



Rodrigo Costa Fernandes

**Modelagem Esparsa e Supertraços para
Deconvolução e Inversão Sísmicas**

Tese de Doutorado

Tese apresentada como requisito parcial para obtenção do grau de Doutor pelo Programa de Pós-graduação em Informática da PUC-Rio.

Orientador : Prof. Marcelo Gattass
Co-orientador: Prof. Hélio Côrtes Vieira Lopes

Rio de Janeiro
Setembro de 2019



Rodrigo Costa Fernandes

**Modelagem Esparsa e Supertraços para
Deconvolução e Inversão Sísmicas**

Tese apresentada como requisito parcial para obtenção do grau de Doutor pelo Programa de Pós-graduação em Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo.

Prof. Marcelo Gattass

Orientador

Departamento de Informática – PUC-Rio

Prof. Hélio Côrtes Vieira Lopes

Co-orientador

Departamento de Informática – PUC-Rio

Prof.^a Marley Maria Bernardes Rebuszi Vellasco

Departamento de Engenharia Elétrica – PUC-Rio

Prof. Sinesio Pesco

Departamento de Matemática – PUC-Rio

Dr. Marcos de Carvalho Machado

Tecnologia Geofísica – Petrobras

Dr. David Sotelo Pinheiro da Silva

Pesquisa Operacional e Ciência de Dados – Petrobras

Dr. Adelson Santos de Oliveira

Tecnologia Geofísica – Petrobras

Rio de Janeiro, 27 de Setembro de 2019

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Rodrigo Costa Fernandes

Graduou-se em Ciência da Computação pela Universidade Federal de Pernambuco. Fez mestrado no Departamento de Informática da PUC-Rio, especializando-se na área de computação gráfica, processamento de imagens, geometria computacional e aprendizado de máquina. Desde 2003, é analista de sistemas na Petróleo Brasileiro S.A. – Petrobras, onde atua em desenvolvimento geocientífico.

Ficha Catalográfica

Fernandes, Rodrigo Costa

Modelagem Esparsa e Supertraços para Deconvolução e Inversão Sísmicas / Rodrigo Costa Fernandes; orientador: Marcelo Gattass; co-orientador: Hélio Côrtes Vieira Lopes. – Rio de Janeiro: PUC-Rio, Departamento de Informática, 2019.

v., 130 f.: il. color. ; 30 cm

Tese (doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui referências bibliográficas.

1. Informática – Teses. 2. Modelagem Esparsa. 3. Deconvolução. 4. Compressão. 5. Agrupamento. 6. Inversão. I. Gattass, Marcelo. II. Côrtes Vieira Lopes, Hélio. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. IV. Título.

Aos meus pais Cícero, Maria do Socorro (*in memoriam*) e Maria Izabel.
À minha esposa Yrama.
Aos meus irmãos Diego e Cícero.
Aos meus filhos Ylana e Cael.

Agradecimentos

Agradecimentos especiais aos meus orientadores, Prof. Marcelo Gattass e Prof. Hélio Lopes. Duas pessoas sem as quais este trabalho não teria sido possível. Professores, pesquisadores, orientadores e, principalmente, seres humanos de cujo entusiasmo e energia me alimentei durante o desenrolar do percurso resultante nesta tese.

Agradeço a Yrama, minha esposa, pelos conselhos, pela paciência, pelo apoio e pela crítica construtiva.

Agradeço aos colegas da PUC-Rio, em especial a Axelle Pochet e Lorenzo Ridolfi. Este último, fornecedor de sementes que acabam germinando e gerando mais Ciência como fruto por onde passa.

Agradeço aos meus colegas da Petrobras, da Tecnologia Geofísica, em especial a Alexandre Sardinha, Daniel de Paula, Tiago Nunes, Ana Paula Müller e Marcos Machado. Em pontos específicos neste caminhar, vocês tiveram valiosa participação.

Especiais agradecimentos ao caro colega Adelson Oliveira e ao caro amigo David Sotelo. Vosso conhecimento, competência técnica, paciência e humanidade me serviram como estímulo e inspiração.

Emocionado agradeço aos meus pais. Sem seu incentivo, seu apoio e sua luta, eu não estaria escrevendo mais um capítulo da **nossa** história. Foram e são tantos os desafios, mas são também valiosas **nossas** conquistas.

Dentre tantas pessoas com as quais partilhei deste período de turbulento caminhar, estas aqui citadas estão na categoria, segundo Eduardo Galeano, de fogos que incendeiam. Vocês me ajudaram muito a manter a firmeza e a esperança.

Agradeço à Petrobras pelo apoio, nas pessoas dos gerentes Neiva Zago e Luiz Henrique Amaral.

Agradeço à PUC-Rio e seu corpo docente pela oportunidade concedida.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. Bolsa de isenção de taxas por desempenho acadêmico.

Resumo

Fernandes, Rodrigo Costa; Gattass, Marcelo; Côrtes Vieira Lopes, Hélio. **Modelagem Esparsa e Supertraços para Deconvolução e Inversão Sísmicas**. Rio de Janeiro, 2019. 130p. Tese de Doutorado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Dados de amplitude sísmica compõem o conjunto de insumos do trabalho de interpretação geofísica. À medida que a qualidade dos sensores sísmicos evoluem, há aumento importante tanto na resolução quanto no espaço ocupado para armazenamento. Neste contexto, as tarefas de deconvolução e inversão sísmicas se tornam mais custosas, em tempo de processamento ou em espaço ocupado, em memória principal ou secundária. Partindo do pressuposto de que é possível assumir, por aproximação, que traços de amplitudes sísmicas são o resultado da fusão entre um conteúdo oscilatório – um pulso gerado por um tipo de explosão, em caso de aquisição marítima – e a presença esparsa de contrastes de impedância e variação de densidade de rocha, pretende-se, neste trabalho, apresentar contribuições quanto à forma de realização de duas atividades em interpretação geofísica: a deconvolução e a inversão de refletividades em picos esparsos. Tomando como inspiração trabalhos em compressão volumétrica 3D e 4D, modelagem esparsa, otimização em geofísica, segmentação de imagens e visualização científica, apresenta-se, nesta tese, um conjunto de métodos que buscam estruturas fundamentais e geradoras das amplitudes: (i) uma abordagem para segmentação e seleção de traços sísmicos como representantes de todo o dado, (ii) uma abordagem para separação de amplitudes em ondaleta e picos esparsos de refletividade via deconvolução e (iii) uma outra para confecção de um operador linear – um dicionário – capaz de representar, parcial e aproximadamente, variações no conteúdo oscilatório – emulando alguns efeitos do subsolo –, com o qual é possível realizar uma inversão de refletividades. Por fim, apresenta-se um conjunto de resultados demonstrando a viabilidade das abordagens, o ganho eventual se aplicadas – incluindo a possibilidade de compressão – e a abertura de oportunidades de trabalhos futuros mesclando geofísica e computação.

Palavras-chave

Modelagem Esparsa; Deconvolução; Compressão; Agrupamento; Inversão.

Abstract

Fernandes, Rodrigo Costa; Gattass, Marcelo (Advisor); Côrtes Vieira Lopes, Hélio (Co-Advisor). **Sparse Modeling and Supertraces for Seismic Deconvolution and Inversion**. Rio de Janeiro, 2019. 130p. Tese de doutorado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Seismic amplitude data are part of the input in a geophysical interpretation pipeline. As seismic sensors evolve, resolution and occupied storage space grows. In this context, tasks as seismic deconvolution and inversion become more expensive, in processing time or in – main or secondary – memory. Assuming that, approximately, seismic amplitude traces result from a fusion between an oscillatory content – a pulse generated by a kind of explosion, in the case of marine acquisition – and the sparse presence of impedance contrasts and rock density variation, this work presents contributions to the way of doing two geophysical interpretation activities: deconvolution and inversion, both targeting sparse-spike reflectivity extraction. Inspired by works in 3D and 4D volumetric compression, sparse modeling, optimization applied to geophysics, image segmentation and scientific visualization, this thesis presents a set of methods that try to fetch fundamental features that generate amplitude data: (i) an approach for seismic traces segmentation and selection, electing them as representatives of the whole data, (ii) an enhancement of an approach for separation of amplitudes into wavelet and sparse-spike reflectivities via deconvolution, and (iii) a way to generate a linear operator – a dictionary – partially and approximately capable of representing variations on the wavelet shape, emulating some effects of the subsoil, from which is possible to accomplish a reflectivity inversion. By the end, it is presented a set of results that demonstrate the viability of such approaches, the possible gain when they are applied – including compression – and some opportunities for future works mixing geophysics and computer science.

Keywords

Sparse Modeling; Deconvolution; Compression; Clustering; Inversion.

Sumário

| | | |
|-----|--|------------|
| 1 | Introdução | 21 |
| 1.1 | Detalhamento do Tema | 22 |
| 1.2 | Detalhamento do Problema de Pesquisa | 23 |
| 1.3 | Objetivo Geral e Objetivos Específicos | 25 |
| 1.4 | Uma Visão sobre os Resultados | 26 |
| 1.5 | Contribuições da Tese | 26 |
| 1.6 | Limitações do Trabalho | 27 |
| 1.7 | Metodologia Utilizada | 28 |
| 1.8 | Justificativa | 28 |
| 1.9 | Organização da Tese | 29 |
| 2 | Conceitos e Trabalhos Relacionados | 31 |
| 2.1 | Conceitos, Problemas e Algoritmos | 31 |
| 2.2 | Amostragem Comprimida - <i>Compressed Sampling</i> | 48 |
| 2.3 | Trabalhos Relacionados | 49 |
| 3 | Deconvolução Sísmica | 53 |
| 3.1 | Relação com codificação esparsa | 57 |
| 3.2 | Uma solução proposta: Lobbes | 58 |
| 4 | Agrupamento | 63 |
| 4.1 | Introdução | 63 |
| 4.2 | Agrupamento como forma de seleção | 64 |
| 4.3 | Agrupamento em supertraços | 66 |
| 4.4 | SLICT: uma proposta para agrupamento em supertraços | 67 |
| 5 | Aprendizado e Extensão de Dicionário para Inversão de Refletividades | 76 |
| 5.1 | Introdução | 76 |
| 5.2 | Aprendizado de um dicionário inicial | 76 |
| 5.3 | Aplicação de regularização aos átomos do dicionário | 82 |
| 5.4 | Aumento artificial do dicionário via filtro da terra | 87 |
| 6 | Resultados | 90 |
| 6.1 | Resultados iniciais: deconvolução por picos esparsos | 90 |
| 6.2 | Resultados em dado real público | 96 |
| 7 | Conclusões e Trabalhos Futuros | 113 |
| 7.1 | Conclusões | 113 |
| 7.2 | Trabalhos Futuros | 114 |
| | Referências bibliográficas | 117 |
| A | Lobbes | 124 |
| B | Trechos de Código Fonte | 130 |

Lista de Figuras

- Figura 1.1 Ilustração dos elementos algébricos do problema de representação esparsa. \mathbf{D} é a matriz para se chegar no vetor esparsa $\tilde{\mathbf{x}}$ tomando por base o vetor denso \mathbf{y} . 24
- Figura 2.1 Exibição de traços de amplitude sísmica do dado F3R. *Inline* 290, e *crosslines* 100 a 190 (saltando de 10 em 10). 32
- Figura 2.2 Diagrama de um levantamento sísmico marinho usando cabos com hidrofones. As linhas tracejadas coloridas representam os caminhos mais curtos percorridos por uma onda sísmica ao ser refletida por camadas de coeficientes de reflexão distintos, captada, em diferentes tempos e posições, por cada hidrofone. Cenário se refere ao disparo de um único tiro de uma pistola de ar. Derivado de trabalho disponível na Wikipedia: By Nwhit - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=18527767>. 33
- Figura 2.3 Navio de levantamento sísmico. Navio Ramform Sovereign, da PGS, na Bacia de Santos. Adaptado do boletim 2.2012 da Sociedade Brasileira de Geofísica. Disponível: <https://sbgf.org.br/noticias/images/Boletim-2-2012.pdf>. Acesso: 19/12/2018. 34
- Figura 2.4 Ilustração da estimativa de solução do problema usando 2-12 (esquerda), usando o Lasso (centro) e usando *ridge*. Supondo $\mathbf{x} = (x_1, x_2)$, as áreas coloridas (em verde ou azul) se referem a $\|\mathbf{x}\|_0 < t$, $\|\mathbf{x}\|_1 < t$ e $\|\mathbf{x}\|_2^2 < t^2$, sendo t uma constante. O ponto $\tilde{\mathbf{x}}$ se refere ao ótimo com restrições, e o ponto $\hat{\mathbf{x}}$ se refere ao ótimo sem as restrições (por mínimos quadrados). Em vermelho, contornos da função de resíduo dos mínimos quadrados. Adaptado a partir de Hastie, Tibshirani e Wainwright (2015), Cap. 2, Figura 2.2. 42
- Figura 3.1 Ondaleta Ricker com frequência dominante de 30Hz, e ângulo (fase) de 30 graus. 53
- Figura 3.2 De cima para baixo, um perfil de refletividades, a convolução com a ondaleta da Figura 3.1 e adição final de ruído gaussiano (com valor absoluto máximo de 0,02). 54
- Figura 3.3 Velocidades de propagação de ondas sísmicas do modelo Marmousi2. Eixo horizontal: posição em metros. Eixo vertical: profundidade em metros. Velocidades em metros/segundo com escala exibida à direita. 55
- Figura 3.4 Densidades do modelo Marmousi2. Eixo horizontal: posição em metros. Eixo vertical: profundidade em metros. Densidade em gramas/mililitro com escala exibida à direita. 56

- Figura 3.5 Refletividade calculada a partir de densidade e velocidade do modelo Marmousi2. Eixo horizontal: posição em metros. Eixo vertical: profundidade em metros. Escala exibida à direita. 56
- Figura 3.6 Amplitudes sísmicas partindo do conteúdo de refletividades do modelo Marmousi2 da Figura 3.5. Cada traço sísmico é resultado da convolução de uma ondaleta Ricker (frequência dominante de 30Hz, ângulo de 30 graus) com a refletividade, e adiciona-se ruído. Eixo horizontal: posição em metros. Eixo vertical: profundidade em metros. 57
- Figura 3.7 Ilustração de uma matriz de Toeplitz. 57
- Figura 4.1 Resultado da aplicação do algoritmo *K-Means* sobre os traços do dado F3 recortado, pedindo de 4 a 1024 grupos de traços. Em cada imagem, eixo horizontal com *crosslines* e eixo vertical com *inlines*. Vista em planta do agrupamento dos traços. 65
- Figura 4.2 Exemplo de aplicação do algoritmo SLIC sobre imagem colorida. Bordas dos superpixels (à esq.) e superpixels (à dir.). 67
- Figura 4.3 Ilustração da janela de limite para cálculo de similaridade. O traço central (t_0 , em vermelho) é a referência contra o qual os traços dentro do limite exterior (em azul) serão comparados. Em verde o limite corresponde ao número de traços pretendido para supertraço, e em azul o limite máximo da busca. Traços t_1 e t_2 nas regiões verde e azul são observados, enquanto o traço t_3 é ignorado. 70
- Figura 4.4 Ilustração do campo escalar usado como base de execução do SLICT. As imagens mostram a mesma informação, mudando somente a escala de cor. A grandeza medida é soma do quadrado da distância euclideana entre traços anterior e posterior, nas duas direções, em cada posição. 72
- Figura 4.5 Ilustração da execução do algoritmo SLICT sobre o dado F3 recortado. Limite de 2000 supertraços. Parâmetro de regularidade = 0,125. Em (a) e (c), blocos com cores distintas. Em (b), somente bordas entre blocos. 73
- Figura 4.6 Ilustração da execução do algoritmo SLICT sobre o dado F3 recortado. Limite de 2000 supertraços. Parâmetro de regularidade = 2,0. Em (a) e (c), blocos com cores distintas. Em (b), somente bordas entre blocos. 73
- Figura 4.7 Ilustração da execução do algoritmo SLICT sobre o dado F3 recortado. Limite de 8000 supertraços. Parâmetro de regularidade = 0,125. Em (a) e (c), blocos com cores distintas. Em (b), somente bordas entre blocos. 74
- Figura 4.8 Ilustração da execução do algoritmo SLICT sobre o dado F3 recortado. Limite de 8000 supertraços. Parâmetro de regularidade = 2,0. Em (a) e (c), blocos com cores distintas. Em (b), somente bordas entre blocos. 74

- Figura 5.1 Resultado inicial da execução de treinamentos de dicionários usando os algoritmos OMP e MOD. Coluna da esquerda, usando parâmetro de regularidade 0,125. Coluna da direita, usando parâmetro de regularidade 2,0. Acima, limite de 2000 supertraços (com 1977 e 1978 selecionados). Abaixo, limite de 8000 supertraços (com 7998 e 8023 selecionados). 80
- Figura 5.2 Resultado da aplicação de poda ao dicionário inicial gerado usando OMP e MOD. Coluna da esquerda, usando parâmetro de regularidade de supertraços com valor igual a 0,125. Coluna da direita, usando parâmetro de regularidade de supertraços com valor igual a 2. Acima, limite de 2000 supertraços (com 1977 e 1978 selecionados). Abaixo, limite de 8000 supertraços (com 7998 e 8023 selecionados). Amostras com distância superior a 100 da diagonal foram eliminadas. Amostras de índices de 50 a 100 distantes da diagonal tiveram atenuação linear até chegar em 0. 81
- Figura 5.3 Visualização da linha de índice 224 de dois dicionários, antes (à esquerda) e depois (ao centro) do recorte e ajuste (normalização), com diferença absoluta mais à direita. Acima, utilizando 2000 supertraços e parâmetro de regularidade com valor 0,125. Abaixo, utilizando 8000 supertraços e parâmetro de regularidade com valor 0,125. Dicionários gerados usando MOD e OMP. 82
- Figura 5.4 Resultado inicial da execução de treinamentos de dicionários usando os algoritmos OMP e SGK com regularização. Coluna da esquerda, usando parâmetro de regularidade de supertraços com valor igual a 0,125. Coluna da direita, usando parâmetro de regularidade de supertraços com valor igual 2. Acima, limite de 2000 supertraços (com 1977 e 1978 selecionados). Abaixo, limite de 8000 supertraços (com 7998 e 8023 selecionados). 84
- Figura 5.5 Resultado da aplicação de poda ao dicionário inicial gerado usando OMP e SGK com regularização. Coluna da esquerda, usando parâmetro de regularidade de supertraços com valor igual a 0,125. Coluna da direita, usando parâmetro de regularidade de supertraços com valor igual a 2. Acima, limite de 2000 supertraços (com 1977 e 1978 selecionados). Abaixo, limite de 8000 supertraços (com 7998 e 58023 selecionados). Amostras com distância superior a 100 da diagonal foram eliminadas. Amostras de índices de 50 a 100 distantes da diagonal tiveram atenuação linear até chegar em 0. 85

- Figura 5.6 Visualização da linha de índice 224 de dois dicionários, antes (à esquerda) e depois (ao centro) do recorte e ajuste (normalização), com diferença absoluta mais à direita. Acima, utilizando 2000 supertraços e parâmetro de regularidade com valor 0,125. Abaixo, utilizando 8000 supertraços e parâmetro de regularidade com valor 0,125. Dicionários gerados usando OMP e SGK com regularização. Observa-se grande diferença em relação ao que mostra a Figura 5.3. 86
- Figura 5.7 Ilustração da matriz de resposta de amplitude segundo o modelo de atenuação e dispersão de Futterman. Da esquerda para a direita, variação da frequência de referência f_r (25 e 35Hz). De cima para baixo, variação do fator de qualidade Q (100 e 60). 88
- Figura 5.8 Dicionário aumentado em 5 vezes usando os seguintes parâmetros para a matriz de resposta de amplitude segundo o modelo de Futterman: ($Q = 100$ e $f_r = 35Hz$), ($Q = 100$ e $f_r = 30Hz$), ($Q = 80$ e $f_r = 35Hz$) e ($Q = 80$ e $f_r = 30Hz$). Dicionário aumentado a partir do extraído com 2000 traços e parâmetro de regularidade de 0,125 como na Figura 5.4. 89
- Figura 6.1 Dado sintético Marmousi2. Acima, conteúdo de velocidade (em m/s). Abaixo, conteúdo de densidade (em g/cm³). 90
- Figura 6.2 Área selecionada do dado Marmousi2. À esquerda, conteúdo de velocidade. À direita, conteúdo de densidade. Eixo horizontal: uma amostra por metro. Eixo vertical: uma amostra a cada 1,25m. 91
- Figura 6.3 Ilustração de um dado de entrada em uma das 10 iterações para extração de ondaleta e refletividade. À esquerda, dado de refletividade sem ruído, resultante da aplicação da fórmula da Equação 2-1. Ao centro, ondaleta usada como gabarito, com frequência dominante 33Hz, fase de 45°, duração de 0,201 s. À direita, resultado da convolução da ondaleta de gabarito com a refletividade e adição de ruído. À esquerda e à direita, imagens mostram 20 traços com 800 amostras. 93
- Figura 6.4 Ilustração da reconstrução dos 200 traços de refletividades. À esquerda, usando ondaleta extraída via OMP (RSR 21,94). Ao centro, usando ondaleta extraída via *Fista TSMF* (RSR 22,089). À direita, usando ondaleta extraída via *Lobbes* (RSR 21,918). 95
- Figura 6.5 Ilustração da diferença da reconstrução dos 200 traços de refletividades. À esquerda, usando ondaleta extraída via OMP (RSR 21,94). Ao centro, usando ondaleta extraída via *Fista TSMF* (RSR 22,089). À direita, usando ondaleta extraída via *Lobbes* (RSR 21,918). As refletividades originais foram escaladas para o valor 1, enquanto as extraídas foram escaladas para -1. Imagens mostram a soma. 95

- Figura 6.6 Ilustração do dado F3R. Acima, exibição da *inline* 200. Abaixo, exibição em 3D com 3 planos paralelos aos eixos do dado. São 588 amostras na *inline*, 939 na *crossline* e 448 amostras na vertical. Cada amostra corresponde a um intervalo de 4ms na vertical e 25m na horizontal. 97
- Figura 6.7 Histogramas a partir do dado de amplitudes F3R. À esquerda, histograma comum com 256 subdivisões. À direita, histograma com escala vertical logarítmica, com 256 subdivisões. Amplitudes normalizadas para o intervalo $[-1, 1]$. 99
- Figura 6.8 Sobreposição de curva normal e de histograma extraídos a partir do resíduo gerado pela representação esparsa usando o dicionário aumentado em 10 vezes, parâmetro de regularização dos supertraços 0,125 e 16000 traços de exemplos, correspondendo ao valor mais distante de zero observado nos experimentos. 104
- Figura 6.9 Exibição de pontos, em três vistas diferentes, que correspondem, em cada traço, ao coeficiente de maior valor absoluto no dado recodificado. Utilizado dicionário de 8000 amostras, parâmetro de regularidade de supertraços 0.125, aumentado em 10 vezes usando filtro da terra. As cores correspondem ao sinal de refletividade (**positivo**, **negativo**). 105
- Figura 6.10 Ilustração do uso dos coeficientes do dicionário ao longo da profundidade usando **2000** traços de entrada para o treinamento. Eixo horizontal: profundidade da amostra. Eixo vertical: contagem dos coeficientes utilizados em cada profundidade. À esquerda, uso do dicionário treinado e aumentado, distinguindo cada trecho do dicionário utilizado. À direita, comparação de uso entre o trecho original (com legenda “0”) e trecho aumentado do dicionário (com legenda “>0”). Duas primeiras linhas: parâmetro de regularidade de supertraços 0,125. Duas últimas linhas: parâmetro de regularidade 2,0. Primeira e terceira linhas: dicionário aumentado para 5 vezes. Segunda e quarta linhas: dicionário aumentado em 10 vezes. 106
- Figura 6.11 Ilustração do uso dos coeficientes do dicionário ao longo da profundidade usando **4000** traços de entrada para o treinamento. Eixo horizontal: profundidade da amostra. Eixo vertical: contagem dos coeficientes utilizados em cada profundidade. À esquerda, uso do dicionário treinado e aumentado, distinguindo cada trecho do dicionário utilizado. À direita, comparação de uso entre o trecho original (com legenda “0”) e trecho aumentado do dicionário (com legenda “>0”). Duas primeiras linhas: parâmetro de regularidade de supertraços 0,125. Duas últimas linhas: parâmetro de regularidade 2,0. Primeira e terceira linhas: dicionário aumentado para 5 vezes. Segunda e quarta linhas: dicionário aumentado em 10 vezes. 107

- Figura 6.12 Ilustração do uso dos coeficientes do dicionário ao longo da profundidade usando **8000** traços de entrada para o treinamento. Eixo horizontal: profundidade da amostra. Eixo vertical: contagem dos coeficientes utilizados em cada profundidade. À esquerda, uso do dicionário treinado e aumentado, distinguindo cada trecho do dicionário utilizado. À direita, comparação de uso entre o trecho original (com legenda “0”) e trecho aumentado do dicionário (com legenda “>0”). Duas primeiras linhas: parâmetro de regularidade de supertraços 0,125. Duas últimas linhas: parâmetro de regularidade 2,0. Primeira e terceira linhas: dicionário aumentado para 5 vezes. Segunda e quarta linhas: dicionário aumentado em 10 vezes. 108
- Figura 6.13 Ilustração do uso dos coeficientes do dicionário ao longo da profundidade usando **16000** traços de entrada para o treinamento. Eixo horizontal: profundidade da amostra. Eixo vertical: contagem dos coeficientes utilizados em cada profundidade. À esquerda, uso do dicionário treinado e aumentado, distinguindo cada trecho do dicionário utilizado. À direita, comparação de uso entre o trecho original (com legenda “0”) e trecho aumentado do dicionário (com legenda “>0”). Duas primeiras linhas: parâmetro de regularidade de supertraços 0,125. Duas últimas linhas: parâmetro de regularidade 2,0. Primeira e terceira linhas: dicionário aumentado para 5 vezes. Segunda e quarta linhas: dicionário aumentado em 10 vezes. 109
- Figura 6.14 Ilustração da inversão da *inline* 284 do dado F3R ao usar dicionários de múltiplos tamanhos treinado usando **2000** amostras. À esquerda, utilizado dicionário treinado e ajustado. Ao centro, utilizado dicionário aumentado para 5 vezes. À direita, utilizado dicionário aumentado para 10 vezes. Primeira linha: parâmetro de regularidade de supertraços 0,125. Segunda linha: parâmetro de regularidade de supertraços 2,0. 110
- Figura 6.15 Ilustração da inversão da *inline* 284 do dado F3R ao usar dicionários de múltiplos tamanhos treinado usando **4000** amostras. À esquerda, utilizado dicionário treinado e ajustado. Ao centro, utilizado dicionário aumentado para 5 vezes. À direita, utilizado dicionário aumentado para 10 vezes. Primeira linha: parâmetro de regularidade de supertraços 0,125. Segunda linha: parâmetro de regularidade de supertraços 2,0. 111
- Figura 6.16 Ilustração da inversão da *inline* 284 do dado F3R ao usar dicionários de múltiplos tamanhos treinado usando **8000** amostras. À esquerda, utilizado dicionário treinado e ajustado. Ao centro, utilizado dicionário aumentado para 5 vezes. À direita, utilizado dicionário aumentado para 10 vezes. Primeira linha: parâmetro de regularidade de supertraços 0,125. Segunda linha: parâmetro de regularidade de supertraços 2,0. 111

Figura 6.17 Ilustração da inversão da *inline* 284 do dado F3R ao usar dicionários de múltiplos tamanhos treinado usando **16000** amostras. À esquerda, utilizado dicionário treinado e ajustado. Ao centro, utilizado dicionário aumentado para 5 vezes. À direita, utilizado dicionário aumentado para 10 vezes. Primeira linha: parâmetro de regularidade de supertraços 0,125. Segunda linha: parâmetro de regularidade de supertraços 2,0. 112

Figura B.1 Código fonte em Python para geração de ondaletas Ricker, convolução com refletividade sintética e exibição gráfica. 130

Lista de Tabelas

- Tabela 5.1 Erro de reconstrução das amostras de entrada do processo de treinamento de dicionário, variando a quantidade aproximada de traços de entrada (2000 e 8000) e o parâmetro de regularidade dos supertraços (0,125 e 2,0). Medidas média (e máxima, em parênteses) percentual. Medidas tomam o valor máximo absoluto do dado de entrada e do erro encontrado. Algoritmos utilizados: OMP e MOD. 80
- Tabela 5.2 Erro de reconstrução das amostras de entrada a partir do uso do dicionário ajustado, variando a quantidade aproximada de traços de entrada (2000 e 8000) e o parâmetro de regularidade dos supertraços (0,125 e 2,0). Medidas média (e máxima, em parênteses) percentual. Medidas tomam o valor máximo absoluto do dado de entrada e do erro encontrado. Observa-se um aumento no erro em todas as medidas. Algoritmos utilizados: OMP e MOD. 81
- Tabela 5.3 Erro de reconstrução das amostras de entrada do processo de treinamento de dicionário, variando a quantidade aproximada de traços de entrada (2000 e 8000) e o parâmetro de regularidade dos supertraços (0,125 e 2,0). Medidas média (e máxima, em parênteses) percentual. Medidas tomam o valor máximo absoluto do dado de entrada e do erro encontrado. Algoritmos utilizados: OMP e SGK com regularização. 85
- Tabela 5.4 Erro de reconstrução das amostras de entrada a partir do uso do dicionário ajustado, variando a quantidade aproximada de traços de entrada (2000 e 8000) e o parâmetro de regularidade dos supertraços (0,125 e 2,0). Medidas média (e máxima, em parênteses) percentual. Medidas tomam o valor máximo absoluto do dado de entrada e do erro encontrado. Observa-se um aumento muito pequeno no erro em todas as medidas, sensivelmente aquém do anteriormente mostrado na Tabela 5.2. Algoritmos utilizados: OMP e SGK com regularização. 86
- Tabela 6.1 Avaliação de qualidade da deconvolução dos traços dados como entrada para o processo de extração de ondaleta e refletividade. MSCRZ: média da similaridade de cosseno das refletividades após a remoção dos zeros. MSCO: média da similaridade do cosseno da ondaleta. RSR: razão sinal-ruído (como na Equação 6-2). Valores arredondados para 3 casas decimais. 94
- Tabela 6.2 Avaliação a partir do índice de qualidade de deconvolução (IQD) como proposto em Fernandes, Lopes e Gattass (2017) e da quantidade de iterações para a convergência dos algoritmos. Valores arredondados para 3 casas decimais. 94

| | | |
|-------------|---|-----|
| Tabela 6.3 | Avaliação de qualidade da reconstrução. Tomando cada uma das matrizes de Toeplitz geradas pelo procedimento da Tabela 6.1 e aplicando unicamente o algoritmo OMP para cálculo da representação esparsa de cada traço de refletividade. RSR: razão sinal-ruído (como na Equação 6-2). Valores arredondados para 3 casas decimais. | 94 |
| Tabela 6.4 | Medidas estatísticas sobre o dado F3R. Exibidas a média, o desvio padrão e a curtose. Valores arredondados para 3 casas decimais. Amplitudes normalizadas para o intervalo $[-1, 1]$. | 99 |
| Tabela 6.5 | Valores utilizados pelo procedimento de distorção de ondaletas via filtro da terra para cada múltiplo do dicionário original. Ampliação para 5 vezes o tamanho original. | 100 |
| Tabela 6.6 | Valores utilizados pelo procedimento de distorção de ondaletas via filtro da terra para cada múltiplo do dicionário original. Ampliação para 10 vezes o tamanho original. | 100 |
| Tabela 6.7 | Resultado numérico para avaliação da reconstrução medindo razão sinal-ruído e pico da razão sinal-ruído. As colunas com rótulo 1x se referem ao uso do dicionário treinado e ajustado. Colunas 5x e 10x se referem ao uso dos dicionários aumentados para 5 e 10 vezes usando deformações provenientes da aplicação do filtro da terra com parâmetros informados nas Tabelas 6.5 e 6.6, respectivamente. | 101 |
| Tabela 6.8 | Avaliação de resíduos quando da utilização de 2000 traços representantes. Em destaque o menor desvio padrão e a menor amplitude. | 102 |
| Tabela 6.9 | Avaliação de resíduos quando da utilização de 4000 traços representantes. Em destaque o menor desvio padrão e a menor amplitude. | 103 |
| Tabela 6.10 | Avaliação de resíduos quando da utilização de 8000 traços representantes. Em destaque o menor desvio padrão e a menor amplitude. | 103 |
| Tabela 6.11 | Avaliação de resíduos quando da utilização de 16000 traços representantes. Em destaque o menor desvio padrão e a menor amplitude. | 103 |

“Cada pessoa brilha com luz própria entre todas as outras. Não existem duas fogueiras iguais. Existem fogueiras grandes e fogueiras pequenas e fogueiras de todas as cores. Existe gente de fogo sereno, que nem percebe o vento, e gente de fogo louco, que enche o ar de chispas. Alguns fogos, fogos bobos, não alumiam nem queimam; mas outros incendeiam a vida com tamanha vontade que é impossível olhar para eles sem pestanejar, e quem chegar perto pega fogo.”

GALEANO, E., *O Livro dos Abraços.*

Lista de Símbolos

x – Escalar x

\mathbf{x} – Vetor \mathbf{x}

$\mathbf{x}_{(t)}$ – Estado do vetor \mathbf{x} na t -ésima iteração de um algoritmo

x_i – O i -ésimo elemento do vetor \mathbf{x}

$\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^T \mathbf{b}$ – Produto escalar de dois vetores \mathbf{a} e \mathbf{b}

\mathbf{A} – Matriz \mathbf{A}

$\mathbf{A}_{i,:}$ – A i -ésima linha da matriz \mathbf{A}

$\mathbf{A}_{:,j}$ – A j -ésima coluna da matriz \mathbf{A}

$a_{i,j}$ – Elemento na i -ésima linha e j -ésima coluna de uma matriz

$\mathbf{v}[i]$ – O i -ésimo elemento num vetor \mathbf{v}

$\mathbf{v}[i, j]$ – O elemento de índices i e j num vetor ou campo escalar bidimensional \mathbf{v}

$f[i]$ – O i -ésimo elemento de uma função discreta f

\mathbf{A}^T – Transposta da matriz \mathbf{A}

\mathbf{A}^{-1} – Inversa da matriz \mathbf{A}

$\mathbf{D}_{:,k}^{(t)}$ – A k -ésima coluna de uma matriz \mathbf{D} na t -ésima iteração de um algoritmo

$\mathbf{D}_{:,k}^{(t+1)}$ – A k -ésima coluna de uma matriz \mathbf{D} na $(t + 1)$ -ésima iteração de um algoritmo

$\mathbf{D}_{(t)}$ – O estado da matriz \mathbf{D} na (t) -ésima iteração de um algoritmo ¹

$\frac{\partial}{\partial \mathbf{A}}(\text{expr})$ – Derivada parcial de uma expressão algébrica em relação à matriz \mathbf{A}

$\|x\|$ – Norma euclideana do vetor $\mathbf{x} = \sqrt{(\sum_i x_i^2)}$

$\|A\|_F$ – Norma de Frobenius da matriz $\mathbf{A} = \sqrt{(\sum_{i,j} |a_{i,j}|^2)}$

Lista de Abreviaturas

CD – *Coordinate Descent*

DL – *Dictionary Learning* ou Treinamento de Dicionário

FISTA – *Fast Iterative Shrinkage and Thresholding Algorithm*

GD – *Gradient Descent* ou Gradiente Descendente

ISTA – *Iterative Shrinkage and Thresholding Algorithm*

K-Means – Algoritmo de agrupamento em K grupos (em torno de K médias)

K-SVD – *K - Singular Value Decomposition*. Referente aos K maiores autovalores.

