\mathbf{d}_{\mathcal{W}}}^T$ e $\mathbf{p}^T\mathbf{d} = {\mathbf{p}_{\mathcal{W}}}^T\mathbf{d}_{\mathcal{W}}$. Se $\mathbf{d}_{\mathcal{W}}$ satisfaz as condições da KKT na Equação (3.58), ${\mathbf{d}_{\mathcal{W}}}{\mathbf{A}_{\mathcal{W}}}^T\mathbf{A}_{\mathcal{W}}\mathbf{d}_{\mathcal{W}} = -{\mathbf{p}_{\mathcal{W}}}^T\mathbf{d}_{\mathcal{W}}$. Então

$$\eta_{\max} = 1, \tag{3.62}$$

o que nos diz que a solução ótima do passo para uma otimização sem restrição é o passo unitário. No entanto, $\mathbf{x} + \eta\mathbf{d} \geq 0$ pode impedir que η^*, a solução ótima da Equação (3.59), seja unitária. Como a função-custo é decrescente no intervalo (0,1), a minimização com restrição será o máximo passo possível

$$\eta_{\text{feas}} = \max_{j \in \mathcal{W}: d_j < 0} \left(\frac{-x_j}{d_j} \right). \tag{3.63}$$

Se η_{feas} não for o passo unitário, ao menos uma variável x_j irá para a fronteira. Nesse caso, o algoritmo deve atualizar $\mathcal{A}$ e $\mathcal{W}$, movendo j de $\mathcal{W}$ para $\mathcal{A}$ e assim, uma restrição é adicionada ao conjunto ativo. O Pseudocódigo 5, adaptado de [13], demonstra como é feita a implementação do método de conjunto ativo.

Inicialmente, o algoritmo postula que todas as variáveis pertencem ao conjunto ativo e a cada iteração determina qual o índice desse conjunto que minimiza o gradiente da função-custo. Então, essa variável é movida para o conjunto $\mathcal{W}$, dando sequência ao algoritmo descrito acima. O processo só termina quando o conjunto

[8]Foi usado "$\mathcal{W}$" por representar o conjunto de variáveis de trabalho (do inglês, *working set*).

ativo estiver vazio ou não contiver mais nenhuma variável que minimize a função-custo.

Uma das desvantagens desse método é permitir somente a mudança de um índice por vez para o conjunto ativo. Um meio de contornar esse problema é apresentado em [48], utilizando o método denominado BPP (do inglês, *block principal pivoting*), baseado no conjunto ativo, que permite que múltiplas variáveis transitem entre os conjuntos ativo e inativo a cada iteração, agilizando o processo.

Pseudocódigo 5 AS-NLS

1: $\mathbf{g} \leftarrow \mathbf{0}$
2: $\mathcal{A} \leftarrow \{1,2,\ldots,R\}$
3: $\mathcal{W} \leftarrow \emptyset$
4: $\mathbf{p} \leftarrow \mathbf{A}^T(\mathbf{y} - \mathbf{Ag})$
5: **enquanto** $\mathcal{A} \neq \emptyset$ **e** $\exists\, j \in \mathcal{A} \mid \mathbf{p}_j > 0$ **faça**
6: $\quad i \leftarrow \{i \in \mathcal{A} \mid \max\ \mathbf{p}_i\}$
7: $\quad$ Mova i de $\mathcal{A}$ para $\mathcal{W}$
8: $\quad$ Resolva $\min_{\mathbf{z}} \|\mathbf{A}_{\mathcal{W}}\mathbf{z} - \mathbf{y}_{\mathcal{W}}\|$
9: $\quad$ **enquanto** $z_j \leq 0\ \forall j \in \mathcal{W}$ **faça**
10: $\quad\quad \eta \leftarrow \max_{j \in \mathcal{W}: z_j \leq 0} \{-g_j/(z_j - g_j)\}$
11: $\quad\quad \mathbf{g} \leftarrow \mathbf{g} + \eta(\mathbf{z} - \mathbf{g})$
12: $\quad\quad$ Mova do conjunto $\mathcal{W}$ para $\mathcal{A}$ todos os índices $j \in \mathcal{W}$ para o qual $g_j = 0$
13: $\quad\quad$ Resolva $\min_{\mathbf{z}} \|\mathbf{A}_{\mathcal{W}}\mathbf{z} - \mathbf{y}_{\mathcal{W}}\|$
14: $\quad$ **fim enquanto**
15: $\quad \mathbf{g} \leftarrow \mathbf{z}$
16: $\quad \mathbf{p} \leftarrow \mathbf{A}^T(\mathbf{y} - \mathbf{Ag})$
17: **fim enquanto**

3.2.5 Outros algoritmos

Como já mencionado, existem diversos modos de realizar a fatoração de matrizes não-negativas. São poucos os trabalhos encontrados na literatura que fazem uma boa comparação entre esses diversos algoritmos, como é feito em [48]. A Tabela 3.1 faz uma breve comparação dos algoritmos citados acima e mais alguns encontrados na literatura.

3.3 Critério de parada

Os algoritmos para a NMF não restrigem qual é o critério de parada a ser utilizado e, além disso, não há estudos específicos para decidir qual é o melhor meio de

Nome	Descrição	Prós	Contras
ALS	Algoritmo de Berry *et al.* [55] baseado em mínimos quadrados alternados	Velocidade	Convergência
MUR	Algoritmo multiplicativo de Lee e Seung [7]	Facilidade de implementação	Velocidade de convergência; Mínimos ruins
SNMF	Algoritmo baseado em MUR com esparsidade [1]	Controle de esparsidade facilmente ajustável	Mínimos ruins
PG	ANLS com gradiente projetado de Lin [44]	Melhores mínimos locais	Projeção; Aproximação de primeira ordem
HALS	Algoritmo baseado em mínimos quadrados alternados [14, 54]	Velocidade; Generalização	Projeção
PQNZC	ANLS de Zdunek e Cichocki [3] utilizando o método de quase-Newton	Melhor convergência	Mau funcionamento para matrizes grandes
PQNK	ANLS de Kim [2] utilizando o método quase-Newton projetado	Convergência garantida	Velocidade
AS	ANLS com método de conjuntos ativos de [13]	Robustez; Convergência	Posto completo da matriz $\mathbf{A}$; Uma atualização por vez
BPP	ANLS com método de *block principal pivoting* de [48]	Velocidade; Convergência	Generalização da função-custo

Tabela 3.1: Comparação entre os diversos métodos para a NMF.

definir quando o algoritmo deve parar ou não. Muitos autores preferem definir um número máximo de iterações, ou então o máximo de tempo decorrido (em segundos), ou a junção de ambos os critérios. Entretanto, com o intuito de evitar excesso de contas sem alguma melhoria aparente, pode-se usar como critério de parada a distância relativa entre duas fatorações:

$$f^{(t+1)} - f^{(t)} = f(\mathbf{W}^{(t+1)}, \mathbf{H}^{(t+1)}) - f(\mathbf{W}^{(t)}, \mathbf{H}^{(t)}) \leq \epsilon, \tag{3.64}$$

sendo $\epsilon \in \mathbb{R}_+$ tão pequeno quanto se queira. Ou então, pode-se expressar a porcentagem desta diferença

$$\frac{|f^{(t+1)} - f^{(t)}|}{f^{(t+1)}} \leq \epsilon. \tag{3.65}$$

Apesar de os critérios de parada citados acima serem muito utilizados, não garantem que o algoritmo convergiu para um ponto estacionário [44], ou seja, o decréscimo da função-custo pode ser pequeno, mas não necessariamente garante que convergiu para um ponto de mínimo local.

Um critério mais rigoroso foi proposto por LIN [44]. De acordo com as condições da KKT (Apêndice B), $\mathbf{W}, \mathbf{H}$ são pontos estacionários se e somente se

$$\mathbf{W} \geq 0 \qquad \mathbf{H} \geq 0, \tag{3.66}$$

$$\nabla_{\mathbf{W}} f \geq 0 \qquad \nabla_{\mathbf{H}} f \geq 0, \tag{3.67}$$

$$\mathbf{W} \odot \nabla_{\mathbf{W}} f = \mathbf{0} \qquad \mathbf{H} \odot \nabla_{\mathbf{H}} f = \mathbf{0}. \tag{3.68}$$

Definindo-se o gradiente projetado $\nabla^p_{\mathbf{W}} f$

$$[\nabla^p_{\mathbf{W}} f]_{i,j} = \begin{cases} [\nabla_{\mathbf{W}} f]_{i,j}, & \text{caso } [\nabla_{\mathbf{W}} f]_{i,j} < 0 \text{ ou } [\mathbf{W}]_{i,j} > 0, \\ 0, & \text{caso contrário,} \end{cases} \tag{3.69}$$

e $\nabla^p_{\mathbf{H}} f$ de modo similar, as condições da KKT podem ser reescritas como

$$\nabla^p_{\mathbf{W}} f = 0, \quad \nabla^p_{\mathbf{H}} f = 0. \tag{3.70}$$

Usando a raiz quadrada da soma das normas ao quadrado dos gradientes projetados, podemos definir o critério de parada como

$$\frac{\Delta}{\Delta_0} \leq \epsilon, \tag{3.71}$$

sendo $\Delta = \sqrt{\|\nabla^p_{\mathbf{W}} f\|^2_{\mathrm{F}} + \|\nabla^p_{\mathbf{H}} f\|^2_{\mathrm{F}}}$, Δ_0 o valor de Δ para $\mathbf{W}$ e $\mathbf{H}$ iniciais e ϵ a

tolerância desejada.

Apesar de esse critério de parada ser útil em quase todos os casos, ele possui uma desvantagem. Considere uma matriz diagonal $\mathbf{D} \in \mathbb{R}_+^{R\times R}$ de modo que $\mathbf{WH} = \mathbf{WD}^{-1}\mathbf{DH}$, não alterando assim o valor da função-custo, porém alterando a norma do gradiente projetado, já que $(\nabla_{\mathbf{WD}^{-1}} f, \nabla_{\mathbf{DH}} f) = \left(\nabla_{\mathbf{W}} f\mathbf{D}, \mathbf{D}^{-1}\nabla_{\mathbf{H}} f\right)$. Então, se uma NMF tende a colocar grande peso em $\mathbf{W}$ e pesos pequenos em $\mathbf{H}$, o método tende a atender o critério de parada. Um modo de contornar esse problema é citado por Ho [56], efetuando um passo de normalização antes de computar a norma.

3.4 Adicionando restrições

Apesar de muitos algoritmos indiretamente lidarem com outras restrições, como a esparsidade, pode ser interessante em muitas aplicações estabelecer regras específicas para garantir que a representação da NMF contenha algum sentido físico como na clusterização, onde é necessário gerar um dicionário esparso para que haja menos informações redundantes, ou em áudio, em que é interessante manter continuidade temporal das emissões de um instrumento para que soe natural.

As restrições podem ser incluídas como

$$\begin{aligned} &\text{minimizar} && f(\mathbf{W},\mathbf{H}) + \boldsymbol{\alpha}_{\mathbf{W}}^T\mathbf{J}_{\mathbf{W}} + \boldsymbol{\alpha}_{\mathbf{H}}^T\mathbf{J}_{\mathbf{H}} \\ &\text{sujeito a} && \mathbf{W} \geq 0, \\ & && \mathbf{H} \geq 0, \end{aligned} \tag{3.72}$$

sendo $\boldsymbol{\alpha}_{\boldsymbol{\chi}} \in \mathbb{R}^{S\times 1}$ os pesos correspondentes às S restrições (penalizações) $\mathbf{J}_{\boldsymbol{\chi}}$.

Dependendo do método a ser utilizado para realizar a fatoração, é necessário que se tenham as informações do gradiente e da Hessiana da função $J_{\boldsymbol{\chi}}$ para serem incorporadas nos diversos métodos supracitados.

3.4.1 Esparsidade

Uma das restrições mais utilizadas é a de esparsidade. As representações de alguns sinais reais, como imagens ou áudio, tendem a ser esparsas, por isso inúmeros métodos tentam lidar com esse critério.

A minimização da norma[9] ℓ_0 seria a melhor escolha para maximizar a esparsidade do sinal desejado; no entanto, a função não é convexa [50].

Outro problema a ser considerado ao se adotar a norma ℓ_0 está ligado às representações de sinais naturais ditos esparsos. Nesse sentido, o sinal esparso, como

[9]Não é formalmente uma norma, pois não obedece a desigualdade triangular nem o critério de homogeneidade; no entanto, tomamos essa liberdade na monografia, como em tantos textos técnicos.

já descrito, não contém zeros de verdade, mas sim valores bem próximos de zero, tornando a contagem que a norma zero realiza impraticável em cenários realistas.

Para contornar esse problema, utiliza-se uma norma convexa "próxima" ℓ_p, com $p \geq 1$.

3.4.1.1 Norma ℓ_p

De forma generalizada, podemos empregar a norma ℓ_p, $p \geq 1$, com $J_{\mathbf{x}}^{\ell_p} = (\sum_i x_i^p)^{1/p} = \|\mathbf{x}\|_p$, onde $\mathbf{x} = \mathbf{vec}(\mathbf{X})$[10]. Seus gradiente e Hessiana são dados por

$$\begin{aligned} \nabla_{\mathbf{x}} J_{\mathbf{x}}^{\ell_p} &= \frac{\mathbf{x}^{\cdot(p-1)}}{\|\mathbf{x}\|_p^{(p-1)}}, \\ \frac{\nabla_{\mathbf{x}}^2 J_{\mathbf{x}}^{\ell_p}}{(p-1)} &= \mathbf{diag}\left(\nabla_{\mathbf{x}} J_{\mathbf{x}}^{\ell_p} \odot \mathbf{x}^{\cdot -1}\right) - \nabla_{\mathbf{x}} J_{\mathbf{x}}^{\ell_p} \frac{\mathbf{x}^{\cdot(p-1)^T}}{\|\mathbf{x}\|_p}, \end{aligned} \tag{3.73}$$

para qualquer $p \geq 1$, onde $\mathbf{x}^{\cdot p}$ eleva cada componente do vetor ao expoente p. Essa medida de esparsidade é utilizada em diversas áreas, sendo o mais usual considerar-se o valor de p igual a 1.

3.4.1.2 Outras medidas de esparsidade

Há diversas outras medidas de esparsidade que podem ser implementadas, tal como a esparsidade de Hoyer [1]

$$\mathbf{spar}(\mathbf{x}) = \frac{\sqrt{n} - ||\mathbf{x}||_1/||\mathbf{x}||_2}{\sqrt{n} - 1}, \tag{3.74}$$

com $\mathbf{x} \in \mathbb{R}^n$. Pode-se reescrever esta restrição para $\mathbf{X}$,

$$J_{\mathbf{X}}^{\mathrm{H}} = (\varsigma \left\|\mathbf{vec}(\mathbf{X})\right\|_2 - \left\|\mathbf{vec}(\mathbf{X})\right\|_1)^2, \tag{3.75}$$

onde $\varsigma = \sqrt{RN} - (\sqrt{RN} - 1)\gamma$ e a esparsidade desejada é especificada através de $\gamma \in [0,1]$. Esta medida é normalizada, *i.e.*, o vetor mais esparso possível que contém somente um elemento não-zero tem $\mathbf{spar}(\mathbf{x}) = 1$, enquanto um vetor com todos os valores iguais tem o valor de esparsidade zero.

Em [57] também é definida como critério de esparsidade uma função logarítmica $J_{\mathbf{X}} = \|\log(1 + \mathbf{X} \odot \mathbf{X})\|_{\mathrm{F}}^2$ tal que, para uma matriz maximamente esparsa ($\mathbf{X} = \mathbf{0}$), a função se aproximará de zero.

[10] Transformação da matriz $\mathbf{X}$ em um vetor, *i.e.*, $\mathbf{vec}(\mathbf{X}) = [X_{1,1} \ \cdots X_{M,1} \ X_{1,2} \ \cdots \ X_{M,N}]^T$.

3.4.2 Continuidade e suavização

Outra característica desejável para uma representação realística de um sinal natural é a continuidade. Um sinal dito contínuo deve ser diferenciável, e por isso é intuitivo considerar $J^{\mathrm{d}}_{\mathbf{X}} = \|\mathbf{TX}\|^2_{\mathrm{F}}$, onde $\mathbf{T}$ é uma matriz de transformação representando a primeira ou segunda derivada das colunas da matriz $\mathbf{X}$. O gradiente e a Hessiana são dados por

$$\nabla_{\mathbf{X}} J^{\mathrm{d}}_{\mathbf{X}} = \mathbf{T}^T\mathbf{TX}, \tag{3.76}$$

$$\nabla^2_{\mathbf{X}} J^{\mathrm{d}}_{\mathbf{X}} = \mathbf{I} \otimes \mathbf{T}^T\mathbf{T}. \tag{3.77}$$

Outro critério de continuidade que pode ser utilizado é dado por $J_{\mathbf{X}} = -|\mathbf{T} \odot (\mathbf{X}^T\mathbf{X})|$, sendo $\mathbf{T}$ uma matriz de Toeplitz, utilizada para fazer a ponderação das correlações entre cada quadro e os demais quadros como uma exponencial decrescente, para que quadros mais próximos sejam maximamente parecidos entre si. Já em [58], forçam a continuidade temporal definindo $J_{\mathbf{X}} = \frac{1}{N}\left\|(\mathbf{I} - \mathbf{T})\mathbf{X}^T\right\|^2_{\mathrm{F}}$, onde $\mathbf{T}$ é o operador de convolução.

Como se pode observar, há muitas restrições que podem ser utilizadas para melhorar a fatoração, e para cada uma dessas restrições há diversas métricas. As melhores estratégias serão sempre ditadas pela natureza da aplicação.

3.5 Outros tipos de função-custo

Por mais que as funções-custo utilizando a norma de Frobenius ou a divergência de Kullback-Leibler sejam as mais empregadas pelos mais diversos pesquisadores, é de senso comum que, em muitas aplicações, elas não se adaptam bem para representar o que se deseja. Como, por exemplo, em aplicações envolvendo áudio é comum utilizarem a distância de Itakura-Saito ou então a divergência de Kullback-Leibler, que parecem se aproximar melhor de como funciona nossa audição. Muitos artigos propõem medidas alternativas de distorção, como, por exemplo, distâncias de Hellinger, de Pearson e alfa-divergente (sendo estas uma subclasse das divergências de Csiszár [59]) ou a divergência de Bregman [60].