LASSO – *Least Absolute Shrinkage and Selection Operator*

MOD – *Method of the Optimal Directions* ou Método das Direções Ótimas

OMP – *Orthogonal Matching Pursuit*

SGK – *Sequential Generalization of K-Means* ou Generalização Sequencial do *K-Means*

SLIC – *Simple Linear Iterative Clustering* ou Agrupamento Iterativo Linear Simples

TD – Treinamento de Dicionário

1

Introdução

Como resultado de uma cadeia de passos de processamento geofísico, os dados tridimensionais de amplitudes sísmicas são parte integrante do trabalho de intérpretes (Yilmaz, 1987). Assim como percebemos no dia-a-dia o aumento gradual de resolução e qualidade de câmeras fotográficas, de câmeras em telefones móveis e de aparelhos de geração e aquisição de imagens médicas, as técnicas de aquisição de dados sísmicos para exploração de petróleo têm evoluído e, conjuntamente, a densidade e o volume de dados é crescente. Sísmica 4D¹ e sísmica de fundo oceânico são exemplos de aquisição geradores de grandes volumes de dados. A aquisição de dados com cada vez maior resolução exige, crescentemente, espaço de armazenamento, o que pode implicar, logo em seguida, na exigência de meios de transmissão de informação, mais pronunciadamente de redes de computadores com ou sem fios, com crescentes velocidades de transmissão.

Num contexto mais geral em que a ocupação de espaço de armazenamento e a demanda por velocidade de transmissão de dados está em elevação, surge a opção imediata por compressão. O estudo de compressão de dados vem de décadas, e há vários algoritmos bem documentados e aplicativos de propósito geral disponíveis, bem como outros que têm desempenho superior em atacar tipos específicos de dados como imagens, áudio e vídeo (Sayood, 2012).

No contexto específico de dados utilizados ao longo da cadeia de processamento sísmico, há trabalhos publicados que se propõem a gerar uma representação compacta para dados sísmicos como, por exemplo Tian e Li (2017) e Li et al. (2017), e outros que partem do pressuposto de que é possível explorar algum tipo de estrutura inerente ao dado sísmico e, a partir dela, pode-se representar trechos do sinal sísmico por uma combinação linear esparsa de elementos multidimensionais fundamentais, como em Zhou, Chen e Gao (2014) e Zhu, Liu e McClellan (2015). Em se tratando de mudar a representação do dado original para uma outra na qual se possa realizar uma compressão, a indicação que surge é de que algoritmos de compressão específicos para tipos específicos de dados poderiam alcançar um desempenho em compressão

¹Chama-se sísmica 4D um levantamento 3D que ocorre sobre a mesma região de interesse em momentos diferentes no tempo.

superior aos algoritmos de propósito geral. Esta especificidade se traduz na suposição, imposição ou percepção de uma **estrutura** que, uma vez levada em conta, pode trazer algum benefício. Este ganho pode ocorrer em termos de qualidade de representação do objeto em estudo e eventual simplicidade na explicação dos fenômenos subjacentes. A qualidade de representação referida pode ser traduzida numericamente, levando em conta, por exemplo, o somatório das diferenças absolutas entre cada amostra do dado reconstruído e o dado original. A simplicidade referida se baseia no Princípio de Occam e, numericamente, e ao longo desta tese, podemos traduzir para a quantidade mínima necessária (ou limitada) de vetores de uma base² de representação, concretizada numa matriz, cuja combinação linear de um subconjunto de suas colunas aproxima um dado sinal³. Um dos desafios explicitados ao longo desta tese é o de calcular pelo menos uma base candidata – uma matriz – que se molde e atenda ao papel de oferecer uma representação esparsa de sinais de amplitude sísmica.

1.1

Detalhamento do Tema

Nesta seção, descreveremos a área de estudo abordada na tese, o escopo do estudo, o problema de pesquisa atacado, os objetivos deste trabalho, que resultados esperamos e o que obtivemos, quais limitações se apresentaram, como podemos observar o rigor da pesquisa metodologicamente, justificativas para a tese e um breve mapa com a estrutura deste trabalho.

1.1.1

Descrição Geral da Área em Estudo

O presente estudo está inserido num contexto em que os problemas inversos de deconvolução e de inversão sísmicas são tratados como o de separação de duas fontes distintas de informação: (i) o conteúdo oscilatório gerado por uma fonte de energia sísmica, cuja duração tem tempo limitado, e (ii) um conteúdo esparsa que demarca as zonas de transição entre camadas de características físicas hipoteticamente distintas, os chamados contrastes de impedância ou refletividade. Neste contexto, a tese traz referências e faz um pequeno avanço em técnicas que se utilizam da modelagem esparsa para alcançar a pretendida separação de fontes.

²Neste trabalho, deixamos menos restrito o conceito de base. Ela não necessariamente é composta por vetores linearmente independentes entre si, ou seja, pode ou não ser concretizada por uma matriz ortogonal.

³O número de elementos deste subconjunto de colunas é, normalmente, inferior a 1/3 do número de linhas da referida matriz.

1.1.2

Abrangência do Estudo

O contexto do trabalho desenvolvido nesta tese abrange geofísica e computação.

Do lado da geofísica, estamos utilizando como insumo de estudo o resultado da aplicação do método de reflexão sísmica, que mede a grandeza física da amplitude da onda sísmica que retorna a partir da subsuperfície após seu disparo, e que viabiliza a inferência de propriedades como estrutura geológica, falhas e tamanho de um reservatório (Aminzadeh e Dasgupta, 2013).

Do lado da computação, fazemos uso de métodos de otimização com regularização. A regularização mais usada é a que usa norma ℓ_1 por poder gerar soluções esparsas. Também fazemos uso de algoritmos gulosos, aproximativos, para obtermos soluções esparsas baseadas na pseudo-norma ℓ_0 . Ambas as estratégias fazem a seleção de dimensões ou características do dado sendo estudado. Aplicamos técnicas de modelagem esparsa, que buscam uma codificação do dado em estudo usando uma base de representação que favorece a redução de dimensionalidade. Também propomos o uso de compressão, sucintamente descrita, para tratamento do resíduo apresentado entre o dado original e o resultado da reconstrução após a recodificação.

1.2

Detalhamento do Problema de Pesquisa

Dada a crescente demanda por espaço derivada da evolução dos métodos de aquisição sísmica, nos parece importante buscar uma forma de mudar a representação do dado sísmico, especificamente o de amplitudes sísmicas⁴, de tal forma que ocupe menos espaço. Este problema poderia ser enquadrado como o de compressão, para o qual há vários métodos de propósito geral disponíveis.

Inicialmente, pensamos em atacar o seguinte problema: como se chegar numa representação de um dado volumétrico de amplitudes sísmicas de tal forma que a informação original seja preservada, a menos de um erro de aproximação tolerado?

Após a leitura do artigo com título *Sparse and Redundant Representation Modeling—What Next?* de Michael Elad (Elad, 2012), houve uma mudança importante no foco do trabalho. Neste referido trabalho, Elad menciona que perceber e impor algum tipo de estrutura ao dado que deseja codificar pode trazer vantagens, e escreve: “(...) *Imposing dependencies between the representation entries adds further constraints on the signal, reduces the effective number of permitted subspaces, and thus leads to deeper dimensionality*

⁴Dado sísmico de amplitudes empilhado e migrado. Sucinta explicação na Subseção 2.1.2.

reduction.” Assim sendo, passamos a observar qual seria a estrutura contida num dado de amplitude e, assim, poderíamos, teoricamente, chegar a uma representação mais compacta do dado original.

A compacidade da representação é objetivo dos algoritmos de representação esparsa. A Figura 1.1 ilustra os elementos algébricos do problema resolvido por tais algoritmos. Na Figura 1.1, \mathbf{D} representa a matriz com a qual se consegue chegar numa representação esparsa como no vetor $\tilde{\mathbf{x}}$ tomando por base um vetor observação \mathbf{y} . Dependendo do algoritmo utilizado e da quantidade de dimensões não zero de $\tilde{\mathbf{x}}$, o produto $\mathbf{D}\tilde{\mathbf{x}}$ aproxima ou reproduz o vetor \mathbf{y} . Normalmente, a quantidade de elementos não nulos no vetor $\tilde{\mathbf{x}}$ é muito inferior ao posto da matriz \mathbf{D} ; e, normalmente, a matriz \mathbf{D} tem mais colunas que linhas e tem posto completo. Chegar a um vetor como $\tilde{\mathbf{x}}$ é o objetivo final de tais algoritmos.

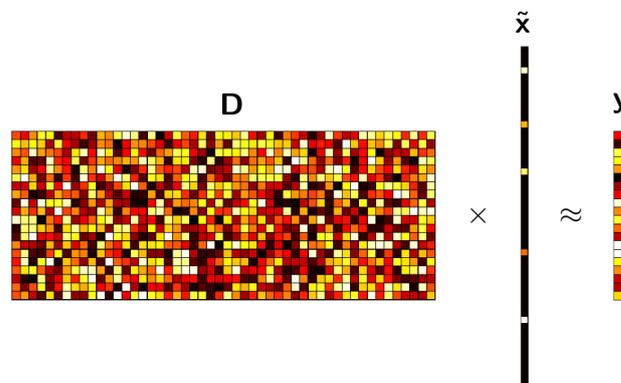


Figura 1.1: Ilustração dos elementos algébricos do problema de representação esparsa. \mathbf{D} é a matriz para se chegar no vetor esparsa $\tilde{\mathbf{x}}$ tomando por base o vetor denso \mathbf{y} .

Em geofísica, o problema de deconvolução por picos esparsos usa o pressuposto de que um dado de amplitudes sísmicas é resultado da junção de um dado ondulatório, representado pela ondaleta sísmica, com deltas de Dirac de amplitudes variadas, esparsamente presentes ao longo da linha vertical de cada traço sísmico. Eis que estas duas suposições se encaixam no que Elad chama de estrutura (Elad, 2012).

A partir de então, o problema que optamos atacar é **como fazer a deconvolução de um volume de amplitudes sísmicas de forma que ela possibilite gerar uma representação mais compacta do dado original?** Mais adiante, este mesmo problema de pesquisa nos permite fazer uma generalização parcial, transicionando de deconvolução para a incorporação de um termo mais genérico utilizado na geofísica: inversão.

1.3

Objetivo Geral e Objetivos Específicos

O objetivo geral desta tese é mostrar uma metodologia para mudar a representação original de um dado volumétrico de amplitudes sísmicas⁵. Tal mudança visa, ao final, alcançar um grau de compressão a partir da separação hipotética do sinal em duas componentes principais: uma componente ondulatória e uma componente de picos esparsos. Tais componentes são, por hipótese, resultado de um processo de geofísica computacional. Quando tal mudança de representação significar uma compressão sem perda ou com perda controlada, o resíduo sinônimo da diferença entre o dado original e o dado reconstruído (a partir das duas componentes) deverá ser observado e, eventualmente, armazenado de maneira apropriada, possivelmente após um processo de quantização. Para maior clareza, entendamos a perda como um erro de aproximação numérica.

São objetivos específicos desta tese mostrar e avaliar a validade das hipóteses de trabalho, quais sejam:

- O emprego de técnicas de agrupamento (*clustering*) para a seleção de traços participantes de processos de deconvolução ou inversão parece ser uma necessidade. Tal seleção, que serve como subamostragem, pode minimizar o impacto no custo dos algoritmos de representação esparsa e treinamento de dicionário usados nesta tese⁶;
- A adição da suposição da variação vertical do conteúdo oscilatório, por um lado, melhora a reconstrução de um dado real, ou seja, diminui o erro da representação esparsa. Por outro lado, há chance de se chegar numa codificação mais esparsa com o mesmo erro de aproximação quando não se supõe a variação vertical;
- A utilização de um conjunto (pequeno) de ondaletas conjuntamente com a adição da suposição de variação vertical como no item anterior podem melhorar ainda mais a reconstrução do dado original. Neste caso, cada elemento do mencionado conjunto deriva de um agrupamento de traços por alguma métrica de similaridade.

⁵Especificamente, o dado utilizado nos experimentos aparenta ter sido gerado pela aplicação de empilhamento e de migração, explicados sucintamente mais à frente.

⁶Dicionário: uma matriz com a qual é possível gerar uma representação esparsa de um vetor de entrada. No Capítulo 2 há explicação mais detalhada e contextualizada sobre treinamento de dicionário de representação esparsa.

1.4

Uma Visão sobre os Resultados

Com a validação das hipóteses, reforça-se a perspectiva de utilização algoritmos de modelagem esparsa em problemas de geofísica. Neste contexto, mostra-se útil a extensão feita de uma técnica de segmentação de imagens. Numericamente, evidencia-se a validade do emprego das técnicas particulares de treinamento e de extensão artificial de dicionário, cuja intenção é de promover uma representação compacta e informativa do dado original de amplitudes. Abre-se também a possibilidade de fazer o movimento inverso, da geofísica para a computação, ao tentar empregar as ideias aqui colocadas, por exemplo, sobre problemas de representação compacta de imagens.

1.5

Contribuições da Tese

Com detalhamento no decorrer do texto, eis algumas contribuições desta tese:

Lobbes. Foi desenvolvido um método de deconvolução de amplitudes sísmicas. A técnica faz uso de métodos de modelagem esparsa, aplicando uma normalização ao dado de entrada, realizando uma busca pelo valor ótimo de um parâmetro de regularização, aplicando restrições à quantidade e posição de picos de refletividade. Em artigo publicado, e nesta tese, mostra-se um comparativo entre métodos já desenvolvidos, e há a demonstração de que houve ganho com o método proposto. Tal contribuição é mostrada no Capítulo 3. Um artigo publicado com esta técnica se encontra no Apêndice A.

SLICT - Supertraços. A partir do conceito de supertraços de dados sísmicos, como o análogo aos superpixels de imagens digitais, foi desenvolvido um método, derivado de um gerador de superpixels, capaz de aglutinar traços sísmicos semelhantes e próximos entre si⁷. Deriva-se a seleção dos representantes destes grupos para um processo posterior de treinamento de dicionário, e mostra-se a semelhança desta seleção com o conceito de conjunto-núcleo. Detalhamento se encontra no Capítulo 4.

SGK com regularização. Tomando como base um método de treinamento de dicionário existente, foi derivada uma extensão capaz de gerar átomos⁸ cujo trecho de validade pode ser determinado com o uso de um operador de regularização. A matriz que representa o conjunto de restrições aplicado

⁷A proximidade leva em conta uma aproximação da posição em mapa em que os traços estão.

⁸Átomo: cada uma das colunas de uma matriz dicionário.

ao dicionário indica, por exclusão, a região de validade do mesmo. Detalhes no Capítulo 5.

Extensão de dicionário com filtro da terra. Cada átomo gerado pelo processo de treinamento de dicionário do item anterior sofre deformações para, em seguida, serem reincorporados ao dicionário original. Tais deformações seguem um modelo de dispersão e atenuação que emula o comportamento da terra, filtrando a onda sísmica propagada. Ao expandir o dicionário desta maneira, promovemos, ao final, uma qualidade superior de reconstrução do dado original. Detalhes no Capítulo 5.

Possibilidade de compressão. Observou-se o resíduo gerado pelo processo de representação esparsa do dado original. A partir dele, é possível perceber uma oportunidade de aplicação de compressão sobre este resíduo utilizando uma quantização ótima para distribuição gaussiana. Detalhes no Capítulo 6.

1.6

Limitações do Trabalho

Vários dos passos tentados ao longo da confecção desta tese esbarraram na limitação de recursos computacionais: experimentos com treinamento de dicionário e representação esparsa de dado sísmico 3D só puderam ser realizados em máquina com muitos núcleos (16 a 40) e com quantidade razoável de memória RAM (mais que 32 gigabytes), e influenciaram a execução de algoritmo de treinamento de dicionário, forçando o estabelecimento de limite de iterações. Além disso, os dados utilizados nos experimentos precisavam ser públicos e sem necessidade de permissão específica para utilização. Para alguns conjuntos de dados sísmicos específicos, havia a necessidade de o autor fornecer informações que poderiam configurar alguma quebra de sigilo de trabalho e, portanto, foram descartados e, neste trabalho, não lhes cabe menção.

Eis que esta tese apresenta, de forma sucinta, uma compilação, ligeiramente multidisciplinar, do que foi possível semear e colher quando da tentativa de alcançar o objetivo geral deste trabalho. Os detalhes de cada tópico são abordados no limite do entendimento claro por parte do autor. Não se trata de uma cápsula ou algo completamente acabado, mas sim de um detalhamento justificado frente a um objetivo importante, espelhando os resultados obtidos até então.

1.7

Metodologia Utilizada

Para a avaliação dos métodos empregados, incluindo os propostos, utilizamos um conjunto muito reduzido de dados de entrada, um sintético bidimensional (Marmousi2) e outro real tridimensional (F3R), descritos mais à frente. Ambos são amplamente utilizados e acessíveis. Os experimentos descritos no Capítulo de Resultados nos dão a possibilidade de inferir algumas conclusões de ordem prática, atestando ou não a validade dos passos realizados. Embora o conjunto de dados de entrada seja reduzido, a quantidade e a complexidade dos experimentos necessários à validação de cada hipótese têm alto custo computacional e, além disso, há chance de favorecer a reprodutibilidade e sua comparação com abordagens futuras. Como contribuição à comunidade científica, opta-se por deixar público um repositório com parte do código-fonte desenvolvido neste trabalho, além de um detalhamento dos dados de entrada utilizados.

1.8

Justificativa

Até onde é de conhecimento do autor desta tese, as referências utilizadas e revisadas neste trabalho não conduziram pesquisa que se utilizasse, ao mesmo tempo: do arcabouço de modelagem esparsa; da segmentação ou seleção de elementos para a geração de um dicionário adequado à codificação esparsa de um dado sísmico 3D pós-empilhado de amplitudes; da utilização de procedimentos de geofísica computacional; e eventual uso de compressão para minimização de perda de precisão numérica, relativa a erro de representação/reconstrução, e atingimento de alto desempenho. Sendo este alvo tão específico e, ao mesmo tempo, importante para utilização em ferramentas de interpretação em geofísica (como é o caso da ferramenta **v3o2**⁹ utilizada pela Petrobras), este trabalho se justifica.

Ao finalizar esta tese, observou-se um conjunto de trabalhos publicados que fazem uso de modelagem esparsa sobre dados sísmicos, com semelhanças ao que se desenvolve nesta tese. Tal fato reforça a relevância do trabalho aqui desenvolvido. Destaque para Tian e Li (2017) e Sui e Ma (2019).

Genericamente, haveria vantagem em uma codificação de um dado de amplitudes sísmicas se esta fosse capaz de reduzir o tamanho do dado original, mesmo aceitando uma perda controlada, e não trouxesse erro ou perda que inviabilizasse processos de interpretação ou visualização posteriores. Além

⁹Vide sítio do Instituto Tecgraf/PUC-Rio: <https://www.tecgraf.puc-rio.br>. Seção Portfolio, Indústria de Óleo e Gás.

disso, um mecanismo de reconstrução associado a tal codificação deveria possibilitar a um usuário tão somente uma reconstrução parcial, local, sem a necessidade de reconstruir o todo para recuperar uma parte. Tal vantagem é uma ambição deste trabalho, que acreditamos ter alcançado, ao menos parcialmente.

Apesar de o dado sísmico não empilhado ocupar espaço significativamente maior do que o tipo de dado aqui utilizado, o emprego das técnicas aqui abordadas podem ser justificadas também para dados empilhados e migrados. Trabalhos publicados por outros autores reforçam este argumento. É da experiência profissional do autor a percepção de que é corriqueira a utilização de volumes crescentes de dados sísmicos em estações de trabalho e em grupos de máquinas para processamento de alto desempenho. Mesmo que, eventualmente, um dado sísmico de amplitude possa caber na memória principal de uma estação de trabalho (com 32, 64, 128, 256 ou até 512 gigabytes de memória principal), é possível que o mesmo não ocorra em todos os ambientes nos quais tal dado precise ser manipulado ou exibido. Além disso, deve-se ter em conta o tempo de transferência de grandes arquivos através de rede e o custo da carga destes para a memória principal de qualquer computador.

1.9 Organização da Tese

Esta tese está organizada em 7 capítulos. Esta introdução é o primeiro capítulo.

No Capítulo 2 é feita uma breve revisão bibliográfica e, para facilitar a compreensão durante a leitura, opta-se por fornecer explicação sobre uma série de conceitos e algoritmos utilizados para resolver problemas, idênticos ou similares, ao que esta tese se propõe a resolver. Dentre estes, pode-se citar de antemão: esparsidade, deconvolução e inversão.

O Capítulo 3 dá ênfase ao problema específico de deconvolução para o qual propusemos uma solução. Inclui insumos e métodos empregados.

Nos Capítulos 4 e 5 são relatados experimentos que reforçam a ideia central da tese de subdivisão de regiões e de aplicação de conceitos como o aprendizado de dicionário com o objetivo de, ao mesmo tempo, chegar a uma representação mais compacta do meio em estudo e facilitar a paralelização das tarefas computacionais. Inclui insumos e métodos empregados.

O Capítulo 6 apresenta resultados do emprego das técnicas mencionadas nos Capítulos 3, 4 e 5 sobre dados sintéticos e reais, e discutimos brevemente tais resultados.

O Capítulo 7 finaliza o texto principal da tese com conclusões, fornecendo um sucinto detalhamento sobre oportunidades para trabalhos futuros.

2

Conceitos e Trabalhos Relacionados

Na primeira parte deste capítulo, trazemos um conjunto de conceitos e algoritmos fundamentais à compreensão da tese. Na segunda parte, mostramos o resultado de um processo de pesquisa e uma breve revisão de trabalhos relacionados a esta monografia.

2.1

Conceitos, Problemas e Algoritmos

Nesta seção são brevemente mostrados alguns conceitos em geofísica, otimização e processamento de sinais no intuito de facilitar o entendimento do conteúdo por vir e de mostrar parcialmente o contexto em que esta tese se insere.

2.1.1

Contraste de Impedância

A técnica de levantamento sísmico por reflexão se utiliza da geração e captação da reflexão de uma onda sísmica, normalmente controlada. As reflexões ou ecos ocorrem quando há variação nas propriedades físicas das rochas ao longo do caminho de propagação da onda sísmica, e a parcela de energia que retorna depois da liberação da onda gerada é percebida por sensores adequados (geofones ou hidrofones, conforme o local do levantamento), registrada e, posteriormente, processada para fins de análise e interpretação.

A impedância acústica é uma grandeza utilizada para estimar os coeficientes de reflexão sísmica nas interfaces entre camadas de subsolo com propriedades físicas distintas. A impedância acústica é uma propriedade da rocha, definida pelo produto da velocidade da onda e da densidade da camada. Neste trabalho, usa-se a suposição, ou simplificação, de que tais camadas são planas e paralelas, com propriedades físicas constantes em cada uma, e que as ondas sísmicas incidem sobre elas perpendicularmente.

Sejam ρ_i e ρ_{i+1} as densidades de duas camadas adjacentes, em que o índice maior se refere à camada imediatamente abaixo, e sejam v_i e v_{i+1} as velocidades de duas camadas adjacentes. Assim, a grandeza *contraste de*

impedância, também chamada de *refletividade*, pode ser definida, na i -ésima posição, por

$$r_i = \frac{\rho_{i+1}v_{i+1} - \rho_i v_i}{\rho_{i+1}v_{i+1} + \rho_i v_i} \quad (2-1)$$

O *traço sísmico* é uma representação computacional, normalmente na forma de um *array*, da sequência vertical ordenada de valores de refletividades, crescente com o tempo de propagação da onda. Cada sensor unitário de captação sísmica acaba por registrar um traço sísmico, de modo que, associado a um tempo a partir do instante de disparo da onda sísmica da fonte de energia, percebe-se um conteúdo de energia relacionado às reflexões desta onda quando chega a uma região de contraste de impedância entre rochas de propriedades físicas distintas.

A Figura 2.1 exhibe o conteúdo de amplitudes sísmicas de 10 traços do dado F3R ¹.

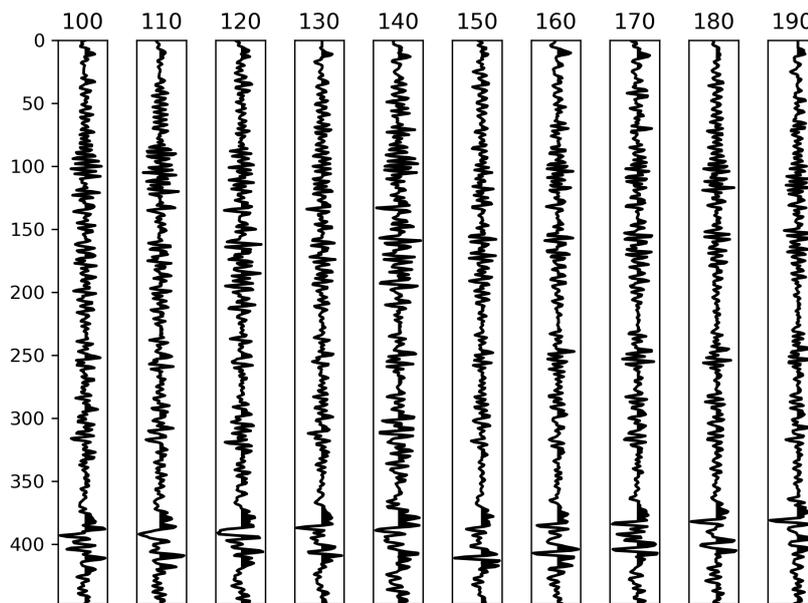


Figura 2.1: Exibição de traços de amplitude sísmica do dado F3R. *Inline* 290, e *crosslines* 100 a 190 (saltando de 10 em 10).

2.1.2

Levantamento sísmico, Empilhamento, Migração e Conversão para Profundidade

O levantamento sísmico convencional usando geofones ou hidrofones ocorre com a coleta de dados a partir de uma sequência de disparos de ondas sísmicas, com a variação da posição ou da fonte ou dos receptores ou de ambos.

¹O dado F3R deriva do bloco F3, descrito no Capítulo 6.

Em terra, um levantamento sísmico pode se utilizar, por exemplo, de cargas explosivas ou de aparelhos vibradores adaptados em veículos de grande porte. No mar, podem ser usados, por exemplo, navios que puxam um conjunto de cabos com hidrofones (os receptores) e, ao mesmo tempo, dispõem de pistolas de ar como geradores das ondas sonoras. A Figura 2.2 ilustra um levantamento sísmico no mar, e a Figura 2.3 mostra um navio de levantamento rebocando cabos com hidrofones.

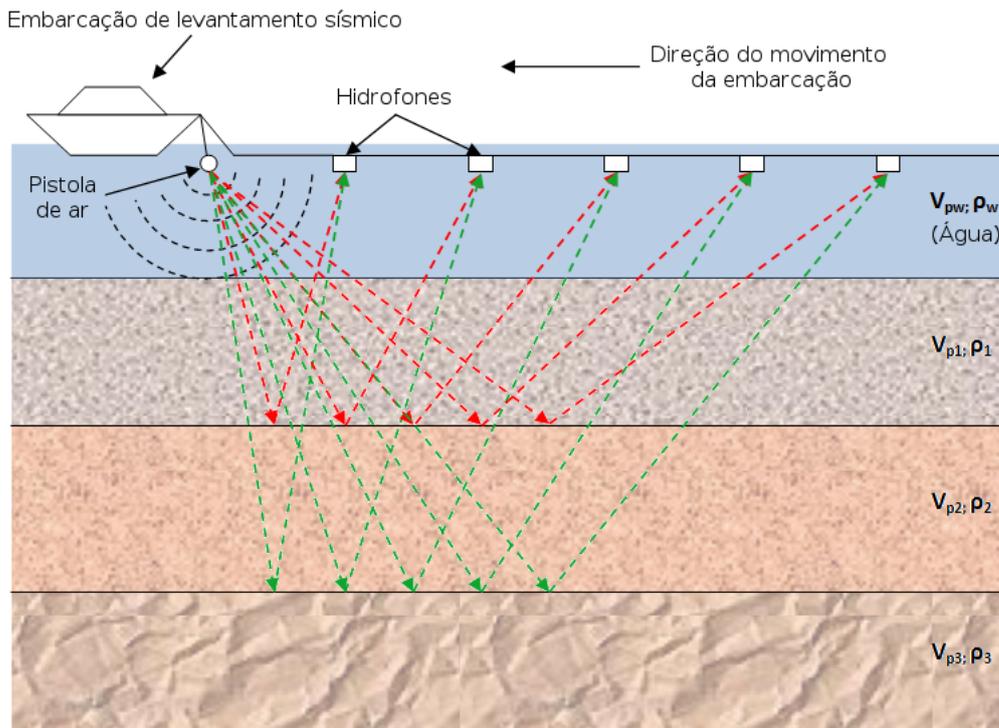


Figura 2.2: Diagrama de um levantamento sísmico marinho usando cabos com hidrofones. As linhas tracejadas coloridas representam os caminhos mais curtos percorridos por uma onda sísmica ao ser refletida por camadas de coeficientes de reflexão distintos, captada, em diferentes tempos e posições, por cada hidrofone. Cenário se refere ao disparo de um único tiro de uma pistola de ar. Derivado de trabalho disponível na Wikipedia: By Nwhit - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=18527767>.

Os dados inicialmente coletados são chamados de pré-empilhados. A partir do acionamento das fontes sísmicas, cada receptor capta as ondas diretamente emitidas pelas fontes, e refletidas e refratadas pelas variações de propriedades físicas das rochas de subsuperfície. As variações de posição das fontes em relação ao receptor e a variação de tempo do registro sísmico geram, para cada receptor, um dado bidimensional. Algumas correções são aplicadas diretamente a este dado antes de ele sofrer o processo chamado de empilhamento. Neste empilhamento, tomando um dado adicional de velocidade de empilhamento capaz de, hipoteticamente, horizontalizar eventos importantes, o dado



Figura 2.3: Navio de levantamento sísmico. Navio Ramform Sovereign, da PGS, na Bacia de Santos. Adaptado do boletim 2.2012 da Sociedade Brasileira de Geofísica. Disponível: <https://sbgf.org.br/noticias/images/Boletim-2-2012.pdf>. Acesso: 19/12/2018.

que antes se configurava bidimensional é projetado para um único traço – unidimensional – na vertical.

Migração é outro processo aplicado a dados geofísicos que tem por propósito mover as reflexões sísmicas para uma localização apropriada na imagem sísmica.

Após os processos de empilhamento e migração é que os eventos sísmicos são posicionados temporalmente de forma correta. O posicionamento de cada amostra de cada traço em profundidade requer outro procedimento, o de conversão tempo-profundidade.

O conjunto de traços empilhados, migrados ou não, convertidos para profundidade ou não, formam um cubo sísmico (em 3D) ou uma seção sísmica (em 2D). Tais conjuntos formam o tipo de insumo fundamental sobre o qual esta tese se aplica.

Para mais detalhes sobre os processos e termos de geofísica, vide Onajite (2014)².

2.1.3 Convolução

A operação matemática da convolução pode ser entendida como a integração, dentro de um intervalo, do produto de duas funções distintas, sendo que uma delas é, propositalmente, invertida e transladada. Trata-se de uma operação particular de composição de duas funções. Para o caso contínuo, tome

²Capítulos 5, 6 e 9.

duas funções $f, g : \mathbb{R} \rightarrow \mathbb{R}$. A *convolução* de f e g , denotada por $(f * g)(x)$, expressa-se como

$$(f * g)(x) = \int_{-\infty}^{+\infty} f(t)g(x-t)dt = \int_{-\infty}^{+\infty} f(x-t)g(t)dt.$$

Note-se a inversão de sinal de uma das funções quando da realização do cálculo.

Para o caso discreto, sejam duas funções $f, g : \mathbb{Z} \rightarrow \mathbb{R}$. Denotamos, então, a convolução como

$$(f * g)[x] = \sum_{i=-\infty}^{+\infty} f[i]g[x-i] = \sum_{i=-\infty}^{+\infty} f[x-i]g[i].$$

E para o caso em que o suporte de ambas as funções é compacto, supondo que ambas as funções estão definidas no intervalo discreto $[-M, M]$ com $2M+1$ elementos, temos

$$(f * g)[x] = \sum_{i=-M}^{+M} f[i]g[x-i] = \sum_{i=-M}^{+M} f[x-i]g[i].$$

O resultado da aplicação de uma série de procedimentos de processamento geofísico sobre o dado do levantamento sísmico traz como resultado um conjunto de traços sísmicos que, idealmente e teoricamente, podem ser interpretados como o resultado da convolução de uma ondaleta sísmica, emitida pela fonte, com os contrastes de impedância, que serviriam como marcadores das regiões vizinhas com características físicas contrastantes. O **modelo convolucional** parte da suposição de que as amplitudes, gravadas nos traços de um dado sísmico, são o resultado da convolução entre os contrastes de impedância e uma ondaleta estacionária, ou seja, que não muda de comportamento com a profundidade.

2.1.4 Deconvolução

A operação de deconvolução é o oposto da convolução, ou seja, a partir de uma função, que chamaremos, aqui, de **composta**, deseja-se determinar quais são suas componentes. Ocorre que este é um problema mal posto, admitindo mais de uma solução. Para se conseguir uma única solução, normalmente usa-se informação adicional, incluindo a suposição de que há algum tipo de estrutura, ou comportamento esperado, de algum dos sinais originários da composta. Deconvolução é um problema inverso, ou seja, desejamos, a partir de um conjunto de observações (as consequências) e de um conjunto de suposições sobre uma das funções componentes, obter as funções originárias (as causas).

A deconvolução é uma das operações executadas nesta tese com o objetivo de estimar quais são as componentes dos traços sísmicos, ou seja, qual é a (ou quais são as) ondaleta(s) e quais são as refletividades encontradas no dado sísmico de entrada.

O principal propósito da deconvolução neste trabalho não é o de aumentar a resolução da sísmica original, mas sim de fazer uma estimativa de qualidade, segundo alguma métrica, de suas componentes. Um dos propósitos é o de mudar a representação do dado original, objetivando uma compressão importante do dado de entrada.