Vale ressaltar que a função-custo f deve ser convexa, ou seja, ter um ponto de mínimo e ser zero se e somente se $\mathbf{V} = \hat{\mathbf{V}}$, porém não necessariamente deve satisfazer os critérios de distância, como a divergência de Kullback-Leibler, que desobedece a desigualdade triangular.

Févotte e Cemgil [61] demonstraram que utilizar como funções-custo distância euclidiana, Kullback-Leibler ou Itakura-Saito equivale a considerar $\mathbf{V} \sim p(\mathbf{V}|\mathbf{WH})$ uma distribuição gaussiana, de Poisson e de Gamma, respectivamente, cobrindo assim, uma grande área de aplicabilidade e modelagem de sistemas.

Escolher a função-custo correta não é um tarefa fácil. Recomenda-se comparar o desempenho do algoritmo variando a função-custo, dado um conjunto de dados de teste. Então, a função que provê os melhores resultados é escolhida. Um resumo de qual função-custo que se deve utilizar para melhor adaptar seus dados pode ser visualizado na Tabela 3.2.

Tipo do dado	Distribuição	$f(\mathbf{W},\mathbf{H})$	Exemplos
Real	Gaussiana	Frobenius	Imagens
Inteiro	Multinomial	KL	Contagem de palavras
Inteiro	Poisson	KL generalizada	Contagem de fótons
Não-negativo	Gamma multiplicativa	Itakura-Saito	Dados espectrais
Não-negativo	Tweedie	β-divergente	Generalização dos modelos acima

Tabela 3.2: Como escolher a função-custo? (Adaptado de Cédric Févotte).

3.5.1 β-divergente

Pelos motivos expostos acima, uma função-custo generalizada muito utilizada é a β-divergente, definida como

$$f_\beta(y||x) = \begin{cases} \frac{1}{\beta(\beta-1)}\left[y^\beta + (\beta-1)x^\beta - \beta y x^{\beta-1}\right], & \beta \in \mathbb{R}\backslash\{0,1\} \\ \frac{y}{x} - \log\frac{y}{x} - 1, & \beta = 0 \\ y\log\frac{y}{x} + x - y, & \beta = 1, \end{cases} \tag{3.78}$$

cujos gradiente e Hessiana são dados por[11]

$$\nabla_{\mathbf{X}} f_\beta(\mathbf{Y}||\mathbf{AX}) = \mathbf{A}^T\left[(\mathbf{AX})^{\cdot(\beta-1)} - \mathbf{Y}\odot(\mathbf{AX})^{\cdot(\beta-2)}\right] \tag{3.79}$$

$$\nabla^2_{\mathbf{X}} f_\beta(\mathbf{Y}||\mathbf{AX}) = \mathbf{diag}([\mathbf{H}]_{*,i}), \quad i \in \{1,\ldots,N\}, \tag{3.80}$$

sendo

$$\mathbf{H}_i = (\beta-1)\mathbf{A}^T\,\mathbf{diag}\left\{[\mathbf{AX}]^{\cdot(\beta-2)}_{*,i}\right\}\mathbf{A} - (\beta-2)\mathbf{A}^T\,\mathbf{diag}\left\{[\mathbf{Y}\odot(\mathbf{AX})^{\cdot(\beta-3)}]_{*,i}\right\}\mathbf{A}.$$

Adotar $\beta = 0$ é o mesmo que utilizar a divergência de Itakura-Saito, $\beta = 1$ a divergência de Kullback-Leibler e $\beta = 2$ a distância euclidiana. Portanto, adotar essa função deixa o problema generalizado para diversos tipos de aplicações.

Adaptações das MUR com a função-custo β-divergente foram apresentadas por Févotte e Idier [62]. Note que os outros métodos também podem ser adaptados, necessitando apenas de algumas modificações.

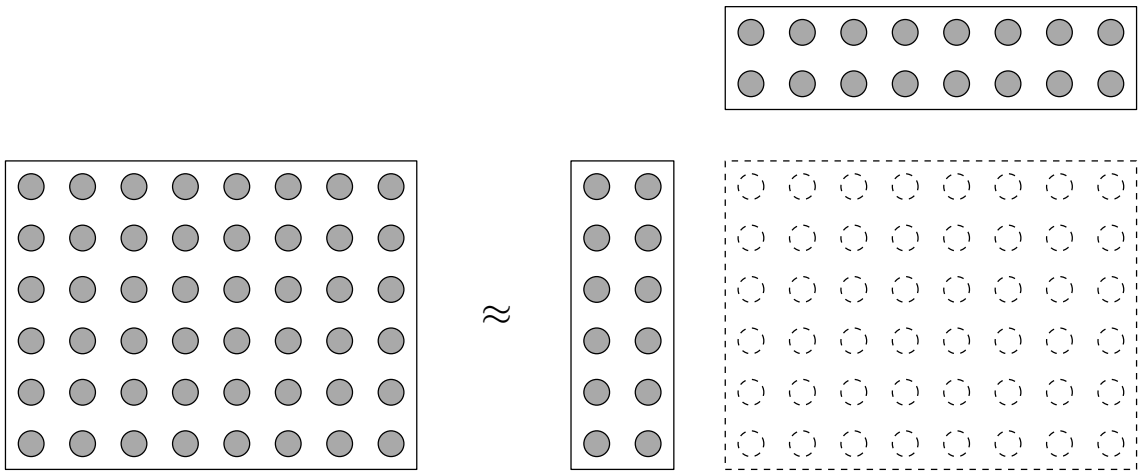
[11] O operador $\mathbf{diag}(\cdot)$ transforma um vetor em uma matriz diagonal.

3.6 Efeitos de regularização

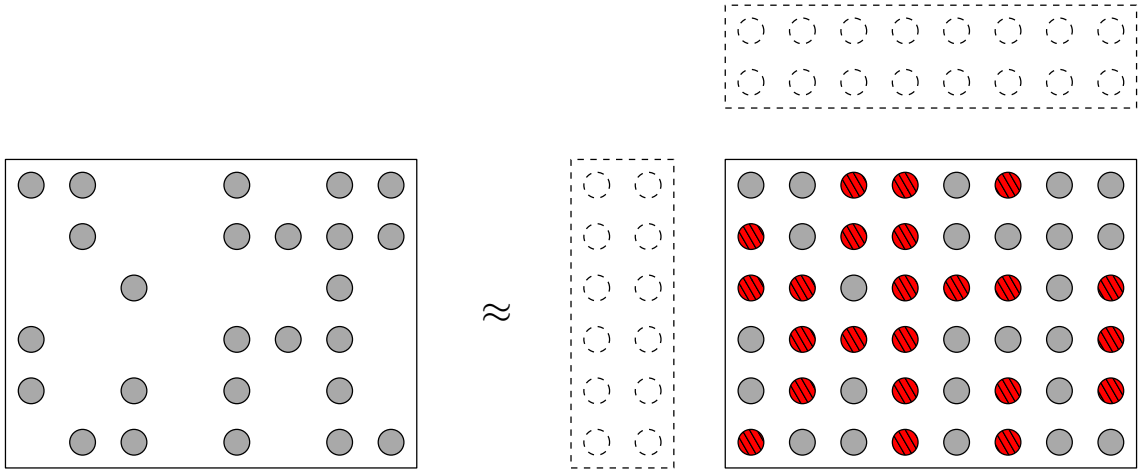
De modo geral, independentemente do algoritmo utilizado, pode-se determinar se a solução encontrada é um ponto de mínimo ou não verificando as condições de otimalidade. No entanto, a NMF contém infinitos pontos de mínimos locais e globais. Dadas uma matriz $\mathbf{D}$ invertível e as soluções $\mathbf{W}$ e $\mathbf{H}$, as novas soluções $\mathbf{WD}$ e $\mathbf{D}^{-1}\mathbf{H}$ também são soluções ótimas do problema, com mesmo custo. Portanto, qualquer permutação e mudança de escala é solução do problema. Para amenizar esse fato, alguns autores realizam alguma regularização, como a normalização das linhas ou colunas dessas matrizes [55], dificultando, assim, a análise de convergência do algoritmo. Por isso, provas sobre a taxa de convergência desses algoritmos ainda são um problema em estudo.

3.7 Interpretação

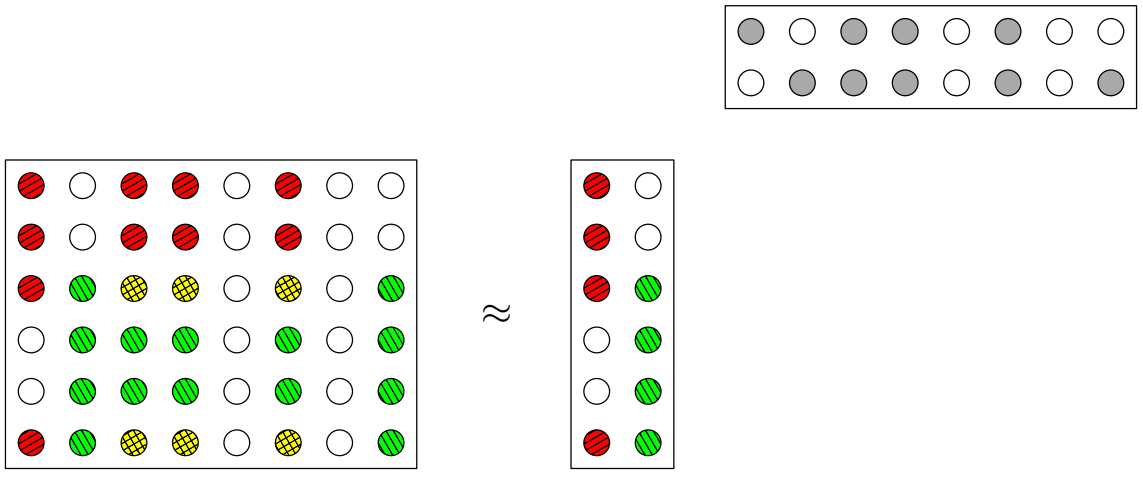
Quando Lee e Seung propuseram seu algoritmo para a fatoração, eles tinham o intuito de representar um conjunto de faces por partes, tais como orelhas, bocas etc (Figura 3.2). Porém, a NMF não é limitada somente a isso, podendo ser empregada em diversas áreas, admitindo diversas interpretações (*e.g.*, Figuras 3.1a, 3.1b e 3.1c). Como a NMF é uma fatoração, pode-se utilizá-la para redução da dimensionalidade do problema (já que $R \ll \min(M,N)$), reduzindo assim o espaço necessário para guardar informações redundantes e o tempo de envio através da web, por exemplo. No plano de mineração de texto, $\mathbf{V}$ é uma matriz que representa a frequência com que cada termo aparece em um determinado documento, a fatoração encontra uma matriz $\mathbf{W}$ que representa um conjunto de palavras e $\mathbf{H}$ é a matriz de documentos codificados. Na área de processamento de imagens, $\mathbf{V}$ pode corresponder aos píxeis de uma imagem corrompida e a fatoração resultante $\hat{\mathbf{V}}$ ser a imagem reconstruída.



(a) Redução de dimensionalidade.



(b) Interpolação, *e.g.*, filtragem colaborativa, reconstrução de imagem [63].



(c) *Unmixing*, *e.g.*, separação de fontes sonoras/imagens.

Figura 3.1: Diferentes usos para a NMF. (Imagens adaptadas de Cédric Févotte, Junho de 2015).

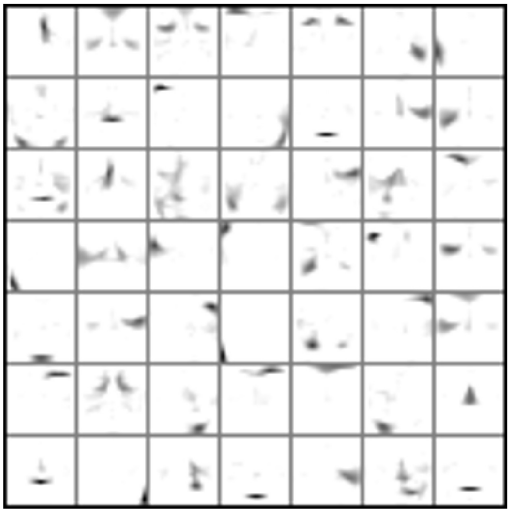


Figura 3.2: Demonstração de Lee e Seung de que a NMF pode aprender formas que relembram partes de rostos. Na figura acima, a matriz $\mathbf{W}$ provém de uma base de dados de mais de dual mil faces condensadas em 49 padrões.

3.7.1 Aplicações em áudio

Em áudio, a matriz $\mathbf{V}$ é um espectograma de magnitude, pois o processamento costuma ocorrer no plano tempo-frequencial e as informações relevantes costumam estar na magnitude, não na fase. Nesse caso, $\mathbf{W}$ contém a base espectral de cada fonte sonora[12] e $\mathbf{H}$ contém as ativações temporais de cada base espectral, como pode ser conferido na Figura 3.3.

Como foi dito, a NMF representa um todo como uma soma ponderada das partes; no caso de áudio, cada base espectral é ponderada por seus ganhos, e cada fonte separada é composta da soma dessas contribuições; o espectograma de magnitude da mistura é visto, então, como uma soma ponderada dos espectogramas de magnitudes das fontes separadas. Temporalmente,

$$\mathbf{y} = \alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2, \quad \alpha_1, \alpha_2 \in \mathbb{R}_+, \tag{3.81}$$

onde $\mathbf{y}$ é a mistura e $\mathbf{x}_1$ e $\mathbf{x}_2$ são as fontes. Realizando a representação tempo-frequencial,

$$\mathbf{Y} = \alpha_1 \mathbf{X}_1 + \alpha_2 \mathbf{X}_2, \tag{3.82}$$

onde $\mathbf{Y}$, $\mathbf{X}_1$ e $\mathbf{X}_2$ são as representações tempo-frequencial aplicando-se a transformada de Fourier de curto termo (STFT, do inglês *Short-Time Fourier Trans-*

[12]Nota-se que o conceito de fonte sonora não está bem definido, podendo ser, por exemplo, um piano de concerto, ou cada uma de suas teclas, ou ainda cada corda percutida pelas teclas (nos casos de notas com 2 ou 3 cordas).

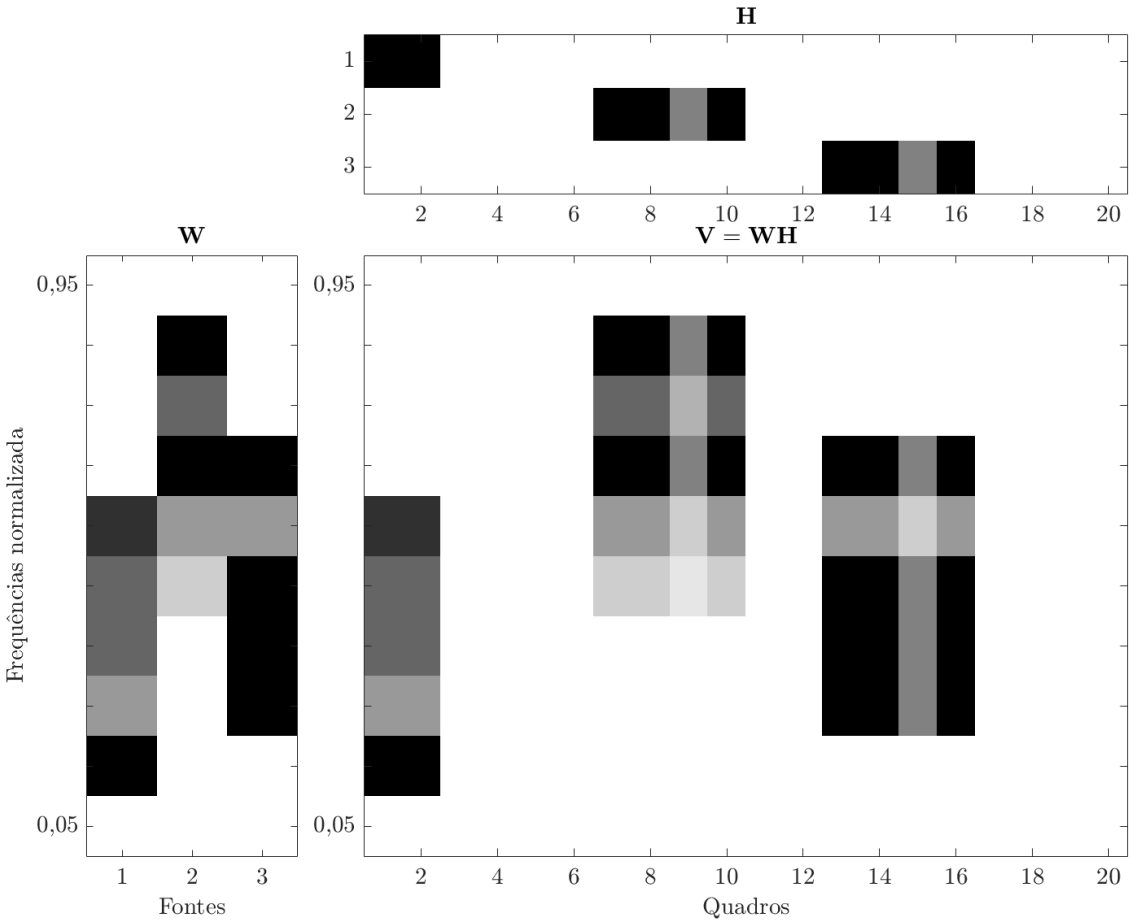


Figura 3.3: Exemplo ilustrativo de como é um gráfico da NMF para a BSS em áudio. A matriz **W** contém os padrões e **H** contém a intensidade de ativações ao longo do eixo das abscissas. Adaptado de [64].

form) [65] nas variáveis $\mathbf{y}$, $\mathbf{x}_1$ e $\mathbf{x}_2$, respectivamente. Portanto, o espectograma de magnitude é dado por

$$|\mathbf{Y}| = |\alpha_1\mathbf{X}_1 + \alpha_2\mathbf{X}_2| \leq |\alpha_1||\mathbf{X}_1| + |\alpha_2||\mathbf{X}_2|, \tag{3.83}$$

o que claramente não corresponde à soma ponderada dos espectogramas de magnitude das fontes separadas. Portanto, é feita uma aproximação, que melhora quanto menor for a sobreposição tempo-frequencial entre as fontes.

A NMF apresentada nesse capítulo consegue realizar uma separação satisfatória de instrumentos percussivos, cujos padrões variam muito pouco ao longo do tempo. Para instrumentos mais complexos, há variantes do método, como a fatoração deconvolutiva de matrizes não-negativas (NMFD, do inglês *Non-negative Matrix Factor Deconvolution*) [11], que permite que um padrão de frequência evolua com um tempo, tal como ocorre com notas de instrumentos, que extrapolam a duração de um quadro. Portanto, a NMFD considera cada nota de um instrumento como uma fonte. Porém, para representar um instrumento como um todo, é interessante

que cada padrão espectral possa ser deslocado no eixo das frequências, modelando assim, o conjunto de notas que podem ser emitidas por um instrumento musical. O método que permite isso é a fatoração duplamente deconvolutiva de matrizes não-negativas (NMF2D, do inglês *Non-Negative Matrix Factor 2-D Deconvolution*) que pode ser conferida em [66]. Caso haja o interesse em aprofundar os conhecimentos dessas representações, vale a pena conferir os trabalhos realizados por Tygel [64] e Almeida [67].

Capítulo 4

Experimentos com NMF

Este capítulo tem o intuito de realizar comparações entre as mais diversas implementações da NMF, visando a determinar quais as qualidades e os defeitos de cada um dos algoritmos.

4.1 Método de avaliação

Existem diversos tipos de abordagens que podem ser adotadas para avaliar se a função-custo é minimizada. No caso, como a função-custo é a distância euclidiana, foi adotada a redução relativa da função-custo

$$\frac{\left\|\mathbf{Y} - \mathbf{A}^{(t)}\mathbf{X}^{(t)}\right\|_{\mathrm{F}}}{\|\mathbf{Y}\|_{\mathrm{F}}} \tag{4.1}$$

onde o índice t corresponde à t-ésima iteração do algoritmo. Vale notar que ao utilizar essa figura de mérito, a qualidade de separação não é considerada, porém o intuito deste capítulo é avaliar a convergência dos algoritmos. Por outro lado, avaliar a qualidade da separação requer a realização de testes objetivos e subjetivos, tema que será abordado mais à frente.

4.2 Base de dados

Como o intuito principal é avaliar a eficácia de diferentes métodos para a fatoração de matrizes não-negativas, foram realizados testes em 5 bases de dados naturais, tal como em [48][1]. Na Tabela 4.1, podem ser conferidas as informações sobre essas bases de dados. Entre elas, há duas bases de texto esparsas. O *corpus* da TDT2 (do inglês *The Topic Detection and Tracking 2*) contém vários artigos de várias fontes como *New York Times*, *Cable News Network* (CNN) e *Voice of America* (VOA),

[1] Porém comparando mais métodos e utilizando métricas de avaliação de qualidade.

todas de 1998. O *corpus* foi marcado manualmente através de 96 tópicos distintos e tem sido amplamente utilizado em mineração de texto. Desse *corpus*, foram selecionados aleatoriamente 40 tópicos nos quais o número de artigos excedesse 10. Contando a frequência de termos em cada um desses documentos, obteve-se uma matriz de tamanho 19.000×3.087. A outra base de texto, *The Newsgroups 20*, é uma coleção de documentos classificados em 20 tópicos, de onde se extraiu uma matriz de documentos 26.214×11.314.

Há também, como pode ser visto na Tabela 4.1, uma base de dados de imagens. A base de dados faciais criada pelo laboratório da AT&T de Cambridge (ATNT) contém 400 imagens de rostos de 40 pessoas diferentes, com 10 imagens por pessoa. Cada imagem tem 92×112 píxeis em tons de cinza representados usando 8 bits, resultando em uma matriz de tamanho 10.304×400.

No campo de separação de sons, a base utilizada será um excerto da compilada pela SiSEC[2] para um dos seus desafios de separar uma ou mais fontes de gravações profissionais. O áudio contém múltiplos instrumentos além de voz cantada, e está amostrado em 44,1 kHz. Esse sinal foi processado calculando-se sua STFT com quadros de 1024 amostras, passos de 256 amostras, uma DFT (do inglês *Discrete Fourier Transform*) de 2048 pontos e utilizando a janela de Hamming. A matriz $\mathbf{V}$ é dada como o módulo desse espectro resultante.

Por fim, para mostrarmos o funcionamento da NMF no campo de misturas aditivas, foi utilizada uma base disponibilizada pela universidade de Stirling[3], contendo 6 fotos com 275×350 píxeis das faces de 2 pessoas com expressões distintas. Foi gerada uma matriz de mistura aditiva $\mathbf{H} \in \mathbb{R}_{+}^{6\times 12}$ que, após ser multiplicada pela matriz $\mathbf{W}$ contendo uma imagem por coluna, resulta na Figura 4.1.

Base de dados	Tamanho	Esparsidade
TDT2	19.000×3.087	99,69%
20Newsgroup	26.214×11.314	99,66%
ATNT	10.304×400	0,0%
SiSEC	513×41.792	66,87%
Stirling	96.250×12	23,71%

Tabela 4.1: Informações das bases de dados utilizadas. A esparsidade foi calculada considerando valores menores que 0,01% do máximo encontrado para cada uma dessas bases de dados.

[2] `https://sisec.inria.fr/home/2016-professionally-produced-music-recordings/`
[3] `http://pics.stir.ac.uk`

Figura 4.1: As imagens contidas na matriz $\mathbf{V}$ para a base Stirling.

4.3 Algoritmos utilizados

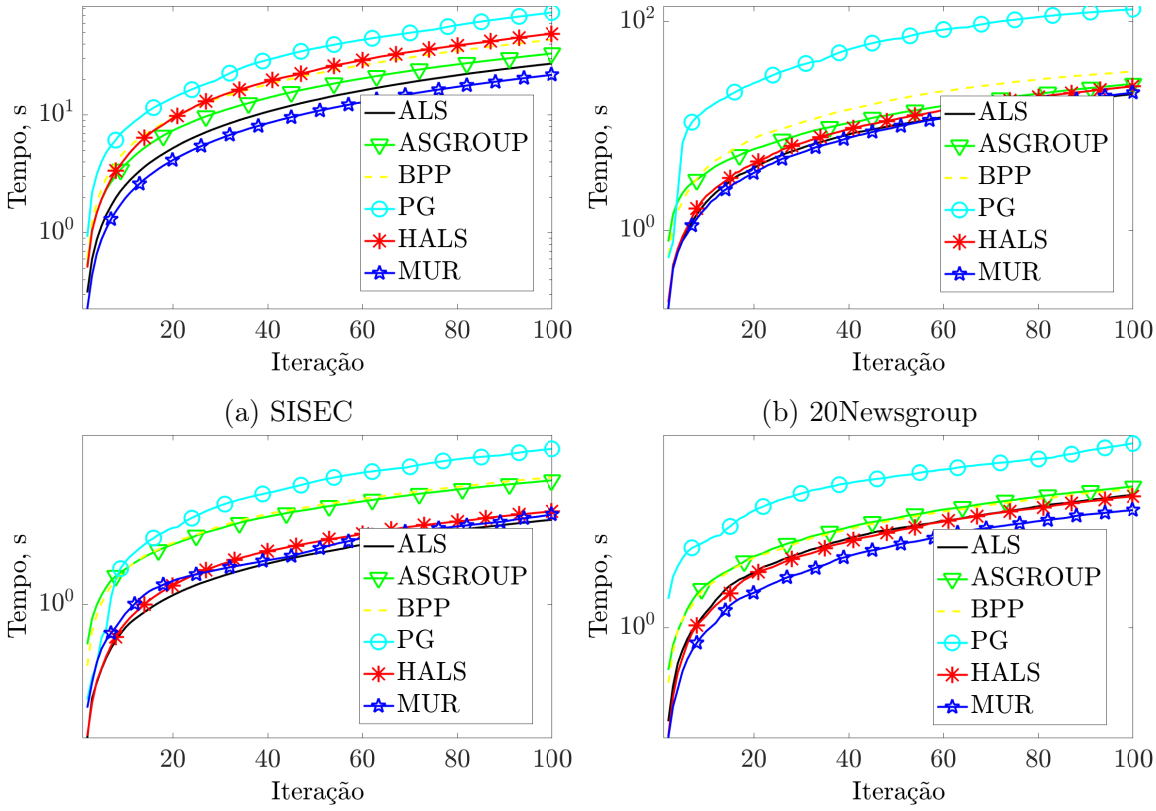
Dentre os algoritmos que serão mostrados, foram implementados pelo o autor o ALS, MUR, PG (com $\sigma = 0{,}01$ e $\mu = 0{,}1$) e o HALS (Equações (3.40) e (3.41)), seguindo como referência o livro "*Nonnegative Matrix and Tensor Factorization*" [58]. O último algoritmo citado foi adaptado de uma implementação feita pelos autores em [48], e os algoritmos ASGROUP (modificação do algoritmo AS para aumentar sua velocidade) e BPP foram disponibilizados pelos mesmos autores. Em todos os resultados que serão mostrados foi realizada a média de 5 simulações para cada algoritmo.

4.4 Resultados

Para bases grandes, como SiSEC, ATNT, TDT2 e 20Newsgroup, o algoritmo PQN demonstrou ser muito ineficiente à medida que a simulação aumentava a quantidade de componentes a serem fatoradas, devido ao mau condicionamento da Hessiana. Por esse motivo, o algoritmo foi excluído dos resultados que serão mostrados adiante.

De todos os algoritmos testados, o clássico MUR é o que apresenta o melhor balanço de tempo gasto por iteração para todos as R componentes, como visto nas Figuras de 4.2 a 4.4, apesar de os outros algoritmos possuírem resultados similares para um R baixo. O PG e o ASGROUP consomem mais tempo por iteração, pois o PG tem um passo interno para determinar o passo ótimo (Equação (3.19)), enquanto o ASGROUP realiza diversas iterações internas para encontrar o conjunto ativo, o que pode ser custoso computacionalmente.

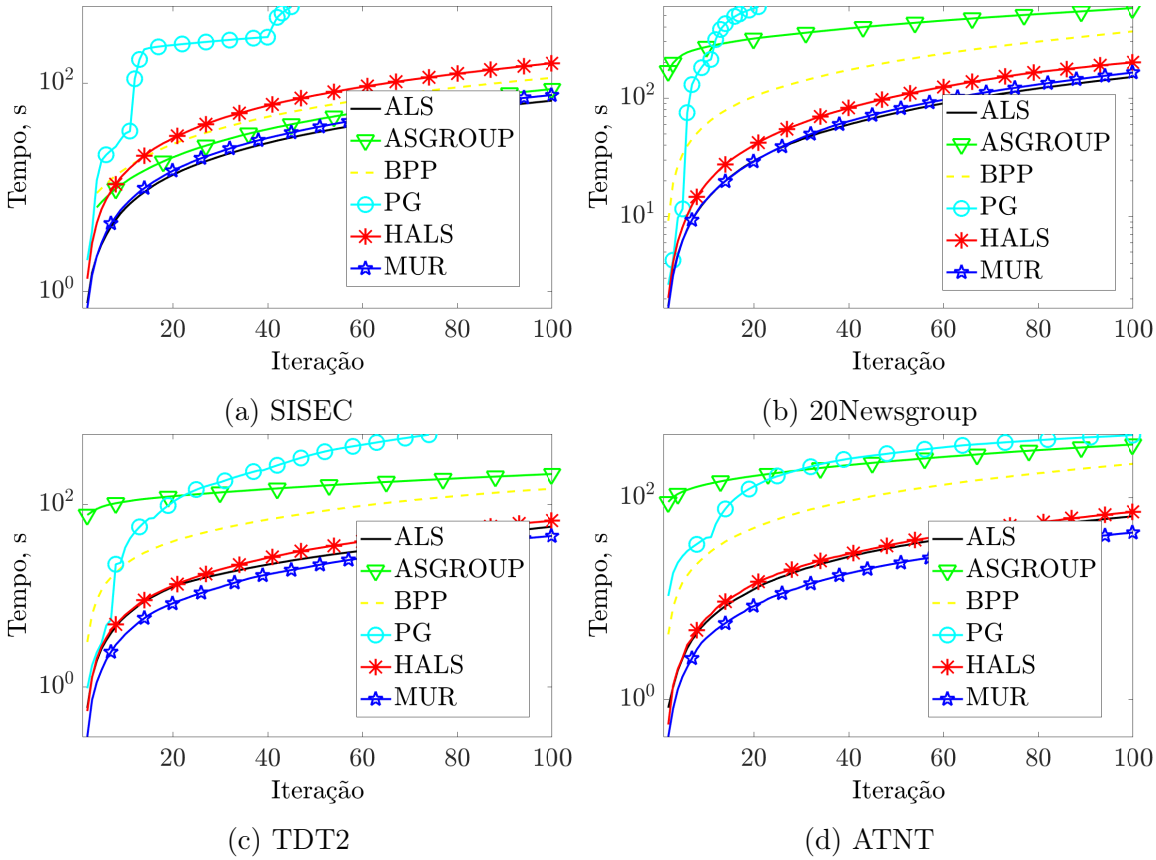
Com o aumento do R, encontrar o passo ótimo para o gradiente descendente projetado se torna uma tarefa cada vez mais difícil, aumentando exponencialmente o tempo por iteração do PG. Em bases esparsas, o algoritmo BPP sobressai ante o ASGROUP, devido à utilização da fatoração de Cholesky [47], economizando processamento por iteração e mantendo-se estável. No entanto, para uma base densa (ATNT), o algoritmo deixa de economizar processamento, aproximando-se do ASGROUP, como pode ser visto na Figura 4.4d.



(a) SISEC (b) 20Newsgroup

(c) TDT2 (d) ATNT

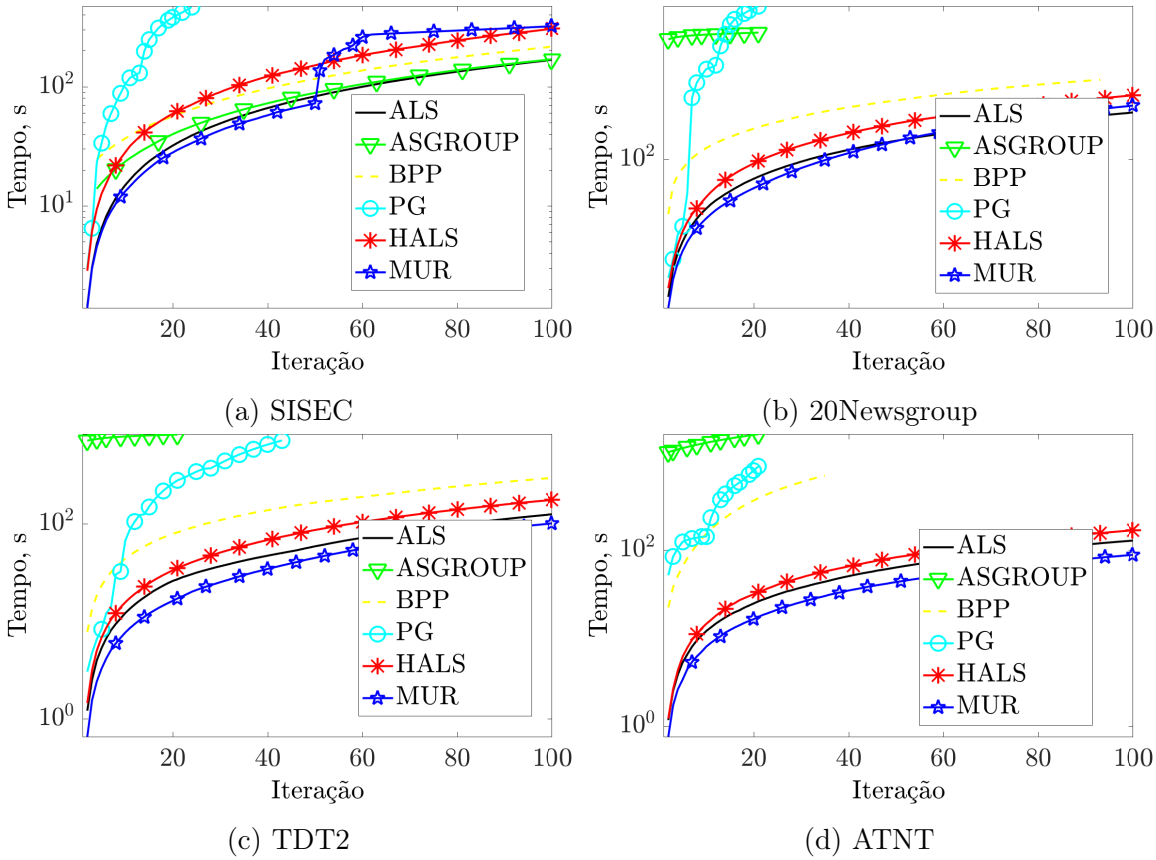
Figura 4.2: Tempo acumulado gasto ao longo das iterações utilizando $R = 10$.



(a) SISEC (b) 20Newsgroup

(c) TDT2 (d) ATNT

Figura 4.3: Tempo acumulado gasto ao longo das iterações utilizando $R = 80$.