A deconvolução de picos esparsos pode ser vista como um problema inverso no qual se deseja determinar a quantidade, a posição, a magnitude e a fase (ou o sinal) de picos esparsos de refletividade para cada traço sísmico. Ao realizar a convolução de tal informação de picos esparsos com a onda sísmica estimada, seria possível aproximar o sinal (ruidoso) de amplitudes sísmicas (Fernandes, Lopes e Gattass, 2017).

2.1.5

Norma

O conceito de norma está ligado a vários algoritmos apresentados nesta tese. Uma norma tem o papel de medir distância ou tamanho, e utilizaremos o conceito de norma para vetores e para matrizes (Golub e Van Loan, 2013).

Dados vetores $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ e um escalar $\alpha \in \mathbb{R}$, uma norma de um vetor em \mathbb{R}^n é uma função $f : \mathbb{R}^n \rightarrow \mathbb{R}$ que atende aos seguintes requisitos:

$$f(\mathbf{x}) \geq 0, \text{ e } f(\mathbf{x}) = 0 \text{ se e somente se } \mathbf{x} = 0,$$

$$f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y}) \text{ (desigualdade triangular),}$$

$$f(\alpha\mathbf{x}) = |\alpha|f(\mathbf{x}).$$

A notação usual para normas é usando barras duplas em torno do elemento a ser medido: $f(\mathbf{x}) = \|\mathbf{x}\|$. Uma norma \mathbf{p} para um elemento $x \in \mathbb{R}^n$ é definida como:

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}, \quad p \geq 1.$$

Utilizaremos a pseudo-norma ℓ_0 , a norma ℓ_1 e a norma ℓ_2 ao longo deste trabalho.

A pseudo-norma ℓ_0 faz a contagem de elementos diferentes de zero do vetor de entrada. Seu uso se dá, ao longo desta tese, principalmente no algoritmo OMP³, explicado mais adiante.

A norma ℓ_1 calcula o somatório dos valores absolutos de um vetor de entrada. Sua utilização na formulação de problemas de otimização em modelagem esparsa permite o uso parcial de uma solução algébrica, e tem por característica gerar soluções esparsas. O uso é parcial porque o domínio da solução ótima é convexo, mas apresenta uma quebra de continuidade por conta do uso do operador de valor absoluto.

A norma ℓ_2 faz o somatório dos quadrados dos elementos do vetor de entrada. Sua larga utilização se dá pela facilidade de aplicação de uma álgebra direta, dado que é possível fazer uma derivação usando cálculo diferencial (derivando e igualando a zero uma função de custo). No entanto, seu uso não gera uma solução esparsa.

Também aplicamos o conceito de norma para matrizes. Sejam \mathbf{A} e \mathbf{B} matrizes do $\mathbb{R}^{m \times n}$. Como o domínio $\mathbb{R}^{m \times n}$ é isomórfico a \mathbb{R}^{mn} , as definições dadas para vetores devem ser equivalentes para matrizes. Assim, tomando agora a função que calcula a norma como $f : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$, temos

$$\begin{aligned} f(\mathbf{A}) &\geq 0, \text{ e } f(\mathbf{A}) = 0 \text{ se e somente se } \mathbf{A} = 0, \\ f(\mathbf{A} + \mathbf{B}) &\leq f(\mathbf{A}) + f(\mathbf{B}) \text{ (desigualdade triangular),} \\ f(\alpha \mathbf{A}) &= |\alpha| f(\mathbf{A}). \end{aligned}$$

A notação usada para norma de vetores também será usada para norma de matrizes, ou seja, as barras verticais duplas: $f(\mathbf{A}) = \|\mathbf{A}\|$.

A norma de matrizes que vamos utilizar é chamada de Frobenius:

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{i,j}|^2}. \quad (2-2)$$

Para detalhamento e propriedades sobre normas de vetores e matrizes, vide Golub e Van Loan (2013)⁴.

³*Orthogonal Matching Pursuit*. Algoritmo guloso que aproxima a solução do problema de representação esparsa, mostrado na Subseção 2.1.8.

⁴Capítulo 2.

2.1.6

Esparsidade

A palavra *esparsidade* parece ainda não estar presente nos dicionários de língua portuguesa⁵, e este é um motivo para que tal termo seja definido. Aproveitamos, então, a definição dada em Foucart e Rauhut (2013). Dado um vetor $\mathbf{v} = \{v_1, v_2, \dots, v_n\}, v_i \in \mathbb{C}$, o *suporte* de \mathbf{v} , $\text{sup}(\mathbf{v}) = \{i \in [1, n], i \in \mathbb{N} \mid v_i \neq 0\}$, isto é, o conjunto dos índices das entradas diferentes de zero de \mathbf{v} . O vetor \mathbf{v} é chamado *s-esparso* se no máximo s entradas de \mathbf{v} forem diferentes de zero, ou seja, $\text{card}(\text{sup}(\mathbf{v})) \leq s$, sendo $\text{card}(\cdot)$ a cardinalidade ou o número de elementos do vetor passado por parâmetro. Ao longo deste trabalho, também usaremos uma forma equivalente à cardinalidade, que é a pseudo-norma ℓ_0 de \mathbf{v} , definida como a seguir:

$$\|\mathbf{v}\|_0^0 \equiv \lim_{p \rightarrow 0} \sum_{i=1}^n |v_i|^p \equiv \sum_{i=1, v_i \neq 0}^n 1.$$

A esparsidade é uma das suposições levadas em conta quando da resolução do problema inverso da deconvolução das amplitudes sísmicas. Tal suposição é aplicada à informação de picos de refletividade que se deseja obter. Isso significa que estamos supondo que há uma quantidade limitada de camadas de rocha na área estudada, cada uma com seu conjunto distinto de propriedades físicas.

2.1.7

Modelagem Esparsa, Dicionário, Átomo e Codificação Esparsa

Sejam $\mathbf{D} \in \mathbb{R}^{m \times n}$, $\mathbf{x} \in \mathbb{R}^n$ e $\mathbf{y} \in \mathbb{R}^m$ e tomemos o problema a seguir como referência:

$$\mathbf{D}\mathbf{x} = \mathbf{y}. \quad (2-3)$$

No problema 2-3 se deseja descobrir o vetor \mathbf{x} dados o vetor de entrada \mathbf{y} e a matriz \mathbf{D} . No caso sem ruído e com $n = m$, bastaria realizar $\mathbf{x} = \mathbf{D}^{-1}\mathbf{y}$, supondo que \mathbf{D} possa ser invertida. No caso com ruído, o problema se torna o de minimizar um erro de aproximação $e(\mathbf{x})$:

$$\begin{aligned} e(\mathbf{x}) &= \mathbf{y} - \mathbf{D}\mathbf{x}, \\ \tilde{\mathbf{x}} &= \arg \min_{\mathbf{x}} \|e(\mathbf{x})\|_2^2. \end{aligned} \quad (2-4)$$

Para este caso, pode-se utilizar o método dos Mínimos Quadrados Ordinário, cuja solução pode ser alcançada, por exemplo, via resolução algébrica

⁵Busca feita nos dicionários online Priberam, Dicio, Infopédia e Michaelis em 23/05/2018.

direta da Equação Normal:

$$\tilde{\mathbf{x}} = (\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \mathbf{y}. \quad (2-5)$$

Há a possibilidade de se usar métodos iterativos como, por exemplo, o de Gradiente Descendente, em que se deseja minimizar $\mathbf{F}(\mathbf{x}) = \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2$ usando seu gradiente $\nabla \mathbf{F}(\mathbf{x}) = 2\mathbf{D}^T(\mathbf{D}\mathbf{x} - \mathbf{y})$ para obter uma solução mais próxima da ótima $\mathbf{x}_{(n+1)}$ a partir da solução obtida na iteração anterior $\mathbf{x}_{(n)}$:

$$\tilde{\mathbf{x}}_{(n+1)} = \tilde{\mathbf{x}}_{(n)} - \gamma_{(n)} \nabla \mathbf{F}(\tilde{\mathbf{x}}_{(n)}). \quad (2-6)$$

O termo $\gamma_{(n)}$ tem o efeito de aproximar a solução momentânea à ótima e, por consequência, apresenta-se como uma sequência de valores decrescentes, desde que a função \mathbf{F} esteja definida e seja diferenciável numa vizinhança do ponto avaliado⁶. Pode-se utilizar como critério de parada, dentre outros, por exemplo, comparando a um limiar previamente estabelecido, a norma euclídeana do gradiente, a norma euclídeana da diferença entre soluções consecutivas, e a norma euclídeana da diferença entre as funções de custo avaliadas em iterações consecutivas.

Para os casos em que não é possível obter uma solução única diretamente a partir da resolução de problemas como nas Equações 2-3, 2-4 ou 2-5, opta-se pelo que se chama de regularização, a adição de uma penalidade $p(\mathbf{x})$ sobre a solução, como na regularização de Tikhonov (Beck e Teboulle, 2009):

$$\begin{aligned} e(\mathbf{x}) &= \mathbf{y} - \mathbf{D}\mathbf{x}, \\ p(\mathbf{x}) &= \mathbf{\Gamma}\mathbf{x}, \\ \tilde{\mathbf{x}} &= \arg \min_{\mathbf{x}} \|e(\mathbf{x})\|_2^2 + \lambda \|p(\mathbf{x})\|_2^2. \end{aligned} \quad (2-7)$$

O fator λ permite um controle entre uma maior fidelidade às medidas e menor sensibilidade a ruído (Beck e Teboulle, 2009). Quando o operador $\mathbf{\Gamma}$ é a identidade, a regularização é chamada de *ridge*. A utilização da regularização de Tikhonov é atrativa, pois possibilita uma solução algébrica:

$$\tilde{\mathbf{x}} = (\mathbf{D}^T \mathbf{D} + \lambda \mathbf{\Gamma}^T \mathbf{\Gamma})^{-1} \mathbf{D}^T \mathbf{y}. \quad (2-8)$$

Além da solução algébrica, também se pode chegar numa solução via métodos iterativos, adicionando o termo da regularização à fórmula em 2-6:

⁶Há técnicas disponíveis para realizar a descoberta do valor ótimo do termo γ . Em Izmailov e Solodov (2007), Capítulo 3, páginas 58 a 74, há detalhamento quanto a técnicas de busca de valores de γ , como as regras de Armijo, de Goldstein e de Wolfe.

$$\begin{aligned}\nabla \mathbf{F}(\mathbf{x}) &= 2\mathbf{D}^T(\mathbf{D}\mathbf{x} - \mathbf{y}) + 2\lambda\mathbf{\Gamma}^T\mathbf{\Gamma}\mathbf{x}, \\ \tilde{\mathbf{x}}_{(n+1)} &= \tilde{\mathbf{x}}_{(n)} - \gamma_{(n)}\nabla \mathbf{F}(\tilde{\mathbf{x}}_{(n)}).\end{aligned}\tag{2-9}$$

Partindo desta breve introdução à solução de problemas algébricos simples que envolvem alguma otimização, passamos, então, a uma breve explicação do termo **modelagem esparsa**.

Modelagem deriva de encontrar um modelo. Por **modelo** podemos entender como um ente matemático que tem, teórica e aproximadamente, a capacidade de explicar algum fenômeno. A matriz \mathbf{D} nas equações anteriores serve como o modelo. Quando \mathbf{D} é multiplicada pelo vetor de parâmetros (de solução) \mathbf{x} , gera-se um resultado que se aproxima do vetor de entrada \mathbf{y} , a amostra multidimensional que exemplifica o fenômeno. O interesse específico em se obter \mathbf{x} com quantidade limitada ou mínima de parâmetros/dimensões não zero, ou de cardinalidade mínima, pode ser traduzido como a busca por uma solução **esparsa**.

Assumindo que se tenha uma matriz de modelo, definida por \mathbf{D} , a busca por uma solução esparsa \mathbf{x} pode ser formulada através dos seguintes problemas de otimização, para os casos sem e com ruído, respectivamente (Rish e Grabarnik, 2015):

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{sujeito a} \quad \mathbf{y} = \mathbf{D}\mathbf{x}, \tag{2-10}$$

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{sujeito a} \quad \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \leq \epsilon. \tag{2-11}$$

Partindo da Equação 2-11 e de uma restrição quanto ao número limite de entradas diferentes de zero no vetor de solução \mathbf{x} e tentando melhor aproximar o vetor de solução ao dado de entrada utilizando a matriz de modelo \mathbf{D} , temos (Dumitrescu e Irofti, 2018):

$$\min_x \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 \quad \text{sujeito a} \quad \|\mathbf{x}\|_0 \leq k. \tag{2-12}$$

Os problemas formulados nas Equações 2-10, 2-11 são combinatoriais, e são classificados como NP-Difíceis⁷ (Natarajan, 1995; Cormen et al., 2001). Em Natarajan (1995), há prova quanto à complexidade da solução do problema na Equação 2-11 e traz um algoritmo para sua solução aproximada.

No linguajar de modelagem esparsa, os problemas 2-10, 2-11 e 2-12 são instâncias de problemas de **representação** ou **recuperação esparsa**.

⁷Não Determinísticos Polinomiais Difíceis. O custo computacional de tais algoritmos cresce exponencialmente com o tamanho da entrada.

Partindo novamente da Equação 2-11 e observando a Equação 2-12, é possível realizar uma recuperação esparsa eficiente resolvendo os seguintes problemas de otimização convexa, sem e com ruído, respectivamente:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{sujeito a} \quad \mathbf{y} = \mathbf{D}\mathbf{x}, \quad (2-13)$$

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{sujeito a} \quad \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \leq \epsilon. \quad (2-14)$$

É possível, ainda, reescrever o problema 2-14 como um problema de otimização sem restrições utilizando um multiplicador de Lagrange λ apropriado definido a partir de ϵ (Rish e Grabarnik, 2015):

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1. \quad (2-15)$$

O problema na Equação 2-15 é chamado de Lasso⁸ (Rish e Grabarnik, 2015), (Hastie, Tibshirani e Wainwright, 2015). Sua utilização e resolução são fundamentais ao tratarmos do problema de deconvolução de picos esparsos de refletividade.

A Figura 2.4 mostra distintas representações de problemas envolvendo as normas ℓ_1 , ℓ_2 e a pseudo-norma ℓ_0 . Ao utilizar a pseudo-norma ℓ_0 , não se tem mais um problema convexo: ele se torna claramente descontínuo. A norma ℓ_2 acaba gerando o menor erro, mas não gera uma solução esparsa. A norma ℓ_1 gera normalmente uma solução com maior erro de aproximação em relação à conseguida via norma ℓ_2 , mas tem a propriedade de gerar uma solução esparsa.

A segunda tarefa principal em modelagem esparsa é a de calcular uma matriz $\mathbf{D} \in \mathbb{R}^{m \times n}$ tal que o vetor \mathbf{y} possa ser codificado por um vetor \mathbf{x} esparsa, ou seja, cujo valor da pseudo-norma ℓ_0 seja inferior a m , ou o menor possível. A esta tarefa chamemos de **aprendizado de dicionário**, em que \mathbf{D} é matriz-dicionário capaz de aproximar o vetor esparsa \mathbf{x} no vetor original \mathbf{y} . As colunas de um dicionário são chamadas de **átomos**. Se tomarmos uma matriz que codifica uma transformada discreta de Fourier, seus átomos correspondem a coeficientes de senóides. O emprego de matrizes-dicionários para transformação de sinais generaliza a ideia das transformadas clássicas como de Fourier ou Wavelet, uma vez que um dicionário normalmente tem quantidade de colunas superior à quantidade de linhas. Grosso modo, quanto maior a quantidade de átomos de um dicionário, maior poder de expressão ele tem e, por consequência, maior será o número de zeros nos vetores que são recodificados quando de sua

⁸Lasso deriva de *Least Absolute Shrinkage and Selection Operator*.

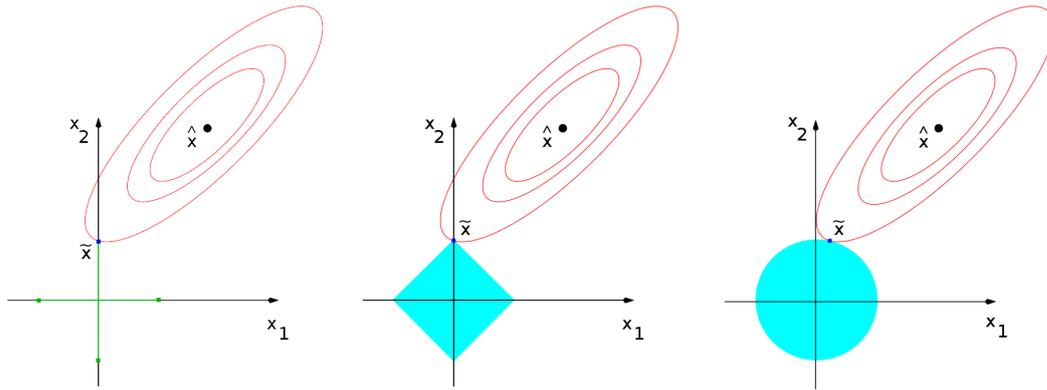


Figura 2.4: Ilustração da estimativa de solução do problema usando 2-12 (esquerda), usando o Lasso (centro) e usando *ridge*. Supondo $\mathbf{x} = (x_1, x_2)$, as áreas coloridas (em verde ou azul) se referem a $\|\mathbf{x}\|_0 < t$, $\|\mathbf{x}\|_1 < t$ e $\|\mathbf{x}\|_2^2 < t^2$, sendo t uma constante. O ponto $\tilde{\mathbf{x}}$ se refere ao ótimo com restrições, e o ponto $\hat{\mathbf{x}}$ se refere ao ótimo sem as restrições (por mínimos quadrados). Em vermelho, contornos da função de resíduo dos mínimos quadrados. Adaptado a partir de Hastie, Tibshirani e Wainwright (2015), Cap. 2, Figura 2.2.

utilização, ou seja, maior é o grau de esparsidade. Observe-se que ao admitir matrizes de codificação com mais colunas do que linhas, e sendo tais matrizes de posto completo (Golub e Van Loan, 2013), elas não mais podem formar uma base ortogonal, como seria o caso, por exemplo, da base para a transformada Fourier; obrigatoriamente, haverá um par de colunas nas matrizes de dicionário com produto escalar maior que zero⁹.

Em Dumitrescu e Irofti (2018) afirma-se que, em geral, um algoritmo para resolver o problema de representação esparsa tem maior chance de encontrar a solução ótima se os átomos do dicionário estiverem distantes (em relação à norma euclidiana) uns dos outros. Por solução ótima entenda-se, neste caso, aquela cujo nível de esparsidade é o maior possível, ou seja, contendo a menor quantidade possível de elementos não zero e que melhor aproxima o dado original.

2.1.8 OMP

O uso da norma ℓ_0 na formulação de problemas de modelagem esparsa, como feito, por exemplo, na Equação 2-12, impede que utilizemos qualquer método algébrico para solução direta do problema, uma vez que tal formulação leva a um problema não convexo, que deriva da seleção do melhor subconjunto de vetores cuja combinação linear melhor aproxima um outro vetor. Para este caso, pode-se optar por um algoritmo aproximativo, que é o caso do OMP.

⁹De fato, tais matrizes de posto completo com mais colunas do que linhas são chamadas de *frames* (quadros). Em Dumitrescu e Irofti (2018) há uma breve explicação sobre *frames*.

O algoritmo OMP (*Orthogonal Matching Pursuit* ou, em tradução livre, busca por casamento ortogonal) é um dos algoritmos capazes de gerar uma codificação esparsa (Rish e Grabarnik, 2015), (Elad, 2010), (Dumitrescu e Irofti, 2018). Trata-se de um algoritmo guloso (Kleinberg e Tardos, 2005, Cap. 4), (Cormen et al., 2001, Cap. 16) em que, a cada iteração, é escolhido o átomo do dicionário cujo módulo do produto escalar com o resíduo seja máximo. Inicialmente, o resíduo é igualado ao vetor de observação (o vetor \mathbf{y} como na Equação 2-3). Uma vez escolhido um átomo, resolve-se um problema de mínimos quadrados ordinário em que somente este átomo é levado em conta para a solução do problema. O resíduo é recalculado e o processo é reiniciado, mantendo, neste instante, o coeficiente do primeiro vetor selecionado fixo. Quando da escolha do segundo vetor em diante, até o limite do grau de esparsidade desejado, os coeficientes dos vetores escolhidos até então sofrem novo ajuste por mínimos quadrados (Rish e Grabarnik, 2015). O Algoritmo 1 sintetiza o funcionamento básico do OMP.

Algoritmo 1: OMP - *Orthogonal Matching Pursuit*

Entrada: dicionário $\mathbf{D} \in \mathbb{R}^{m \times n}$

sinal observado $\mathbf{y} \in \mathbb{R}^m$

nível de esparsidade desejado k

Saída: Suporte à representação S e a solução $\mathbf{x} \in \mathbb{R}^n$ com $\|\mathbf{x}\|_0 \leq k$.

- 1 $S = \emptyset, \mathbf{r} = \mathbf{y};$
 - 2 **enquanto** $|S| < k$ e *seguirAFrente*(r) **faça**
 - 3 $j \leftarrow$ índice da coluna de D que maximiza $|\mathbf{r}^T \mathbf{D}_{:,j}|, j \notin S;$
 - 4 $S \leftarrow S \cup \{j\};$
 - 5 Calcule a nova solução com este suporte $\mathbf{x}_S = (\mathbf{D}_S^T \mathbf{D}_S)^{-1} \mathbf{D}_S^T \mathbf{y};$
 - 6 Atualize o resíduo $\mathbf{r} = \mathbf{y} - \mathbf{D}_S \mathbf{x}_S;$
 - 7 **fim**
 - 8 **retorna** (S, \mathbf{x}_S)
-

No Algoritmo 1, a condição de permanência no laço principal do algoritmo OMP contém, além da verificação se o suporte tem o tamanho igual ao nível de esparsidade desejado, a avaliação de uma condição de parada adicional, representada pela avaliação da função *seguirAFrente*, que poderia ser, por exemplo, se o módulo do resíduo está abaixo de um limiar previamente fornecido (ocultado da lista de parâmetros de entrada).

Para o caso exato, sem ruído, em Dumitrescu e Irofti (2018) afirma-se que, exceto no caso degenerado em que o espaço gerado pelas colunas de $\mathbf{D} \in \mathbb{R}^{m \times n}$ não for \mathbb{R}^m , o algoritmo OMP obtém a solução exata se o critério de verificação do resíduo não for atendido antes.

O ajuste da solução ao suporte obtido até então (linha 5) por mínimos

quadrados ordinário, concretizado pela equação normal, é o ponto de distinção entre este algoritmo e aquele do qual ele deriva (*Matching Pursuit*, (Rish e Grabarnik, 2015)).

2.1.9 ISTA

O algoritmo ISTA (*Iterative Shrinkage-Thresholding Algorithm* ou, em tradução livre, algoritmo iterativo de achatamento e limiarização) é uma estratégia clássica para realizar a codificação esparsa. O ISTA resolve o problema na Equação 2-15 e pode ser derivado a partir da aplicação de um operador de achatamento ao resultado de um passo de otimização do algoritmo de Gradiente Descendente (Beck e Teboulle, 2009)¹⁰:

Algoritmo 2: ISTA

Entrada: Matriz dicionário $\mathbf{D} \in \mathbb{R}^{m \times n}$

Função de erro de aproximação $\mathbf{F}(\mathbf{x}) = \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2$

Gradiente $\nabla \mathbf{F}(\mathbf{x}) = 2\mathbf{D}^T(\mathbf{D}\mathbf{x} - \mathbf{y})$

Constante de Lipschitz L relativa a ∇F

Parâmetro de regularização λ

Vetor de entrada $\mathbf{y} \in \mathbb{R}^m$

Operador de achatamento e limiarização \mathcal{T}

Saída: Vetor de saída \mathbf{x}

1 $q = 0$;

2 $\mathbf{x}_{(0)} = \mathbf{0}$;

3 **enquanto** *não convergir* **faça**

4 $q = q + 1$;

5 $\mathbf{x}_{(q)} = \mathcal{T}_{\lambda, \frac{1}{L}}\left(\mathbf{x}_{(q-1)} - \frac{1}{L}\nabla \mathbf{F}(\mathbf{x}_{(q-1)})\right)$;

6 **retorna** *melhor solução até o momento* $\mathbf{x}_{(q)}$;

em que $\mathcal{T}_{\lambda, \gamma}$ é o operador de achatamento, dado por:

$$\mathcal{T}_{\lambda, \gamma}(x_i) = \begin{cases} x_i - (\lambda\gamma) & , \text{ se } x_i > \lambda\gamma, \\ 0 & , \text{ se } |x_i| \leq \lambda\gamma, \\ x_i + (\lambda\gamma) & , \text{ se } x_i < -\lambda\gamma, \end{cases} \quad (2-16)$$

O mesmo operador pode ser simplificado para $\mathcal{T}_{\lambda, \gamma}(x) = \text{sinal}(x)(|x| - \lambda\gamma)_+$, em que “sinal(x)” corresponde ao sinal do operando (+1, se positivo; -1, se negativo), e $(x)_+$ corresponde à parte maior do que zero do operando x .

Como em Beck e Teboulle (2009), assume-se:

¹⁰Vide referências citadas no referido trabalho.

- (a) A parcela da regularização da função de custo de otimização (a que penaliza o próprio vetor de solução) é uma função convexa, possivelmente não suave¹¹;
- (b) A parcela do erro de aproximação da função de custo de otimização (a que calcula a norma quadrática da diferença entre o vetor original e o produto do dicionário pela solução atualmente encontrada) é suave, continuamente diferenciável com gradiente $L(\mathbf{F})$ Lipschitz-contínuo¹²:

$$\|\nabla F(\mathbf{a}) - \nabla F(\mathbf{b})\| \leq L(\mathbf{F})\|\mathbf{a} - \mathbf{b}\|, \forall \mathbf{a}, \mathbf{b} \in \mathbb{R}^n.$$

Para que a apresentação fique mais simples e didática, faremos uso de uma constante L em substituição à função $L(\mathbf{F})$, cujo valor mínimo é $2\lambda_{max}(\mathbf{D}^T\mathbf{D})$, em que λ_{max} significa o maior autovalor associado à matriz $\mathbf{D}^T\mathbf{D}$.

ISTA é um dos algoritmos capazes de resolver o Lasso. Um algoritmo derivado do ISTA, o FISTA, tem convergência provada mais rápida (Beck e Teboulle, 2009), e é apresentado a seguir.

2.1.10 FISTA

O algoritmo FISTA (*Fast Iterative Shrinkage-Thresholding Algorithm* ou, em tradução livre, algoritmo rápido iterativo de achatamento e limiarização) (Beck e Teboulle, 2009) é outro algoritmo clássico para realizar a codificação esparsa, dado que tenha um dicionário. Em Beck e Teboulle (2009) há prova de convergência comparando este algoritmo com o ISTA. A principal mudança está no uso de suas soluções parciais anteriores para se chegar na próxima. Assumindo-se as condições (a) e (b) descritas anteriormente para o ISTA, o FISTA pode ser descrito como no Algoritmo 10.

2.1.11 CD

A estratégia de otimização *Coordinate Descent* (ou coordenada descendente) (CD) é uma abordagem simples de implementar, eficiente e efetiva quando de sua aplicação específica para resolver o problema do Lasso (Rish e Grabarnik, 2015). A técnica funciona deixando todas as variáveis (coordenadas) fixas exceto uma, sobre a qual se aplica uma otimização. Proceda-se

¹¹Sua primeira derivada pode não existir em algum ponto. Ocorre, por exemplo, com a função $f(x) = |x|$ (valor absoluto de x), que é contínua mas não é continuamente diferenciável quando $x = 0$.

¹²Vide conceitos de subgradiente e subdiferencial em Izmailov e Solodov (2007), Cap. 1, seção 1.3 (Convexidade).

Algoritmo 3: FISTA

Entrada: Matriz dicionário $\mathbf{D} \in \mathbb{R}^{m \times n}$
 Função de erro de aproximação $\mathbf{F}(\mathbf{x}) = \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2$
 Gradiente $\nabla \mathbf{F}(\mathbf{x}) = 2\mathbf{D}^T(\mathbf{D}\mathbf{x} - \mathbf{y})$
 Constante de Lipschitz L relativa a ∇F
 Parâmetro de regularização λ
 Vetor de entrada $\mathbf{y} \in \mathbb{R}^m$
 Operador de achatamento e limiarização \mathcal{T} como no ISTA

Saída: Vetor de saída \mathbf{x}

```

1  $q = 0$ ;
2  $\mathbf{x}_{(0)} = \mathbf{0}$ ;
3  $\mathbf{w}_{(1)} = \mathbf{x}_{(0)}$ ;
4  $t_{(1)} = 1$ ;
5 enquanto não convergir faça
6    $q = q + 1$ ;
7    $\mathbf{x}_{(q)} = \mathcal{T}_{\lambda, \frac{1}{L}} \left( \mathbf{w}_{(q-1)} - \frac{1}{L} \nabla \mathbf{F}(\mathbf{w}_{(q-1)}) \right)$ ;
8    $t_{(q+1)} = \frac{1 + \sqrt{1 + 4t_{(q)}^2}}{2}$ ;
9    $\mathbf{w}_{(q+1)} = \mathbf{x}_{(q)} + \left( \frac{t_{(q)} - 1}{t_{(q+1)}} \right) (\mathbf{x}_{(q+1)} - \mathbf{x}_{(q)})$ ;
10 retorna melhor solução até o momento  $\mathbf{x}_{(q)}$ ;
```

com a otimização de cada uma das variáveis de forma cíclica até que se atenda algum critério de convergência.

O funcionamento desta técnica pode ser resumido pelo Algoritmo 4.

A cada iteração, um índice j é selecionado ciclicamente (linha 4, usando o operador de resto da divisão “mod”), e o contador q é atualizado. Nas linhas 6 e 7, cada dimensão do resíduo é atualizada. Na linha 8, a variável v guarda o produto escalar da j -ésima coluna do dicionário com o resíduo atualizado, representando a solução de mínimos quadrados univariada. Sobre v , na linha 9, aplica-se o operador de achatamento e limiarização e se guarda o resultado na j -ésima dimensão do vetor de resposta \mathbf{x} . Ao final, o resíduo é avaliado para decidir sobre a necessidade de uma nova iteração. A versão geral, não específica para o Lasso, é alcançada eliminando-se a limiarização e achatamento na linha 9, e atribuindo diretamente a variável v à dimensão a atualizar no vetor de solução \mathbf{x} .

Algoritmo 4: CD - *Coordinate Descent* para Lasso

Entrada: Matriz dicionário $\mathbf{D} \in \mathbb{R}^{m \times n}$
 Parâmetro de regularização λ
 Vetor de entrada $\mathbf{y} \in \mathbb{R}^m$
 Limiar como critério de parada ϵ

Saída: Vetor de saída $\mathbf{x} \in \mathbb{R}^n$

```

1  $q = 0$ ;
2  $\mathbf{x} = \mathbf{0}$ ;
3 Resíduo  $\mathbf{r} = \mathbf{0}$ ,  $\mathbf{r} \in \mathbb{R}^m$ ;
4 enquanto não convergir faça
5    $j = (q \bmod n) + 1$ ;
6    $q = q + 1$ ;
7   para  $i = 1..m$  faça
8      $\mathbf{r}_i = \mathbf{y}_i - \sum_{w \neq j} d_{i,w} x_w$ ;
9      $v = \mathbf{D}_{:,j}^T \mathbf{r}$ ;
10     $x_j = \text{sinal}(v)(|v| - \lambda)_+$ ;
11    se  $\|\mathbf{r} - \mathbf{D}_{:,j} x_j\| < \epsilon$  então
12       $\perp$  Convergiu. Saia do laço.
13 retorna melhor solução até o momento  $\mathbf{x}$ ;
```

2.1.12**Treinamento de dicionário**

Podemos entender treinamento de dicionário (TD) como o processo pelo qual calculamos um conjunto de colunas de uma matriz que formam a uma base sobrecompleta de vetores¹³. Se formos calcular o produto escalar máximo entre distintas colunas de um dicionário padrão – com mais colunas do que linhas –, ele será estritamente maior que zero. Dois dos algoritmos mais conhecidos para TD são o MOD (de Método das Direções Ótimas, ou *Method of the Optimal Directions*), explicado brevemente a seguir, e o K-SVD (de Decomposição em K Valores Singulares, ou *K Singular Values Decomposition*) (Aharon, Elad e Bruckstein, 2006).

2.1.13**Método das Direções Ótimas**

O Método das Direções Ótimas (MOD) (Engan, Aase e Husoy, 1999) pode ser resumido como a solução algébrica de um problema de mínimos quadrados ordinário em que o conjunto de variáveis a otimizar corresponde aos

¹³Chama-se sobrecompleta porque a quantidade de colunas é maior que a quantidade de linhas. Uma nomenclatura mais correta levaria em conta não a quantidade de linhas e colunas, mas sim a quantidade de colunas superior ao posto da matriz.

átomos do dicionário sendo buscado. Sejam \mathbf{D} um dicionário; \mathbf{Y} o conjunto dos exemplos, um exemplo por coluna, cada linha correspondendo a uma dimensão de um exemplo; \mathbf{X} a matriz de codificação esparsa de \mathbf{Y} usando \mathbf{D} . Tomando o erro da codificação esparsa \mathbf{E} como $\mathbf{E} = \mathbf{Y} - \mathbf{DX}$, o MOD tem por finalidade minimizar $\|\mathbf{E}\|_F^2 = \|\mathbf{Y} - \mathbf{DX}\|_F^2 = (\mathbf{Y} - \mathbf{DX})^T (\mathbf{Y} - \mathbf{DX})$. Tomando a derivada desta última expressão em relação ao que queremos otimizar, a matriz \mathbf{D} , e observando a regra adequada em Petersen e Pedersen (2012), chegamos em:

$$\frac{\partial}{\partial \mathbf{D}} \left((\mathbf{Y} - \mathbf{DX})^T (\mathbf{Y} - \mathbf{DX}) \right) = -2(\mathbf{Y} - \mathbf{DX})\mathbf{X}^T. \quad (2-17)$$

Igualando a zero a Equação 2-17, chegamos em $\mathbf{D}_{(t+1)} = \mathbf{Y}\mathbf{X}_{(t)}(\mathbf{X}_{(t)}\mathbf{X}_{(t)}^T)^{-1}$, com o subscrito evidenciando que o dicionário da próxima iteração $\mathbf{D}_{(t+1)}$ sofre a influência da matriz de codificação esparsa $\mathbf{X}_{(t)}$ da iteração anterior. Como observado em Dumitrescu e Irofti (2018), os autores do trabalho original não se utilizaram da álgebra de matrizes diretamente, mas sim dos itens em cada uma das matrizes envolvidas no processo. Nota-se também que não é necessário que os átomos tenham tamanho unitário¹⁴, e se deixa este requisito por conta do algoritmo de codificação esparsa ou da aplicação específica.

2.2

Amostragem Comprimida - *Compressed Sampling*

Cabe escrever um pouco sobre Amostragem Comprimida¹⁵. Esta área do conhecimento tem alta simbiose com a área de modelagem esparsa, aproveitando desta última todo desenvolvimento. Uma das ideias mais estimulantes da amostragem comprimida está na não necessidade de se fazer amostragem segundo o teorema de Shannon-Nyquist, em que se diz que para se recuperar perfeitamente um sinal, é requerido que façamos uma amostragem regular com o dobro da largura de banda do sinal original. A área de amostragem comprimida estabelece um conjunto de teoremas sobre a matriz dicionário, sobre o vetor original e sobre esparsidade do vetor de representação esparsa de tal forma que, atendidas as condições, pode-se recuperar o sinal original usando uma amostragem muito inferior à estabelecida por Shannon-Nyquist. Isto é especialmente valioso para áreas em que a aquisição e visualização de dados são caras, difíceis ou arriscadas como, por exemplo, ressonância magnética (Chen e Huang, 2014), (Lustig, Donoho e Pauly, 2007), aquisição sísmica (Vera Ro-

¹⁴Norma ℓ_2 igual a 1.

¹⁵O termo é comumente referenciado por *compressive sensing*, *compressive sampling*, *compressed sensing* ou *compressed sampling*.

driguez, Sacchi e Gu, 2012), (Villarreal et al., 2017) e astrofísica (Honma et al., 2014), (Akiyama et al., 2019).

Para aprofundamento neste assunto, do qual não tratamos diretamente nesta tese, o leitor é convidado a ler Foucart e Rauhut (2013) e Cheng (2016)¹⁶.

2.3

Trabalhos Relacionados

Como o problema de pesquisa da tese agrega mais de um assunto, agrupamos os trabalhos relacionados em temas: deconvolução e inversão, compressão volumétrica e aplicações de modelagem esparsa em geociências.

2.3.1

Deconvolução e Inversão

Em Liu, Fomel e Chen (2011) os autores definem um mapa que vincula tempo e frequência e associa os valores neste mapa aos coeficientes (os componentes do vetor de valores) de uma ondaleta variante no tempo. Sua abordagem depende da utilização do trabalho em Fomel (2007), que propõe uma regularização levando em conta a forma de curvas, com a intenção de impor continuidade e suavidade à curva (ondaleta) final. Tal regularização foi uma das motivações para implementarmos a regularização por átomo de dicionário que propomos no Capítulo 5.

Em Wang et al. (2016) são desenvolvidos algoritmos para, ao mesmo tempo, (i) extrair uma única ondaleta e (ii) um conjunto de refletores. Unindo ondaleta e refletores, via convolução, seria possível aproximar o dado de amplitudes sísmicas. Os autores se utilizaram do pressuposto do modelo convolucional e de a ondaleta única ser estacionária, que não deve sofrer alterações ao longo da profundidade. O trabalho se desenvolve fazendo uso de algoritmos como o FISTA para selecionar os picos de refletividade ao longo dos traços e Fused-Lasso¹⁷ para calcular uma ondaleta, num esquema de otimização em duas etapas. O Capítulo 3 mostra com maiores detalhes o desenvolvimento feito a partir deste trabalho e o avanço proposto. Os resultados mostrados no Capítulo 6 atestam que nossa abordagem fornece melhorias concretas em relação à abordagem de representação esparsa mostrada em Wang et al. (2016).

Em Pereira-Dias et al. (2017) é feita aplicação de modelagem esparsa para obter uma migração no domínio da imagem, com resolução aumentada, utilizando restrições de esparsidade, ampliando a largura de banda do espectro do dado original. Os autores também enfatizam que levaram em conta a natureza

¹⁶Capítulo 2.

¹⁷Vide Hoefling (2010).

não estacionária da aplicação das funções de espalhamento de pontos, usadas em processamento de imagens, para restaurar um conteúdo de refletividades.

Em Sui e Ma (2019), os autores promoveram uma extensão do método em Wang et al. (2016) através da descoberta de um fator de qualidade que altera uma matriz, com uma estrutura particular, contendo uma ondaleta inicial para agregar fenômenos de atenuação e dispersão. Há semelhanças deste recente trabalho com a tese aqui desenvolvida. Os autores também se utilizaram da aceleração de desempenho promovida pelo mesmo algoritmo de representação esparsa utilizado nesta tese (OMP) e também fazem uso dos conceitos de atenuação e dispersão usando uma teoria similar à que foi usada nesta tese. Dentre as distinções com esta tese, podemos afirmar que os autores não exploraram o potencial da técnica de treinamento de dicionário para extrair uma representação de ondaleta do dado original como feito aqui, não deixaram livre a utilização de diversos valores de fator de qualidade para observar qual ou quais valores são mais relevantes em regiões distintas do dado a mudar de representação, e não deixaram clara a necessidade de se fazer uma amostragem mais rica do que, a priori, uma amostragem regular no intuito de acelerar o mecanismo de extração de ondaleta.

2.3.2

Compressão Volumétrica

Em Gobbetti, Guitián e Marton (2012) há a apresentação de uma abordagem chamada COVRA para compressão e visualização baseada no conceito de *coresets* (Agarwal, Har-Peled e Varadarajan, 2005), em que utilizam um algoritmo de TD (K-SVD (Aharon, Elad e Bruckstein, 2006)) sobre dados volumétricos, incluindo volumes variantes no tempo, particionados em blocos pequenos para servirem como amostras de entrada, cuja reconstrução para visualização ocorre em processadores gráficos. Resultados mostram alto grau de compressão mantendo qualidade razoável de visualização, mas admite perda, sendo o objetivo maior a rapidez na visualização. Em seguida, em Rodríguez et al. (2014) é feita uma revisão de literatura sobre compressão volumétrica em que o trabalho em Gobbetti, Guitián e Marton (2012) está inserido e são comparadas abordagens com dicionários prontos, derivados de transformadas como Fourier, cosseno, Wavelet e outras. Resultados comparativos envolvendo largura de banda e memória gráfica necessárias, e escalabilidade de qualidade usando abordagem de nível de detalhes favorecem a abordagem em Gobbetti, Guitián e Marton (2012) dentre outras citadas.

Em Fout e Ma (2012) há a apresentação de um mecanismo de compressão volumétrica de dados em ponto flutuante que se baseia numa lista de preditores,

cada qual com um identificador específico e uma forma de realizar a predição polinomial. Os autores acabam por fazer uma compressão cujo algoritmo é escolhido como sendo o que menos resíduo gera e, ao mesmo tempo, se preocupam em codificar o resíduo gerado. Este trabalho serviu como fonte de inspiração e de software de compressão de ponto flutuante que chegamos a experimentar ao final dos processos desenvolvidos nesta tese partindo de Burtcher (2009) e Lindstrom e Isenburg (2006).

2.3.3

Aplicações de Modelagem Esparsa em Geociências

Em Zhou et al. (2013) e Zhou, Chen e Gao (2014) é desfeita a sobreposição de registros em tempos próximos de dados pré-empilhados para que se recupere um dado sísmico pré-empilhado de um único receptor. Há o emprego de TD para favorecer tanto uma representação mais esparsa quanto um conteúdo extraído com razão sinal ruído superior em relação à utilização de um dicionário fixo de *curvelets*. No primeiro trabalho foi feito uso de OMP e no segundo foi utilizada a regressão de ângulo mínimo para o Lasso.

Em Beckouche e Ma (2014), os autores fazem uso de técnicas de treinamento de dicionário e recuperação esparsa para atenuar ruído em imagens sísmicas. Os autores usam explicitamente o método das direções ótimas para treinar o dicionário e o OMP para realizar a recuperação esparsa. O dado é subdividido em retalhos quadrados, e não em traços, e usado para treinar um dicionário. Experimentos realizados mostram uma qualidade superior na reconstrução do dado original. Os autores parecem ter aplicado os mesmos princípios de representação esparsa e TD aplicados para imagens convencionais, nas quais não há distinção a priori quanto a características variáveis com a posição original dos exemplos usados no treinamento. Esta abordagem não parece adequada se a intenção for a de extrair estrutura de picos esparsos de refletividade, mas demonstra o poder, mesmo de forma genérica, dos processos de modelagem esparsa.

Em Boßmann e Ma (2015) há o uso de um modelo gerador de curvas oscilatórias, separadas em fase e envelope, capazes de melhor representar, de forma esparsa, um dado sísmico utilizado nos experimentos. Neste trabalho, leva-se em conta a característica de que as ondas sísmicas podem ser não estacionárias, ou seja, podem sofrer efeitos de atenuação e distorção com o aumento da profundidade. Este trabalho acaba por fazer uma adaptação do algoritmo OMP, fazendo-o incorporar a base da representação esparsa.

Em Tian e Li (2017) os autores propõem um mecanismo de compressão de dados sísmicos, subdivididos em retalhos. Os retalhos são agrupados e usados

como entrada para um processo de TD *online*. Os autores argumentam que poderiam gerar mais de um dicionário por grupo, o que melhoraria a qualidade da reconstrução do dado original. Para finalizar o processo de compressão, os autores se utilizam de quantização uniforme e codificação aritmética (Sayood, 2012)¹⁸.

¹⁸Capítulo 4

3 Deconvolução Sísmica

Neste capítulo, apresentamos uma abordagem que se propõe a resolver o problema de deconvolução do dado de amplitudes sísmicas, mais especificamente o de deconvolução por picos esparsos de refletividades.

Como um modelo matemático que tenta representar um fenômeno físico subjacente, há de se aceitar ou impor um conjunto de restrições ou suposições. Para o caso presente, parte-se da restrição de que cada traço vertical disponível num dado volumétrico de amplitudes sísmicas é, por um lado, resultado da convolução de uma ondaleta com um conjunto de picos, ou deltas de Dirac e, por outro lado, pode-se impor ou supor que haja limitação na quantidade ou na posição relativa destes picos. A ondaleta representaria, num curto intervalo de tempo, o comportamento da amplitude de uma onda sísmica emitida por uma fonte. A natureza desta fonte emissora de energia sísmica varia de acordo com o local do levantamento (se em terra, se no mar). Os picos mencionados representariam coeficientes de reflexão, e marcariam a transição entre camadas de solo (terrestre ou abaixo do leito do oceano) com distintas características de densidade e de velocidade de propagação de sismos Onajite (2014).

A Figura 3.1 ilustra uma ondaleta sintética e a Figura 3.2 ilustra um perfil sintético de refletividades, a convolução com a ondaleta da figura anterior e o resultado final com adição de ruído gaussiano¹.

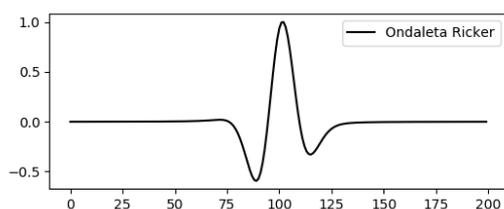


Figura 3.1: Ondaleta Ricker com frequência dominante de 30Hz, e ângulo (fase) de 30 graus.

Por definição, deconvolução em picos esparsos é um processo usado para separar o conteúdo de refletividade entre as camadas do solo e o conteúdo da ondaleta Onajite (2014). Neste processo de separação, as duas partes componentes, refletividade e ondaleta, não estão presentes de forma pura.

¹No Apêndice B, há o código-fonte B.1 que gera a Figura 3.1.

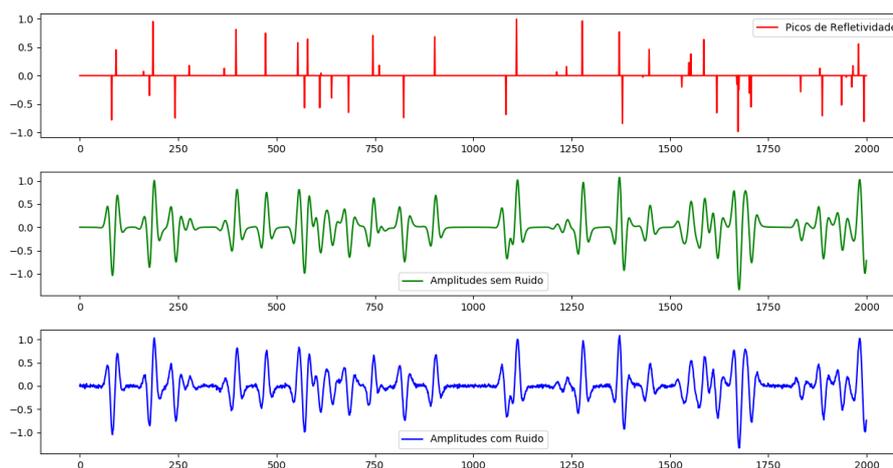


Figura 3.2: De cima para baixo, um perfil de refletividades, a convolução com a ondaleta da Figura 3.1 e adição final de ruído gaussiano (com valor absoluto máximo de 0,02).

Se há a presença de ruído, a deconvolução se configura como um problema inverso de aproximação, partindo do pressuposto de que não é possível modelar precisamente o ruído.

Na determinação aproximada dos picos esparsos, há pelo menos quatro variáveis de interesse Fernandes, Lopes e Gattass (2017), Velis (2008):

- a **quantidade** de picos, que é a medida da esparsidade do sinal extraído, cuja característica almejada é o de ter uma maior largura de banda do espectro de amplitude, derivado da transformada de Fourier;
- a **posição** em que tais picos ocorrem. Ao unir vários traços adjacentes depois de terem passado por deconvolução, é possível que fique mais clara a presença de horizontes sísmicos, o que pode facilitar um trabalho de interpretação por especialista ou a execução de um algoritmo de rastreamento de horizonte sísmico;
- a **magnitude** de tais picos, que deriva do grau de dissimilaridade de características físicas entre níveis verticalmente adjacentes de subsolo; e
- a **fase**, ou o sinal de tais picos, que pode refletir a variação positiva ou negativa do coeficiente de refletividade, podendo indicar, assim, uma região de transição, evidenciando corpos de interesse como, por exemplo, regiões de aprisionamento de hidrocarbonetos ou corpos salinos.

A deconvolução por picos esparsos é uma das estratégias possíveis para desfazer a fusão ou mistura entre ondaleta e perfil de refletividade em traços verticais.

Para facilitar o entendimento do processo, tomemos por base o modelo sintético Marmousi2 Martin, Wiley e Marfurt (2006). Neste modelo, há 2801 amostras na vertical e 13601 amostras na horizontal. Nas duas direções, a amostragem é de 1,25m por amostra, perfazendo 3500m de profundidade e 17000m de largura. A Figura 3.3 apresenta o conteúdo de velocidade de propagação de ondas sísmicas; a Figura 3.4 apresenta as densidades no material rochoso; a Figura 3.5 mostra conteúdo de refletividade como resultado da aplicação direta da fórmula na Equação 2-1.

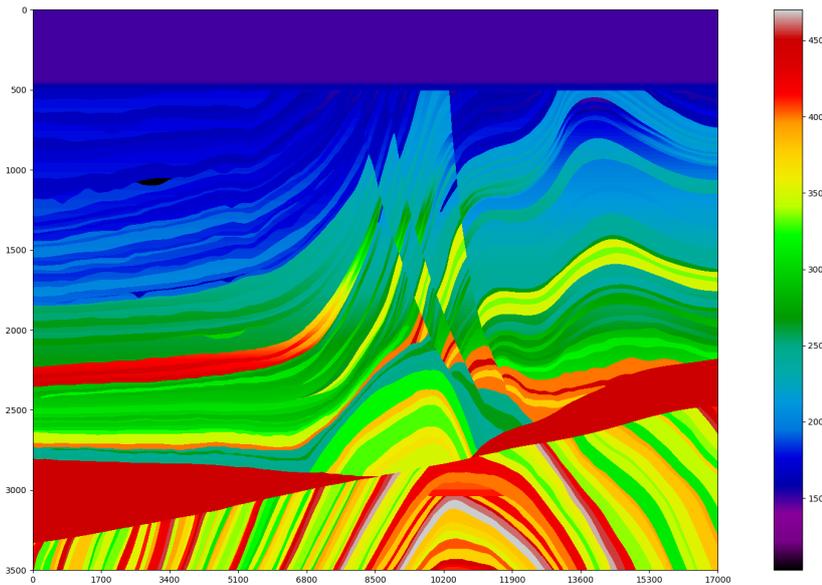


Figura 3.3: Velocidades de propagação de ondas sísmicas do modelo Marmousi2. Eixo horizontal: posição em metros. Eixo vertical: profundidade em metros. Velocidades em metros/segundo com escala exibida à direita.

De posse da informação de refletividade, é possível gerar um conteúdo de amplitudes sísmicas. Partindo do pressuposto do modelo convolucional, toma-se uma ondaleta como sendo o pulso emitido por uma fonte sísmica, faz-se a convolução de tal ondaleta com o conteúdo de refletividade e, ao final, adiciona-se ruído, o que pode resultar num conteúdo similar ao exibido pela Figura 3.6.

Ocorre que, em casos reais, o que se tem é um dado de amplitudes sísmicas a partir do qual se deseja extrair o conteúdo oscilatório e o conteúdo hipoteticamente esparsos de refletividades. O processo inverso, a depender do modelo de trabalho, impõe restrições. O modelo convolucional, por exemplo, impõe que a ondaleta tenha um caráter estacionário em sua zona de atuação. Esta afirmativa só parece ser válida num intervalo normalmente pequeno de profundidade, algo em torno de menos de 1000m, ou um equivalente em tempo se houver um modelo de velocidade estimado em mãos.

Nesta abordagem de deconvolução por picos esparsos, quanto melhor for o posicionamento dos refletores, melhor poderemos descrever e entender

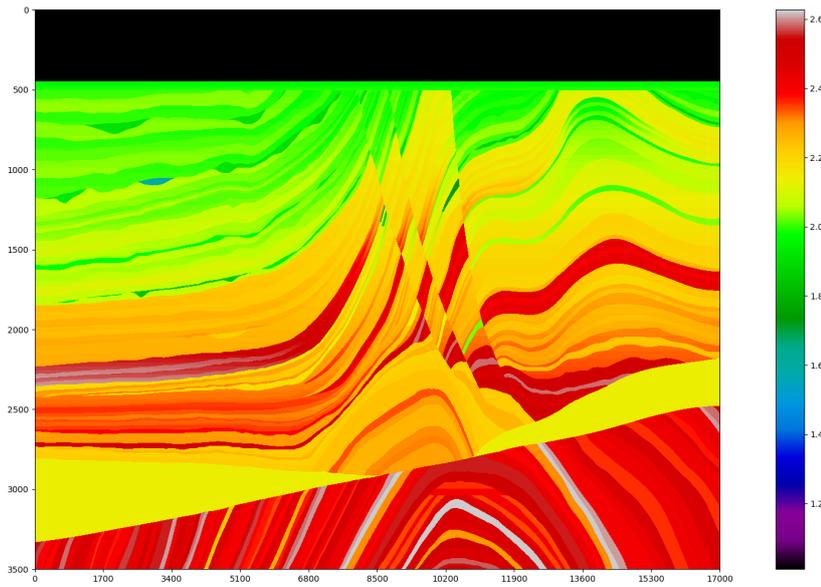


Figura 3.4: Densidades do modelo Marmousi2. Eixo horizontal: posição em metros. Eixo vertical: profundidade em metros. Densidade em gramas/mililitro com escala exibida à direita.

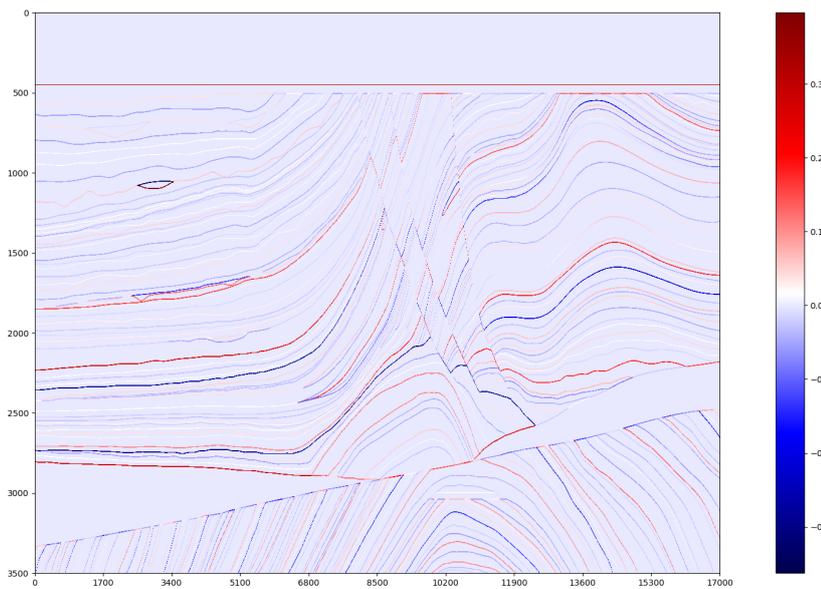


Figura 3.5: Refletividade calculada a partir de densidade e velocidade do modelo Marmousi2. Eixo horizontal: posição em metros. Eixo vertical: profundidade em metros. Escala exibida à direita.

PUC-Rio - Certificação Digital N° 1421617/CA

a geologia da região estudada Fernandes, Lopes e Gattass (2017). Em Velis (2008) há a afirmação de que a busca por armadilhas sutis de hidrocarbonetos tinha se tornado uma tarefa importante em exploração sísmica.

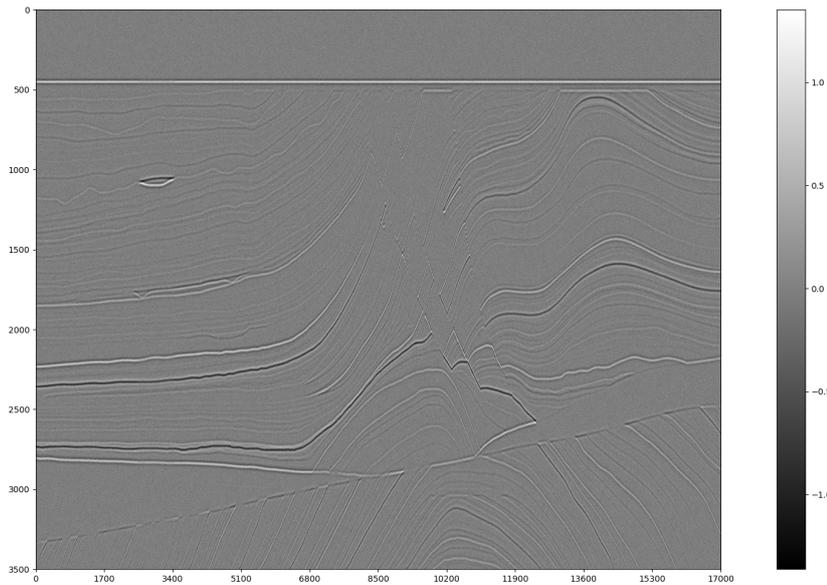


Figura 3.6: Amplitudes sísmicas partindo do conteúdo de refletividades do modelo Marmousi2 da Figura 3.5. Cada traço sísmico é resultado da convolução de uma ondaleta Ricker (frequência dominante de 30Hz, ângulo de 30 graus) com a refletividade, e adiciona-se ruído. Eixo horizontal: posição em metros. Eixo vertical: profundidade em metros.

3.1 Relação com codificação esparsa

Para estabelecer um elo entre a descoberta de uma ondaleta, um pulso, para convolução e algoritmos de treinamento de dicionário, é necessário mostrar o que vem a ser uma matriz de Toeplitz. Se \mathbf{D} é uma matriz Toeplitz, então $d_{i,j} = d_{i-1,j-1}$, sendo i e j os índices das linhas e colunas, respectivamente, com $i > 1$ e $j > 1$. Dito de outro modo, numa matriz de Toeplitz basta se fazer o preenchimento da primeira linha e da primeira coluna; os demais elementos seguem a regra $d_{i,j} = d_{i-1,j-1}$, com $i > 1$ e $j > 1$. A Figura 3.7 ilustra uma matriz de Toeplitz, em que as cores diferentes representam valores diferentes.

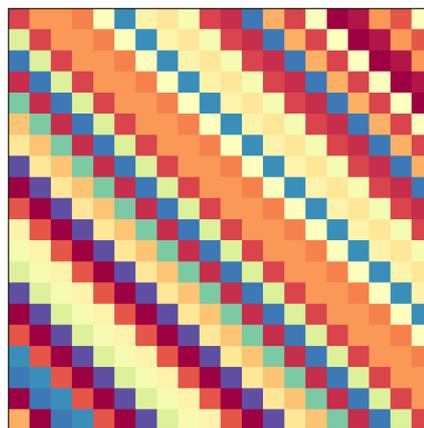


Figura 3.7: Ilustração de uma matriz de Toeplitz.

Agora é possível estabelecer uma relação prática entre o processo de extração de pulso e refletividades a partir das amplitudes sísmicas e algoritmos de treinamento de dicionário. Tomemos, então, num primeiro momento, um dicionário que tem como características (i) ser uma matriz quadrada, de dimensão igual ao número de amostras dos traços e (ii) ter uma estrutura interna de uma matriz de Toeplitz. Tais características conferem a esta matriz a característica de ser um operador de convolução. Ao aplicar tal matriz a um traço de picos de refletividade, obteríamos um conteúdo de amplitudes sísmicas. Num segundo momento, mantendo fixo o operador de convolução, codificado na matriz-dicionário, poderíamos nos utilizar de algoritmos para computação de representação esparsa como OMP ou FISTA para obter um vetor esparsa de refletividades que melhor aproximasse um traço de amplitudes. Ao mantermos o conteúdo de refletividade fixo, o processo de estimação de uma ondaleta com a qual se possa, via convolução, recuperar o conteúdo de amplitudes é um caso particular um treinamento de dicionário, em que uma matriz-dicionário tem a estrutura de uma matriz de Toeplitz.

3.2

Uma solução proposta: Lobbés

Lobbés, um acrônimo para *Lasso-based binary search for parameter selection*, foi o nome dado a uma estratégia que realiza uma busca pelo valor de um parâmetro de modelo que interfere na esparsidade da solução.

Em modelagem esparsa, normalmente se aplica um tipo de normalização sobre a matriz-dicionário. Tal normalização se concretiza, por exemplo, aplicando uma ou mais das seguintes transformações aos átomos de um dicionário:

- Todos os átomos, ou colunas da matriz-dicionário, devem ter norma euclideana igual a 1;
- Os valores do vetor de cada átomo têm média igual 0;
- Os valores do vetor de cada átomo têm variância igual a 1;
- Os valores do vetor de cada átomo estão confinados ao intervalo $[-1,1]$ escalado pelo inverso do maior valor absoluto.

A normalização que se propõe aplicar no contexto do **Lobbés** é a mencionada em Hastie, Tibshirani e Wainwright (2015)². Sejam $\mathbf{A} \in \mathbb{R}^{m \times n}$ uma matriz dicionário e $\mathbf{y} \in \mathbb{R}^{m \times 1}$ um vetor de entrada. Daí, a normalização utilizada tem os seguintes efeitos sobre os elementos algébricos:

²Cap. 2, páginas 8 e 9.

- Cada átomo $\mathbf{a}_j \in \mathbb{R}^m$ da matriz dicionário \mathbf{A} é mudado para ter média 0 ($\frac{1}{m} \sum_{i=1}^m a_{i,j} = 0$) e variância 1 ($\frac{1}{m} \sum_{i=1}^m a_{i,j}^2 = 1$);
- O vetor de entrada é mudado para ter média 0 ($\frac{1}{n} \sum_{i=1}^n y_{i,1} = 0$).

Sem esta normalização, a solução gerada pelo Lasso fica na dependência das unidades utilizadas para medir os preditores (as dimensões ou colunas da matriz dicionário). A normalização traz o benefício de revelar o valor máximo do parâmetro λ da Equação 2-15 acima do qual o vetor de solução é anulado. Assim, tendo um valor limite do parâmetro de regularização após a transformação da matriz-dicionário, pode-se realizar uma busca pelo valor do parâmetro de regularização que, ao final, pode resultar numa solução cuja quantidade de dimensões não zero seja a desejada.

3.2.1

Um arcabouço para deconvolução

Num procedimento convencional de treinamento de dicionário, ocorre a descoberta de um operador linear com o qual é feita uma codificação esparsa de um conteúdo de entrada. Tal operador linear, neste contexto, é uma matriz de Toeplitz que codifica uma convolução com uma ondaleta única. O conteúdo esparsa gerado pela recodificação é o conteúdo de refletividades, e o conteúdo de entrada é o dado de amplitudes sísmicas.

Em Wang et al. (2016) há a apresentação de um algoritmo para fatoração do conteúdo de amplitudes descrito parcialmente no Algoritmo 5.

O vetor $\boldsymbol{\alpha}$ representa a ondaleta. Fazendo uso da notação em Wang et al. (2016), $\boldsymbol{\alpha} = (a_{-(n-1)}, \dots, a_0, \dots, a_{n-1})^T$, e a matriz de Toeplitz \mathbf{A} associada a $\boldsymbol{\alpha}$ tem a seguinte configuração:

$$\mathbf{A} = \begin{pmatrix} a_0 & a_1 & \cdots & a_{n-2} & a_{n-1} \\ a_{-1} & a_0 & \cdots & a_{n-3} & a_{n-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{-(n-2)} & a_{-(n-3)} & \cdots & a_0 & a_1 \\ a_{-(n-1)} & a_{-(n-2)} & \cdots & a_{-1} & a_0 \end{pmatrix} \quad (3-1)$$

Na linha 6 do Algoritmo 5, as matrizes I com subscritos se referem à matriz

identidade com deslocamento da diagonal. Sendo a identidade $I \in \mathbb{R}^{n \times n}$, i e j índices de linhas e colunas, respectivamente, a matriz I_w , com $w \in [-(n-1), (n-1)]$, $w \in \mathbb{Z}$, tem, em seu conteúdo, o valor 1 quando $|j - i| = w$, e 0 nas demais posições.

Na mesma linha 6 usa-se o operador vec , que transforma uma matriz com m linhas e n colunas em uma matriz com mn linhas e 1 coluna, empilhando

Algoritmo 5: Fatoração de matriz de amplitude Wang et al. (2016)

Entrada: Matriz de Toeplitz $\mathbf{A}^{(0)} \in \mathbb{R}^{n \times n}$ inicializada
 Matriz com sinais de entrada $\mathbf{B} \in \mathbb{R}^{m \times n}$
 Esparsidade k
 Parâmetros de penalização β_0, β_1

Saída: Matriz de Toeplitz \mathbf{A}
 Matriz esparsa \mathbf{R} com as refletividades

```

1  $t = 1$ ;
2 enquanto não convergir faça
3   para cada coluna  $\mathbf{b}_j$  de  $\mathbf{B}$  faça
4      $\mathbf{r}_j^{(t)} = \arg \min_{\mathbf{r}_j} \frac{1}{2} \|\mathbf{b}_j - \mathbf{A}^{(t-1)} \mathbf{r}_j\|_2^2 + \lambda_k \|\mathbf{r}_j\|_1$ ;
5      $\mathbf{R}^{(t)} = (r_1^{(t)}, \dots, r_n^{(t)})$ ;
6      $\tilde{\mathbf{R}}^{(t)} = (\text{vec}(I_{-(n-1)} \mathbf{R}^{(t)}), \dots, \text{vec}(I_{(n-1)} \mathbf{R}^{(t)}))$ ;
7      $\tilde{\mathbf{b}} = \text{vec}(\mathbf{B})$ ;
8      $\boldsymbol{\alpha}^{(t)} =$ 
        $\arg \min_{\boldsymbol{\alpha}} \frac{1}{2} \|\tilde{\mathbf{b}}_j - \tilde{\mathbf{R}}^{(t)} \boldsymbol{\alpha}^{(t-1)}\|_2^2 + \beta_0 \|\boldsymbol{\alpha}^{(t-1)}\|_1 + \beta_1 \sum_q |\alpha_{q+1} - \alpha_q|$ ;
9     Transforme o vetor  $\boldsymbol{\alpha}^{(t)}$  na matriz  $\mathbf{A}^{(t)}$ ;
10     $t = t + 1$ ;
11 retorna melhor solução até o momento  $\mathbf{A}^{(t-1)}$  e  $\mathbf{R}^{(t-1)}$ ;
```

ordenadamente as colunas da matriz original. Sendo $\mathbf{A} \in \mathbb{R}^{m \times n}$ a matriz original com seus elementos $a_{i,j}$, a matriz resultante $\mathbf{B} \in \mathbb{R}^{mn \times 1}$ tem cada um de seus elementos com valor $b_{i+(j-1)m,1} = a_{i,j}$.

Na linha 8 há a utilização de uma implementação particular do Lasso em que se inclui, na parcela de regularização, uma penalização que leva em conta a diferença entre valores consecutivos do vetor α de uma ondaleta candidata. Tal penalidade tem o efeito de forçar a presença de trechos de valor constante, e esta é chamada de penalidade de fusão. Os autores se referem a Liu, Yuan e Ye (2010) e a Hoeffling (2010) para resolução deste problema, chamado de *Fused Lasso*.

No trecho em que há o cálculo da refletividade no Algoritmo 5 (linha 4), outros algoritmos de representação esparsa podem ser usados. Em Wang et al. (2016) há utilização de um algoritmo proposto derivado do FISTA, em que o parâmetro de achatamento e limiarização λ_k varia a cada iteração, e seu valor corresponde à metade do valor absoluto do $(k+1)$ -ésimo maior valor, em módulo, do vetor de resultado parcial r_j . Tal abordagem foi baseada pelos autores como *Fista for TSMF* (Fista para Fatoração Esparsa de Matriz de Toeplitz).

Em substituição ao *Fista for TSMF*, propomos o *Lobbes*, descrito pelo

Algoritmo 6 a seguir, e com maior detalhamento no Apêndice A.