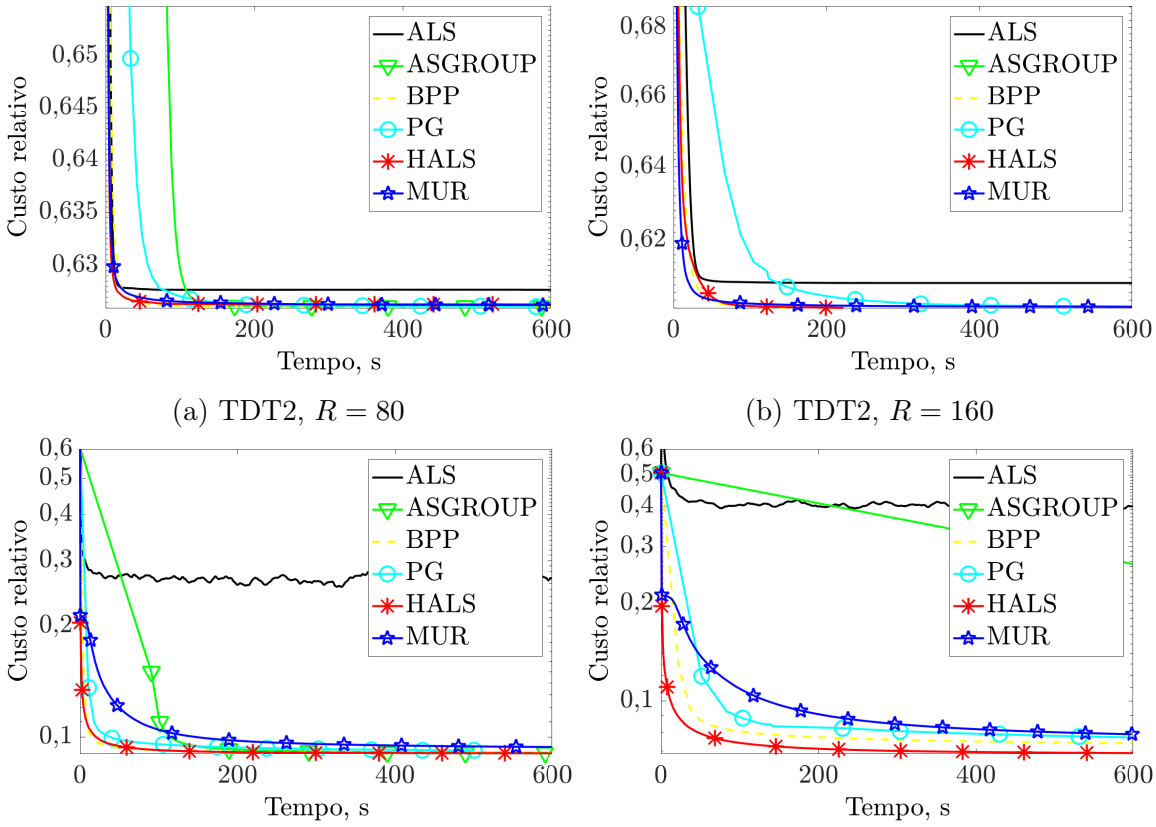
(a) SISEC (b) 20Newsgroup

(c) TDT2 (d) ATNT

Figura 4.4: Tempo acumulado gasto ao longo das iterações utilizando $R = 160$.

Apesar de os algoritmos ALS, MUR e HALS terem desempenho semelhante nessa avaliação, ao compararmos a redução da função-custo ao longo do tempo, como se vê na Figura 4.5b, o ALS falha em acompanhar os outros algoritmos para o mesmo ponto de mínimo, até mesmo divergindo em alguns casos, vide Figuras 4.5c e 4.5d, devido a sua aproximação rudimentar ao problema de fatoração de matrizes não-negativas. Em problemas com posto alto, os algoritmos possuem desempenho similar, apesar de MUR, HALS e BPP chegarem no mínimo mais rápido.

À medida que o problema se torna denso e com grandes números de componentes o HALS consegue se sobressair, como pode ser visto na Figura 4.5d.



(a) TDT2, $R = 80$

(b) TDT2, $R = 160$

(c) ATNT, $R = 80$

(d) ATNT, $R = 160$

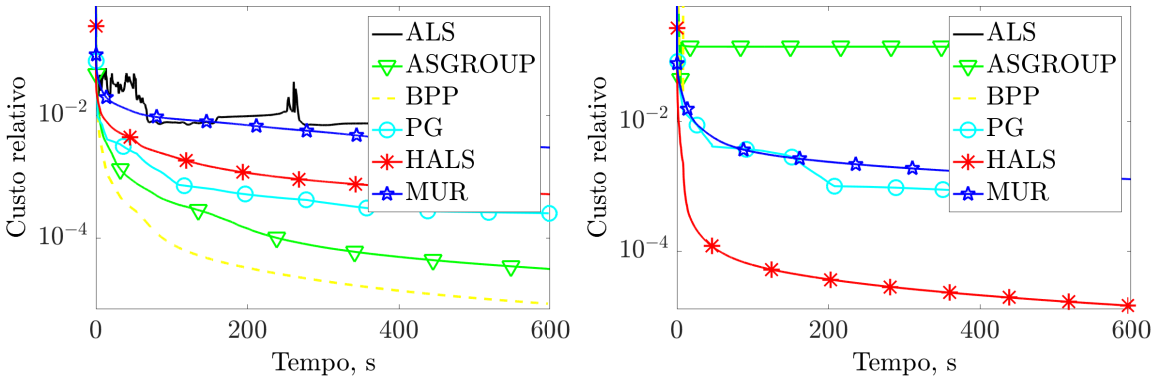
Figura 4.5: O custo relativo, $\|\mathbf{V} - \mathbf{WH}\|_{\mathrm{F}} / \|\mathbf{V}\|_{\mathrm{F}}$, em função do tempo. Os resultados para o SiSEC e o 20Newsgroup foram similares ao do TDT2 e por isso foram omitidos.

Até agora foi mostrado que os algoritmos, com exceção de PG e ASGROUP, são bastante baratos por iteração e que para um problema denso e com muitas componentes para fatorar, o BPP tem custo computacional elevado. No entanto, apesar de ser rápido, o algoritmo ALS pode falhar e até mesmo divergir em algumas situações. O MUR, apesar de simples e fácil de implementar, não garante uma rápida convergência, necessitando às vezes de muito mais iterações que outros algoritmos, um revés que ocorre devido ao uso de uma regra fixa para a escolha do passo, que garante a não-negatividade.

Em problemas pequenos, utilizando a base Stirling, podemos notar que, fatorando em 6 componentes (o posto exato da matriz), os algoritmos convergem com sucesso, minimizando a função-custo, e com exceção do ALS, conseguem ser mais rápidos que o MUR, como é mostrado na Figura 4.6a. Uma "foto" do algoritmo feito em 30 segundos de simulação pode ser visualizada na Figura 4.7, onde podemos constatar que os algoritmos BPP e ASGROUP conseguem reconstruir as imagens originais com maior velocidade que o MUR.

Ao realizarmos o mesmo procedimento fatorando em 12 componentes, os métodos ALS, ASGROUP e BPP falham, vide Figura 4.6b, pois a matriz $\mathbf{W}$ não terá posto

completo, o que é um dos pré-requisitos desses algoritmos. Já os outros, por não terem essa restrição, conseguem com sucesso fatorar o problema.



(a) $R = 6$ (b) $R = 12$

Figura 4.6: Redução da função-custo utilizando a base Stirling.

Figura 4.7: Imagens contidas na matriz **W** resultante. De cima para baixo: ALS, MUR, PG, ASGROUP, BPP, HALS.

Capítulo 5

Aplicação: Transcrição de Instrumentos Percussivos

A detecção de eventos de *onset* é considerada um processamento de baixo nível, porém extremamente importante. Serve de prelúdio para diversos algoritmos de *music information retrieval* (MIR), como *query-by-humming* [68], reconhecimento de instrumentos musicais [69] e em particular a transcrição de instrumentos percussivos [70, 71]. O intuito desse capítulo é revisar como pode ser realizada essa tarefa e então utilizar a NMF para auxiliar na transcrição de um ou mais instrumentos percussivos contidos em uma mistura.

5.1 Transitório x *onset* x ataque

Antes de se realizar a detecção de *onsets* é necessário defini-los corretamente. Quando uma nota é emitida, como pode ser observado na Figura 5.1, em geral há quatro características temporais importantes: *onset*, ataque, transitório e decaimento.

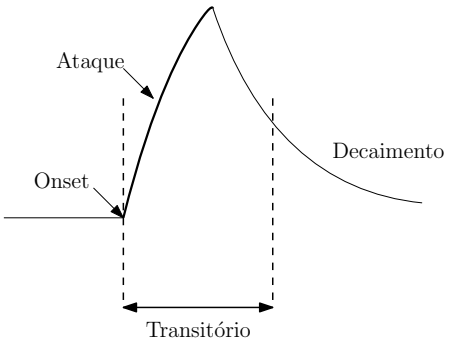


Figura 5.1: Diferença entre transitório, *onset* e ataque.

O ataque é definido como o intervalo de tempo no qual a amplitude do envelope é crescente. O transitório possui uma definição mais imprecisa, porém é de senso

comum que ele representa a parte na qual o sinal apresenta comportamentos que não são previsíveis, como por exemplo, no caso de um bumbo quando há a excitação com a aplicação da baqueta. Apesar de essa definição não ser bem exata, é importante citar que o ouvido humano não é capaz de diferenciar dois transitórios com intervalo menor que 10 ms. Finalmente, o *onset* é definido como um momento único no tempo que delimita o começo da percepção do transitório.

5.2 Esquema geral dos algoritmos de detecção de *onset*

Pode-se dizer que o requisito mais importante dos sistemas de transcrição de instrumentos percussivos é detectar o *onset* corretamente, pois todo o processo a posteriori é sensível a essa etapa.

Para uma simples nota em uma gravação de alta qualidade, esses eventos são todos fáceis de serem detectados, no entanto, em um sinal com mais de um instrumento, ruído de fundo e outros fatores, essas definições ficam mais difusas.

Realizar essa detecção não é tão simples quanto diferenciar o sinal no tempo e definir as regiões de maiores energia como sendo o momento que um instrumento foi acionado. É necessário mapear o sinal utilizando uma função denominada função de detecção. Como pode ser observado na Figura 5.2, a partir do sinal original, que pode ser pré-processado para aumentar a eficiência dos métodos posteriores, gera-se uma função de detecção de taxa reduzida em relação à taxa de amostragem do sinal, sobre a qual serão procurados os picos que determinam os *onsets*.

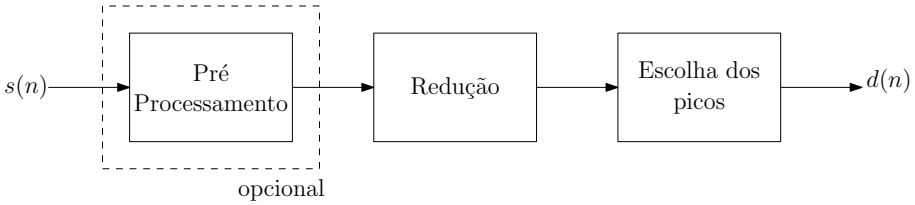


Figura 5.2: Diagrama de blocos para detecção de *onsets*.

5.3 Pré-processamento

O pré-processamento é uma etapa que tem o intuito de acentuar propriedades do sinal que sejam de interesse e atenuar o que não se deseja. Há diversos métodos existentes para serem aplicados a sinais de áudio, porém [70] classifica esta etapa em duas categorias distintas: separação do sinal em múltiplas bandas, ou separação em regimes permanente e transitório.

5.3.1 Múltiplas bandas

Vários estudos nesta área propõem a análise de diversas sub-bandas independentemente [70, 72]. Normalmente, aplicam-se filtros, separando o sinal em múltiplas bandas de frequência e descobre-se o *onset* de cada banda, combinando as informações ao final do processo, tal como é feito em [73]. De outra forma, pode-se utilizar as informações de sub-banda para dar uma complementariedade à detecção global de *onset*.

5.4 Redução

No contexto de detecção de *onsets*, o conceito de redução está relacionado ao fato de transformar o sinal em outro, acentuando suas características transitórias para facilitar a detecção de *onsets*. Nessa categoria, há diversos algoritmos, alguns dos quais serão citados nesse projeto.

5.4.1 Características temporais

Quando se observa a evolução temporal de sinais musicais, fica claro que a ocorrência de algum *onset* é seguida de um acréscimo na amplitude do sinal. Os primeiros métodos para detecção de *onset* baseavam-se nessa ideia de criar uma função de detecção que seguisse a amplitude do sinal. Tal seguidor de envelope pode ser construído retificando e suavizando o sinal,

$$E_0(n) = \frac{1}{L} \sum_{l=-\frac{L}{2}}^{\frac{L}{2}-1} |x(n+l)|w(l), \tag{5.1}$$

onde $w(m)$ é uma janela de N pontos centrada em 0. Esse tipo de algoritmo funciona bem para sinais que contêm sons percussivos fortes e baixo ruído de fundo. Em vez de seguir o envelope, podemos seguir a energia local,

$$E(n) = \frac{1}{L} \sum_{l=-\frac{L}{2}}^{\frac{L}{2}-1} [x(n+l)]^2 w(l). \tag{5.2}$$

Apesar da suavização que é realizada, o sinal reduzido proveniente dos métodos citados acima não é usualmente utilizado na seleção de picos por não retornar resultados confiáveis. Como um refinamento, pode-se utilizar a derivada da energia do sinal de modo que transições abruptas na energia são transformadas em picos estreitos. A energia e sua derivada normalmente são utilizadas em combinação com alguma técnica de pré-processamento discutida anteriormente.

Outro refinamento é aproveitado da psicoacústica. Sabendo que a intensidade do áudio é percebida logaritmicamente, e que $\partial(\log E)/\partial t = (1/E)\partial E/\partial t$, então calcular a diferença do log da energia simula grosseiramente a percepção humana de audibilidade.

5.4.2 Características espectrais

Sem dúvidas, a maior parte das funções de detecção foram propostas na área de características espectrais, pois garantem mais confiabilidade nos resultados. Além disso, permitem reduzir o pré-processamento (como a remoção da parte tonal) e são bem sucedidas em muitos cenários, incluindo detecção de *onsets* em músicas polifônicas com instrumentos de diversos tipos [70].

5.4.2.1 *High frequency content*

No domínio espectral, um aumento de energia ligado ao transitório tende a aparecer como um evento em todas as faixas de frequências. Como a energia do sinal normalmente fica concentrada em baixas frequências, alterações realizadas pelo transitório são mais visíveis em altas frequências. Para realçar essa característica, o espectro pode ser ponderado antes da soma para se obter uma medida de energia ponderada

$$\tilde{E}(n) = \frac{1}{K} \sum_{k=-\frac{K}{2}}^{\frac{K}{2}-1} W_k |X(n,k)|^2 \tag{5.3}$$

onde W_k é o peso dependente de frequência. Por Parseval, se $W_k = 1 \ \forall \ k$, $\tilde{E}(n)$ é igual ao método de energia local, definido anteriormente. Uma boa escolha é $W_k = |k|$, ponderando linearmente cada raia proporcionalmente à sua frequência. Tal método com essa escolha de peso é chamado *high frequency content* (HFC). A função HFC produz picos proeminentes durante os ataques e é conhecida pelo sucesso em detectar *onsets* percussivos.

5.4.2.2 *Spectral Difference*

Apesar de esses métodos serem bons, são medidas baseadas no espectro de curto-termo, omitindo qualquer consideração de evolução temporal. Uma abordagem mais abrangente que considera a evolução temporal do espectro pode ser feita formulando-se a função de detecção como uma distância entre sucessivos espectros, ou seja, considerando-os como pontos em um espaço K-dimensional. Dependendo da métrica utilizada, distintas diferenças espectrais (SF, do inglês *spectral flux*) podem

ser construídas. Uma bastante utilizada é construída utilizando a norma ℓ_2 de um retificador

$$\mathrm{SD}(n) = \sum_{k=-\frac{K}{2}}^{\frac{K}{2}-1} H[|X(n,k)| - |X(n-1,k)|]^2, \tag{5.4}$$

onde $H(x) = (x + |x|)/2$, ou seja, zero para argumentos negativos. A retificação tem o efeito de contar somente aquelas frequências que contribuem para o aumento de energia, enfatizando o *onset* mas não o *offset*. Como apontado em [71], testes empíricos favorecem o uso da norma ℓ_1 e o uso de uma magnitude linear em vez da logarítmica proposta, por Klapuri [72].

5.4.3 Características espectrais utilizando a fase

Todos os métodos supracitados utilizam a informação de magnitude do espectro como única fonte de informação. No entanto, informações de fase também podem ser utilizadas na análise de *onset*. Seja $\phi(n,k)$ a fase de um coeficiente STFT $X(n,k)$. Para uma senoide, a fase $\phi(n,k)$ e a fase da janela anterior $\phi(n-1,k)$ são utilizados para calcular o valor da frequência instantânea; uma estimação da frequência do k-ésimo componente da STFT é dada por

$$f(n,k) = \left(\frac{\phi(n,k) - \phi(n-1,k)}{2\pi h} \right) f_s, \tag{5.5}$$

onde h é o salto e f_s é a taxa de amostragem. É esperado que, para uma senoide estacionária, a frequência instantânea seja aproximadamente constante ao longo das janelas, isto é

$$\phi(n,k) - \phi(n-1,k) \approx \phi(n-1,k) - \phi(n-2,k). \tag{5.6}$$

Podemos reescrever a Equação (5.6) de modo a enfatizar o desvio de fase que ocorre:

$$\Delta\phi(n,k) = \phi(n,k) - 2\phi(n-1,k) + \phi(n-2,k) \approx 0. \tag{5.7}$$

5.4.3.1 Desvio de Fase

No transitório, a frequência instantânea não é bem definida, então $\Delta\phi_k(n)$ tende a ser bem maior em magnitude. Em [70] é proposto um método que realiza uma análise da distribuição instantânea dos desvios de fase sobre o domínio da frequência. Durante a parcela de regime permanente do som, esse desvio tende a zero. Durante

ataques $\Delta\phi(n,k)$ aumenta, e esse efeito pode ser calculado de forma simples:

$$\mathrm{PD}(n) = \frac{1}{K}\sum_{k=1}^{K}|\Delta\phi(n,k)|, \tag{5.8}$$

denotando o desvio médio absoluto da fase.

5.4.3.2 Desvio de Fase ponderado

Apesar de o método anterior alcançar algumas melhorias sobre sinais complexos, ele é suscetível a distorções de fase e ruídos. Uma alternativa é utilizar o algoritmo proposto em [71], chamado de *weighted phase deviation* (WPD):

$$\mathrm{WPD}(n) = \frac{1}{K}\sum_{k=1}^{K}|X(n,k)\Delta\phi(n,k)|. \tag{5.9}$$

5.5 Pós-processamento

Assim como o pré-processamento, essa etapa é opcional e dependerá do tipo de método que foi utilizado na fase de redução. O pós-processamento tem como intuito facilitar a tarefa de seleção de limiares e seleção de picos, transformando o sinal resultante da função de detecção em formas mais facilmente detectáveis. Porém, é de consenso que utilizar suavizadores [74], e realizar a normalização, como utilizado em [71], facilita a tarefa de escolha de picos.