Algoritmo 6: Lobbes: Seleção de parâmetros baseada no *Lasso*

Entrada: Matriz-dicionário $\mathbf{A} \in \mathbb{R}^{m \times n}$
 Sinal de entrada $\mathbf{b} \in \mathbb{R}^m$
 Esparsidade desejada $k < m$
 Número de intervalos p

Saída: Sinal $\mathbf{x} \in \mathbb{R}^n$ com esparsidade k (k não zeros)

```

1  $\mathbf{x} \leftarrow \mathbf{0} \in \mathbb{R}^n$ ;
2  $(\hat{\mathbf{A}}, \hat{\mathbf{b}}, \lambda_{\max}) \leftarrow \text{normalizar}(\mathbf{A}, \mathbf{b})$ ;
3  $\delta_\lambda \leftarrow \lambda_{\max}/(p+1)$ ;
4 para  $i \leftarrow [0..p]$  faça
5    $\lambda \leftarrow i\delta_\lambda$ ;
6   se  $i == 0$  então
7      $\hat{\mathbf{x}}_{\text{novo}} \leftarrow (\hat{\mathbf{A}}^T \hat{\mathbf{A}} + \epsilon \mathbf{I})^{-1} \hat{\mathbf{A}}^T \hat{\mathbf{b}}$ ;
8   senão
9      $\hat{\mathbf{x}}_{\text{novo}} \leftarrow \text{spams.fistaFlat}(\hat{\mathbf{A}}, \hat{\mathbf{b}}, \mathbf{x}, \lambda)$ ;
10     $(b_0, \mathbf{x}_{\text{novo}}) \leftarrow \text{desnormalizar}(\hat{\mathbf{x}}_{\text{novo}})$ ;
11     $\mathbf{x}_{\text{novo}} \leftarrow \text{centralizarPicos}(\mathbf{x}_{\text{novo}})$ ;
12    se  $\text{esparsidade}(\mathbf{x}_{\text{novo}}) = k$  então
13      retorna  $\mathbf{x}_{\text{novo}}$ 
14    senão
15      se  $\text{esparsidade}(\mathbf{x}_{\text{novo}}) > k$  então
16         $\lambda_{\text{sup}} \leftarrow \lambda$ ;
17         $\mathbf{x} \leftarrow \mathbf{x}_{\text{novo}}$ ;
18      senão
19         $\lambda_{\text{inf}} \leftarrow \lambda$ ;
20      Saia do laço;
21 enquanto  $s \neq k$  faça
22    $\lambda \leftarrow (\lambda_{\text{inf}} + \lambda_{\text{sup}})/2$ ;
23    $\hat{\mathbf{x}}_{\text{novo}} \leftarrow \text{spams.fistaFlat}(\hat{\mathbf{A}}, \hat{\mathbf{b}}, \mathbf{x}, \lambda)$ ;
24    $(b_0, \mathbf{x}_{\text{novo}}) \leftarrow \text{desnormalizar}(\hat{\mathbf{x}}_{\text{novo}})$ ;
25    $\mathbf{x}_{\text{novo}} \leftarrow \text{centralizarPicos}(\mathbf{x}_{\text{novo}})$ ;
26   se  $\text{esparsidade}(\mathbf{x}_{\text{novo}}) = k$  então
27     retorna  $\mathbf{x}_{\text{novo}}$ 
28   senão
29     se  $\text{esparsidade}(\mathbf{x}_{\text{novo}}) > k$  então
30        $\lambda_{\text{sup}} \leftarrow \lambda$ ;
31        $\mathbf{x} \leftarrow \mathbf{x}_{\text{novo}}$ ;
32     senão
33        $\lambda_{\text{inf}} \leftarrow \lambda$ ;
34   se atingir limite de iterações então
35     retorna melhor solução até o momento  $\mathbf{x}$ 

```

Na inicialização, a matriz \mathbf{A} contém o conteúdo de uma única ondaleta com estrutura de uma matriz de Toeplitz. O sinal de entrada corresponde a um traço de amplitudes sísmicas. A esparsidade k corresponde à quantidade de elementos diferentes de zero do vetor esparso de refletividades. O número de intervalos corresponde à quantidade de subdivisões do valor do parâmetro de regularização λ , entre 0 e λ_{max} , que serve como valor inicial de busca do valor ótimo do mesmo parâmetro que resultará na esparsidade desejada k .

Inicialmente, o vetor esparso inicial \mathbf{x} é zerado. Procede-se com a normalização da matriz-dicionário, da qual também se extrai o valor máximo do parâmetro de regularização (chamemos de λ_{max}). Na primeira iteração, quando não se leva em consideração o parâmetro de regularização, com o contador da linha i zerado, calculamos uma solução inicial para traço esparso de refletividades $\hat{\mathbf{x}}_{novo}$ como sendo a solução algébrica de pseudo-inversão com um pequeno fator de regularização $\epsilon \ll 1$; senão, a solução de $\hat{\mathbf{x}}_{novo}$ é tomada como sendo a do Lasso, aqui realizada pelo procedimento *fistaFlat* disponível no pacote de software *SPAMS*.

Em seguida, é desfeita a normalização. No passo seguinte, o vetor \mathbf{x}_{novo} sofre um ajuste denotado pela função *centralizarPicos*, que pode ser explicado como segue. Para cada elemento de índice i de \mathbf{x}_{novo} , se nele estiver o valor de maior amplitude em relação a valores de índice $i - 1$ e $i + 1$, altera-se o valor de índice i para a soma dos elementos nos índices $i - 1$, i e $i + 1$, e se atribui zero aos valores dos índice $i - 1$ e $i + 1$. Desta forma, picos consecutivos de refletividade são eliminados. Em seguida, avalia-se se a solução sem normalização \mathbf{x}_{novo} tem a esparsidade desejada. Dependendo do número de não-zeros do vetor, é feito um aumento ou diminuição do parâmetro de regularização momentâneo λ numa busca binária.

Ao finalizar o Lobbes, a solução encontrada tem, no máximo, a quantidade requerida de não zeros. Sobre esta solução é aplicado o algoritmo *Coordinate Descent* puro (sem a limiarização e achatamento) somente sobre as dimensões não zero, resolvendo um problema de mínimos quadrados, algo análogo ao que é feito ao final de cada iteração do algoritmo OMP quando da seleção de uma nova coluna da matriz dicionário.

No Capítulo de resultados explicitamos mais de uma comparação deste algoritmo com seu antecessor em Wang et al. (2016) e com o OMP.

4 Agrupamento

4.1 Introdução

A partir da solução proposta em Wang et al. (2016), observa-se que há a necessidade de se limitar a quantidade de elementos de entrada do procedimento principal de deconvolução. A parte da otimização encarregada de descobrir a representação esparsa domina o tempo de execução, que é proporcional à quantidade de exemplos (traços) de entrada para o procedimento. Assim sendo, no intuito de diminuir o tempo de execução deste passo, parece importante adotar um critério de seleção das *amostras de entrada*, muito embora tal escolha possa remeter a um novo problema combinatorial. As *amostras de entrada* aqui mencionadas são os próprios traços de amplitudes sísmicas alvo de algoritmos de agrupamento, e não cada um dos valores componentes de cada traço.

Para o caso sintético 2D em Wang et al. (2016), optou-se por tomar um subconjunto de traços. Tal subconjunto foi obtido por amostragem regular, a partir de um recorte retangular da seção de amplitudes do modelo Marmousi2. Para o caso da aplicação em um dado de campo real, não se percebe critério de escolha dos traços, mas sim uma subdivisão do trabalho de extração de ondaleta e refletividade em blocos de traços. Assim, o problema acaba sendo resolvido por uma abordagem por partes. Para cada trecho do dado sísmico formado por um número determinado de traços contíguos, aplica-se a técnica de deconvolução. A ondaleta extraída do conjunto anterior é usada como ponto de partida da ondaleta no conjunto corrente. Ao final da extração de todas as ondaletas, uma para cada parte, os autores realizam uma média destas ondaletas extraídas. Com tal ondaleta média, realiza-se uma extração das refletividades no dado completo. Usar uma única ondaleta se justifica por se almejar a característica de continuidade lateral dos refletores, que poderia ser quebrada caso ondaletas extraídas de regiões adjacentes fossem muito diferentes entre si, significando diferença entre fases ou ângulos. Tal continuidade, se alcançada, poderia apontar para a presença de horizontes sísmicos, que podem servir como marcadores da transição entre camadas de solo verticalmente adjacentes, mas

com características físicas distintas. No procedimento de codificação esparsa que, no caso, é sinônimo de extração de conteúdo de refletividades, ao usar ondaletas distintas, cada qual sobre sua respectiva região de extração, poderia causar quebra perceptível de continuidade de refletores em traços de região limítrofe entre blocos.

No trabalho descrito em Rodríguez et al. (2014), o problema atacado é o de se realizar uma compressão de um dado volumétrico a partir de um treinamento de dicionário. Os autores utilizam um mecanismo de seleção das amostras de entrada para a etapa de criação do dicionário que se inspira no conceito de *coreset* (ou conjunto-núcleo), que seria, a depender do algoritmo aplicado ao dado de entrada, um subconjunto do conjunto inicial de amostras de entrada capaz de aproximar o resultado da aplicação do mesmo algoritmo sobre o conjunto completo dos dados. Exemplificando de forma simples usando um problema geométrico, um *coreset* para o problema de fecho convexo (em 2D) seria formado por um subconjunto de amostras de entrada que aproximam o polígono de perímetro mínimo do conjunto original.

De forma abstrata, ao observar os elementos deste conjunto-núcleo, o conceito parece carregar semântica similar à observação em Han, Kamber e Peir (2012) sobre o que torna um padrão, ou exemplar, **interessante**: fácil ao entendimento de um humano (interpretabilidade), potencialmente útil, detentor de características únicas que o mantém válido até certo grau de correção em comparação com dados novos (representatividade) e, eventualmente, podendo confirmar uma hipótese desejada.

4.2

Agrupamento como forma de seleção

Algoritmos de agrupamento são métodos de aprendizado de máquina não supervisionados capazes de atribuir diferentes amostras de entrada a grupos. Nestes grupos, elas partilham de algum tipo de similaridade. Com tais algoritmos, dados de entrada são aglutinados a partir do princípio da maximização da similaridade intragrupos e da minimização da similaridade intergrupos. O funcionamento desta classe de algoritmos pode ser visto como uma forma de organização das observações, dos exemplos, em hierarquias que agrupam eventos similares entre si. Agrupamento (ou *clustering*) é uma classe de técnicas que podem ser sinônimo de reconhecimento não supervisionado de padrões (Theodoridis e Koutroumbas, 2008). Uma técnica de agrupamento também pode ser interpretada como uma forma de alcançar uma redução de dados (Han, Kamber e Peir, 2012)¹. E é este sentido que justifica o uso deste

¹Capítulo 3, Seção 3.4.7.

tipo de técnica para atingir o objetivo da seleção das amostras da entrada em que, no presente caso, tal seleção resulta num subconjunto de traços com os quais pretende-se desfazer a fusão de uma informação ondulatória com uma informação esparsa de refletividade.

Algoritmos de agrupamento como o *K-Means* dependem de uso de uma distância euclidiana entre as amostras de entrada e, a priori, todos os pares possíveis são usados para o cálculo dos grupos.

Como ponto de partida, resolvemos, inicialmente, realizar um conjunto de agrupamentos levando em conta todo o dado recortado do bloco F3 via *K-Means*. São 552.132 traços de amplitude (588 *inlines* vezes 939 *crosslines*), cada qual com 448 amostras. A Figura 4.1 exibe ilustrações feitas a partir deste conjunto de agrupamentos².

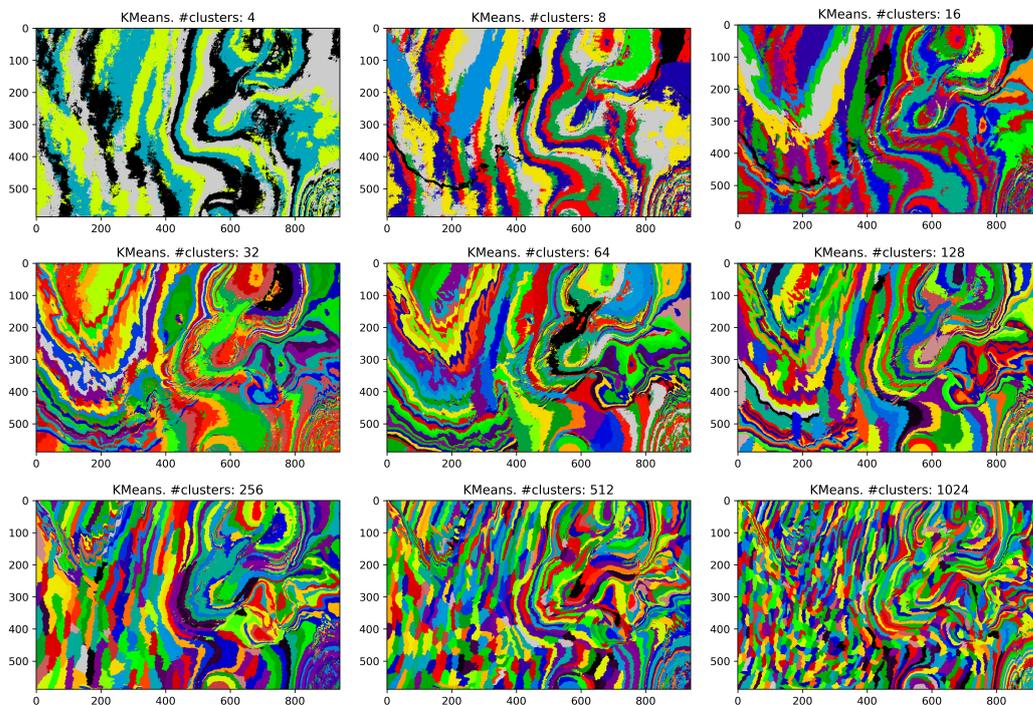


Figura 4.1: Resultado da aplicação do algoritmo *K-Means* sobre os traços do dado F3 recortado, pedindo de 4 a 1024 grupos de traços. Em cada imagem, eixo horizontal com *crosslines* e eixo vertical com *inlines*. Vista em planta do agrupamento dos traços.

Em sua forma original, o *K-Means* dá a possibilidade de se gerar grupos com elementos disjuntos, consequência da comparação entre todos os pares de elementos.

Traços sísmicos vizinhos podem apresentar uma semelhança que, eventualmente, pode não ser medida adequadamente se utilizarmos a distância euclidiana ou outra métrica de similaridade entre pares de amostras de mesmo

²Foi utilizada implementação disponível no pacote de software **scikit-learn**, em Python.

índice. Por outro lado, parece adequado levar em conta a complexidade computacional de qualquer medida de dissimilaridade ou de distância entre amostras que sofrerão comparação quando da realização do agrupamento.

Além da escolha da métrica de similaridade subjacente, limitar a comparação entre traços dentro de um raio ou uma janela de distância (em mapa 2D) poderia acelerar o processo de agrupamento. A possibilidade de se chegar a um agrupamento em que haja amostras de entrada de um grupo cercada por amostras de entrada de outro também deveria ser inibida. Assim sendo, não parece adequado utilizar o algoritmo *K-Means* puro na intenção de segmentar traços e, posteriormente, selecionar amostras de entrada de cada segmento para o processo posterior de treinamento de dicionário.

No presente caso, estamos lidando com dados de alta dimensionalidade, pois o tamanho de um traço sísmico tem centenas ou milhares de amostras. A dificuldade de se selecionar dados de alta dimensionalidade sem que se perca as características principais do dado completo pode ser encarada como um sintoma da chamada de **maldição da dimensionalidade**, citada, por exemplo, em Herrmann (2010).

4.3

Agrupamento em supertraços

Chamamos aqui de **supertraço** um conjunto formado por traços contíguos. A partir de uma métrica de similaridade previamente estabelecida, o conjunto dos supertraços, por hipótese, seria capaz de recortar o dado sísmico de amplitude de forma que traços num mesmo supertraço se assemelham e, ao mesmo tempo, são mais dissimilares aos traços dos supertraços da adjacência imediata. Consideramos supertraço como extensão e adaptação do conceito de superpixels, que são blocos de pixels em imagens que facilitam o trabalho de segmentação (Achanta et al., 2012; Van den Bergh et al., 2015). É uma extensão por abrir a possibilidade de utilização de múltiplas métricas de similaridade; e é uma adaptação por se tratar do caso específico de agrupamento de traços de amplitudes sísmicas.

O algoritmo SLIC (*Simple Linear Iterative Clustering*) (Achanta et al., 2012) deriva do *K-Means* e se propõe a realizar uma segmentação de imagens convertendo pixels para o espaço de cores LAB³. Ele limita a região de busca para comparação entre pixels e, ao mesmo tempo, faz uma ponderação entre as similaridades de cores dos pixels e a distância entre eles. De modo mais simples, o SLIC pode ser entendido como um *K-Means* com atuação local. Dois

³Espaço de cor CIE Lab. Para uma visão rápida sobre espaços de cor, consultar Tkalcic e Tasic (2003).

parâmetros guiam a segmentação de uma imagem em superpixels: a quantidade máxima de superpixels e o parâmetro de *compacidade*. Este último, ao ter seu valor aumentado, força cada superpixel a ficar com formato mais próximo de um círculo. Uma variação deste mesmo algoritmo usa somente o parâmetro de quantidade de superpixels, e é chamado de SLIC0 ou SLICO. Na Figura 4.2, que exemplifica o funcionamento do algoritmo SLIC sobre uma imagem colorida, são exibidas as bordas dos superpixels e os superpixels em si.



Figura 4.2: Exemplo de aplicação do algoritmo SLIC sobre imagem colorida. Bordas dos superpixels (à esq.) e superpixels (à dir.).

A partir da implementação disponibilizada pelos autores, fizemos extensões que julgamos adequadas para tratar do agrupamento de traços, detalhada a seguir.

4.4

SLICT: uma proposta para agrupamento em supertraços

Derivado do SLIC, o algoritmo proposto de agrupamento de traços, aqui chamado de SLICT (para *Simple Linear Interactive Clustering of Traces*), promove uma alteração fundamental no algoritmo original e dá possibilidade de uma extensão. A alteração diz respeito à natureza das amostras de entrada, que passam a ter dimensão igual ao tamanho de cada traço. Manteve-se a métrica da soma das distâncias euclidianas quadráticas nas duas direções em mapa, que, no caso, se referem às coordenadas de *inline* e *crossline* para geração do mapa de arestas e para a realização da comparação entre traços para separação em grupos. A extensão está na ampliação do propósito do algoritmo em si que, ao invés de somente realizar a segmentação dos traços, passa a dar a possibilidade de, a partir dos grupos, selecionar um representante. A coleção de representantes dos supertraços é o que utilizaremos como entrada do processo posterior de treinamento de dicionário.

Por um representante de um supertraço entenda-se como sendo o traço com dissimilaridade mínima contra os demais traços. Desta forma, abre-se, mais uma vez, a possibilidade de alterar a forma de comparação entre traços.

Além do quadrado da distância euclideana e da dissimilaridade do cosseno, averiguamos a possibilidade de se usar uma função de casamento de padrões.

A função de casamento de padrões avaliada é baseada no algoritmo DTW (*dynamic time warping*) original. Há uma versão mais rápida deste algoritmo, mas tal versão deixa explícita a possibilidade de erro (Salvador e Chan, 2007).

O cálculo de dissimilaridade entre traços está ilustrado no Algoritmo 7. Neste algoritmo, o peso atribuído aos trechos de caminho de custo mínimo deriva do mecanismo de pesos encontrado no trabalho sobre invariantes integrais (Manay et al., 2006).

Algoritmo 7: Casamento de padrões por DTW

Entrada: traço de referência $\mathbf{t}_{\text{ref}} \in \mathbb{R}^m$
 traço na janela de comparação $\mathbf{t}_1 \in \mathbb{R}^m$
 salto máximo entre amostras s
 parâmetro de força de alinhamento em diagonal α

Saída: Função de correspondência F
 Dissimilaridade acumulada (custo do caminho) C

```

1 início
2   Mapa de custos  $q \leftarrow 0$ ;
3   Posição pivô  $p \leftarrow 0$ ;
4   para  $j \in [0..m - 1]$  faça
5     minIndx  $\leftarrow \max(0, j - s)$ ;
6     maxIndx  $\leftarrow \min(j + s + 1, m - 1)$ ;
7     para  $i \in [\text{minIndx}..\text{maxIndx}]$  faça
8        $c_r \leftarrow \text{diferença}(t_{\text{ref}}[j], t_1[i])$ ;
9        $c_1 \leftarrow (c_r + \alpha)/2 + q[p - 1]$ ;
10       $c_2 \leftarrow (c_r + \alpha)/2 + q[p - m]$ ;
11       $c_3 \leftarrow (\sqrt{2}c_r)/2 + q[p - 1 - m]$ ;
12       $mc \leftarrow \min(c_1, c_2, c_3)$ ;
13       $mi \leftarrow \text{índice correspondente } amc$ ;
14       $q[p] \leftarrow mc$ ;
15       $\pi[p] \leftarrow mi$ ;
16    $p \leftarrow m^2 - 1$ ;
17    $F \leftarrow \emptyset$ ;
18   enquanto pivô  $p$  for válido faça
19      $F \leftarrow F \cup \{(p \bmod m, \lfloor p/m \rfloor)\}$ ;
20      $C \leftarrow C + q[p]$ ;
21      $p \leftarrow \pi[p]$ ;
22 retorna  $(F, C)$ ;
```

Observando o Algoritmo 7, percebe-se custo computacional $O(sm)$ em tempo e em espaço, sendo s a diferença máxima entre índices dos vetores comparados e m o tamanho de cada vetor. No caso, o parâmetro s pode assumir um significado geofísico, que seria o tempo máximo ou profundidade máxima tolerada de deslocamento entre traços vizinhos a considerar o casamento de padrões (ou mergulho máximo). Tais custos podem tornar seu uso proibitivo.

Da mesma forma que o SLIC, o SLICT também faz uso das janelas dentro das quais são limitados os cálculo de distância para os traços vizinhos. A Figura

4.3 ilustra a janela.

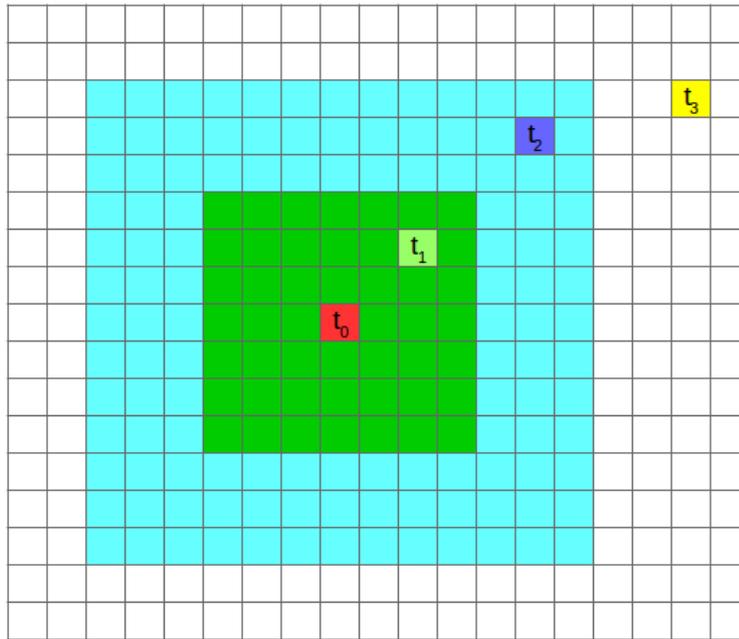


Figura 4.3: Ilustração da janela de limite para cálculo de similaridade. O traço central (t_0 , em vermelho) é a referência contra o qual os traços dentro do limite exterior (em azul) serão comparados. Em verde o limite corresponde ao número de traços pretendido para supertraço, e em azul o limite máximo da busca. Traços t_1 e t_2 nas regiões verde e azul são observados, enquanto o traço t_3 é ignorado.

Foi feita alteração a partir do código-fonte disponibilizado pelos autores do SLIC0 de forma que fosse restaurada a utilização de dois parâmetros que influenciam tanto na quantidade de elementos dentro de um mesmo grupo quanto no formato dos grupos, derivado do peso dado à aproximação do módulo do gradiente em cada posição em mapa. O formato varia conforme um parâmetro análogo ao de compacidade do SLIC, que é chamado de parâmetro de regularidade dos supertraços.

Algoritmos como o *K-Means* fazem uso intensivo do cálculo de distância entre os elementos do dado de entrada. O SLIC e, conseqüentemente, o SLICT não desvia deste comportamento. O uso de uma métrica que custa mais caro em termos algorítmicos sugere que calculemos a priori uma matriz de dissimilaridade entre os possíveis pares usados pelo algoritmo de supertraços. A distância máxima (em mapa) entre traços, como ilustrado na Figura 4.3, é o fator que limita o tamanho máximo de cada supertraço. Quanto maior a distância máxima permitida entre traços, maior a quantidade de valores calculados e armazenados. Uma maior distância entre traços dá a possibilidade de geração de supertraços maiores e com formato menos regular (significando menos semelhantes a círculos ou polígonos convexos). No entanto, somente este

cálculo *offline* de uma matriz de distâncias, dependendo do algoritmo utilizado, pode não ser prático ⁴.

Podemos, arbitrariamente, selecionar um único representante de cada supertraço como a amostra de entrada ao treinamento de dicionário. Assim sendo, a quantidade final requerida de supertraços é igual à quantidade de amostras de entrada para o treinamento de dicionário. Em procedimentos de treinamento de aprendizado de máquina, normalmente se requer que a quantidade de elementos de entrada seja (muito) maior que a quantidade de parâmetros do modelo. No presente caso não é diferente. Herda-se do SLIC o cálculo da distância máxima de busca (como ilustrado na Figura 4.3) como função da quantidade de grupos (supertraços) requerida, dada por $s = \sqrt{\frac{N}{K}}$, em que s é o lado do quadrilátero cuja área corresponde à quantidade de elementos requerida K , e sendo N a quantidade disponível de amostras (no caso, corresponderia à quantidade de traços no dado sísmico de amplitudes).

O passo inicial de execução do algoritmo SLIC é o de calcular um campo escalar de gradiente de cores, no espaço Lab, por diferenças finitas. Valores mais altos neste campo escalar evitam que representantes de superpixels sejam ali posicionados. Seja p um ponto da imagem com coordenadas (x, y) numa imagem colorida. Seja $lab(x, y)$ o valor da cor na posição do ponto p usando o sistema de cores Lab. O campo escalar $e(x, y)$ na posição do ponto p , de acordo com o código-fonte do SLIC é dado por $e(x, y) = dx + dy$, com $dx = (lab(x + 1, y) - lab(x - 1, y))^2$ e $dy = (lab(x, y + 1) - lab(x, y - 1))^2$. O SLICT funciona de forma análoga, trocando a cor (3 dimensões) pelo traço (várias dimensões, uma para cada posição do vetor (vertical) correspondente ao traço de amplitudes).

Para geração do campo escalar Δ em cada posição $[i, j]$ de diferenças usado pelo SLICT, fazemos

$$\Delta[i, j] = d(\mathbf{T}[i, j + 1], \mathbf{T}[i, j - 1])^2 + d(\mathbf{T}[i + 1, j], \mathbf{T}[i - 1, j])^2,$$

sendo $\mathbf{T}[i, j]$ o traço na i -ésima *inline* e j -ésima *crossline* e $d(\mathbf{a}, \mathbf{b})$ sendo a distância euclideana entre traços \mathbf{a} e \mathbf{b} .

A Figura 4.4 ilustra um campo escalar gerado pelos traços do dado F3 recortado usado para gerar os supertraços via SLICT.

⁴Realizamos dois experimentos. O primeiro deles usando janela com raio de 50 amostras de entrada, levou cerca de 160 horas e ocupou 10.2Gb de espaço em disco, num computador com Intel Core i7 7500U (4 threads). O segundo experimento usou janela de raio de 25 amostras de entrada, levando cerca de 26 horas e ocupou 2.5Gb, num computador com 2 processadores Intel Xeon E5-2687W (16 threads). A título de comparação, o dado recortado a partir do bloco F3 de amplitudes sísmicas utilizado ocupa cerca de 990Mb.

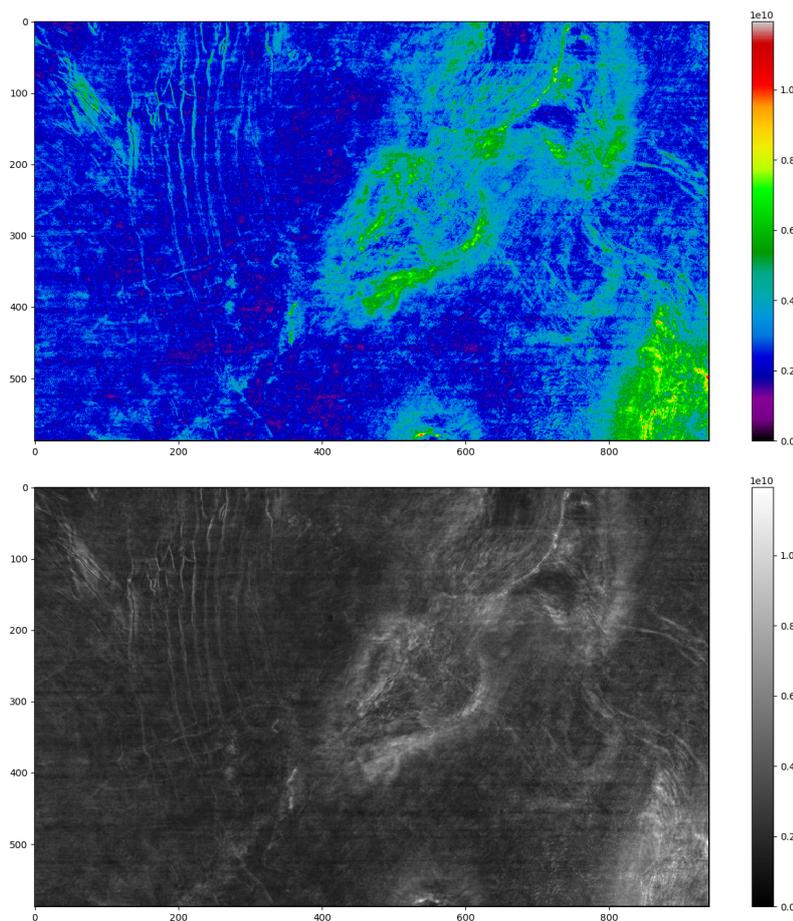


Figura 4.4: Ilustração do campo escalar usado como base de execução do SLICT. As imagens mostram a mesma informação, mudando somente a escala de cor. A grandeza medida é soma do quadrado da distância euclidiana entre traços anterior e posterior, nas duas direções, em cada posição.

As Figuras 4.5, 4.6, 4.7 e 4.8 ilustram os resultados da execução do algoritmo SLICT sobre o dado recortado a partir do bloco F3 recortado, variando tanto o limite da quantidade de supertraços (2000 ou 8000) quanto o valor do peso dado à diferença entre os traços (ou o parâmetro de regularidade, com valores 0,125 ou 2,0). As imagens das respectivas figuras mostram a mesma informação, mudando a apresentação. Nas Figuras 4.5(a), 4.6(a), 4.7(a) e 4.8(a), as cores correspondem aos índices de cada grupo (supertraço), e são mostradas as bordas entre eles. Nas Figuras 4.5(b), 4.6(b), 4.7(b) e 4.8(b), somente as bordas são mostradas. E nas Figuras 4.5(c), 4.6(c), 4.7(c) e 4.8(c), faz-se uma atribuição aleatória de cores a cada grupo na intenção de destacar cada um em relação à sua vizinhança.

A partir de resultados de segmentação usando o SLICT, pode-se realizar

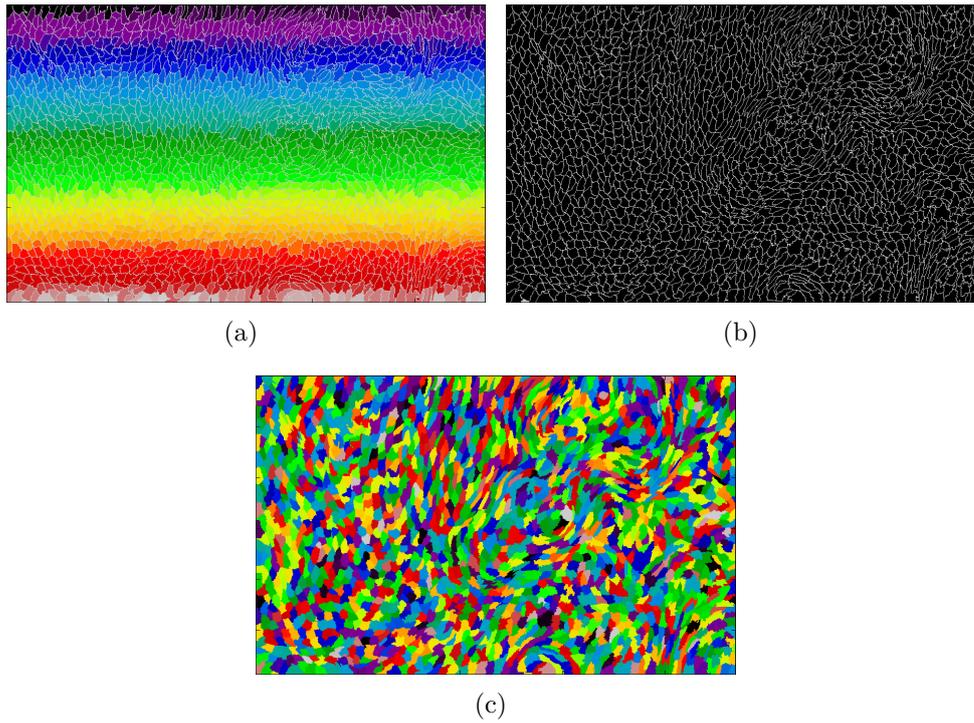


Figura 4.5: Ilustração da execução do algoritmo SLICT sobre o dado F3 recortado. Limite de 2000 supertraços. Parâmetro de regularidade = 0,125. Em (a) e (c), blocos com cores distintas. Em (b), somente bordas entre blocos.

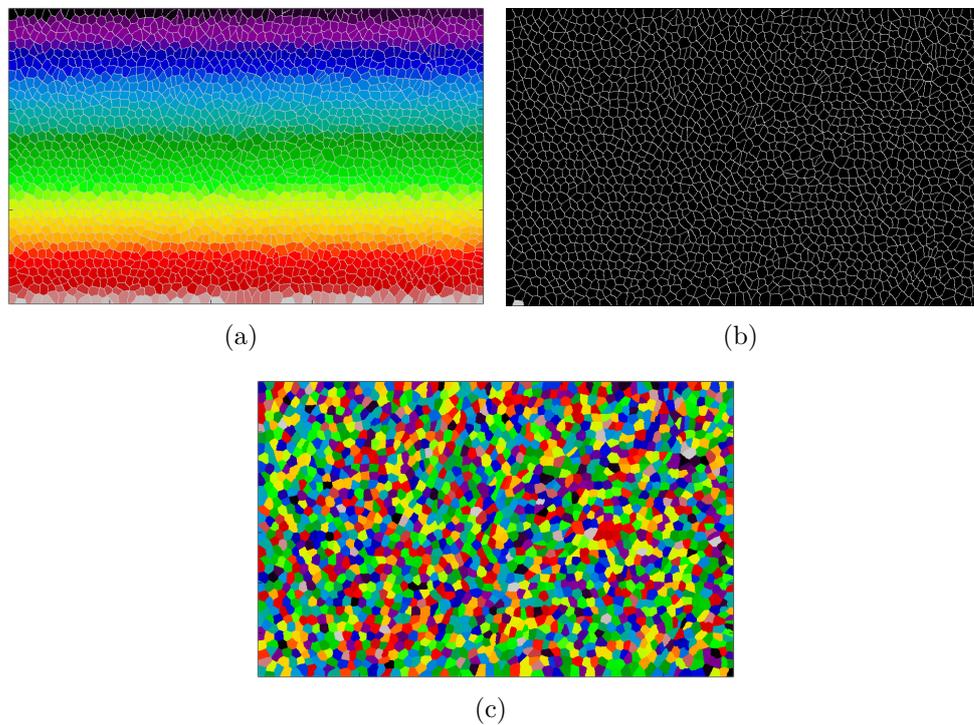


Figura 4.6: Ilustração da execução do algoritmo SLICT sobre o dado F3 recortado. Limite de 2000 supertraços. Parâmetro de regularidade = 2,0. Em (a) e (c), blocos com cores distintas. Em (b), somente bordas entre blocos.