5.6 Escolha de picos

Mesmo após todas as etapas descritas, podem existir picos que não correspondem a *onsets*. Portanto, é necessário definir um limiar responsável por separar os picos relevantes dos não-relevantes. Ingenuamente, podemos escolher um limiar fixo tal que todo $d(n) \geq \delta$ é considerado um evento de *onset*, onde δ é uma constante positiva e $d(n)$ é a função de detecção (como visto na Figura 5.2). Apesar de esse método poder funcionar em sinais com pouca dinâmica e alta SNR (do inglês, *signal-to-noise ratio*), sinais musicais apresentam grande variação de audibilidade ao longo do tempo. Nessas situações, esse tipo de limiar irá deixar passar *onsets* em períodos de baixa excitação e irá detectar uma grande quantidade de *onsets* em períodos agitados.

Usualmente é empregado um limiar adaptativo $\hat{\delta}(n)$, calculado como uma versão suavizada da função de detecção. Essa suavização pode ser linear, utilizando um

filtro passa-baixas FIR

$$\hat{\delta}(n) = \delta + \sum_{l=0}^{L} a_i d(n-l), \tag{5.10}$$

onde $a_0 = 1$. Seguindo a lógica, a suavização pode ser não-linear

$$\hat{\delta}(n) = \delta + \xi \sum_{l=-L}^{L} w_i d^2(n+l), \tag{5.11}$$

onde ξ é uma constante positiva e w_i é a janela de suavização. No entanto, esse tipo de limiar adaptativo pode apresentar grandes flutuações quando há picos largos na função de detecção, ocultando picos menores adjacentes. Para isso são utilizados métodos do tipo

$$\hat{\delta}(n) = \delta + \xi f(\bar{m}), \quad m - \beta L \leq \bar{m} \leq m + L, \tag{5.12}$$

onde f é uma função como a média local ou mediana local da função de detecção.

Após o pós-processamento e a escolha de limiar, a seleção de picos se limita a selecionar máximos locais que estão acima de um limiar. No entanto, após todo esse processo ainda há vários parâmetros que devem ser selecionados. Como solução, realizam-se diversos experimentos variando esses parâmetros até achar um número que maximize a quantidade de acertos.

5.7 NMF aplicada à transcrição

Nessa categoria, também conhecida como separação e detecção, são separados os instrumentos através do uso da NMF de forma a permitir que se realize a detecção de *onsets* individualmente. Como exemplificado na Figura 5.3, pode-se realizar o processo de transcrição somente com a gravação da mistura ou utilizando informações a priori dos instrumentos de interesse, construindo uma base espectral $\mathbf{W}_p$ que facilite o processo de separação realizado pela NMF. Essa informação a priori pode ser uma base com os instrumentos separados, ou até a gravação onde os músicos estão testando seus instrumentos isoladamente. Após a separação, é utilizado o espectro do sinal separado para a detecção de *onset* e sua série temporal é recuperada através da reconstrução de fase seguida de uma inversa da STFT.

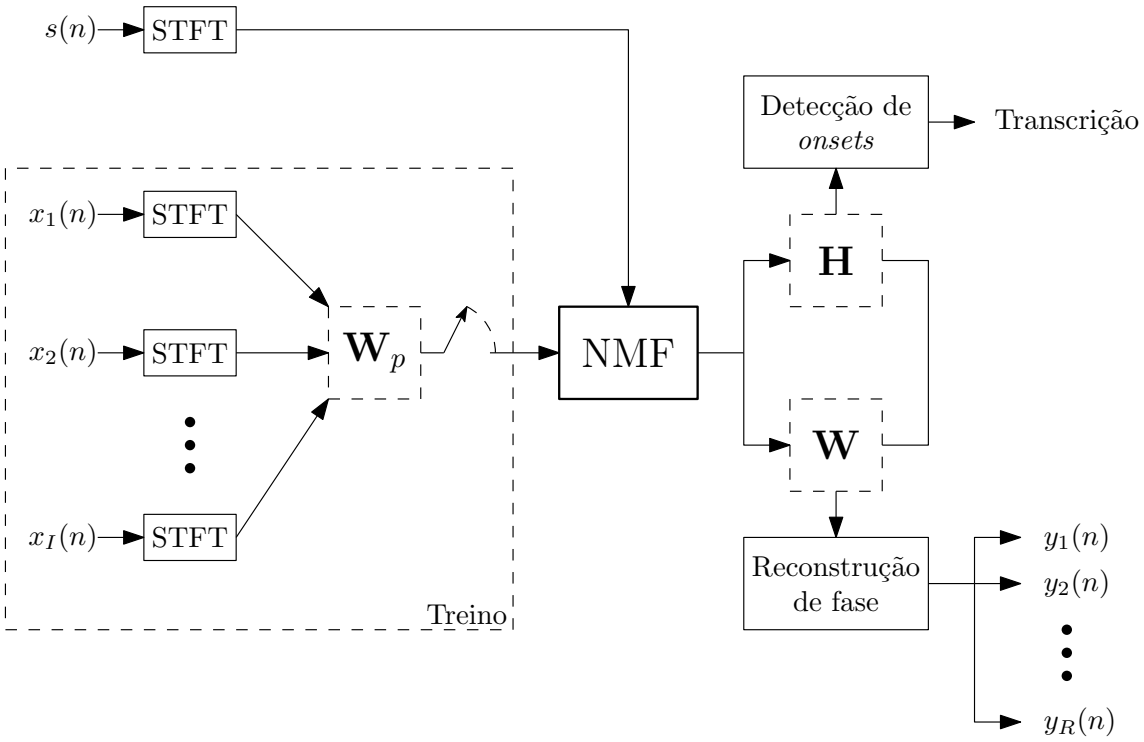


Figura 5.3: Exemplificação da transcrição de instrumentos percussivos utilizando a NMF. As matrizes resultantes servem tanto para a transcrição quanto para obter as gravações dos instrumentos separados. (Adaptado de [75]).

5.7.1 NMF com bases adaptativas

Pode-se utilizar qualquer algoritmo descrito nos capítulos anteriores para realizar a separação de instrumentos percussivos; no entanto, inicializar a base espectral $\mathbf{W}^{(0)}$ com $\mathbf{W}_p$ facilita o processo de convergência do algoritmo.

5.7.2 NMF com bases fixas

Outro meio de realizar a separação é omitir a parte da atualização de $\mathbf{W}$, deixando-o fixo, esperando que o espectro será fatorado para indicar apenas as ativações em $\mathbf{H}$. Desse modo, a NMF com uma base fixa não será capaz de generalizar bem a dinâmica temporal do instrumento. Um modo de contornar esse problema é separar a base espectral dos instrumentos percussivos em ataque e decaimento, fixando apenas as bases que contem o ataque. Podemos notar que esse método é mais sensível à presença de picos na matriz de ativações, devido a outros sinais presentes (ou variações no modo de tocar o instrumento) que não estão presentes na base espectral.

5.7.3 NMF com bases semiadaptativas

Um meio termo entre os dois métodos é impor uma base semiadaptativa [75]. Em vez de inicializar $\mathbf{W}$ com $\mathbf{W}_p$ e deixar o algoritmo modificá-lo livremente, podemos modificar a base espectral de acordo com as quantidades de iterações. Dado que $\mathbf{W}^{(0)} = \mathbf{W}_p$, na t-ésima iteração teremos

$$\mathbf{W}^{(t)} = \nu\mathbf{W}^{(t-1)} + (1-\nu)\mathbf{W}^{(t)}, \tag{5.13}$$

$$\nu = (1 - t/t_{\max})^{\varphi}. \tag{5.14}$$

Como pode ser visto na Equação (5.13), inicialmente a NMF é calculada deixando $\mathbf{W}$ o mais próximo possível do calculado $\mathbf{W}_p$ e ao final das iterações um ajuste fino é realizado, adaptando-o para o espectro real. Em [75] foi adotado $\varphi = 2$.

Capítulo 6

Experimentos com Transcrição

Neste capítulo, iremos avaliar as diversas funções de detecção de *onset* para determinar qual é o melhor algoritmo para ser usado na transcrição de instrumentos percussivos. A maior dificuldade para a avaliação gira em torno de se obter uma base de dados anotada suficientemente extensa, cobrindo diversas variedades de estilos, complexidades etc.

Outro problema envolvendo esse experimento consiste em como será realizada a avaliação, já que há alguns parâmetros que podem ser ajustados de modo a controlar a quantidade de falsos positivos e falsos negativos. Mais abaixo será explicado como vamos endereçar esse problema e um outro que vem da psicoacústica: a percepção dos *onsets*.

6.1 Método de avaliação

Dependendo do tempo decorrido entre duas notas consecutivas, um ou mais *onsets* podem ser percebidos, no entanto isso depende do instrumento tocado e de outros fatores externos, como a presença de ruídos e sons simultâneos. Por isso, será considerado corretamente anotado aquele *onset* detectado que estiver dentro de um intervalo de 50 ms da anotação real.

Esse número arbitrado segue o valor encontrado na literatura [70, 71]; no entanto, testes feitos demonstraram que esse valor é grande demais, pois dependendo das características do sinal é possível diferenciar dois ataques com intervalos menores de 50 ms.

Para avaliar os resultados serão utilizadas três estatísticas: precisão P, sensibili-

dade S e medida F1 (média harmônica entre P e S):

$$\mathrm{P} = \frac{\mathrm{VP}}{\mathrm{VP} + \mathrm{FP}} \tag{6.1}$$

$$\mathrm{S} = \frac{\mathrm{VP}}{\mathrm{VP} + \mathrm{FN}} \tag{6.2}$$

$$\mathrm{F1} = \frac{2\mathrm{PS}}{\mathrm{P} + \mathrm{S}} = \frac{2\mathrm{VP}}{2\mathrm{VP} + \mathrm{FP} + \mathrm{FN}}, \tag{6.3}$$

onde VP é a quantidade de detecções corretas, FN é o número de falsos negativos e FP o número de falsos positivos (veja a Figura 6.1).

Os parâmetros foram escolhidos de modo que F1 fosse maximizado; porém, dependendo do tipo de aplicação, os falsos positivos e falsos negativos não são igualmente desejáveis, portanto outra escolha de parâmetros poderia ser mais adequada.

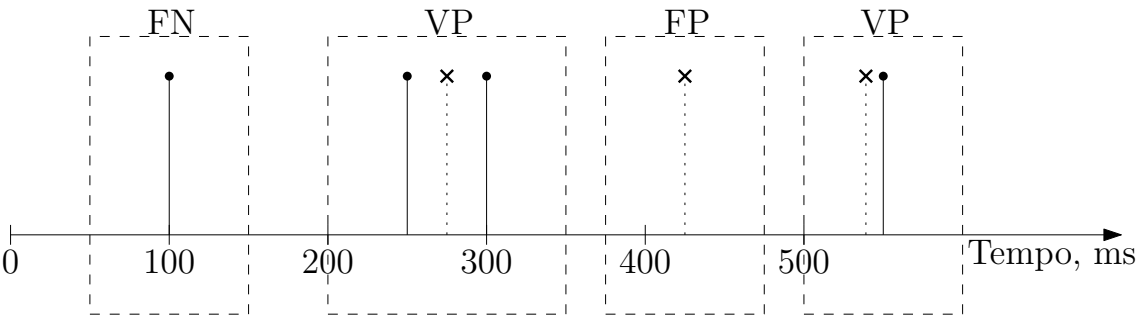


Figura 6.1: Exemplo de detecção de VP, FP e FN. Note que duas notas próximas são mescladas.

6.1.1 Avaliação da separação: Medidas baseadas em SNR

Além da qualidade da transcrição, também será avaliada a qualidade dos sons separados dos instrumentos. No entanto, medidas subjetivas automáticas de qualidade para o caso de separação cega de fontes ainda não existem ou não são amplamente divulgadas. Utilizar a figura de mérito do capítulo anterior também não é adequado para avaliar o quão bem o áudio foi separado. Para isso, existe um método baseado na razão sinal-ruído (SNR).

Apesar de ser uma métrica ingênua para o caso de separação de fontes sonoras por não considerar as especificidades do aparelho auditivo humano, ela é capaz de mensurar a quantidade de interferências que são adicionadas ao sinal separado, podendo ser utilizada em diversos tipos de dados (audio, imagens etc), em diferentes tipos de misturas (instantânea, convolutiva etc) e para qualquer tipo de algoritmo.

É possível modelar o sinal separado como

$$\hat{\mathbf{s}} = \mathbf{s} + \mathbf{e}_i + \mathbf{e}_a + \mathbf{e}_r,$$

onde $\hat{\mathbf{s}}$ é a fonte separada, $\mathbf{s}$ a fonte original, $\mathbf{e}_i$ é a interferência causada por outras fontes, $\mathbf{e}_a$ são os defeitos possivelmente inseridos devido à separação e finalmente $\mathbf{e}_r$ é o ruído presente na mistura. Nota-se a necessidade de que se conheça a fonte separada assim como todas as fontes originais, incluíndo o ruído inserido na mistura.

A partir daí, podem-se definir quatro medidas de qualidade:

6.1.1.1 Razão Fonte-Distorção (SDR, do inglês *Source-to-Distortion ratio*)

Essa figura de mérito dá uma ideia geral da qualidade da separação, e é definida como

$$\mathrm{SDR} = 10 \log \frac{\|\mathbf{s}\|_2^2}{\|\mathbf{e}_i + \mathbf{e}_a + \mathbf{e}_r\|_2^2}. \tag{6.4}$$

6.1.1.2 Razão Fonte-Interferência (SIR, do inglês *Sources-to-Inteferences Ratio*)

Essa medida avalia a qualidade da separação em si, medindo o quanto há das outras fontes inseridas na fonte de interesse, e pode ser calculada como

$$\mathrm{SIR} = 10 \log \frac{\|\mathbf{s}\|_2^2}{\|\mathbf{e}_i\|_2^2}. \tag{6.5}$$

6.1.1.3 Razão Fontes-Artefatos (SAR, do inglês *Sources-to-Artifacts Ratio*)

A razão fontes-artefatos é uma métrica que tenta estimar a quantidade de defeitos que foram inseridos devido ao processo de separação e que não estavam na mistura original. A SAR é calculada como

$$\mathrm{SAR} = 10 \log \frac{\|\mathbf{s} + \mathbf{e}_i + \mathbf{e}_r\|_2^2}{\|\mathbf{e}_a\|_2^2}. \tag{6.6}$$

Vale a pena notar que essas medidas são invariantes ao ganho e à ordenação das fontes sonoras, então cada estimativa de fonte separada é comparada com as demais fontes originais, e aquela que possuir maior SDR é considerada. Essas métricas foram desenvolvidas em [76] e o pacote com as implementações está disponível

publicamente[1].

6.2 Base de dados

Serão utilizadas duas bases de dados. A primeira tem o intuito de avaliar o uso de diferentes algoritmos de detecção de *onsets* e, assim, determinar qual é o melhor para cada tipo de conjunto de instrumentos musicais. Trata-se da mesma base de [70] disponibilizada pelos autores com a ressalva de que havia algumas divergências em suas anotações, cujas correções por sua vez se acham disponibilizadas num *link*[2]. Essa base consiste em arquivos gravados em 44,1 kHz, divididos em quatro grupos de acordo com suas características: com *pitch* não-percussivo, com *pitch* percussivo (piano), sem *pitch* percussivo (tambores) e misturas complexas (músicas polifônicas), totalizando 1151 *onsets*.

Uma segunda base de dados será utilizada para avaliar o uso da NMF na transcrição de instrumentos percussivos. Essa base é disponibilizada publicamente pelo instituto Fraunhover de Tecnologia de Mídia Digital[3] (denominada de IDMT) com intuito de separação e transcrição de três instrumentos percussivos: chimbau (*hi-hat*, em inglês), caixa (*snare drum*, em inglês) e bumbo (*kick drum*, em inglês). O arranjo desses instrumentos em uma bateria pode ser visualizado na Figura 6.2. São 560 arquivos em 44,1 kHz, totalizando 2 horas de áudio. Desses arquivos, há 95 misturas e para cada mistura há 3 arquivos de treino para os instrumentos contidos na mistura, totalizando 258 arquivos para serem utilizados na transcrição. Essas gravações estão dividas em três categorias:

- Instrumento real, mistura acústica (RealDrum);
- Instrumento real, mistura aditiva (WaveDrum);
- Instrumento sintético (bateria eletrônica com distorções ativadas), mistura aditiva (TechnoDrum).

São disponibilizados os *onsets* de cada um dos instrumentos contidos na mistura. Além disso, há 60 áudios polifônicos com mistura aditiva da qual se dispõe de cada um dos instrumentos separados, provendo 180 sinais de referência para os experimentos na separação de fontes.

[1] `http://bass-db.gforge.inria.fr/bss_eval/`
[2] `https://github.com/CPJKU/onset_db`
[3] `http://www.idmt.fraunhofer.de/en/business_units/smt/drums.html`

Figura 6.2: Uma bateria composta por (1) chimbau, (2) caixa e (3) bumbo, além de outros tambores e pratos.

6.3 Resultados

6.3.1 Avaliação das funções de detecção

Primeiramente, será investigado qual é a melhor função de detecção para a transcrição de instrumentos percussivos. Utilizando a base disponibilizada por Juan Pablo Bello (denominada de MULT), foi realizada a etapa de detecção de *onset*. Dentre os algoritmos citados, foram comparados o HFC (Seção 5.4.2.1), o SD-1 (Seção 5.4.2.2, utilizando a norma ℓ_1), o SD-2 (Seção 5.4.2.2, utilizando a norma ℓ_2) e o WPD (Seção 5.4.3.2) [70, 71]. A STFT do sinal foi calculada com quadros de 1024 amostras, passos de 256 amostras, uma DFT de 2048 pontos e utilizando a janela de Hamming.