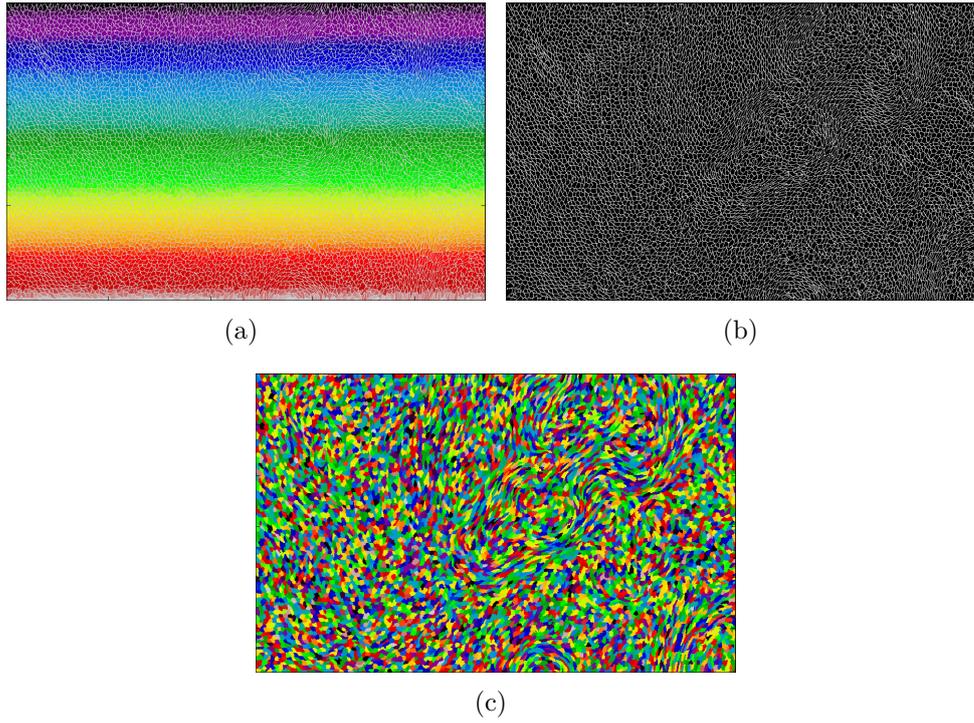


Figura 4.7: Ilustração da execução do algoritmo SLICT sobre o dado F3 recortado. Limite de 8000 supertraços. Parâmetro de regularidade = 0,125. Em (a) e (c), blocos com cores distintas. Em (b), somente bordas entre blocos.

PUC-Rio - Certificação Digital Nº 1421617/CA

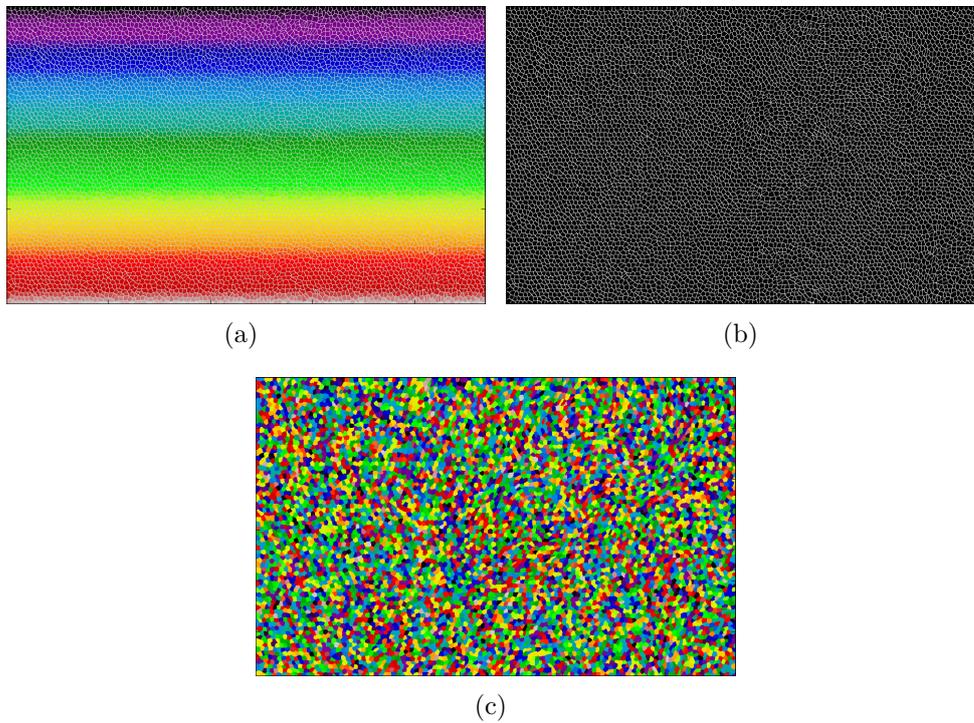


Figura 4.8: Ilustração da execução do algoritmo SLICT sobre o dado F3 recortado. Limite de 8000 supertraços. Parâmetro de regularidade = 2,0. Em (a) e (c), blocos com cores distintas. Em (b), somente bordas entre blocos.

a seleção dos traços que comporão o conjunto de entrada para o processo de treinamento de dicionário. Observando a estratégia adotada no algoritmo *K-Means* para o cálculo do representante de cada grupo, optou-se por realizar algo similar para o caso dos supertraços: dentro de cada supertraço, realizamos a busca do elemento cuja soma das dissimilaridades para os demais seja mínima. Sendo Θ um supertraço e $\mathbf{t}_i, \mathbf{t}_j$ traços em Θ , com $i \neq j$ (ou $\mathbf{t}_i \neq \mathbf{t}_j$), pode-se formular o cálculo do representante, com índice i de Θ como na Equação 4-1 a seguir.

$$\arg \min_i \sum_{i \neq j} d(\mathbf{t}_i, \mathbf{t}_j), \quad \mathbf{t}_i, \mathbf{t}_j \in \Theta \quad (4-1)$$

Para as Figuras 4.5, 4.6, 4.7 e 4.8, a dissimilaridade usada foi a distância euclidiana. A partir destas figuras, percebe-se que, ao aumentar o valor do parâmetro de regularidade dos supertraços, as arestas dos supertraços mais facilmente deixam de respeitar o campo escalar previamente calculado, deixando, assim, de mostrar alguma característica geológica da área em estudo. Mesmo com uma parametrização que faz o algoritmo não mais marcar alguma característica geológica, o SLICT parece servir como método de suporte para amostragem de traços do dado sísmico para aplicações que não a feita nesta tese.

No capítulo seguinte será detalhado o procedimento de geração do dicionário que, com base nos traços selecionados usando os supertraços gerados pelo SLICT, será usado para realizar a representação esparsa do conteúdo de amplitude do dado completo.

5

Aprendizado e Extensão de Dicionário para Inversão de Refletividades

5.1

Introdução

Neste capítulo mostraremos o processo de geração de um dicionário, representado por uma matriz com mais colunas do que linhas, que servirá para se realizar o passo final de representação esparsa do dado de amplitude. Podemos entender tal representação como uma aproximação da inversão sísmica para se chegar em um conteúdo de picos esparsos de refletividade. Para se chegar ao dicionário final, inicialmente optamos por extrair um dicionário quadrado, ou seja, com a quantidade de átomos (colunas) coincidindo com a quantidade de amostras de um traço de amplitudes (linhas). Tal escolha foi motivada pela característica do operador de Toeplitz utilizado em Wang et al. (2016) e Fernandes, Lopes e Gattass (2017). O passo seguinte é um corte ao redor da diagonal da matriz para eliminar informação ondulatória de forma a representar a natureza de tempo limitado das ondaletas. Em seguida, aplica-se um aumento artificial do dicionário gerado usando a abordagem em Oliveira (2005) para aplicar alterações em ondaletas sísmicas na tentativa de emular o efeito aplicado pela terra às ondas sísmicas.

5.2

Aprendizado de um dicionário inicial

O capítulo anterior mostrou a abordagem utilizada para se chegar num subconjunto de traços que, hipoteticamente, poderia representar todo o dado sísmico. De posse deste subconjunto, fazemos uso da técnica de aprendizado de dicionário, em que se opta por gerar um dicionário, com forma de uma matriz quadrada, em que cada átomo tem norma euclideana 1.

O Algoritmo 8 descreve o comportamento deste passo.

Algoritmo 8: Treinamento de dicionário

Entrada: \mathcal{S}_0 : conjunto de traços representantes dos supertraços
 k : número máximo de picos em cada traço
 $maxIters$: número máximo de iterações
 ϵ : limiar para avanço na aproximação
 \mathcal{D}_0 : dicionário inicial

Saída: dicionário \mathcal{D}
refletores \mathcal{R}

```

1 início
2    $\mathcal{D} \leftarrow \mathcal{D}_0, \mathcal{S} \leftarrow \mathcal{S}_0;$ 
3    $P_{\mathcal{S}} \leftarrow$  número de picos de cada traço de  $\mathcal{S}$  limitado a  $k$ ;
4    $d \leftarrow \infty, f_0 \leftarrow \infty, it \leftarrow 0;$ 
5   enquanto ( $d > \epsilon$ ) e ( $it < maxIters$ ) faça
6      $\mathcal{R} \leftarrow$  representação esparsa de  $\mathcal{S}$  usando  $\mathcal{D}$ ;
7     atualize  $\mathcal{D}$  usando  $\mathcal{R}$  e  $\mathcal{S}$ ;
8      $f_1 \leftarrow \|\mathcal{D}\mathcal{R}\|_F^2;$ 
9      $d \leftarrow |f_0 - f_1|/f_1;$ 
10     $f_0 \leftarrow f_1;$ 
11 retorna ( $\mathcal{D}, \mathcal{R}$ )

```

Na linha 6 é que se usam os algoritmos de representação esparsa como, por exemplo, OMP, FISTA, CD para Lasso.

Na linha 7 é que se aplica uma estratégia para atualização de dicionário como, por exemplo, MOD, K-SVD, e SGK.

Na linha 8 é realizada uma reconstrução, e na linha 9 se avalia a mudança relativa à última iteração.

No Algoritmo 8, percebe-se um conjunto de fatores que influenciam no resultado (dicionário e conteúdo de refletividades):

- Traços de entrada e sua quantidade;
- Quantidade limite de picos por traço (linha 3);
- Número máximo de iterações do algoritmo de treinamento (linha 5);
- Limiar do avanço iterativo do treinamento (linha 5);
- Condição inicial do dicionário;
- Algoritmo de contagem da quantidade máxima de picos por traço (linha 3);
- Algoritmo para representação esparsa (linha 6);

- Algoritmo para atualização do dicionário (linha 7).

Mantendo a quantidade limite de iterações constante, cabe observar que, em relação ao OMP, o algoritmo LS-OMP tenderá a um tempo de execução superior e a um resultado numérico ligeiramente melhor em termos de erro de reconstrução. Isso se deve ao fato de que, para escolher qual novo átomo comporá a representação esparsa de um traço de entrada, é feito, no LS-OMP, um cálculo de mínimos quadrados, ao invés de fazer simplesmente um produto escalar. Muito embora a literatura¹ mencione e se constate na prática que o resultado numérico do LS-OMP seja superior ao OMP, o uso do OMP se mostra mais viável devido à sua complexidade algorítmica inferior.

O limiar de avanço iterativo permite que o processo de treinamento seja interrompido caso o avanço conseguido seja inferior ao valor desejado.

Um aspecto importante observado foi o da sensibilidade do resultado do dicionário extraído às condições iniciais. Quando inicializamos o dicionário com uma matriz **identidade**, emulando o comportamento ideal de um pulso sísmico como um delta de Dirac, sem variação ao longo da profundidade, obtivemos dicionários que, também, têm os picos de amplitude das ondaletas extraídas ao longo da diagonal da matriz. Este comportamento não se repete quando não se inicializa o dicionário como uma matriz identidade. Assumindo que na diagonal principal da matriz do dicionário está presente o valor de maior amplitude, não ocorre deslocamento vertical de um evento significativo num traço de amplitude em relação ao seu par recodificado de refletividade. A opção por esta estrutura de matriz de dicionário deriva dos trabalhos de deconvolução por picos esparsos em Wang et al. (2016) e em Fernandes, Lopes e Gattass (2017).

Para a contagem de picos, usa-se uma abordagem de supressão de não máximos. Parte-se do pressuposto de haver um limite máximo para a quantidade de picos por traço. Dentro deste limite, o Algoritmo 9 busca picos levando em conta que, ao redor de cada pico, ou seja, dentro de uma janela de busca centrada na amostra em avaliação, não deve existir nenhum outro elemento de valor absoluto maior. Tal janela é aumentada quando se encontra, ao final do processo de contagem para cada traço, um número maior de picos que o desejado.

¹Vide Elad (2010) e Dumitrescu e Irofti (2018)

Algoritmo 9: Contagem de picos por traço

Entrada: conjunto de traços \mathcal{S}_0 representantes dos supertraços
quantidade k de limite de picos em cada traço

Saída: picos por traço \mathcal{P}

```

1 início
2   para cada traço  $\mathbf{t}$  em  $\mathcal{S}_0$  faça
3     Seja  $w = 1$  o tamanho da meia janela;
4     traçoProcessado  $\leftarrow$  falso;
5     enquanto não(traçóProcessado) faça
6        $c \leftarrow 0$ ;
7       para cada amostra  $t_{[i]}$  em  $\mathbf{t}$  faça
8         se  $|[t_{[i-w]}..t_{[i-1]}, t_{[i+1]}..t_{[i+w]}]| < |t_{[i]}|$  então
9            $c \leftarrow c + 1$ ;
10        se  $c > k$  então
11           $w \leftarrow w + 1$ ;
12        senão
13          traçoProcessado  $\leftarrow$  verdadeiro;
14           $P[i] \leftarrow c$ ;
15 retorna  $\mathcal{P}$ 

```

Os algoritmos de atualização do dicionário implementados foram o MOD e o SGK com regularização. Todos utilizaram como método de codificação esparsa o OMP.

A Figura 5.1 exibe resultados de treinamento de dicionário usando o par de algoritmos OMP e MOD com 2 tamanhos de entrada e dois valores de parâmetro de regularidade de supertraços do SLICT.

A partir dos resultados parciais exibidos na Figura 5.1 montamos a Tabela 5.1 contendo o erro de reconstrução (ou resíduo) das amostras de entrada do processo de treinamento.

Em inversão sísmica, as ondaletas obtidas (ou desejadas) neste processo devem ter um limite de duração e outro limite dentro do qual está a parte mais significativa do pulso. Na intenção de emular tal comportamento, avaliamos inicialmente uma espécie de poda do conteúdo da matriz de dicionário, eliminando os valores mais distantes da diagonal principal e, mais próximo a este limite, realizamos uma atenuação arbitrária do sinal. A atenuação segue, como mostrada na Figura 5.4, uma função linear decrescente.

Procedendo tal corte, ocorre a remoção de parte da informação codificável

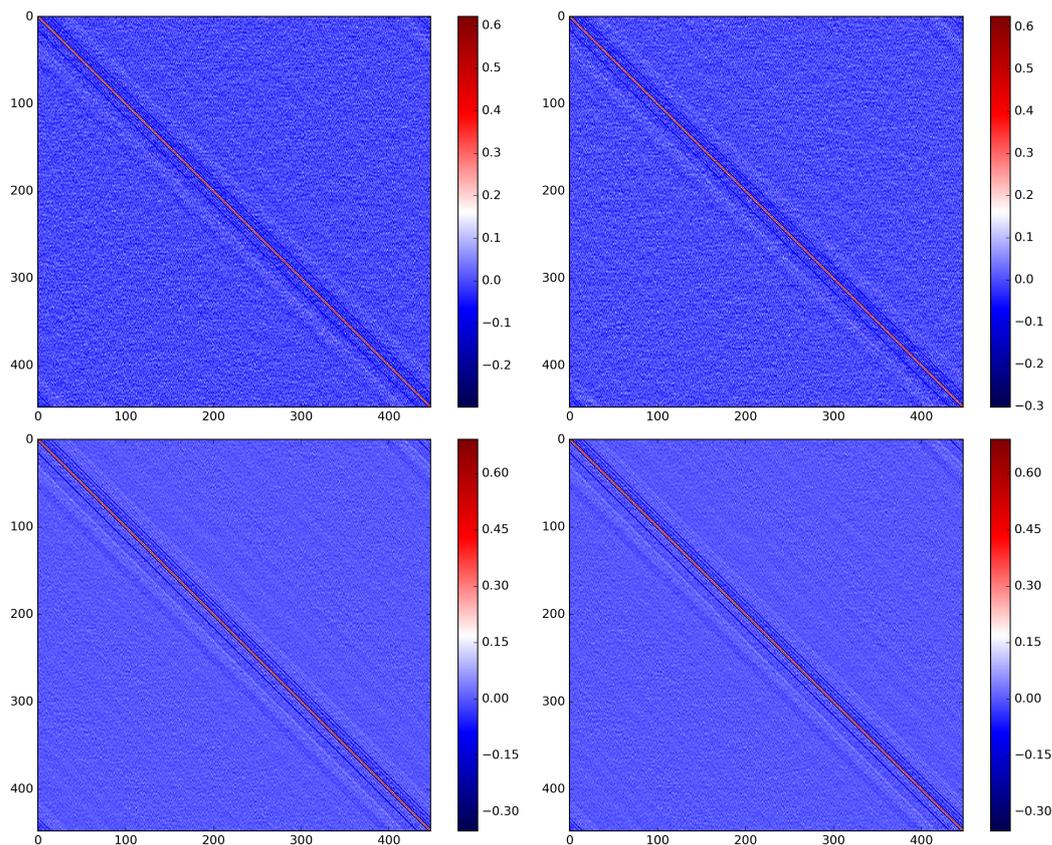


Figura 5.1: Resultado inicial da execução de treinamentos de dicionários usando os algoritmos OMP e MOD. Coluna da esquerda, usando parâmetro de regularidade 0,125. Coluna da direita, usando parâmetro de regularidade 2,0. Acima, limite de 2000 supertraços (com 1977 e 1978 selecionados). Abaixo, limite de 8000 supertraços (com 7998 e 8023 selecionados).

| Erro de reconstrução | 0,125 | 2,0 |
|-----------------------------|-----------------|-----------------|
| 2000 | 1,157% (9,844%) | 1,153% (8,497%) |
| 8000 | 1,76% (14,397%) | 1,914% (15,13%) |

Tabela 5.1: Erro de reconstrução das amostras de entrada do processo de treinamento de dicionário, variando a quantidade aproximada de traços de entrada (2000 e 8000) e o parâmetro de regularidade dos supertraços (0,125 e 2,0). Medidas média (e máxima, em parênteses) percentual. Medidas tomam o valor máximo absoluto do dado de entrada e do erro encontrado. Algoritmos utilizados: OMP e MOD.

pelo dicionário original. Após tal corte, cada átomo é ajustado para que volte a ter norma euclidiana igual a 1. A Tabela 5.2 mostra o resultado do cálculo do erro de reconstrução fazendo uso dos dicionários após tal ajuste.

A Figura 5.3 ilustra a diferença de conteúdo de dicionários, gerados usando OMP e MOD, antes e depois do recorte e ajuste (normalização dos átomos), selecionando somente uma linha da matriz.

A partir das Tabelas 5.1 e 5.2, observamos que tal procedimento simpli-

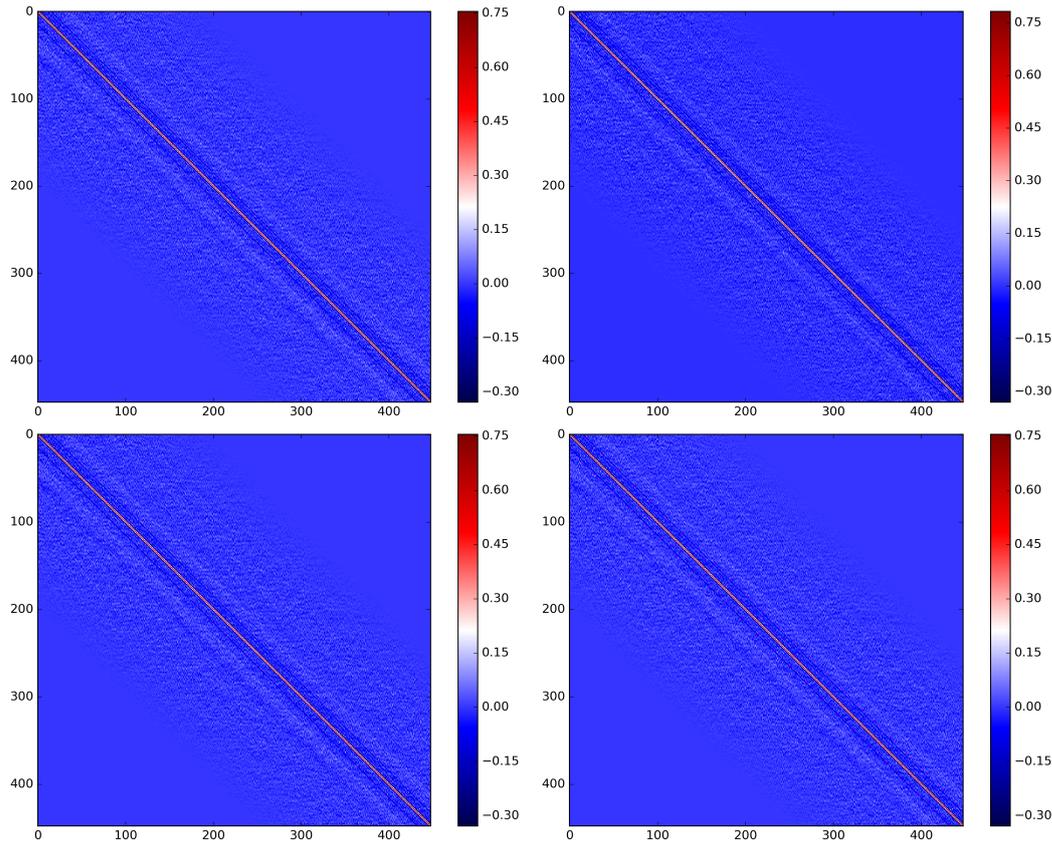


Figura 5.2: Resultado da aplicação de poda ao dicionário inicial gerado usando OMP e MOD. Coluna da esquerda, usando parâmetro de regularidade de supertraços com valor igual a 0,125. Coluna da direita, usando parâmetro de regularidade de supertraços com valor igual a 2. Acima, limite de 2000 supertraços (com 1977 e 1978 selecionados). Abaixo, limite de 8000 supertraços (com 7998 e 8023 selecionados). Amostras com distância superior a 100 da diagonal foram eliminadas. Amostras de índices de 50 a 100 distantes da diagonal tiveram atenuação linear até chegar em 0.

| Erro de reconstrução | 0,125 | 2,0 |
|-----------------------------|----------------|-----------------|
| 2000 | 3,49% (36,97%) | 3,47% (29,605%) |
| 8000 | 2,7% (39,97%) | 2,815% (45,97%) |

Tabela 5.2: Erro de reconstrução das amostras de entrada a partir do uso do dicionário ajustado, variando a quantidade aproximada de traços de entrada (2000 e 8000) e o parâmetro de regularidade dos supertraços (0,125 e 2,0). Medidas média (e máxima, em parênteses) percentual. Medidas tomam o valor máximo absoluto do dado de entrada e do erro encontrado. Observa-se um aumento no erro em todas as medidas. Algoritmos utilizados: OMP e MOD.

ficado acaba por introduzir um montante importante de erro na reconstrução do próprio sinal de treinamento. Assim sendo, parece ser mais adequado aplicarmos algum tipo de regularização sobre os próprios átomos a fim de que tal corte não acarrete na introdução de erros significativos. Na seção a seguir,

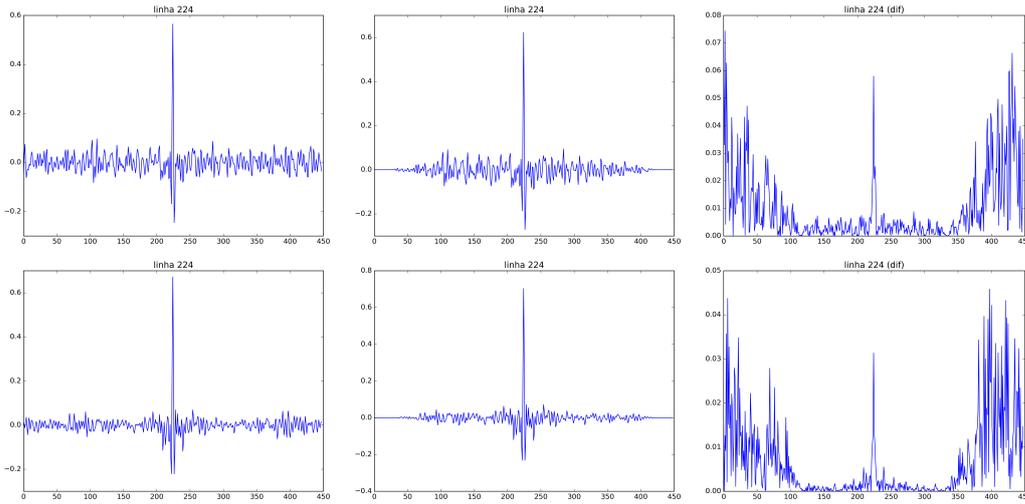


Figura 5.3: Visualização da linha de índice 224 de dois dicionários, antes (à esquerda) e depois (ao centro) do recorte e ajuste (normalização), com diferença absoluta mais à direita. Acima, utilizando 2000 supertraços e parâmetro de regularidade com valor 0,125. Abaixo, utilizando 8000 supertraços e parâmetro de regularidade com valor 0,125. Dicionários gerados usando MOD e OMP.

investigamos uma proposta de solução para este problema.

5.3

Aplicação de regularização aos átomos do dicionário

Na seção anterior, percebemos que a remoção dos valores mais distantes da diagonal da matriz de dicionário acarreta em perda de informação. Nesta seção, explicamos um procedimento para que tal perda seja minimizada a partir da aplicação de uma regularização sobre a forma de cada átomo.

5.3.1

Generalização Sequencial do *K-Means*: SGK

Em Sahoo e Makur (2013), citado por Dumitrescu e Irofti (2018), é sugerido um método de ajuste de dicionário, átomo a átomo, derivado do *K-Means*, chamado SGK (*Sequential Generalization of K-Means*) ou Generalização Sequencial do *K-Means*. Este algoritmo faz uma otimização de um átomo por vez, e em sequência, mantendo os demais átomos e seus coeficientes (na matriz de representação esparsa) intactos.

Partindo da Equação 2-17, fazemos um desmembramento para deixar o k -ésimo átomo do dicionário separado:

$$\|\mathbf{E}\|_F^2 = \left\| \left(\mathbf{Y} - \sum_{j \neq k} \mathbf{D}_{:,j} \mathbf{X}_{j,:} \right) - \mathbf{D}_{:,k} \mathbf{X}_{k,:} \right\|_F^2. \quad (5-1)$$

Chamemos de $\mathbf{E}_k = \mathbf{Y} - \sum_{j \neq k} \mathbf{D}_{:,j} \mathbf{X}_{j,:}$ o erro ao deixar de considerar o k -ésimo átomo de \mathbf{D} . Desta forma, a Equação 5-1 do erro de codificação esparsa pode ser reescrita da seguinte forma:

$$\|\mathbf{E}\|_F^2 = \|\mathbf{E}_k - \mathbf{D}_{:,k} \mathbf{X}_{k,:}\|_F^2. \quad (5-2)$$

Para simplificar momentaneamente a notação, usemos $\mathbf{d}_k = \mathbf{D}_{:,k}$. O que se deseja é que o k -ésimo átomo seja otimizado mantendo o restante fixo:

$$\mathbf{d}_k^{(t+1)} = \arg \min_{\mathbf{d}_k} \|\mathbf{E}_k^{(t)} - \mathbf{d}_k \mathbf{X}_{k,:}^{(t)}\|_F^2. \quad (5-3)$$

De forma similar à Equação 2-17, a solução para a Equação 5-3 é dada por:

$$\mathbf{d}_k^{(t+1)} = \mathbf{E}_k^{(t)} \mathbf{X}_{k,:}^{(t)} \left(\mathbf{X}_{k,:}^{(t)} \mathbf{X}_{k,:}^{(t)T} \right)^{-1}. \quad (5-4)$$

O átomo recém-atualizado interfere na atualização do próximo átomo na sequência. Observa-se, como no MOD, que não há necessidade explícita de realizar a normalização de cada átomo. Caso seja necessária, tal normalização só pode ser feita quando da finalização da atualização de todos os átomos do dicionário (Sahoo e Makur, 2013).

5.3.2 SGK com regularização por átomo

Já foi observado que informação é perdida quando se corta as bandas mais afastadas da diagonal da matriz de dicionário. Com o objetivo de minimizar esta perda causada pelo corte da matriz de dicionário, propõe-se o uso de regularização de cada átomo de forma que os elementos mais afastados da diagonal da matriz tenham seu valor diminuído ou anulado. Partindo da Equação 5-2, adicionamos um termo para penalizar um átomo, uma matriz \mathbf{P}_k , e um fator λ resultando numa função alvo de minimização, agregando erro e penalização \mathbf{F}_{ep} :

$$\mathbf{F}_{ep} = \|\mathbf{E}_k - \mathbf{d}_k \mathbf{X}_{k,:}\|_F^2 + \lambda \|\mathbf{P}_k \mathbf{d}_k\|_F^2. \quad (5-5)$$

Observando a derivação feita na Equação 2-17, chegamos em:

$$\begin{aligned} \frac{\partial}{\partial \mathbf{d}_k} (\mathbf{F}_{ep}) &= \frac{\partial}{\partial \mathbf{d}_k} \left((\mathbf{E}_k - \mathbf{d}_k \mathbf{X}_{k,:})^T (\mathbf{E}_k - \mathbf{d}_k \mathbf{X}_{k,:}) + \lambda (\mathbf{P}_k \mathbf{d}_k)^T (\mathbf{P}_k \mathbf{d}_k) \right) \\ &= -2(\mathbf{E}_k - \mathbf{d}_k \mathbf{X}_{k,:}) \mathbf{X}_{k,:}^T + 2\lambda \mathbf{P}_k^T \mathbf{P}_k \mathbf{d}_k \\ &= -2\mathbf{E}_k \mathbf{X}_{k,:}^T + 2\|\mathbf{X}_{k,:}\|_2^2 \mathbf{I} \mathbf{d}_k + 2\lambda \mathbf{P}_k^T \mathbf{P}_k \mathbf{d}_k \end{aligned} \quad (5-6)$$

Igualando a Equação 5-6 a zero, chegamos em:

$$\mathbf{d}_k = \left(\|\mathbf{X}_{k,:}\|_2^2 I + \lambda \mathbf{P}_k^T \mathbf{P}_k \right)^{-1} \mathbf{E}_k \mathbf{X}_{k,:}^T \quad (5-7)$$

Por simplicidade, podemos impor que a matriz \mathbf{P}_k seja uma matriz diagonal, com zeros onde não se quer penalizar e não zero onde se deseja penalizar. Desta forma, a inversão de matriz na Equação 5-7 fica trivial. O parâmetro λ controla o efeito da regularização.

Repetimos o processo de treinamento de dicionário como na Seção 5.2 usando o SGK com regularização. A Figura 5.4 exibe resultados de treinamento de dicionário usando o par de algoritmos OMP e SGK com regularização com 2 tamanhos de entrada e dois valores de parâmetro de regularidade de supertraços do SLICT.

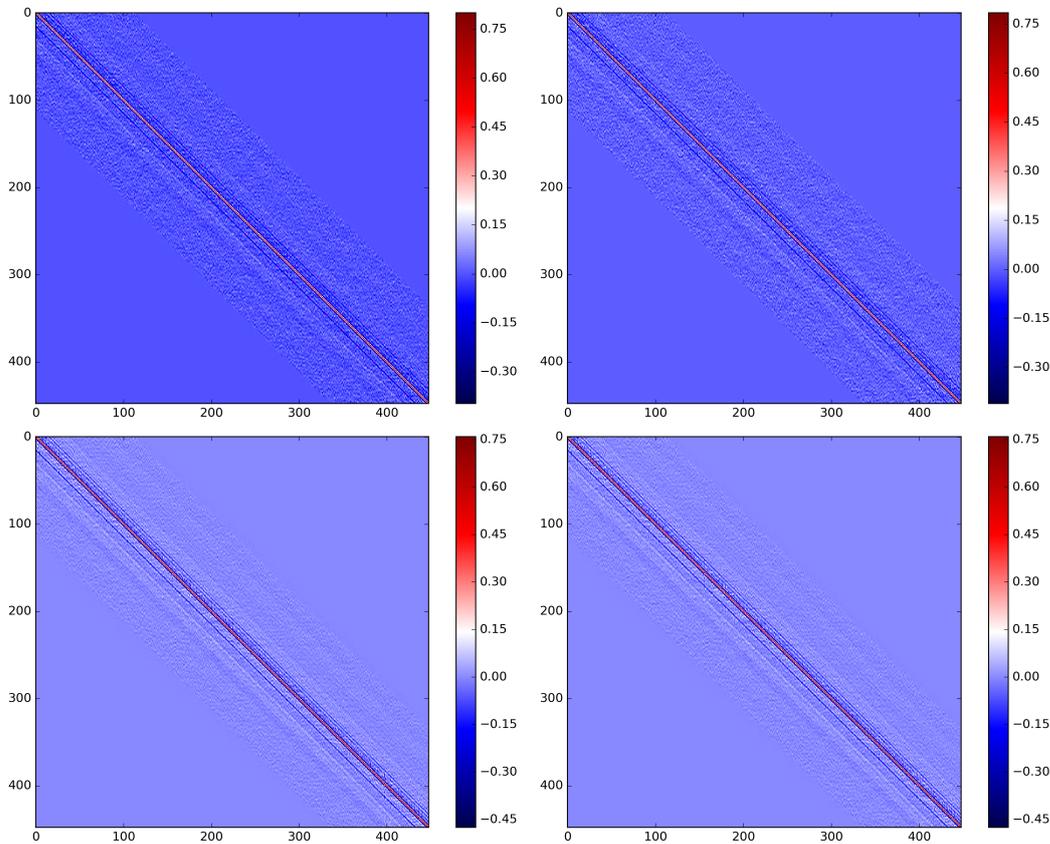


Figura 5.4: Resultado inicial da execução de treinamentos de dicionários usando os algoritmos OMP e SGK com regularização. Coluna da esquerda, usando parâmetro de regularidade de supertraços com valor igual a 0,125. Coluna da direita, usando parâmetro de regularidade de supertraços com valor igual 2. Acima, limite de 2000 supertraços (com 1977 e 1978 selecionados). Abaixo, limite de 8000 supertraços (com 7998 e 8023 selecionados).

Como na Seção 5.2, montamos a Tabela 5.3 mostrando o erro de reconstrução (ou resíduo) das amostras de entrada do processo de treinamento

utilizando SGK com regularização.

| Erro de reconstrução | 0,125 | 2,0 |
|-----------------------------|------------------|------------------|
| 2000 | 1,6% (10,806%) | 1,596% (11,727%) |
| 8000 | 1,912% (13,947%) | 1,914% (15,12%) |

Tabela 5.3: Erro de reconstrução das amostras de entrada do processo de treinamento de dicionário, variando a quantidade aproximada de traços de entrada (2000 e 8000) e o parâmetro de regularidade dos supertraços (0,125 e 2,0). Medidas média (e máxima, em parênteses) percentual. Medidas tomam o valor máximo absoluto do dado de entrada e do erro encontrado. Algoritmos utilizados: OMP e SGK com regularização.

Aplicando a eliminação dos valores do dicionário como anteriormente, podemos chegar na configuração ilustrada na Figura 5.5.

PUC-Rio - Certificação Digital Nº 1421617/CA

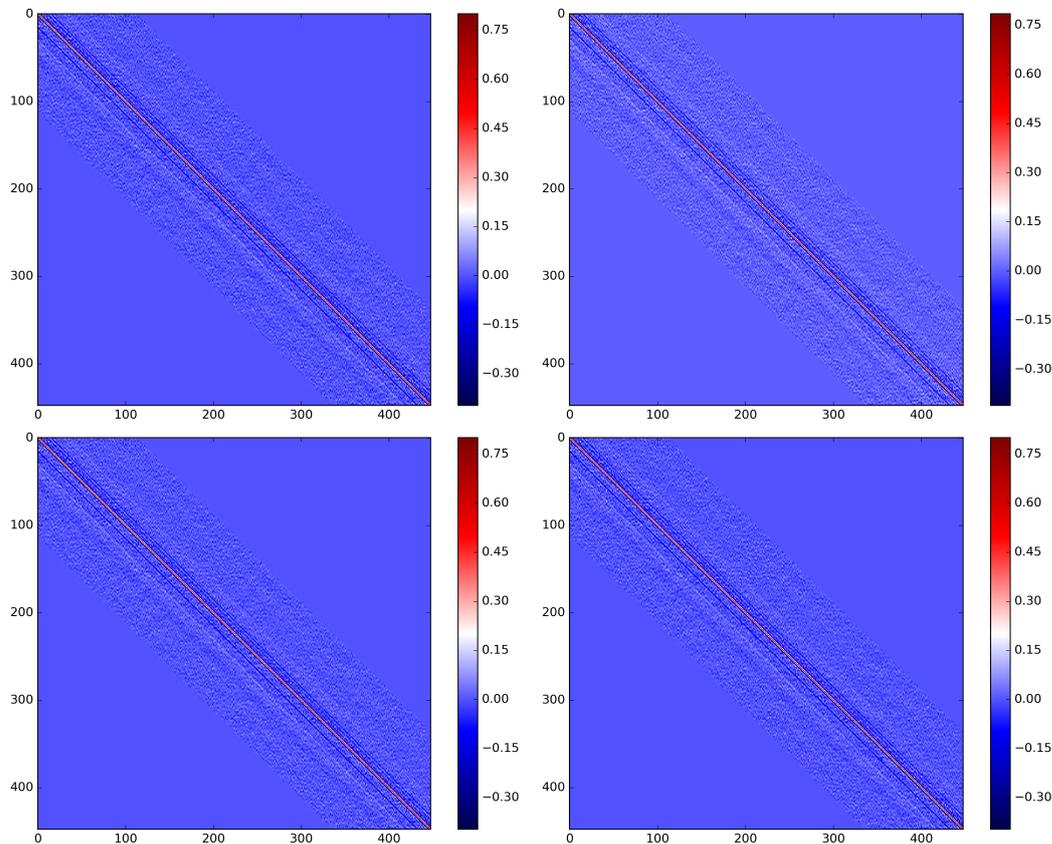


Figura 5.5: Resultado da aplicação de poda ao dicionário inicial gerado usando OMP e SGK com regularização. Coluna da esquerda, usando parâmetro de regularidade de supertraços com valor igual a 0,125. Coluna da direita, usando parâmetro de regularidade de supertraços com valor igual a 2. Acima, limite de 2000 supertraços (com 1977 e 1978 selecionados). Abaixo, limite de 8000 supertraços (com 7998 e 58023 selecionados). Amostras com distância superior a 100 da diagonal foram eliminadas. Amostras de índices de 50 a 100 distantes da diagonal tiveram atenuação linear até chegar em 0.

Procedendo com a eliminação da banda do dicionário como na Seção 5.2, chegamos na Tabela 5.4.

| Erro de reconstrução | 0,125 | 2,0 |
|----------------------|------------------|------------------|
| 2000 | 1,605% (10,821%) | 1,602% (11,733%) |
| 8000 | 1,917% (14,076%) | 1,919% (15,31%) |

Tabela 5.4: Erro de reconstrução das amostras de entrada a partir do uso do dicionário ajustado, variando a quantidade aproximada de traços de entrada (2000 e 8000) e o parâmetro de regularidade dos supertraços (0,125 e 2,0). Medidas média (e máxima, em parênteses) percentual. Medidas tomam o valor máximo absoluto do dado de entrada e do erro encontrado. Observa-se um aumento muito pequeno no erro em todas as medidas, sensivelmente aquém do anteriormente mostrado na Tabela 5.2. Algoritmos utilizados: OMP e SGK com regularização.

A Figura 5.6 ilustra a diferença de conteúdo de dicionários, gerados usando OMP e SGK com regularização, antes e depois do recorte e ajuste (normalização dos átomos), selecionando somente uma linha da matriz.

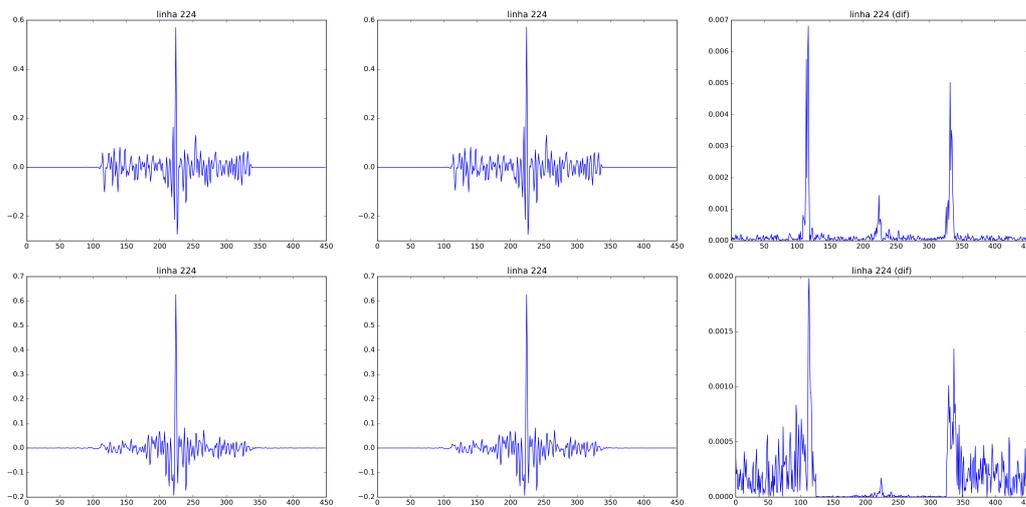


Figura 5.6: Visualização da linha de índice 224 de dois dicionários, antes (à esquerda) e depois (ao centro) do recorte e ajuste (normalização), com diferença absoluta mais à direita. Acima, utilizando 2000 supertraços e parâmetro de regularidade com valor 0,125. Abaixo, utilizando 8000 supertraços e parâmetro de regularidade com valor 0,125. Dicionários gerados usando OMP e SGK com regularização. Observa-se grande diferença em relação ao que mostra a Figura 5.3.

Claramente, o método com regularização (OMP, SGK com regularização e corte) apresenta um ganho importante em relação ao procedimento inicial (OMP, MOD e corte), observando o resíduo da reconstrução das amostras de treinamento.

Neste instante, está finalizada a primeira etapa de geração de um dicionário que pode servir como operador da transformação do conteúdo de picos esparsos de refletividade em amplitudes sísmicas. A estrutura gerada desta forma carrega uma propriedade de localidade. Quando o i -ésimo átomo, ou a i -ésima coluna da matriz de dicionário, é escolhido para compor a combinação linear esparsa que resulta num traço, é a i -ésima amostra vertical do traço de refletividade que aparecerá diferente de zero, ou seja, é na posição correspondente à i -ésima amostra que a representação esparsa posiciona um pico esparsos e, compondo tais marcações em traços vizinhos, pode-se perceber a presença de trechos de horizontes sísmicos. Esta marcação dos picos esparsos tende a ressaltar os eventos sísmicos mais significativos em cada traço, visto que a própria abordagem “gulosa” dos algoritmos OMP e LS-OMP acabam por fazer este tipo de escolha, mesmo que haja variação na quantidade de picos entre traços vizinhos.

A partir de então, tenta-se aumentar o poder de codificação do dicionário extraído através da inclusão de cópias modificadas dos átomos de forma que estes carreguem em si a simulação do efeito de distorções de sinal provocados pelo próprio subsolo e, para tal, fazemos um uso particular da técnica de geração de filtro da terra, abordado na seção seguinte.

5.4

Aumento artificial do dicionário via filtro da terra

O aumento artificial do dicionário se dá pela aplicação de uma matriz que modela dispersão e atenuação seguindo o modelo de Futterman descrito em Oliveira (2005). Tal matriz pode ser calculada como na Equação 5-8.

$$A[t, f] = \exp\left(\frac{-\pi f t}{Q}(1 + i \ln(f/f_r))\right), \quad i = \sqrt{-1} \quad (5-8)$$

Na Equação 5-8, a matriz A codifica a resposta de amplitude, que sofre os efeitos simultâneos de atenuação e dispersão, em função do tempo de trânsito do pulso (índice t , crescente ao longo das linhas ou do eixo vertical descendente) e da frequência do sinal (índice f) ao longo das colunas. Na mesma equação, o fator Q é entendido como o “fator de qualidade”, cuja faixa de valor parece ser a mencionada em Oliveira (2005), algo entre 20 e 100, sendo valores abaixo de 70 considerados baixos. Quanto menor o valor deste parâmetro, mais facilmente perceptível seu efeito de atenuação. Há também o uso do parâmetro f_r : uma frequência de referência. Na prática, o modelo de Futterman é uma tentativa de expressar que, quanto maior a frequência do sinal de entrada e quanto maior

a profundidade, indiretamente calculada pelo tempo em que o pulso teve seu retorno lido, maior serão os efeitos de atenuação e dispersão aplicados pela terra.

A Figura 5.7 mostra, a título de ilustração, matrizes de resposta de amplitude de acordo com o modelo de Futterman.

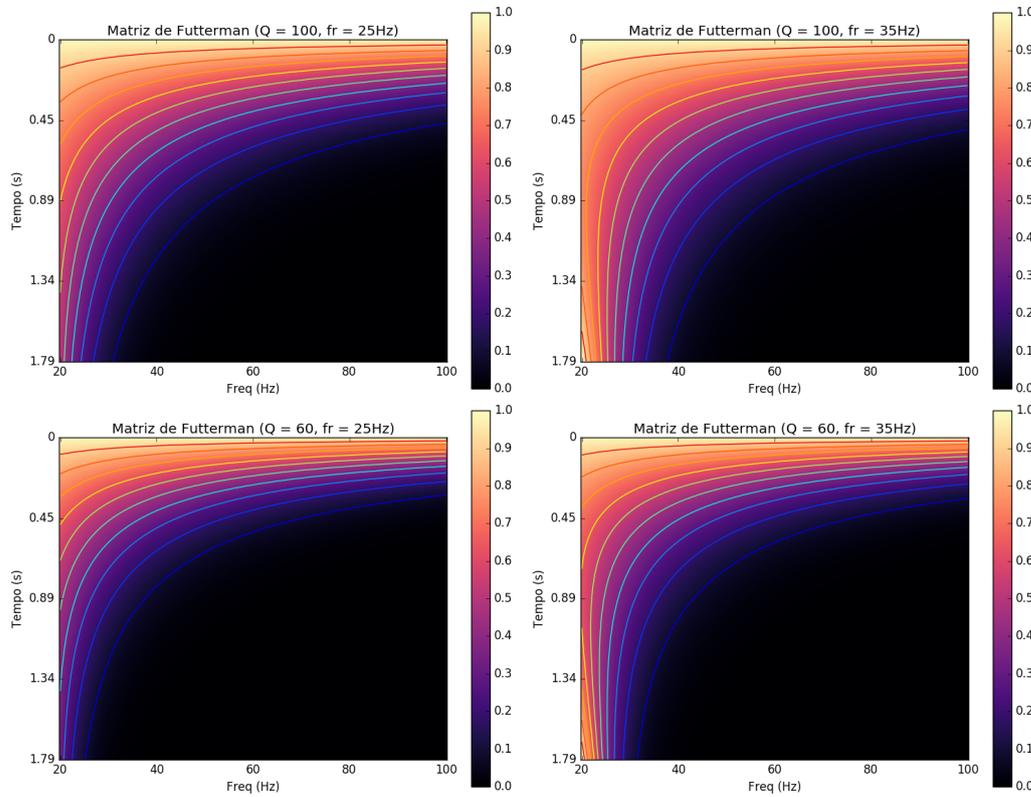


Figura 5.7: Ilustração da matriz de resposta de amplitude segundo o modelo de atenuação e dispersão de Futterman. Da esquerda para a direita, variação da frequência de referência f_r (25 e 35Hz). De cima para baixo, variação do fator de qualidade Q (100 e 60).

A partir do dicionário inicialmente extraído, passamos a aumentá-lo na tentativa de fazer com que seus átomos possam melhor refletir o comportamento da ondaleta sísmica conforme se aumenta a profundidade ou se varia o conteúdo de frequência de cada ondaleta codificada pelos átomos do dicionário. Este aumento se dá pela aplicação do modelo de Futterman, parametrizado pelo fator de qualidade Q e pela frequência de referência f_r , aos átomos do dicionário.

Como ilustração deste processo, a Figura 5.8 exhibe o resultado da ampliação de um dicionário após a aplicação de filtros da terra com parâmetros variados.

O processo final de representação esparsa do dado de amplitudes sísmicas poderá utilizar este dicionário aumentado. Vale pontuar que o processo de

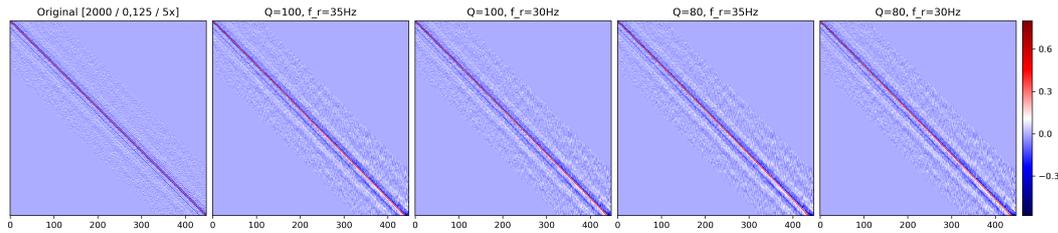


Figura 5.8: Dicionário aumentado em 5 vezes usando os seguintes parâmetros para a matriz de resposta de amplitude segundo o modelo de Futterman: ($Q = 100$ e $f_r = 35Hz$), ($Q = 100$ e $f_r = 30Hz$), ($Q = 80$ e $f_r = 35Hz$) e ($Q = 80$ e $f_r = 30Hz$). Dicionário aumentado a partir do extraído com 2000 traços e parâmetro de regularidade de 0,125 como na Figura 5.4.

representação esparsa, e por conseguinte o de aprendizado de dicionário, tem custo proporcional à quantidade de amostras de entrada. Assim como em outros procedimentos de aprendizado de máquina, como as máquinas vetor de suporte (Han, Kamber e Peir, 2012)², quanto maior a quantidade de dados de entrada, maior será o poder de generalização do preditor e menor será a chance de *overfitting*. Por outro lado, pelo menos para o caso apresentado nesta tese, parece prudente avaliar a quantidade de amostras de entrada para o procedimento de treinamento de dicionário. Isso se justifica pela, a priori, alta dimensionalidade do dado de entrada, pela dificuldade em se chegar a um grupo de representantes do dado completo e, sobretudo, pelo custo computacional do treinamento do dicionário, majoritariamente dominado pelo procedimento de representação esparsa³. O processo final de representação esparsa do dado completo tem como atenuante a possibilidade de ser executado de forma massivamente paralela, assumindo que não há dependência de resultado entre traços⁴.

O capítulo seguinte vincula a obtenção de um dicionário aumentado, a representação esparsa do dado original na forma de picos esparsos de refletividade, a possibilidade de compressão e uma discussão sobre a aplicação dos artefatos gerados neste processo por uma ferramenta de interpretação em geofísica.

²Capítulo 9, Seção 9.3.

³Há procedimento de aprendizado de dicionário *online*, como descrito em Mairal et al. (2009). Em avaliação inicial, demonstrou resíduo de reconstrução do sinal de treinamento maior e, por isso, não foi utilizado nesta tese.

⁴Neste contexto, o resultado é sinônimo do conteúdo esparsa de refletividades de um traço.

6 Resultados

6.1

Resultados iniciais: deconvolução por picos esparsos

O dado utilizado nesta etapa é o modelo Marmousi2 (Martin, Wiley e Marfurt, 2006) – já ilustrado nas Figuras 3.3 e 3.4 no Capítulo 3 – reapresentado na Figura 6.1. Tentamos nos utilizar de uma área do dado utilizada em Wang et al. (2016). A área em recorte é composta por 200 traços, cada qual com 800 amostras, e está ilustrada na Figura 6.2. Nesta configuração, escolhemos alguns parâmetros antes de rodar os experimentos. Primeiramente, para a ondaleta, selecionamos uma frequência dominante (33Hz) e um ângulo (fase) de rotação da ondaleta (45°). Em seguida, fazemos a convolução desta wavelet com os traços de refletividade extraídos do modelo. Adiciona-se, finalmente, um ruído cujo valor absoluto corresponde a 5% do valor máximo absoluto de amplitude obtida com a convolução. O ruído não tem distribuição normal, mas sim uniforme.

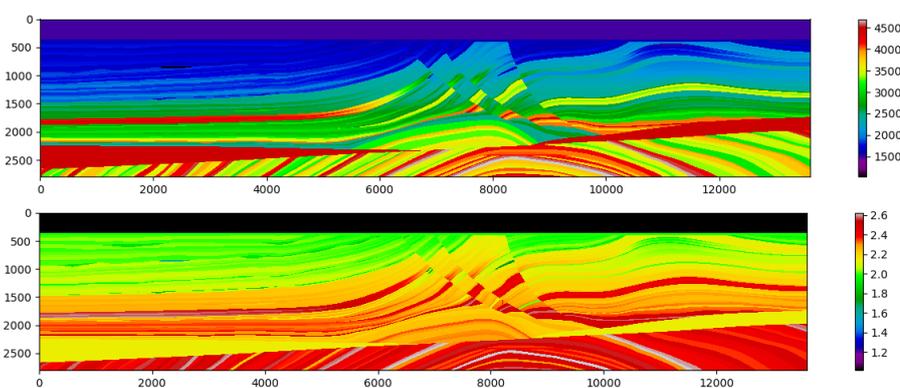


Figura 6.1: Dado sintético Marmousi2. Acima, conteúdo de velocidade (em m/s). Abaixo, conteúdo de densidade (em g/cm^3).

6.1.1

Métricas

Fizemos uso de algumas métricas para comparar resultados de algoritmos ou estratégias diferentes (Sayood, 2012; Gonzales, 2007).

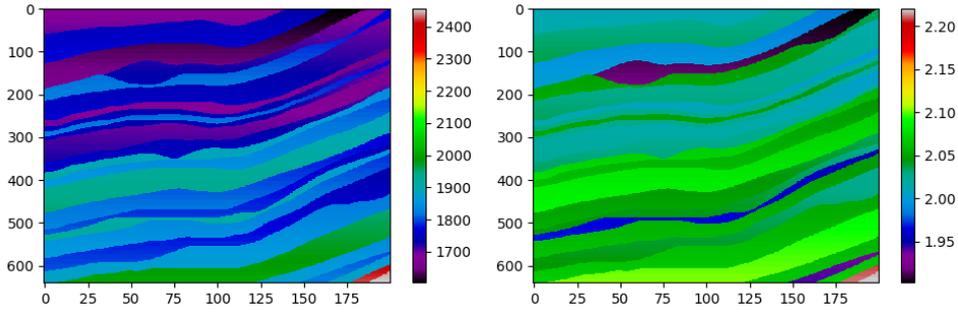


Figura 6.2: Área selecionada do dado Marmousi2. À esquerda, conteúdo de velocidade. À direita, conteúdo de densidade. Eixo horizontal: uma amostra por metro. Eixo vertical: uma amostra a cada 1,25m.

Erro médio quadrático. Dados dos vetores \mathbf{x} e \mathbf{y} de tamanho n , sendo \mathbf{x} o vetor de entrada original e \mathbf{y} o vetor reconstruído, o **erro médio quadrático** (EMQ) é dado por $EMQ = \frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2$.

Razão sinal-ruído. Sendo o valor médio quadrático do sinal de entrada (VMQSE) igual a $\frac{1}{n} \sum_{i=1}^n (x_i)^2$, podemos calcular a **razão sinal-ruído** (RSR) dada por $RSR = VMQSE/EMQ$. É usual calcular a RSR em escala logarítmica, medida em decibéis (dB):

$$RSR = 10 \log_{10}(VMQSE/EMQ) \quad (6-1)$$

Em Tian e Li (2017), há outra fórmula para a razão sinal ruído dada por

$$RSR = 10 \log_{10} \left(\frac{\text{Var}^2(\mathbf{x})}{\text{Var}^2(\mathbf{y})} \right), \quad (6-2)$$

em que $\text{Var}(\cdot)$ se refere à variância.

Pico da razão sinal-ruído. O pico da razão sinal-ruído (PRSR) aqui utilizada, também com unidade em decibéis (dB), é calculada da seguinte forma:

$$PRSR = 20 \log_{10}((\max(x) - \min(x))/EMQ). \quad (6-3)$$

em que $\max(\cdot)$ e $\min(\cdot)$ se referem ao valor máximo e mínimo num conjunto, respectivamente. O pico da razão sinal-ruído é também utilizada em processamento de imagens para avaliar a qualidade da imagem após descompressão, e aqui foi feita uma adaptação ao que está descrito em Sayood (2012)¹ ao usar a faixa de valores $(\max(x) - \min(x))$.

Similaridade do cosseno. Sejam $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$. A similaridade do cosseno entre os vetores \mathbf{a} e \mathbf{b} é dada por:

$$S_{cos}(\mathbf{a}, \mathbf{b}) = \frac{\langle \mathbf{a}, \mathbf{b} \rangle}{\|\mathbf{a}\| \|\mathbf{b}\|} = \frac{\mathbf{a}^T \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} \quad (6-4)$$

¹Capítulo 8, pág. 222.

A similaridade do cosseno tem valores na faixa $[-1, 1]$. Vetores idênticos têm similaridade do cosseno igual a 1.

Esta métrica é aplicada para comparar uma ondaleta extraída contra uma ondaleta previamente conhecida.

Para o caso de comparação entre picos de refletividade extraídos contra conhecidos, para um único traço, é feito uso de uma derivação desta métrica, aqui chamada de **similaridade do cosseno removidos os zeros** (SCRZ). Normalmente, a quantidade de valores iguais a zero ao longo de um traço de refletividades é muito superior à quantidade de não zeros. Removendo os zeros e, em seguida, aplicando a métrica convencional de similaridade do cosseno, fazemos uma comparação semelhante à que se faz entre *strings* de caracteres. A SCRZ admite uma similaridade máxima se a sequência de valores de refletividade for idêntica, embora possa haver, no perfil original de refletividade, uma quantidade variável de valores iguais a zero entre os picos. Pode-se interpretar a SCRZ como uma simplificação de uma métrica de similaridade por casamento de padrões como o DTW (Algoritmo 7).

Experimentos a seguir extraem uma ondaleta e vários picos de refletividade para cada grupo de traços de amplitude. Assim, calcula-se um valor médio da SCRZ e um único valor da similaridade do cosseno da ondaleta (SCO) para cada grupo de traços processados.

Índice de qualidade de deconvolução. Como mostrado em Fernandes, Lopes e Gattass (2017), parece interessante observar a combinação de resultados de refletividade e ondaleta e, por isso, foi proposto o índice de qualidade da deconvolução (IQD), dado por:

$$IQD = \frac{\sqrt{\max(0, MSCRZ(\mathbf{r}_e, \mathbf{r}_o))^2 + \max(0, MSCO(\mathbf{w}_e, \mathbf{w}_o))^2}}{\sqrt{2}}, \quad (6-5)$$

em que MSCRZ é a média das medidas SCRZ, e MSCO é a média das medidas SCO, levando em conta a qualidade da refletividade e da ondaleta. Os termos \mathbf{r}_e e \mathbf{r}_o se referem à refletividade extraída e à original, respectivamente. Os termos \mathbf{w}_e e \mathbf{w}_o se referem à ondaleta extraída e à original, respectivamente. O IQD só cabe quando se conhece tanto a ondaleta quanto o conteúdo de refletividade e, portanto, somente aplicável para dados sintéticos. Tem faixa de valores no intervalo $[0, 1]$. Quando há coincidência entre valores conhecidos e extraídos, tanto de ondaleta quanto de refletividade, o IQD assume o valor máximo (1).

6.1.2 Experimento inicial

Inicialmente, avaliamos a qualidade da recuperação/reconstrução dos próprios traços utilizados no processo de separação da amplitude em ondaleta e refletividade. Dos 200 traços disponíveis, fizemos uma amostragem uniforme, com salto de 20 em 20 traços. Assim, este dado foi dividido em 10 partes de 20 traços cada. Sendo $i \in [0, 9]$ o índice da parte, a i -ésima parte utiliza os traços de índice $i + 10j$, com $j \in [0, 19]$. A Figura 6.3 ilustra um exemplo de dado de entrada de uma destas iterações. Tal experimento foi repetido 10 vezes. Em cada repetição, um novo dado era gerado, com a mesma ondaleta (fase de 45° , frequência dominante de 33Hz) e refletividade originais, mas com mudança provocada por ruído uniforme de amplitude máxima de 5% do valor original da amostra. Os algoritmos avaliados foram inicializados com a mesma ondaleta inicial (fase de 30° , frequência dominante de 30Hz). Como se tinha o gabarito para comparar os resultados, observamos tanto o conteúdo dos refletores quanto da ondaleta extraída ao final, tomando como referência os próprios traços usados como entrada. Assim sendo, tal avaliação equivale, em aprendizado de máquina, a uma medida de qualidade do modelo usando dado de treinamento. A Tabela 6.1 resume tais resultados coletados.

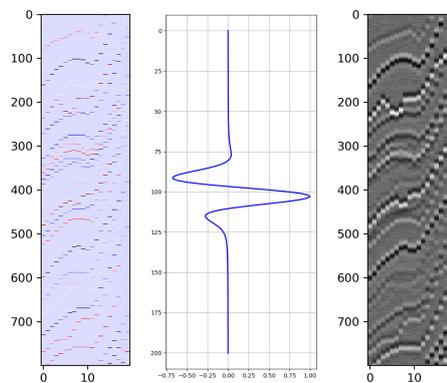


Figura 6.3: Ilustração de um dado de entrada em uma das 10 iterações para extração de ondaleta e refletividade. À esquerda, dado de refletividade sem ruído, resultante da aplicação da fórmula da Equação 2-1. Ao centro, ondaleta usada como gabarito, com frequência dominante 33Hz, fase de 45° , duração de 0,201 s. À direita, resultado da convolução da ondaleta de gabarito com a refletividade e adição de ruído. À esquerda e à direita, imagens mostram 20 traços com 800 amostras.

A Tabela 6.2 apresenta valores de IQD e quantidade de iterações sobre os mesmos dados usados para gerar a Tabela 6.1.

Em Dumitrescu e Irofti (2018) há a recomendação de que, uma vez que um método tenha sido escolhido para se gerar a fatoração de uma matriz, o mesmo método deve ser usado para recuperar o dado original. Contrariando

| Avaliação 1 | MSCRZ | | MSCO | | RSR | |
|-------------|--------------|----------|--------------|----------|---------------|----------|
| | μ | σ | μ | σ | μ | σ |
| OMP | 0,395 | 0,171 | 0,791 | 0,035 | 21,635 | 1,375 |
| Fista TSMF | 0,146 | 0,091 | 0,873 | 0,031 | 20,477 | 1,427 |
| Lobbes | 0,361 | 0,237 | 0,896 | 0,063 | 22,346 | 2,539 |

Tabela 6.1: Avaliação de qualidade da deconvolução dos traços dados como entrada para o processo de extração de ondaleta e refletividade. MSCRZ: média da similaridade de cosseno das refletividades após a remoção dos zeros. MSCO: média da similaridade do cosseno da ondaleta. RSR: razão sinal-ruído (como na Equação 6-2). Valores arredondados para 3 casas decimais.

| Avaliação 2 | IQD | | Iterações | |
|-------------|--------------|----------|-----------|----------|
| | μ | σ | μ | σ |
| OMP | 0,62 | 0,049 | 14,39 | 3,37 |
| Fista TSMF | 0,624 | 0,029 | 12,6 | 3,153 |
| Lobbes | 0,686 | 0,082 | 25,58 | 8,004 |

Tabela 6.2: Avaliação a partir do índice de qualidade de deconvolução (IQD) como proposto em Fernandes, Lopes e Gattass (2017) e da quantidade de iterações para a convergência dos algoritmos. Valores arredondados para 3 casas decimais.

esta recomendação, experimentamos a reconstrução dos 200 traços utilizando somente o algoritmo OMP para a reconstrução esparsa e as 300 matrizes de Toeplitz (ou ondaletas) geradas pelos 3 métodos já descritos. A escolha do algoritmo OMP foi motivada pelo desempenho (em tempo de execução) superior aos demais. Usou-se uma única ondaleta de gabarito e o mesmo conteúdo de refletividade, ambos idênticos ao do experimento inicial. Houve 5 repetições deste experimento e, para cada uma, aplicou-se ruído aleatório uniforme de intensidade máxima de 5% sobre o valor de amplitude. A Tabela 6.3 mostra o que foi obtido. Tal resultado pode servir como meio de avaliação dos métodos quanto à qualidade das ondaletas extraídas, uma vez que são determinantes para a recuperação esparsa das refletividades.

| Avaliação 3 | RSR | | | |
|-------------|---------------|--------------|---------------|---------------|
| | μ | σ | máx. | mín. |
| OMP | 21,141 | 0,548 | 21,94 | 20,198 |
| Fista TSMF | 21,393 | 0,363 | 22,089 | 20,641 |
| Lobbes | 21,461 | 0,258 | 21,918 | 20,877 |

Tabela 6.3: Avaliação de qualidade da reconstrução. Tomando cada uma das matrizes de Toeplitz geradas pelo procedimento da Tabela 6.1 e aplicando unicamente o algoritmo OMP para cálculo da representação esparsa de cada traço de refletividade. RSR: razão sinal-ruído (como na Equação 6-2). Valores arredondados para 3 casas decimais.

A Figura 6.4 ilustra a extração de refletividades dos 200 traços usados no experimento anterior. Para cada método, foi utilizada a matriz de Toeplitz que gerou a melhor razão sinal ruído na reconstrução de todo o dado.

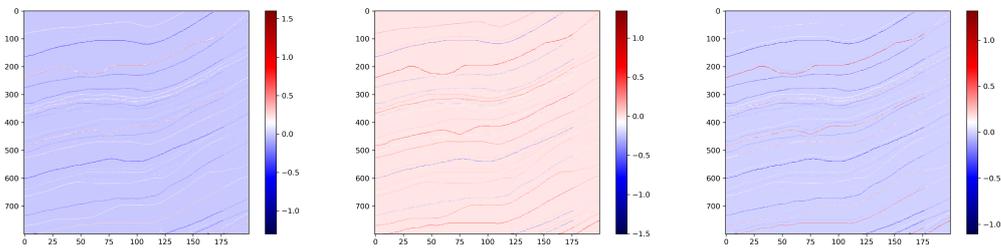


Figura 6.4: Ilustração da reconstrução dos 200 traços de refletividades. À esquerda, usando ondaleta extraída via OMP (RSR 21,94). Ao centro, usando ondaleta extraída via *Fista TSMF* (RSR 22,089). À direita, usando ondaleta extraída via *Lobbes* (RSR 21,918).

A Figura 6.5 ilustra a diferença de posicionamento entre refletores originais e os obtidos como na Figura 6.4. Percebe-se que as refletividades extraídas acompanham as originais, embora haja deslocamento vertical, possivelmente causado pela rotação de fase das ondaletas.

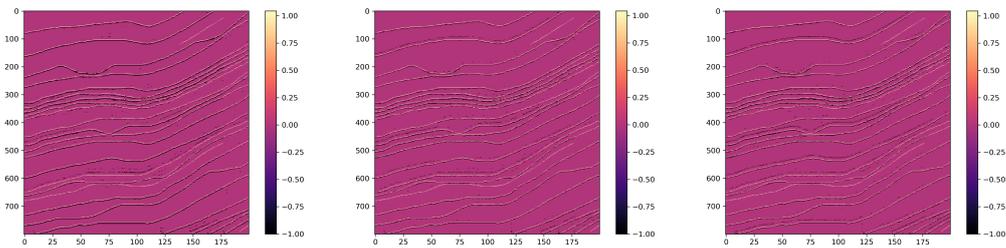


Figura 6.5: Ilustração da diferença da reconstrução dos 200 traços de refletividades. À esquerda, usando ondaleta extraída via OMP (RSR 21,94). Ao centro, usando ondaleta extraída via *Fista TSMF* (RSR 22,089). À direita, usando ondaleta extraída via *Lobbes* (RSR 21,918). As refletividades originais foram escaladas para o valor 1, enquanto as extraídas foram escaladas para -1. Imagens mostram a soma.

As avaliações tomadas pelas Tabelas 6.1, 6.2 e 6.3 expõem o emprego do modelo convolucional, para o qual basta extrair uma ondaleta para o dado inteiro. Neste modelo, a reconstrução se consegue ao se realizar convolução da ondaleta extraída com os picos de refletividade gerados por um algoritmo de representação esparsa.

Elaboramos, então, uma hipótese de trabalho, em que supomos que, ao se obter mais de uma ondaleta para o dado, podemos reconstruí-lo com qualidade superior. Não parece usual aceitar tal hipótese para dados sintéticos. Assim,

a hipótese de trabalho passa a ser **obter** e **avaliar** uma matriz similar à de Toeplitz², para dado real, com obtenção e avaliação divididas em duas etapas.