Como recomendado em [71], não foi utilizada compressão logarítmica [72]. Como pós-processamento, o sinal resultante da função de detecção, d, é normalizado para que tenha média zero e valor máximo absoluto unitário. Para retirar falsos positivos, o sinal foi filtrado por um passa-baixas de primeira ordem

$$H(z) = 1 - 0.4z^{-1}, \tag{6.7}$$

e para a escolha dos picos foi utilizado o filtro de mediana móvel (Equação (5.12)), utilizando $L = 1$, $\beta = 5$ e $\xi = 1$, como recomendado em [74].

O único parâmetro em aberto é a escolha do limiar inicial δ. No entanto, testes preliminares mostraram que o algoritmo é muito sensível a esse parâmetro, cujo problema também foi apontado pelos autores em [70, 71]. Portanto, seguindo as

mesmas recomendações, o limiar foi ajustado de modo que maximizasse a medida F1 para cada áudio. Após a mediana móvel, foram selecionados os máximos locais. A Tabela 6.1 mostra o resultado para 4 diferentes funções de detecção de *onset* utilizando a base MULT.

Comparados aos resultados contidos em [70, 71], notam-se pequenas divergências devido a algumas diferenças no algoritmo de escolha de picos.

De modo geral, podemos observar que os métodos de deteção de *onset* conseguem atingir alto grau de desempenho nessas bases de dados. Além do mais, observa-se que o algoritmo SD-1 obtém os melhores resultados de F1 para cada um das bases de dados, além de ter a vantagem de ser um algoritmo fácil e rápido de se implementar.

	PNP			PP			NPP			MIX		
	P	S	F1	P	S	F1	P	S	F1	P	S	F1
HFC	**0.946**	0.897	0.921	0.926	0.954	0.935	0.965	0.938	0.947	0.902	**0.886**	0.892
SD-1	0.939	**0.948**	**0.944**	**0.979**	0.961	**0.969**	0.996	**0.981**	**0.988**	**0.959**	0.882	**0.918**
SD-2	0.800	**0.948**	0.868	0.899	0.925	0.909	0.968	0.896	0.924	0.860	0.759	0.800
WPD	0.867	0.938	0.901	0.950	**0.963**	0.955	**0.997**	0.975	0.986	0.909	0.872	0.888

Tabela 6.1: Resultados do teste de detecção de *onsets* utilizando a base MULT, mostrando a precisão (P), sensibilidade (S) e a medida F1 (F1) para 4 diferentes funções de detecção. Os melhores resultados de cada coluna estão em negrito.

Uma outra simulação foi gerada com o intuito de descobrir quais foram os valores da média e do desvio padrão de δ dados a classificação do sinal e o algoritmo. Infelizmente, como pode ser observado na Figura 6.3, o valor do δ ótimo para cada par algoritmo-mistura possui um alto desvio padrão, mostrando a sensibilidade do sistema ao ajuste desse parâmetro. Para a função de detecção SD-1, que foi a melhor para os instrumentos percussivos, o valor médio do δ é 0,0567.

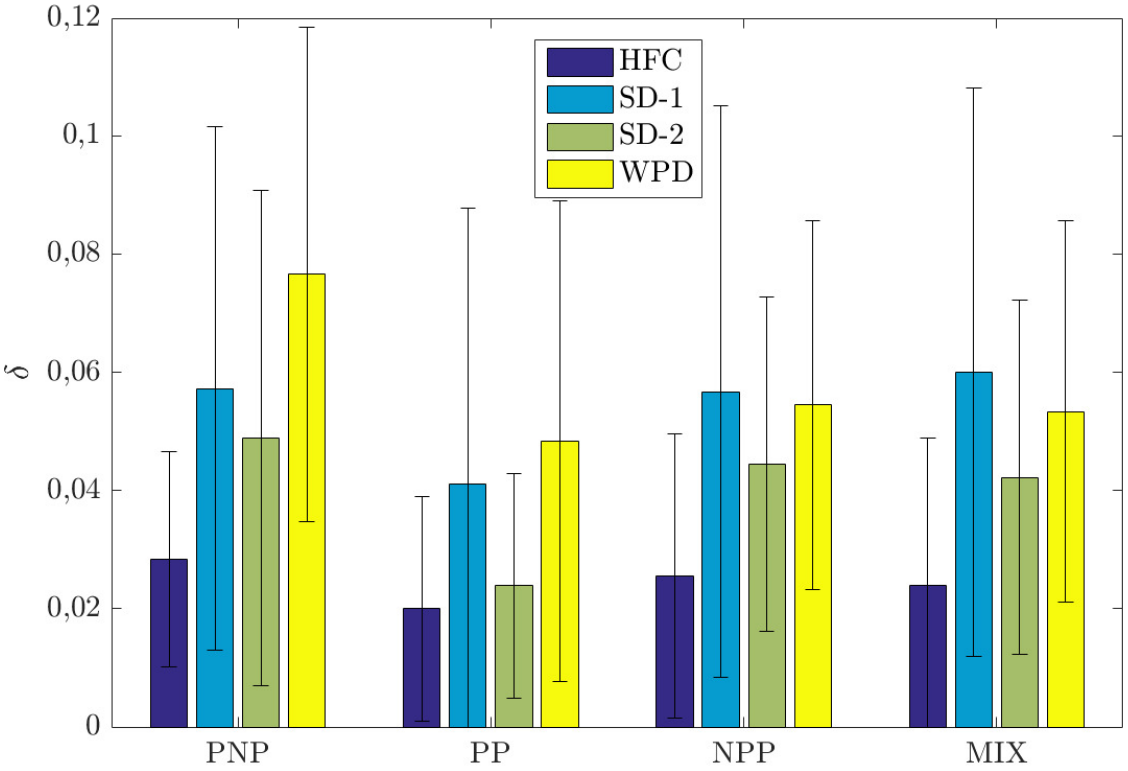


Figura 6.3: Valor médio e desvio padrão de δ para cada função de detecção.

6.3.2 Separação, transcrição e ressíntese

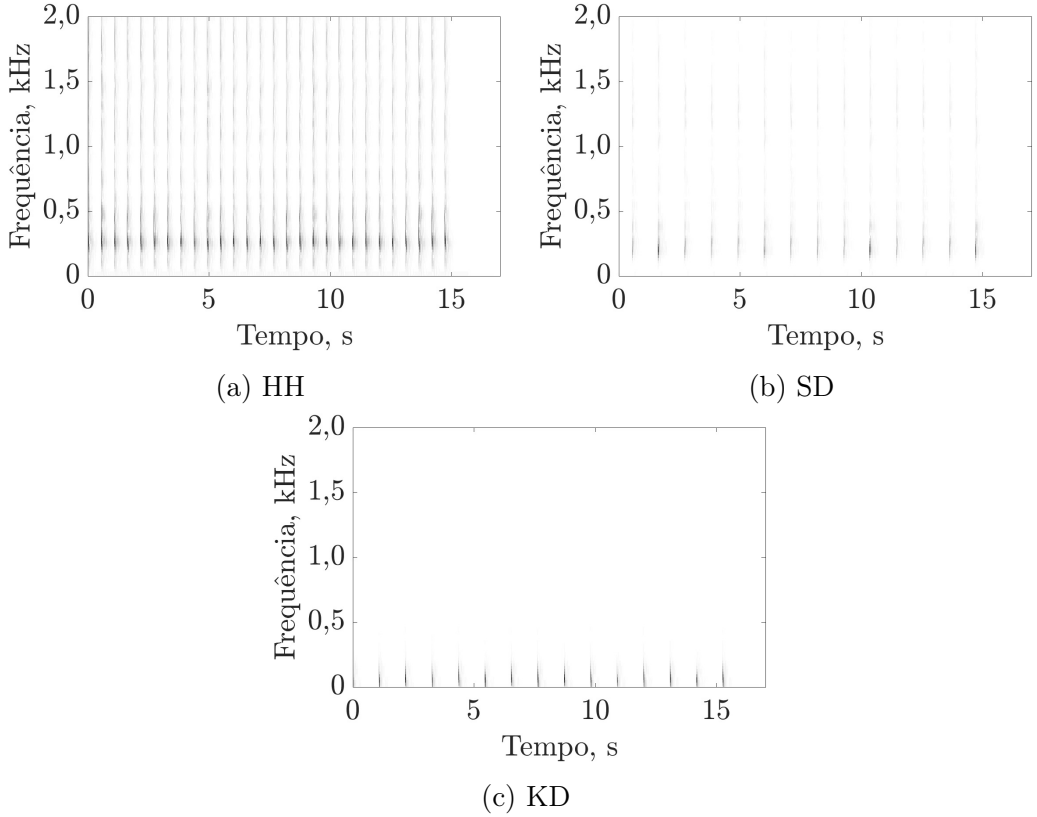
A segunda parte do experimento utiliza a base IDMT para avaliar a separação, transcrição e ressíntese de instrumentos percussivos utilizando gravações acústicas, misturas aditivas e instrumentos sintéticos, utilizando a separação com bases fixas, adaptativas ou semiadaptativas (Seção 5.7) e o algoritmo SD-1. Foi utilizado o método BPP para o cálculo da NMF com critério de suavização na base espectral e critério de esparsidade nas ativações temporais.

Para a reconstrução do instrumento musical r, foi realizada a multiplicação de $\mathbf{V}_r = [\mathbf{W}]_{*,r}[\mathbf{H}]_{r,*}$ e agregada a fase da mistura $\mathbf{V}$ a $\mathbf{V}_r$. Como refinamento final, é utilizada uma filtragem de Wiener como descrito em [64] em cada um dos espectrogramas estimados, utilizando o espectrograma da mistura. Após esse processo foi calculada a STFT inversa, recuperando a série temporal do instrumento separado.

No caso de bases fixas e semiadaptativas, determinar quem são os instrumentos separados é trivial, já que a base $\mathbf{W}$ foi formada utilizando uma ordem pré-estabelecida dos instrumentos. Porém, determinar os instrumentos provenientes da separação cega, isto é, utilizando bases adaptativas, é mais complicado, e a solução encontrada foi aferir o instrumento através da detecção de *onsets*. Foi realizada a transcrição dos instrumentos e foi calculada a medida de F1 para cada uma das três anotações (uma para cada instrumento); aferiu-se o tipo de instrumento pelo melhor

valor de F1 obtido. Esse método não é infalível, pois há casos de separações mal realizadas, e dois sinais separados podem acusar ser do mesmo instrumento. Para esses casos, a solução é realizar a marcação manual.

A Tabela 6.2 mostra os valores de precisão, sensibilidade e medida F1 para os três instrumentos separados (chimbau, bumbo e caixa) de cada uma das três categorias existentes (TechnoDrum, WaveDrum e RealDrum), utilizando os três métodos distintos para a separação com a NMF (bases adaptativas, semiadaptativas e fixas). Para cada instrumento foi maximizado o valor de F1 variando-se o δ. De um modo geral, a detecção de *onsets* do bumbo (KD) e do chimbau (HH) é mais precisa que a da caixa (SD), pois esse último instrumento apresenta muita informação na faixa intermediária de frequências, o que causa maior interferência com os outros dois instrumentos, degradando a qualidade da separação e gerando maior quantidade de falsos positivos. A Figura 6.4 mostra os espectrogramas dos instrumentos da categoria WaveDrum. Observe que o HH e o SD possuem muita interseção em suas raias de frequência, enquanto o KD, instrumento bem mais grave, possui menos interseção quando comparado aos demais instrumentos. Foi observado que para esses casos a separação do SD foi degradada.



(a) HH

(b) SD

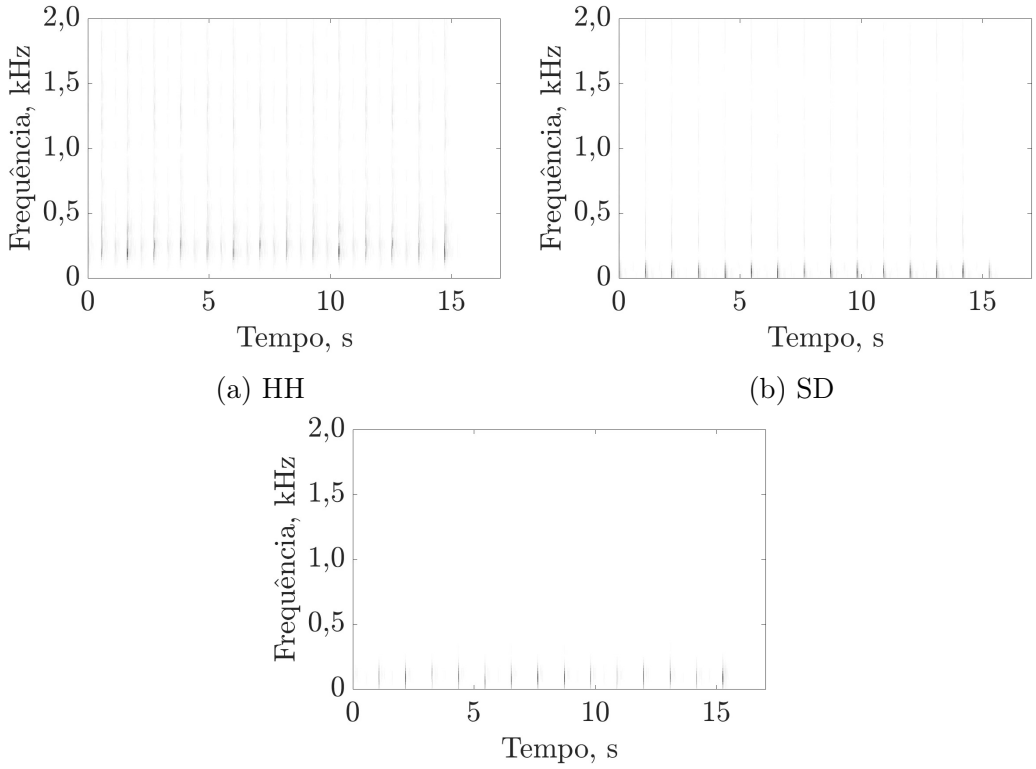
(c) KD

Figura 6.4: Espectrogramas dos instrumentos da categoria WaveDrum. Observe a interseção nas frequências entre o HH e o SD. Por outro lado, o KD tem frequências bem mais baixas, garantindo menor interseção e por consequência facilitando a separação.

As misturas realizadas com sinais sintéticos possuem maior erro de localização de *onsets*, pois as fontes originais estão com diversas distorções eletrônicas (como alta reverberação), poluindo o espectro e dificultando a separação. No caso de altos níveis de reverberação, as repetições ocasionadas pelo eco confundiram o algoritmo de transcrição, fazendo com que fossem detectados *onsets* oriundos dessa distorção.

A Tabela 6.3 foi construída de forma similar à tabela anterior; no entanto, foi escolhido um limiar fixo $\delta = 0{,}0567$. Comparando a Tabela 6.2, gerada para maximizar F1 para cada instrumento, com a Tabela 6.3, feita com um limiar fixo, observamos nitidamente uma piora nos resultados. A medida F1 teve uma piora máxima de 20%. Para alguns casos o limiar ficou grande demais, gerando muitos falsos negativos (piorando a sensibilidade), como por exemplo o caso do KD na categoria TechnoDrum com bases semiadaptativas, em que a piora foi de 27%; em outros, ficou muito baixo, deixando passar mais picos espúrios, gerando mais falsos positivos (piorando a precisão), como pode ser visto para o instrumento SD na categoria RealDrum com bases adaptativas, em que a piora foi de 37%.

As Figuras 6.6, 6.7 e 6.8 mostram, respectivamente, os valores de SIR, SAR e SDR para os três instrumentos separados da categoria WaveDrum utilizando a NMF com bases adaptativas. Analisando essas figuras, podemos observar que o bumbo e a caixa em si possuem altos valores de SIR, SAR e SDR (acima de 3dB), a menos de algumas exceções, que são os casos em que dois instrumentos são tocados ao mesmo tempo. Quando isso ocorre, a NMF não é capaz de determinar que há dois instrumentos distintos, juntando a informação espectral de dois instrumentos em um e deixando a outra componente com a cauda dos espectros. Esse fenômeno pode ser observado na Figura 6.5, que contém o espectrograma dos sinais estimados. Pode-se observar que as ativações temporais do chimbau e do bumbo ocorrem corretamente quando comparadas com as da Figura 6.4. Por outro lado, o espectrograma estimado da caixa não corresponde ao original, pois as ativações temporais ocorrem nos mesmos instantes de tempo que as do HH. Essa interseção faz com que o algoritmo separe os dois instrumentos em um só, deixando o SD com a cauda do KD e algumas informações de mais alta frequência do HH.



(a) HH

(b) SD

(c) KD

Figura 6.5: Espectrogramas dos instrumentos estimados da categoria WaveDrum. Comparados com os espectrogramas originais, Figura 6.4, indicam que o HH e o KD contêm os *onsets* nas posições corretas. No entanto, como as ativações do SD ocorrem sempre nos mesmos instantes de tempo que as do HH, a caixa acabou ficando com a representação da cauda do bumbo e algumas componentes espectrais em mais altas frequências do HH. Quanto ao chimbau, ficou com a junção dos dois espectrogramas (a menos de algumas informações de mais alta frequência).

O que chama mais atenção é o péssimo resultado de SIR encontrado para o HH, indicando que há muita interferência entre fontes; no entanto, verificando-se a Tabela 6.2, a taxa de acerto foi de 85,8% para a separação com base adaptativa e 91,2% para bases fixas, valores esses bem altos. Analisando o áudio desse instrumento, percebe-se que a separação realmente contém muita interferência de outras fontes; mas como a ativação temporal da fonte de interferência ocorre em conjunto com o chimbau, atrapalha a qualidade da separação mas não a detecção de *onsets*.

Apesar de a qualidade da separação ser melhor em testes objetivos para o caso de bases fixas, subjetivamente a separação é pior. O que ocorre é que a base espectral não está corretamente ajustada ao instrumentista, ocasionando uma má representação daquele instrumento dada a mistura, gerando mais transições abruptas devido a perdas de informações sobre a cauda do espectrograma. No entanto, como a base é fixa, há menos *cross-talk* na separação, gerando menos interferências, como pode ser visto na Figura 6.9.

Resumindo, o uso de bases fixas melhora a detecção de *onsets*, ao passo que

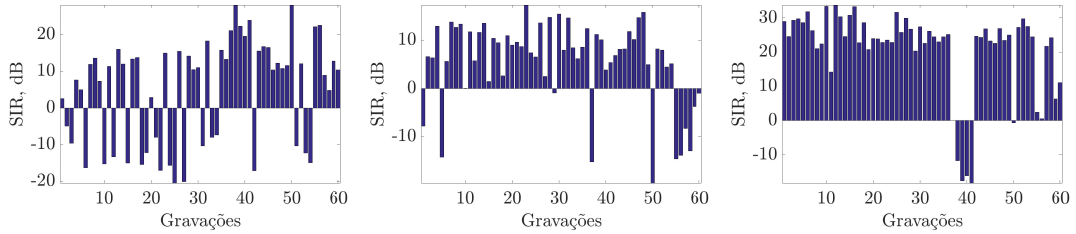
utilizar as bases adaptativas ou semiadaptativas geram instrumentos que soam mais naturais. Globalmente, a tarefa de detecção de *onset* utilizando δ fixo apresentou uma medida F1 de 83,9%. Apesar de a adição de interferência ser alta em alguns casos, não houve adição significativa de artefatos. Além do mais, o bumbo obteve a melhor separação por possuir frequências extremamente baixas, gerando menos interferências entre as outras duas fontes, que possuem maior interseção espectral entre si, e como consequência obteve maiores valores de SDR.

		TechnoDrum			WaveDrum			RealDrum			Todos		
		P	S	F1	P	S	F1	P	S	F1	P	S	F1
HH	adap.	0,883	0,967	0,923	0,852	0,865	0,858	0,973	0,972	0,973	0,876	0,896	0,886
	semi.	0,884	0,967	0,924	0,870	0,905	0,887	0,972	0,971	0,971	0,889	0,924	0,906
	fixa	0,960	0,892	0,925	0,967	0,863	0,912	0,922	0,910	0,916	0,958	0,875	0,914
SD	adap.	0,315	0,945	0,472	0,560	0,925	0,698	0,996	0,996	0,996	0,548	0,940	0,692
	semi.	0,336	0,935	0,495	0,655	0,959	0,778	0,996	0,996	0,996	0,617	0,962	0,752
	fixa	0,788	0,930	0,853	0,979	0,978	0,979	0,996	1,000	0,998	0,953	0,976	0,964
KD	adap.	0,870	0,949	0,908	0,992	0,978	0,985	0,905	0,993	0,947	0,962	0,976	0,969
	semi.	0,867	0,948	0,906	0,992	0,979	0,986	0,905	0,993	0,947	0,964	0,978	0,970
	fixa	0,938	0,993	0,965	0,977	0,980	0,979	0,970	0,980	0,975	0,971	0,981	0,976
HH+SD+KD	adap.	0,652	0,958	0,776	0,808	0,908	0,855	0,961	0,982	0,971	0,805	0,926	0,861
	semi.	0,667	0,956	0,785	0,849	0,936	0,890	0,960	0,981	0,971	0,836	0,946	0,888
	fixa	0,915	0,926	0,921	0,972	0,918	0,944	0,949	0,945	0,947	0,961	0,923	0,942
		0,726	0,947	0,822	0,871	0,921	0,895	0,957	0,969	0,963	0,862	0,932	0,895

Tabela 6.2: Resultados para a base IDMT maximizando o valor de F1. Foi utilizado o algoritmo SD-1 para a deteção de *onsets*. A linha HH+SD+KD não é a média das outras linhas, mas sim os valores de P, S, e F1 recalculados considerando o número total de VP, FN, FP dos três instrumentos. O mesmo é feito na coluna "Todos", nesse caso considerando todos os tipos de instrumentos/misturas.

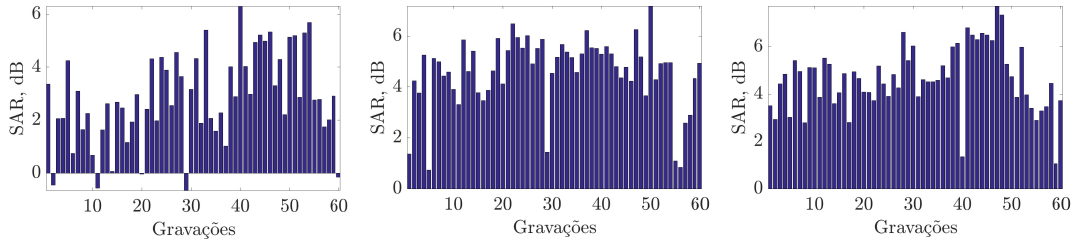
		TechnoDrum			WaveDrum			RealDrum			Todos		
		P	S	F1	P	S	F1	P	S	F1	P	S	F1
HH	adap.	0.800	0.912	0.852	0.863	0.763	0.810	0.915	0.978	0.945	0.863	0.818	0.840
	semi.	0.802	0.914	0.854	0.881	0.825	0.852	0.913	0.978	0.944	0.875	0.862	0.869
	fixa	0.900	0.881	0.890	0.943	0.837	0.887	0.784	0.934	0.853	0.904	0.859	0.881
SD	adap.	0.337	0.688	0.452	0.466	0.927	0.620	0.626	1.000	0.770	0.479	0.919	0.630
	semi.	0.338	0.699	0.456	0.504	0.942	0.657	0.639	1.000	0.780	0.507	0.928	0.656
	fixa	0.569	0.965	0.716	0.901	0.920	0.910	0.985	1.000	0.992	0.847	0.940	0.891
KD	adap.	0.620	0.996	0.764	0.858	0.968	0.909	0.882	0.953	0.916	0.820	0.969	0.888
	semi.	0.596	0.992	0.744	0.847	0.969	0.904	0.882	0.953	0.916	0.811	0.969	0.883
	fixa	0.743	0.996	0.851	0.876	0.974	0.923	0.753	1.000	0.859	0.838	0.981	0.903
HH+SD+KD	adap.	0.649	0.906	0.756	0.745	0.850	0.794	0.827	0.976	0.895	0.745	0.878	0.806
	semi.	0.634	0.902	0.745	0.766	0.887	0.822	0.831	0.976	0.898	0.758	0.903	0.824
	fixa	0.763	0.928	0.838	0.914	0.891	0.902	0.812	0.963	0.881	0.872	0.908	0.889
		0.680	0.912	0.779	0.803	0.876	0.838	0.823	0.972	0.891	0.789	0.896	0.839

Tabela 6.3: Resultados para a base IDMT utilizando $\delta = 0{,}0567$. Foi utilizado o algoritmo SD-1 para a deteção de *onsets*. A linha HH+SD+KD não é a média das outras linhas, mas sim os valores de P, S, e F1 recalculados considerando o número total de VP, FN, FP dos três instrumentos. O mesmo é feito na coluna "Todos", nesse caso considerando todos os tipos de instrumentos/misturas.



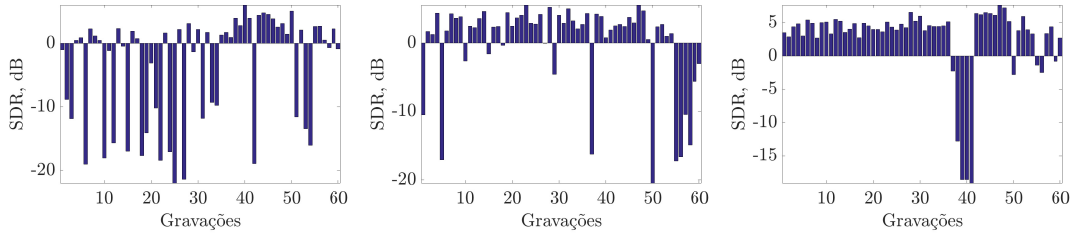
(a) HH (b) SD (c) KD

Figura 6.6: Valores de SIR utilizando bases adaptativas.



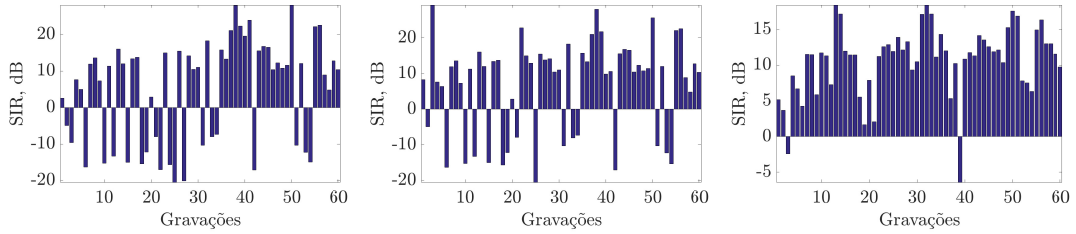
(a) HH (b) SD (c) KD

Figura 6.7: Valores de SAR utilizando bases adaptativas.



(a) HH (b) SD (c) KD

Figura 6.8: Valores de SDR utilizando bases adaptativas.



(a) Adaptativa (b) Semiadaptativa (c) Fixa

Figura 6.9: Valores de SIR do chimbau para cada tipo de separação. Bases fixas garantem que não irá ocorrer *cross-talk*.

Capítulo 7

Conclusão e Trabalhos Futuros

Nesse projeto final foram revisados diversos algoritmos para a fatoração de matrizes não-negativas. Além dos algoritmos, foram mostrados métodos de inicialização, adição de restrições como esparsidade e suavização, critérios de parada, e ainda outras funções-custo que normalmente são utilizadas para melhor se adequar ao problema.

Foi mostrado que há métodos, tais como PG [44], HALS [14], ASGROUP [13] e BPP [48], que possuem melhor convergência com o mesmo custo do que utilizando as famosas regras de atualizações multiplicativas, propostas por Lee e Seung [12] há mais de uma década. Para isso, foram utilizadas diversas bases de dados, mostrando que a NMF pode ser usada em áudio, imagem, texto etc.

Como análise, vimos que para problemas esparsos, o algoritmo *block principal pivoting* [48] se sobressai, ao passo que se a matriz $\mathbf{V}$ se torna mais densa, deve-se preferir o algoritmo baseado em mínimos quadrados hierárquicos [14]. Uma solução para esse problema seria utilizar um sistema híbrido, baseado na estimativa do posto da matriz e no tamanho do problema, e assim determinar o melhor algoritmo a ser utilizado.

Em MIR [70, 71], foi demonstrado como a NMF pode ser inserida na área de transcrição de instrumentos percussivos, detectando *onsets* provenientes de uma separação de um áudio polifônico. Foi desenvolvido um arcabouço genérico para essa tarefa e diversas funções de detecção foram descritas e testadas.

Dentre as funções de detecção, a que se sobressaiu para os instrumentos percussivos utilizava a diferença espectral com norma ℓ_1. Infelizmente, a transcrição demonstrou ter uma forte sensibilidade à escolha do parâmetro δ, o limiar inicial para o uso da mediana móvel.

A segunda etapa consistiu em realizar o mesmo processo de transcrição utilizando uma segunda base de dados, contendo sons percussivos provenientes de gravações acústicas, misturas aditivas e sinais sintéticos. Foi mostrado que utilizando-se um δ médio, obtido no experimento anterior, e utilizando as diferenças espectrais é possí-

vel separar instrumentos percussivos com acurácia acima de 80%. O melhor caso de separação para transcrição se obteve utilizando-se bases fixas; porém, para a reconstrução das fontes sonoras é melhor o uso de bases adaptativas ou semiadaptativas, aumentando a qualidade de separação dos instrumentos. Aliado a isso, é sabido que utilizar a fase da mistura para a reconstrução do instrumento separado não é a melhor estratégia: existem algoritmos específicos para esse caso [77–79] que não foram abordados nesse projeto.

Um modo de incrementar a separação de instrumentos musicais poderia ser utilizar a inicialização em múltiplas camadas e a adaptação dos algoritmos HALS e BPP para funções-custo com base nas divergências de Kullback-Liebler ou Itakura Saito.

Outra possível área de pesquisa seria a adaptação dos algoritmos que realizam a fatoração de matrizes não-negativas citados nesse trabalho a métodos de fatoração mais complexos, como NMFD ou NMF2D, garantindo maior rapidez e acurácia na convergência na separação de sinais complexos.

Referências Bibliográficas

[1] HOYER, P. O. "Non-negative matrix factorization with sparseness constraints", *The Journal of Machine Learning Research*, v. 5, pp. 1457–1469, Dezembro 2004.

[2] KIM, D., SRA, S., DHILLON, I. S. "Fast Projection-Based Methods for the Least Squares Nonnegative Matrix Approximation Problem", *Statistical Analysis and Data Mining*, v. 1, n. 1, pp. 38–51, Fevereiro 2008.

[3] ZDUNEK, R., CICHOCKI, A. "Non-negative Matrix Factorization with Quasi-Newton Optimization". In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L. A., et al. (Eds.), *Artificial Intelligence and Soft Computing – ICAISC 2006*, 1 ed., Springer Berlin Heidelberg, pp. 870–879, Heidelberg, 2006.

[4] L. GORSUCH, R. *Factor Analysis*. 2 ed. New York, Lawrence Erlbaum Associates, 1983.

[5] DONOHO, D., STODDEN, V. "When Does Non-Negative Matrix Factorization Give a Correct Decomposition into Parts?" In: Thrun, S., Saul, L. K., Schölkopf, B. (Eds.), *Advances in Neural Information Processing Systems 16*, MIT Press, pp. 1141–1148, New York, 2004.

[6] PAATERO, P., TAPPER, U. "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values", *Environmetrics*, v. 5, n. 2, pp. 111–126, Junho 1994.

[7] SEUNG, H. S., LEE, D. D. "Learning the parts of objects by non-negative matrix factorization", *Nature*, v. 401, n. 6755, pp. 788–791, Outubro 1999.

[8] XU, W., LIU, X., GONG, Y. "Document clustering based on non-negative matrix factorization". In: *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp. 267–273, Toronto, Agosto 2003. ACM SIGIR.

[9] PAUCA, V. P., PIPER, J., PLEMMONS, R. J. "Nonnegative matrix factorization for spectral data analysis", *Linear Algebra and its Applications*, v. 416, n. 1, pp. 29–47, Julho 2006.

[10] WILSON, K. W., RAJ, B., SMARAGDIS, P., et al. "Speech denoising using nonnegative matrix factorization with priors". In: *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 4029–4032, Las Vegas, Março 2008. IEEE.

[11] SMARAGDIS, P. "Non-negative Matrix Factor Deconvolution; Extraction of Multiple Sound Sources from Monophonic Inputs". In: Puntonet, C. G., Prieto, A. (Eds.), *Independent Component Analysis and Blind Signal Separation*, Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 494–499, Granada, 2004.

[12] LEE, D. D., SEUNG, H. S. "Algorithms for non-negative matrix factorization". In: *Advances in neural information processing systems*, pp. 556–562. Neural Information Processing Systems Foundation, 2001.

[13] KIM, H., PARK, H. "Nonnegative Matrix Factorization Based on Alternating Nonnegativity Constrained Least Squares and Active Set Method", *SIAM Journal on Matrix Analysis and Applications*, v. 30, n. 2, pp. 713–730, Julho 2008.

[14] CICHOCKI, A., ZDUNEK, R., AMARI, S.-I. "Hierarchical ALS Algorithms for Nonnegative Matrix and 3D Tensor Factorization", *Independent Component Analysis and Signal Separation*, v. 4666, n. 1, pp. 169–176, Setembro 2007.

[15] PAULUS, J., VIRTANEN, T. "Drum transcription with non-negative spectrogram factorisation". In: *13th European Signal Processing Conference*, pp. 1–4, Antália, Setembro 2005. IEEE.

[16] PEDERSEN, M. S., LARSEN, J., KJEMS, U., et al. "Convolutive Blind Source Separation Methods". In: Benesty, J., Sondhi, M. M., Huang, Y. (Eds.), *Springer Handbook of Speech Processing*, Springer Press, cap. 52, pp. 1065–1094, Heidelberg, 2008.

[17] HYVÄRINEN, A., KARHUNEN, J., OJA, E. *Independent Component Analysis*, v. 46, *Adaptive and Learning Systems for Signal Processing, Communications, and Control*. New York, John Wiley & Sons, Inc., 2001.

[18] JOLLIFFE, I. *Principal Component Analysis*. Springer Series in Statistics. 2 ed. Chichester, Springer-Verlag, 2002.

[19] PEARSON, K. "LIII. On lines and planes of closest fit to systems of points in space", *Philosophical Magazine Series 6*, v. 2, n. 11, pp. 559–572, Novembro 1901.

[20] HOTELLING, H. "Analysis of a complex of statistical variables into principal components", *Journal of Educational Psychology*, v. 24, n. 6, pp. 417–441, Setembro 1933.

[21] LEE, T.-W., GIROLAMI, M., BELL, A., et al. "A unifying information-theoretic framework for independent component analysis", *Computers & Mathematics with Applications*, v. 39, n. 11, pp. 1–21, Junho 2000.

[22] HYVARINEN, A. "Gaussian moments for noisy independent component analysis", *IEEE Signal Processing Letters*, v. 6, n. 6, pp. 145–147, Junho 1999.

[23] COMON, P. "Independent component analysis, A new concept?" *Signal Processing*, v. 36, n. 3, pp. 287–314, Abril 1994.

[24] HYVÄRINEN, A., OJA, E. "Independent component analysis: algorithms and applications", *Neural Networks*, v. 13, n. 4-5, pp. 411–430, Junho 2000.

[25] OJA, E. "Applications of Independent Component Analysis". In: Pal, N. R., Kasabov, N., Mudi, R. K., et al. (Eds.), *Neural Information Processing*, Springer Berlin Heidelberg, pp. 1044–1051, Heidelberg, 2004.

[26] BOFILL, P., ZIBULEVSKY, M. "Underdetermined blind source separation using sparse representations", *Signal Processing*, v. 81, n. 11, pp. 2353–2362, Novembro 2001.

[27] PLUMBLEY, M. "Algorithms for nonnegative independent component analysis", *IEEE Transactions on Neural Networks*, v. 14, n. 3, pp. 534–543, Maio 2003.

[28] JETER, M., PYE, W. "A note on nonnegative rank factorizations", *Linear Algebra and its Applications*, v. 38, pp. 171–173, Junho 1981.

[29] CHEN, J.-C. "The nonnegative rank factorizations of nonnegative matrices", *Linear Algebra and its Applications*, v. 62, pp. 207–217, Novembro 1984.

[30] BARMAN, P. C., IQBAL, N., LEE, S.-Y. "Non-negative Matrix Factorization Based Text Mining: Feature Extraction and Classification". In: King, I., Wang, J., Chan, L.-W., et al. (Eds.), *Neural Information Processing*, Springer Berlin Heidelberg, pp. 703–712, Heidelberg, 2006.

[31] PAUCA, V. P., PIPER, J., PLEMMONS, R. J. "Nonnegative matrix factorization for spectral data analysis", *Linear Algebra and its Applications*, v. 416, n. 1, pp. 29–47, Julho 2006.

[32] MONGA, V., MIHCAK, M. K. "Robust and Secure Image Hashing via Non-Negative Matrix Factorizations", *IEEE Transactions on Information Forensics and Security*, v. 2, n. 3, pp. 376–390, Setembro 2007.

[33] BRUNET, J.-P., TAMAYO, P., GOLUB, T. R., et al. "Metagenes and molecular pattern discovery using matrix factorization", *Proceedings of the National Academy of Sciences*, v. 101, n. 12, pp. 4164–4169, Março 2004.

[34] GAO, Y., CHURCH, G. "Improving molecular cancer class discovery through sparse non-negative matrix factorization", *Bioinformatics*, v. 21, n. 21, pp. 3970–3975, Novembro 2005.

[35] KIM, H., PARK, H. "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis", *Bioinformatics*, v. 23, n. 12, pp. 1495–1502, Junho 2007.

[36] HELÉN, M., VIRTANEN, T. "Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine". In: *Proceedings of the 13th European Signal Processing Conference*, pp. 1–4, Antália, Setembro 2005. EUSIPCO.

[37] VAN BENTHEM, M. H., KEENAN, M. R. "Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems", *Journal of Chemometrics*, v. 18, n. 10, pp. 441–450, Outubro 2004.

[38] CICHOCKI, A., AMARI, S.-I. *Adaptive Blind Signal and Image Processing*. Chichester, John Wiley & Sons, Ltd, 2002.

[39] WILD, S., CURRY, J., DOUGHERTY, A. "Improving non-negative matrix factorizations through structured initialization", *Pattern Recognition*, v. 37, n. 11, pp. 2217–2232, Novembro 2004.

[40] BOUTSIDIS, C., GALLOPOULOS, E. "SVD based initialization: A head start for nonnegative matrix factorization", *Pattern Recognition*, v. 41, n. 4, pp. 1350–1362, Abril 2008.

[41] CICHOCKI, A., AMARI, S.-I., ZDUNEK, R., et al. "Extended SMART Algorithms for Non-negative Matrix Factorization". In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L. A., et al. (Eds.), *Artificial Intelligence and Soft*

Computing – ICAISC 2006, Springer Berlin Heidelberg, pp. 548–562, Heidelberg, 2006.

[42] BERGMANN, S., IHMELS, J., BARKAI, N. "Iterative signature algorithm for the analysis of large-scale gene expression data", *Physical Review E*, v. 67, n. 3, Março 2003.

[43] SMARAGDIS, P., BROWN, J. "Non-negative matrix factorization for polyphonic music transcription". In: *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 177–180, New Paltz, Outubro 2003. IEEE.

[44] LIN, C.-J. C.-B. "Projected gradient methods for nonnegative matrix factorization", *Neural computation*, v. 19, n. 10, pp. 2756–2779, Outubro 2007.

[45] BELLAVIA, S., MACCONI, M., MORINI, B. "An interior point Newton-like method for non-negative least-squares problems with degenerate solution", *Numerical Linear Algebra with Applications*, v. 13, n. 10, pp. 825–846, Dezembro 2006.

[46] ZDUNEK, R. "Spectral Signal Unmixing with Interior-Point Nonnegative Matrix Factorization". In: Mladenov, V., Koprinkova-Hristova, P., Palm, G., et al. (Eds.), *23rd International Conference on Artificial Neural Networks, ICANN 2013*, v. 8131, *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 65–72, Heidelberg, 2012.

[47] KIM, J., PARK, H. "Toward faster nonnegative matrix factorization: A new algorithm and comparisons". In: *Eighth IEEE International Conference on Data Mining (ICDM'08)*, pp. 353–362, Pisa, Dezembro 2008. IEEE.

[48] KIM, J., PARK, H. "Fast Nonnegative Matrix Factorization: An Active-Set-Like Method and Comparisons", *SIAM Journal on Scientific Computing*, v. 33, n. 6, pp. 3261–3281, Janeiro 2011.

[49] KIM, J., HE, Y., PARK, H. "Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework", *Journal of Global Optimization*, v. 58, n. 2, pp. 285–319, Fevereiro 2014.

[50] BOYD, S. P., VANDENBERGHE, L. *Convex Optmization*. Cambridge University Press, 2004.

[51] GONZALEZ, E., ZHANG, Y. *Accelerating the Lee-Seung algorithm for nonnegative matrix factorization*. Relatório técnico, Dept. Comput. & Appl. Math., Rice Univ., 2005.

[52] STRANG, G. *Álgebra Linear e Suas Aplicações*. Cengage Learning, 2010.

[53] KIRKPATRICK, S., GELATT, C. D., VECCHI, M. P. "Optimization by Simulated Annealing", *Science*, v. 220, n. 4598, pp. 671–680, 1983.

[54] CICHOCKI, A., PHAN, A. H. "Fast local algorithms for large scale nonnegative matrix and tensor factorizations", *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, v. 92, n. 3, pp. 708–721, 2009.

[55] BERRY, M. W., BROWNE, M., LANGVILLE, A. N., et al. "Algorithms and applications for approximate nonnegative matrix factorization", *Computational Statistics & Data Analysis*, v. 52, n. 1, pp. 155–173, Junho 2007.

[56] HO, N. D. *Nonnegative matrix factorization algorithms and applications*. Tese de doutorado, Universidade Católica da Lovaina, Lovaina, 2008.

[57] RACZYŃSKI, S., ONO, N. "Multipitch analysis with harmonic nonnegative matrix approximation". In: *8th International Conference on Music Information Retrieval*, pp. 281–386, Viena, September 2007. International Society for Music Information Retrieval.

[58] CICHOCKI, A., ZDUNEK, R., PHAN, A. H., et al. *Nonnegative Matrix and Tensor Factorizations*. Chichester, John Wiley & Sons, Ltd, 2009.

[59] CICHOCKI, A., ZDUNEK, R., AMARI, S.-I. "Csiszar's divergences for nonnegative matrix factorization: Family of new algorithms". In: Rosca, J., Erdogmus, D., Príncipe, J. C., et al. (Eds.), *Independent Component Analysis and Blind Signal Separation*, v. 3889, Springer Berlin Heidelberg, pp. 32–39, Heidelberg, 2006.

[60] DHILLON, I. S., SRA, S. "Generalized nonnegative matrix approximations with Bregman divergences". In: *Advances in neural information processing systems 18*, v. 18, pp. 283–290, Vancouver, 2005. NIPS.

[61] FÉVOTTE, C., CEMGIL, A. T. "Nonnegative matrix factorizations as probabilistic inference in composite models". In: *Proceedings of the 17th European Signal Processing Conference*, pp. 1913–1917, Glasgow, Agosto 2009. IEEE.

[62] FÉVOTTE, C., IDIER, J. "Algorithms for nonnegative matrix factorization with the beta-divergence", *Neural computation*, v. 23, n. 9, pp. 2421–2456, Março 2011.

[63] WANG, Y.-X., ZHANG, Y.-J. "Image inpainting via Weighted Sparse Non-negative Matrix Factorization". In: *2011 18th IEEE International Conference on Image Processing*, pp. 3409–3412, Bruxelas, Setembro 2011. IEEE.

[64] TYGEL, A. F. *Métodos de fatoração de matrizes não-negativas para separação de sinais musicais*. Tese de mestrado, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2009.

[65] RABINER, L. R., JUANG, B. H. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[66] SCHMIDT, M. N., MØRUP, M. "Nonnegative matrix factor 2-D deconvolution for blind single channel source separation". In: Davies, M. E., James, C. J., Abdallah, S. A., et al. (Eds.), *Independent Component Analysis and Blind Signal Separation*, Springer, pp. 700–707, Heidelberg, 2006.

[67] MARIANO ALMEIDA, R. *Separação de Fontes Sonoras por Fatoração Duplamente Deconvolutiva de Matrizes Não-Negativas com uso de Restrições*. Tese de mestrado, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2014.

[68] THOSHKAHNA, B., RAMAKRISHNAN, K. R. "An onset detection algorithm for query by humming (QBH) applications using psychoacoustic knowledge". In: *17th European Signal Processing Conference*, pp. 939–942. European Association for Signal, Agosto 2009.

[69] ZHU, B., GAN, J., CAI, J., et al. "Adaptive onset detection based on instrument recognition". In: *12th International Conference on Signal Processing*, pp. 2416–2421. IEEE, Outubro 2014.

[70] BELLO, J., DAUDET, L., ABDALLAH, S., et al. "A tutorial on onset detection in music signals", *IEEE Transactions on Speech and Audio Processing*, v. 13, n. 5, pp. 1035–1047, Setembro 2005.

[71] DIXON, S. "Onset detection revisited". In: *Proceedings of the International Conference on Digital Audio Effects*, pp. 133–137, Montreal, Setembro 2006. Music Technology Group of the Schulich School of Music, McGill University.

[72] KLAPURI, A. "Sound onset detection by applying psychoacoustic knowledge". In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*, v. 6, pp. 3089–3092, Phoenix, Março 1999. IEEE.

[73] KLAPURI, A. *Automatic Transcription of Music*. Tese de mestrado, Universidade de Tecnologia de Tampere, Tampere, 1997.

[74] MASRI, P. *Computer modelling of sound for transformation and synthesis of musical signals*. Tese de doutorado, Universidade de Bristol, Bristol, 1996.

[75] DITTMAR, C., GÄRTNER, D., FRAUNHOFER, I., et al. "Real-Time Transcription and Separation of Drum Recordings Based on NMF Decomposition". In: *Proceedings of the 17th International Conference on Digital Audio Effects*, v. Setembro, pp. 187–194, Erlangen, Setembro 2014. Fraunhofer IIS and Friedrich-Alexander-Universität Erlangen-Nürnberg.

[76] VINCENT, E., JAFARI, M., ABDALLAH, S., et al. *Blind Audio Source Separation*. Relatório técnico, Queen Mary Universidade de Londres, Londres, 2005.

[77] GRIFFIN, D., JAE LIM. "Signal estimation from modified short-time Fourier transform", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 32, n. 2, pp. 236–243, Abril 1984.

[78] ZHU, X., BEAUREGARD, G. T., WYSE, L. L. "Real-Time Signal Estimation From Modified Short-Time Fourier Transform Magnitude Spectra", *IEEE Transactions on Audio, Speech and Language Processing*, v. 15, n. 5, pp. 1645–1653, Julho 2007.

[79] GUNAWAN, D., SEN, D. "Music source separation synthesis using Multiple Input Spectrogram inversion". In: *2009 IEEE International Workshop on Multimedia Signal Processing*, pp. 1–5, Rio de Janeiro, Outubro 2009. IEEE.

[80] PETERSEN, K. B., PEDERSEN, M. S. M. S. "The Matrix Cookbook". 2012.

Apêndice A

Cálculo Matricial

A.1 Propriedades básicas do Traço

$$\mathrm{Tr}(\mathbf{A}) = \sum_i [\mathbf{A}]_{i,i} \tag{A.1}$$

$$\mathrm{Tr}(\mathbf{A}) = \sum_i \lambda_i, \quad \boldsymbol{\lambda} = \mathrm{eig}(\mathbf{A}) \tag{A.2}$$

$$\mathrm{Tr}(\mathbf{A}) = \mathrm{Tr}(\mathbf{A}^T) \tag{A.3}$$

$$\mathrm{Tr}(\mathbf{AB}) = \mathrm{Tr}(\mathbf{BA}) \tag{A.4}$$

$$\mathrm{Tr}(\mathbf{A} + \mathbf{B}) = \mathrm{Tr}(\mathbf{A}) + \mathrm{Tr}(B) \tag{A.5}$$

$$\mathrm{Tr}(\mathbf{ABC}) = \mathrm{Tr}(\mathbf{BCA}) = \mathrm{Tr}(\mathbf{CAB}) \tag{A.6}$$

$$\mathbf{a}^T\mathbf{a} = \mathrm{Tr}(\mathbf{a}\mathbf{a}^T) \tag{A.7}$$

A.2 Derivadas

Esta subseção irá cobrir um grande número de expressões envolvendo diferenciações com respeito a uma matrix $\mathbf{X}$ arbitrária. Vale a pena ressaltar que não é assumida nenhuma propriedade especial, isto é, colunas linearmente independentes ou matrizes positivas semidefinidas. Caso contrário, será prontamente mencionado ao longo do texto.

A.2.1 Derivadas de matrizes, vetores e escalares

$$\frac{\partial \mathbf{x}^T\mathbf{a}}{\partial \mathbf{x}} = \frac{\partial \mathbf{a}^T\mathbf{x}}{\partial \mathbf{x}} = \mathbf{a} \tag{A.8}$$

$$\frac{\mathbf{a}^T\mathbf{X}\mathbf{b}}{\partial \mathbf{X}} = \mathbf{a}\mathbf{b}^T \tag{A.9}$$

$$\frac{\mathbf{a}^T\mathbf{X}^T\mathbf{b}}{\partial \mathbf{X}} = \mathbf{b}\mathbf{a}^T \tag{A.10}$$

$$\frac{\partial \mathbf{a}^T\mathbf{X}\mathbf{a}}{\partial \mathbf{X}} = \frac{\partial \mathbf{a}^T\mathbf{X}^T\mathbf{a}}{\partial \mathbf{X}} = \mathbf{a}\mathbf{a}^T \tag{A.11}$$

A.2.2 Derivada do Traço

Assuma que existe uma função $F(\mathbf{X})$ diferenciável para cada elemento de $\mathbf{X}$. Sabe-se que

$$\frac{\partial \operatorname{Tr}(F(\mathbf{X}))}{\partial \mathbf{X}} = f(\mathbf{X})^T,$$

onde $f(\cdot)$ é a derivada escalar de $F(\cdot)$.

Então:

$$\frac{\partial}{\partial \mathbf{X}} \operatorname{Tr}(\mathbf{X}) = \mathbf{I} \tag{A.12}$$

$$\frac{\partial}{\partial \mathbf{X}} \operatorname{Tr}(\mathbf{X}\mathbf{A}) = \mathbf{A}^T \tag{A.13}$$

$$\frac{\partial}{\partial \mathbf{X}} \operatorname{Tr}(\mathbf{A}\mathbf{X}^T\mathbf{B}) = \mathbf{B}\mathbf{A} \tag{A.14}$$

$$\frac{\partial}{\partial \mathbf{X}} \operatorname{Tr}(\mathbf{X}^T\mathbf{A}) = \mathbf{A} \tag{A.15}$$

$$\frac{\partial}{\partial \mathbf{X}} \operatorname{Tr}(\mathbf{A}\mathbf{X}^T) = \mathbf{A} \tag{A.16}$$

$$\frac{\partial}{\partial \mathbf{X}} \operatorname{Tr}(\mathbf{A} \otimes \mathbf{X}) = \operatorname{Tr}(\mathbf{A})\mathbf{I} \tag{A.17}$$

Para mais derivadas, incluíndo de determinantes, inversas etc., recomenda-se a leitura de [80].

Apêndice B

Condições de Karush-Kuhn-Tucker (KKT)

As condições KKT, uma generalização dos multiplicadores de Lagrange, são as condições necessárias para que uma solução de um problema de programação não-linear seja ótima. Caso o problema não seja convexo, um ponto da KKT pode ser um mínimo global, um mínimo local ou um ponto de sela. Então dado um problema de otimização

$$\begin{aligned} &\text{minimizar} && f_0(\mathbf{x}) \\ &\text{sujeito a} && f_i(\mathbf{x}) \geq b_i, \quad i = 1, \ldots, m, \\ &&& h_j(\mathbf{x}) = 0, \quad j = 1, \ldots, p, \end{aligned} \tag{B.1}$$

onde o vetor $\mathbf{x} = (x_1, \ldots, x_n)$ é a variável a ser otimizada do problema, a função $f_0 : \mathbb{R}^n \to \mathbb{R}$ é a função objetivo, as funções $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$, são as restrições de desigualdade, sendo $b_1, \ldots, b_m$ os limites, e as funções $h_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, p$, são as restrições de igualdade. Dizemos que $\mathbf{x}^*$ é um ponto ótimo se todos os pontos satisfazem as restrições: dado um $\mathbf{z}$ na vizinhança com $f_1(\mathbf{z}) \leq b_1, \ldots, f_m(\mathbf{z}) \leq b_m$, $h_1(\mathbf{z}) = 0, \ldots, h_p(\mathbf{z}) = 0$, temos que $f_0(\mathbf{z}) \geq f_0(\mathbf{x}^*)$. Assumindo que as funções $f_0, \ldots, f_m, h_1, \ldots, h_p$ são diferenciáveis e se $\mathbf{x}^*$ é um mínimo local, então existe um $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ para o qual

$$\nabla f_0(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i^* f_i(\mathbf{x}^*) + \sum_{j=1}^{p} \nu_i^* \nabla h_i(\mathbf{x}^*) = 0 \tag{B.2}$$

$$f_i(\mathbf{x}^*) \leq 0, \quad i = 1, \ldots, m \tag{B.3}$$

$$h_j(\mathbf{x}^*) = 0, \quad j = 1, \ldots, p \tag{B.4}$$

$$\lambda_i^* \geq 0, \quad i = 1, \ldots, m \tag{B.5}$$

$$\lambda_i^* f_i(\mathbf{x}^*) = 0, \quad i = 1, \ldots, m. \tag{B.6}$$

Em outras palavras, para qualquer problema de otimização com funções diferenciáveis, $\mathbf{x}^*$ será ótimo se satisfizer as condições KKT.

Essas condições têm um papel importante na área de otimização e, como será mostrado, são utilizadas como critério de convergência para os algoritmos da NMF, além de o problema poder ser resolvido dessa forma, como mostrado em [46]. Para informações mais detalhadas, recomenda-se a leitura de [50].