Na primeira etapa, o operador linear do dicionário pode reter informação sem a restrição da ondaleta única como ocorre em Wang et al. (2016) e em Fernandes, Lopes e Gattass (2017). Ao incorporar informação oscilatória não mais restrita a uma única ondaleta, opta-se por obedecer a uma formulação na qual os átomos do dicionário têm uma faixa de seus coeficientes anulada e outra livre. A faixa livre fica em torno da diagonal da matriz dicionário inicialmente gerada pelo processo de treinamento. Esta abordagem se assemelha à adotada em Fomel (2007), em que se busca aproximar os coeficientes requeridos a uma curva específica. No caso presente, utilizamos o método SGK com regularização.

Na segunda etapa, é feita expansão artificial de um dicionário recém gerado, adicionando variações que poderiam melhor representar os efeitos de filtro que a própria terra exerce sobre as ondas sísmicas. Resultados empíricos da aplicação de tais dicionários expandidos poderiam demonstrar a validade de tais hipóteses, o que é feito na seção a seguir.

6.2

Resultados em dado real público

Para o conjunto de resultados obtidos a seguir, utilizamos um recorte do dado público de amplitudes sísmicas do bloco F3³, do qual utilizamos as primeiras 588 *inlines*, as primeiras 939 *crosslines* e as últimas 448 amostras verticais. A partir daqui, tal dado recortado será referenciado como **F3R**. Também a partir deste ponto, referências a índices de uma das três dimensões deste dado F3R estão indexados a partir do 0 (zero). A Figura 6.6 ilustra o dado F3R.

A partir do trabalho em Wang et al. (2016), observamos que o processo de deconvolução pede que façamos uma amostragem do dado original a fim de extrairmos um pulso, uma ondaleta e, a partir daí, se faz a extração da refletividade do dado completo. O passo de se extrair uma única ondaleta no trabalho de Wang e outros equivale a extrair um dicionário cuja estrutura é o de uma matriz de Toeplitz. Em algoritmos de treinamento (não *online*) de dicionário, um passo essencial é o de representação esparsa usando um

²A expansão da matriz de dicionário usando filtro da terra acaba por gerar, em cada trecho da expansão, uma matriz similar à matriz de Toeplitz, como já se mostrou no capítulo anterior.

³O dado sísmico de amplitude está disponível para utilização na versão gratuita do aplicativo OpendTect, acessado a partir do sítio <https://www.dgbes.com/index.php/software#free>. O bloco F3 compõe um pacote com dados de naturezas diversas, e está disponível a partir do sítio <https://terrannubis.com/datainfo/Netherlands-Offshore-F3-Block-Complete>.

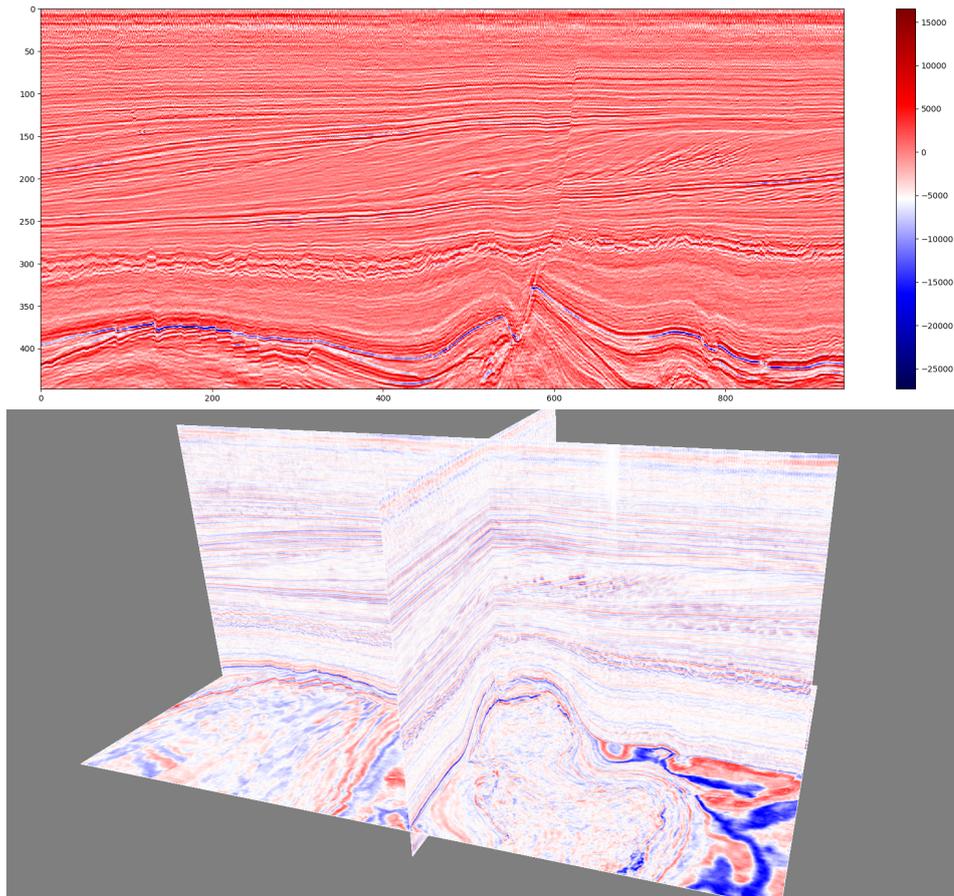


Figura 6.6: Ilustração do dado F3R. Acima, exibição da *inline* 200. Abaixo, exibição em 3D com 3 planos paralelos aos eixos do dado. São 588 amostras na *inline*, 939 na *crossline* e 448 amostras na vertical. Cada amostra corresponde a um intervalo de 4ms na vertical e 25m na horizontal.

dicionário candidato provisório. Tal passo tem custo proporcional à quantidade de elementos de entrada a terem sua representação transformada. Daí, percebe-se a oportunidade ou a necessidade de uma técnica de amostragem mais apurada a fim de reduzir o conjunto de entrada e, ao mesmo tempo, gerar um resultado satisfatório de dicionário treinado e de representação esparsa posterior. Em Rodríguez et al. (2014) cita-se o que chama de *coreset* (ou conjunto-núcleo), um subconjunto das amostras de entrada que podem servir como representantes do dado original, tomando como meio de avaliação da qualidade de tal subamostragem um algoritmo específico⁴. Parece haver um vínculo teórico entre tal subconjunto representante com algoritmos de agrupamento, uma vez que, nestes algoritmos, elementos que se assemelham tendem a estar no mesmo grupo, e, reciprocamente, elementos diferentes entre

⁴Por exemplo, tomando o algoritmo de fecho convexo, um conjunto-núcleo do dado original seria um subconjunto capaz de aproximar o resultado do algoritmo. Um efeito prático óbvio da seleção de um conjunto-núcleo é o de redução de tempo de execução do algoritmo selecionado, uma vez que se reduz o tamanho do conjunto de entrada.

si tendem a pertencer a grupos distintos. Algum ou alguns dos elementos de cada um destes grupos poderiam ser os elementos de um *coreset* a depender da aplicação.

Voltando ao caso da amostragem de traços de amplitudes sísmicas, nota-se que a execução do *K-Means* requer o cálculo de distância entre todas as amostras (todos os traços). Daí, optou-se, então, pela aplicação do SLICT, derivado do SLIC, para execução da tarefa de agrupamento para posterior seleção de representantes de cada grupo.

Os representantes derivados dos grupos são usados como amostras de entrada para o processo posterior de treinamento de dicionário. Neste ponto, utilizamos o método SGK com regularização para extração do dicionário inicial. Observando a formulação da Equação 5-7, a matriz de penalização \mathbf{P} , que é quadrada de lado M , tem seus elementos $p_{i,j}$ calculados como segue:

- Se o $|i - j| < M/4$, $p_{i,j} = 0$,
- Senão $p_{i,j} = |i - j| - M/4$,

em que i e j são índices, respectivamente, de linha e de coluna de \mathbf{P} , e o parâmetro λ da mesma equação tem valor grande o suficiente ⁵.

De posse do dicionário inicial, aumentamos artificialmente sua largura a partir de múltiplas aplicações do filtro da terra, usando poucas variações dos parâmetros de fator de qualidade e frequência de referencia.

Com os dicionários aumentados, partimos, então para a extração dos picos de refletividade em cada traço. A quantidade de picos em cada traço é uma variável de entrada, e cabe a análise de um especialista para determinar, em cada caso em estudo, qual a faixa de valores válidos. No presente estudo, atribuímos, arbitrariamente, uma quantidade limite correspondente a 20% do número de amostras de cada traço.

6.2.1

Uma Análise Numérica

Foi feita, inicialmente, uma análise puramente numérica a fim de, eventualmente, identificar algum padrão que pudesse relacionar as diversas variáveis envolvidas nesta inversão. Nesta etapa, as seguintes variáveis foram levadas em conta:

- O parâmetro de regularidade dos supertraços utilizado pelo SLICT (0,125 e 2);

⁵Foram realizados experimentos exploratórios tomando valores crescentes de potências de 10 para este parâmetro, e o valor 10^8 foi primeiro valor adequado ao propósito específico da regularização dos átomos do dicionário tomando como exemplos os traços do dado F3R. Tal valor foi obtido sem fazer a normalização das amplitudes do dado F3R.

- Quantidade de supertraços dos quais são extraídos os traços representativos, que são usados para treinar um dicionário (2000, 4000, 8000 e 16000);
- Multiplicidade do aumento do dicionário inicial via aplicações parametrizadas do filtro da terra (5 e 10).

Inicialmente, foram feitas algumas medidas do dado original a fim de comparar com os dados derivados. A Figura 6.7 exibe histogramas do dado utilizado nos experimentos. A Tabela 6.4 lista estatísticas extraídas do dado original.

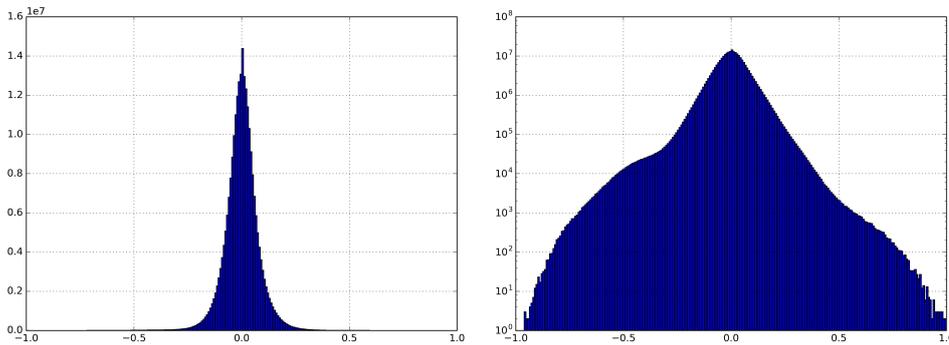


Figura 6.7: Histogramas a partir do dado de amplitudes F3R. À esquerda, histograma comum com 256 subdivisões. À direita, histograma com escala vertical logarítmica, com 256 subdivisões. Amplitudes normalizadas para o intervalo $[-1, 1]$.

| Avaliação do dado F3R | | |
|-----------------------|---|------------------------|
| Métrica | Fórmula | Valor obtido |
| Média | $\mu = \frac{\sum_{i=1}^n x_i}{n}$ | $2,544 \times 10^{-5}$ |
| Desvio padrão | $\sigma = \sqrt{\left(\frac{\sum_{i=1}^n (x_i - \mu)^2}{n}\right)}$ | $7,717 \times 10^{-2}$ |
| Curtose | $kur = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^4}{\sigma^4}$ | 5,204 |

Tabela 6.4: Medidas estatísticas sobre o dado F3R. Exibidas a média, o desvio padrão e a curtose. Valores arredondados para 3 casas decimais. Amplitudes normalizadas par o intervalo $[-1, 1]$.

Com o dicionário gerado a partir da etapa anterior de treinamento, na qual foram utilizados as amostras apontadas pelos representantes dos supertraços via SLICT, foi feita sua expansão. Cada coluna deste dicionário inicial (átomo inicial) passou pelo processo de deformação e atenuação via filtro da terra múltiplas vezes, cada uma com um valor de parâmetro distinto, seja o fator de qualidade, seja a frequência de referência. Tal dicionário expandido foi utilizado na fase de recodificação ou representação esparsa do dado de

amplitudes sísmicas original. A partir do dicionário original, fez-se a ampliação para 5 vezes e para 10 vezes. As Tabelas 6.5 e 6.6 mostram a correspondência do múltiplo do dicionário e os valores dos parâmetros utilizados em cada ampliação. Em cada caso, o dicionário original é mantido como o índice de múltiplo 0 (zero), perfazendo, ao final, 5 ou 10 múltiplos.

| Índice da Ampliação do Dicionário | Fator de Qualidade Q | Frequência de Referência (Hz) |
|-----------------------------------|----------------------|-------------------------------|
| 1 | 100 | 35 |
| 2 | 100 | 30 |
| 3 | 80 | 35 |
| 4 | 80 | 30 |

Tabela 6.5: Valores utilizados pelo procedimento de distorção de ondaletas via filtro da terra para cada múltiplo do dicionário original. Ampliação para **5** vezes o tamanho original.

| Índice da Ampliação do Dicionário | Fator de Qualidade Q | Frequência de Referência (Hz) |
|-----------------------------------|----------------------|-------------------------------|
| 1 | 100 | 35 |
| 2 | 100 | 32,5 |
| 3 | 100 | 30 |
| 4 | 90 | 35 |
| 5 | 90 | 32,5 |
| 6 | 90 | 30 |
| 7 | 80 | 35 |
| 8 | 80 | 32,5 |
| 9 | 80 | 30 |

Tabela 6.6: Valores utilizados pelo procedimento de distorção de ondaletas via filtro da terra para cada múltiplo do dicionário original. Ampliação para **10** vezes o tamanho original.

A Tabela 6.7 lista um resultado numérico levando em conta as variáveis mencionadas.

A partir da Tabela 6.7, pode-se afirmar que houve ganho de qualidade da reconstrução ao aumentar a quantidade de múltiplos do dicionário inicial usando as distorções geradas pela aplicação do filtro da terra. Este ganho percebido está de acordo com o que Elad informa⁶: a redundância traz a possibilidade de enriquecer o modelo, permitindo mais combinações para construção dos sinais e soluções mais esparsas. Percebe-se ganho na reconstrução ao se aumentar a quantidade de amostras de entrada, sendo a exceção a última

⁶Curso com título *Sparse Representations in Signal and Image Processing: Fundamentals*, disponível via aplicativo **edX**.

| Avaliação da reconstrução | | | | Parâmetro de regularidade | | | | | |
|---------------------------|-------|--------|------|-----------------------------|--------|---------------|--------|--------|---------------|
| | | | | 0,125 | | | 2,0 | | |
| | | | | Tamanho final do dicionário | | | | | |
| | | | | 1x | 5x | 10x | 1x | 5x | 10x |
| Amostras de entrada | 2000 | Medida | RSR | 18,764 | 22,782 | 22,978 | 18,814 | 22,828 | 23,022 |
| | | | PRSR | 37,757 | 39,666 | 39,763 | 37,682 | 39,688 | 39,786 |
| | 4000 | | RSR | 19,232 | 22,982 | 23,186 | 19,274 | 22,017 | 23,227 |
| | | | PRSR | 37,89 | 39,766 | 39,868 | 37,912 | 39,784 | 39,888 |
| | 8000 | | RSR | 19,584 | 23,186 | 23,411 | 19,532 | 23,131 | 23,358 |
| | | | PRSR | 38,067 | 39,868 | 39,98 | 38,041 | 39,841 | 39,954 |
| | 16000 | | RSR | 19,518 | 22,749 | 22,953 | 19,566 | 22,634 | 22,822 |
| | | | PRSR | 38,034 | 39,65 | 39,751 | 38,058 | 39,593 | 39,686 |

Tabela 6.7: Resultado numérico para avaliação da reconstrução medindo razão sinal-ruído e pico da razão sinal-ruído. As colunas com rótulo **1x** se referem ao uso do dicionário treinado e ajustado. Colunas **5x** e **10x** se referem ao uso dos dicionários aumentados para 5 e 10 vezes usando deformações provenientes da aplicação do filtro da terra com parâmetros informados nas Tabelas 6.5 e 6.6, respectivamente.

quantidade de 16000⁷. Em Dumitrescu e Irofti (2018) (pg. 79, seção 3.11.1) é relatado um experimento em que a raiz quadrada do erro médio quadrático, medido a partir da reconstrução de sinais de entrada, tem uma tendência de decréscimo com as iterações do algoritmo de treinamento de dicionário, o que pode sugerir que, ao limitar as iterações em uma quantidade baixa, estamos limitado o poder de representação de um dicionário. Ao mesmo tempo, não podemos concluir, dada a amostra experimental, qual valor do parâmetro de regularidade de supertraço pode gerar melhor qualidade de reconstrução do dado de entrada⁸.

6.2.2

Compressibilidade e Análise de Resíduos

Os processos de treinamento de dicionário e de representação esparsa baseada num dicionário podem ser entendidos como processos de aproximação e, por isso, é comum a geração de resíduo ao transformar um dado em sua aproximação via representação esparsa.

Sejam n_i , n_c e n_t , relativamente ao dado de amplitude sísmica disponível, respectivamente, o número de amostras na direção *inline*, o número de amostras na direção *crossline* e o número de amostras na direção vertical (tempo

⁷Para a realização deste experimento, optou-se por limitar a quantidade de iterações, o que pode ter impedido ganho de qualidade ao usar a maior quantidade de amostras de entrada.

⁸Cada experimento de reconstrução requer uma etapa de treinamento de dicionário e outra etapa de recodificação esparsa, cada uma levando cerca de 3 a 15 horas em um computador equipado com processador Intel Core i7 7500 @ 2.7GHz e 16Gb de memória.

ou profundidade). Tomando por base os experimentos realizados, nos quais forçamos a quantidade de não zeros para, no máximo, $1/5$ do número de amostras num traço, podemos estimar o espaço ocupado pela representação esparsa. Usando o tipo de dado original de amplitude sísmica (ponto flutuante de 4 bytes) nos códigos da representação esparsa, temos:

- Número de amostras = $n_i * n_c * n_t$
- Espaço ocupado por um dicionário = $n_t^2 * 4$ bytes * número de múltiplos, ocupando cerca de até 2% do dado original
- Espaço ocupado pelos códigos: $1/5$ das amostras * 4 bytes
- Espaço ocupado pelas posições dos códigos: $1/5$ das amostras * 2 bytes
- **Total aproximado:** $1/5$ das amostras * 6 bytes + 2% das amostras * 4 bytes = 1,28 bytes por amostra
- Ocupação de espaço original: 4 bytes por amostra
- Taxa de compressão (com perda) (Salomon, 2007): $1,28 / 4 = 0,32$ (32%)

Não foi aqui considerada qualquer compressão posterior ou qualquer otimização de representação compacta, o que poderia diminuir ainda mais o espaço efetivamente ocupado.

Neste ponto, é possível fazer uma avaliação da qualidade da representação esparsa observando o resíduo como sendo a diferença entre o dado original e o dado reconstruído após a representação esparsa. As Tabelas 6.8, 6.9, 6.10 e 6.11 trazem informação sobre os resíduos gerados pela reconstrução usando os vários dicionários treinados. Não são exibidas as médias dos resíduos pois, em todos os experimentos, seus valores absolutos foram inferiores a 10^{-4} , e não parecem fundamentais a uma avaliação. Também estão omitidos os valores obtidos de assimetria, pois tiveram valores absolutos inferiores a 10^{-3} .

| Número de amostras | 2000 | | | | | |
|--|---------|--------|--------------|--------|---------|---------------|
| | 0,125 | | | 2,0 | | |
| Parâm. de regularidade dos supertraços | | | | | | |
| Tamanho do dicionário | 1x | 5x | 10x | 1x | 5x | 10x |
| Desvio padrão | 0,0262 | 0,0208 | 0,0206 | 0,0261 | 0,0207 | 0,0205 |
| Curtose | 0,054 | 0,097 | 0,105 | 0,052 | 0,094 | 0,102 |
| Mínimo | -0,1636 | -0,131 | -0,1225 | -0,154 | -0,1366 | -0,1436 |
| Máximo | 0,1772 | 0,127 | 0,1285 | 0,1698 | 0,1235 | 0,1204 |
| Amplitude | 0,3408 | 0,258 | 0,251 | 0,3238 | 0,26011 | 0,264 |

Tabela 6.8: Avaliação de resíduos quando da utilização de 2000 traços representantes. Em destaque o menor desvio padrão e a menor amplitude.

Observando os valores de assimetria e curtose das amostras de resíduo, pode-se assumir que estamos diante de uma distribuição simétrica e muito

| Número de amostras | 4000 | | | | | |
|--|---------|---------------|---------|---------|---------|---------------|
| Parâm. de regularidade dos supertraços | 0,125 | | | 2,0 | | |
| Tamanho do dicionário | 1x | 5x | 10x | 1x | 5x | 10x |
| Desvio padrão | 0,0255 | 0,0206 | 0,0203 | 0,0254 | 0,0205 | 0,0202 |
| Curtose | 0,065 | 0,122 | 0,132 | 0,064 | 0,115 | 0,126 |
| Mínimo | -0,1654 | -0,1259 | -0,1395 | -0,1758 | -0,1258 | -0,1343 |
| Máximo | 0,1562 | 0,1257 | 0,1204 | 0,163 | 0,1321 | 0,1231 |
| Amplitude | 0,3216 | 0,2516 | 0,2598 | 0,3387 | 0,2579 | 0,2574 |

Tabela 6.9: Avaliação de resíduos quando da utilização de 4000 traços representantes. Em destaque o menor desvio padrão e a menor amplitude.

| Número de amostras | 8000 | | | | | |
|--|---------|---------|---------------|---------|---------|---------|
| Parâm. de regularidade dos supertraços | 0,125 | | | 2,0 | | |
| Tamanho do dicionário | 1x | 5x | 10x | 1x | 5x | 10x |
| Desvio padrão | 0,025 | 0,0203 | 0,0201 | 0,0251 | 0,0204 | 0,0202 |
| Curtose | 0,087 | 0,145 | 0,157 | 0,081 | 0,149 | 0,162 |
| Mínimo | -0,1625 | -0,1274 | -0,1259 | -0,1603 | -0,1279 | -0,1304 |
| Máximo | 0,1667 | 0,1352 | 0,1278 | 0,1717 | 0,1309 | 0,1251 |
| Amplitude | 0,3292 | 0,2626 | 0,2537 | 0,332 | 0,2588 | 0,2555 |

Tabela 6.10: Avaliação de resíduos quando da utilização de 8000 traços representantes. Em destaque o menor desvio padrão e a menor amplitude.

| Número de amostras | 16000 | | | | | |
|--|---------|---------------|---------------|---------|---------|---------|
| Parâm. de regularidade dos supertraços | 0,125 | | | 2,0 | | |
| Tamanho do dicionário | 1x | 5x | 10x | 1x | 5x | 10x |
| Desvio padrão | 0,0251 | 0,0208 | 0,0206 | 0,025 | 0,021 | 0,0207 |
| Curtose | 0,092 | 0,166 | 0,182 | 0,081 | 0,159 | 0,173 |
| Mínimo | -0,1621 | -0,134 | -0,1315 | -0,1554 | -0,1368 | -0,1366 |
| Máximo | 0,1619 | 0,1262 | 0,129 | 0,1598 | 0,1276 | 0,1266 |
| Amplitude | 0,324 | 0,2602 | 0,2605 | 0,3152 | 0,2643 | 0,2632 |

Tabela 6.11: Avaliação de resíduos quando da utilização de 16000 traços representantes. Em destaque o menor desvio padrão e a menor amplitude.

próxima a uma distribuição gaussiana. Em Hosking (1992) é exibida uma tabela em que constam valores de curtose para distribuições simétricas e, com base nela, podemos aproximar uma função de densidade de probabilidade gaussiana para descrever o comportamento do resíduo em cada exemplo.

A título de ilustração e de reforço desta última observação, a Figura 6.8 mostra um histograma normalizado de resíduo e uma curva normal usando a média e desvio padrão do mesmo resíduo a partir do maior valor de curtose calculado (16000 exemplos, parâmetro de regularização dos supertraços 0,125 e aumento de dicionário para 10 vezes).

Assim sendo, há a possibilidade de se realizar uma compressão com

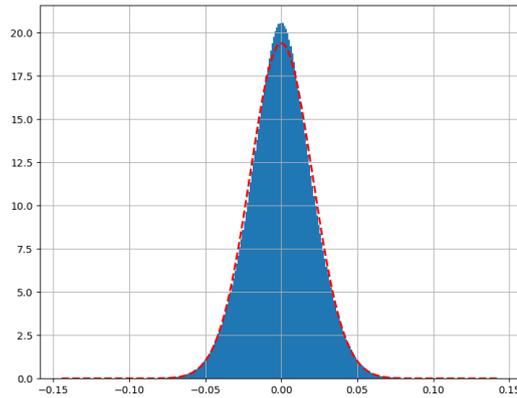


Figura 6.8: Sobreposição de curva normal e de histograma extraídos a partir do resíduo gerado pela representação esparsa usando o dicionário aumentado em 10 vezes, parâmetro de regularização dos supertraços 0,125 e 16000 traços de exemplos, correspondendo ao valor mais distante de zero observado nos experimentos.

perda controlada aproveitando parcialmente o conteúdo do resíduo. Podemos aplicar uma quantização de Lloyd-Max, que é ótima em termos do erro médio quadrático sobre o conteúdo do resíduo (Max, 1960; Gonzales, 2007; Sayood, 2012). Tal quantização tem a característica de usar uma tabela de códigos de tamanho fixo (em bits), associando cada código binário a um valor representante de um intervalo. A partir da aproximação a uma gaussiana, torna-se mais simples o cálculo dos intervalos e dos valores representantes dos mesmos.

6.2.3 Uma Análise Não Numérica

Gostaríamos de avaliar, também, se a metodologia aqui mostrada poderia ser usada como um auxiliar na extração de feições importantes do dado. É importante salientar que não se fez uso de qualquer dado derivado de processos de perfuração de poços, como perfis de poços.

Tomemos, então, o dado recodificado, ou seja, o resultante da codificação esparsa do dado original quando se utiliza um dicionário treinado. Ao selecionar somente o elemento de máxima amplitude, o pico de refletividade de norma máxima, é possível observar trechos de superfícies que se assemelham a horizontes, como mostra a Figura 6.9. Tal resultado pode ser usado, por exemplo, para controle de qualidade do próprio processo de inversão, uma vez que pode haver zonas do dado original de amplitude sísmica com valores absolutos grandes e, com eles, pode ser possível verificar a validade do processo de treinamento de dicionário e de recodificação.

Observe-se que a Figura 6.9 acaba mostrando não somente trechos de

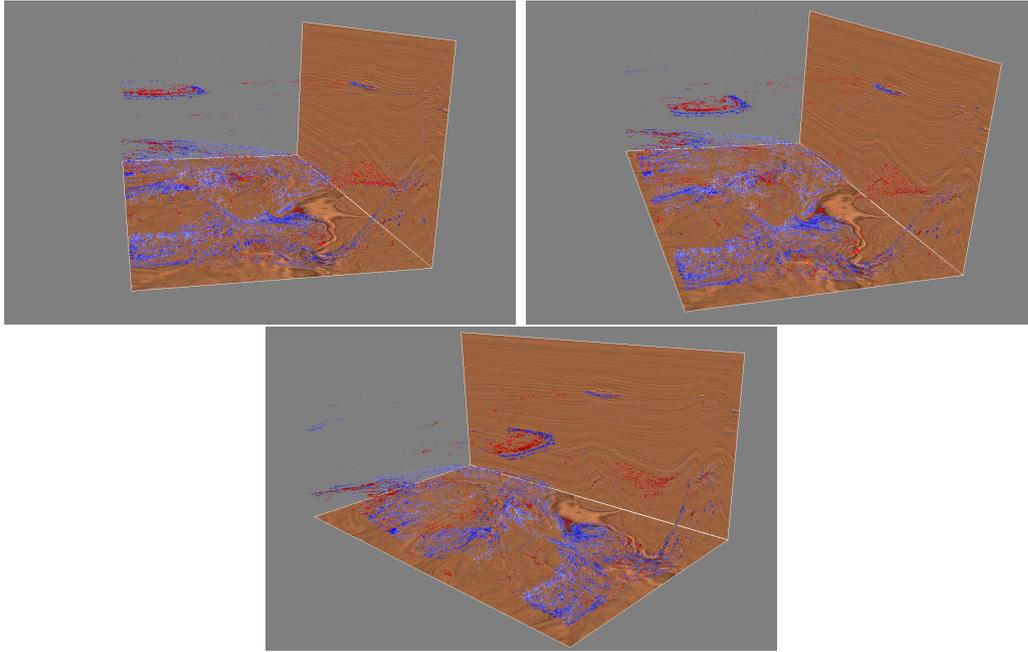


Figura 6.9: Exibição de pontos, em três vistas diferentes, que correspondem, em cada traço, ao coeficiente de maior valor absoluto no dado recodificado. Utilizado dicionário de 8000 amostras, parâmetro de regularidade de supertraços 0.125, aumentado em 10 vezes usando filtro da terra. As cores correspondem ao sinal de refletividade (**positivo**, **negativo**).

horizontes, mas também o sinal do coeficiente de refletividade, que serve como indicativo de variação da densidade do meio.

6.2.4 Contribuição Efetiva do Trecho Aumentado dos Dicionários

Cabe uma análise da real contribuição da adição de múltiplas cópias ajustadas do dicionário inicialmente calculado/treinado. Como tal processo de aumento é artificial, ou seja, não passa pelo processo convencional de treinamento de dicionário, seria importante observar se uma posterior representação esparsa faz uso efetivo do trecho aumentado do dicionário.

As Figuras 6.10, 6.11, 6.12 e 6.13 a seguir demonstram que a parte aumentada do dicionário é usada e, em alguns trechos, é mais usada do que o trecho original do dicionário. O trecho original de cada dicionário está com legenda “0”, e demais trechos da parte aumentada tem legenda com valores de 1 a 4, no caso de um dicionário aumentado para 5 vezes, e de 1 a 9, no caso de um dicionário aumentado para 10 vezes.

Percebe-se que alguns dos trechos aumentados do dicionário (aqueles que têm legenda com valor acima de zero) foram pouco usados, o que sugere que se pode ou se deve mudar a parametrização do filtro da terra para a geração do trecho aumentado do dicionário.

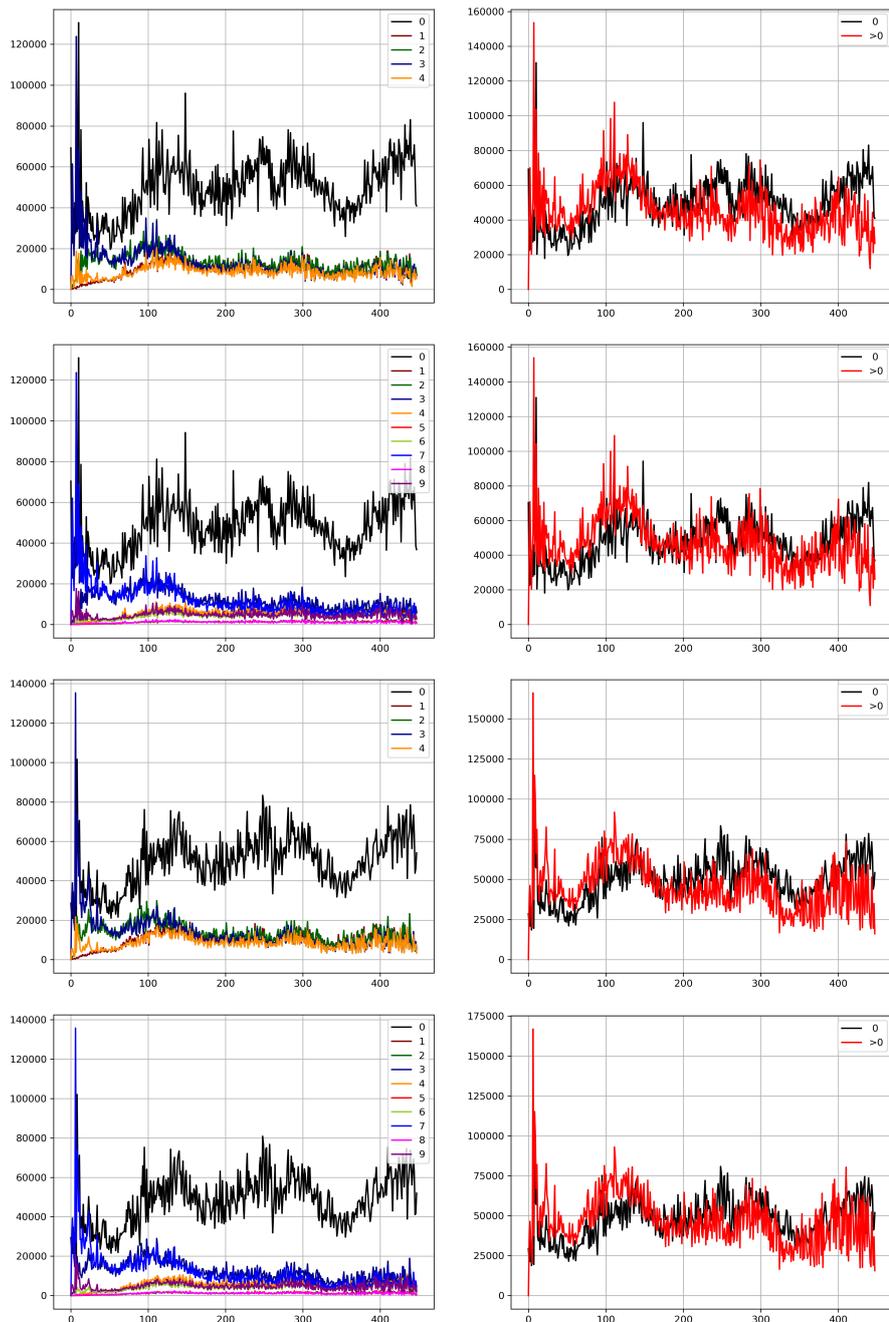


Figura 6.10: Ilustração do uso dos coeficientes do dicionário ao longo da profundidade usando **2000** traços de entrada para o treinamento. Eixo horizontal: profundidade da amostra. Eixo vertical: contagem dos coeficientes utilizados em cada profundidade. À esquerda, uso do dicionário treinado e aumentado, distinguindo cada trecho do dicionário utilizado. À direita, comparação de uso entre o trecho original (com legenda “0”) e trecho aumentado do dicionário (com legenda “>0”). Duas primeiras linhas: parâmetro de regularidade de supertraços 0,125. Duas últimas linhas: parâmetro de regularidade 2,0. Primeira e terceira linhas: dicionário aumentado para 5 vezes. Segunda e quarta linhas: dicionário aumentado em 10 vezes.

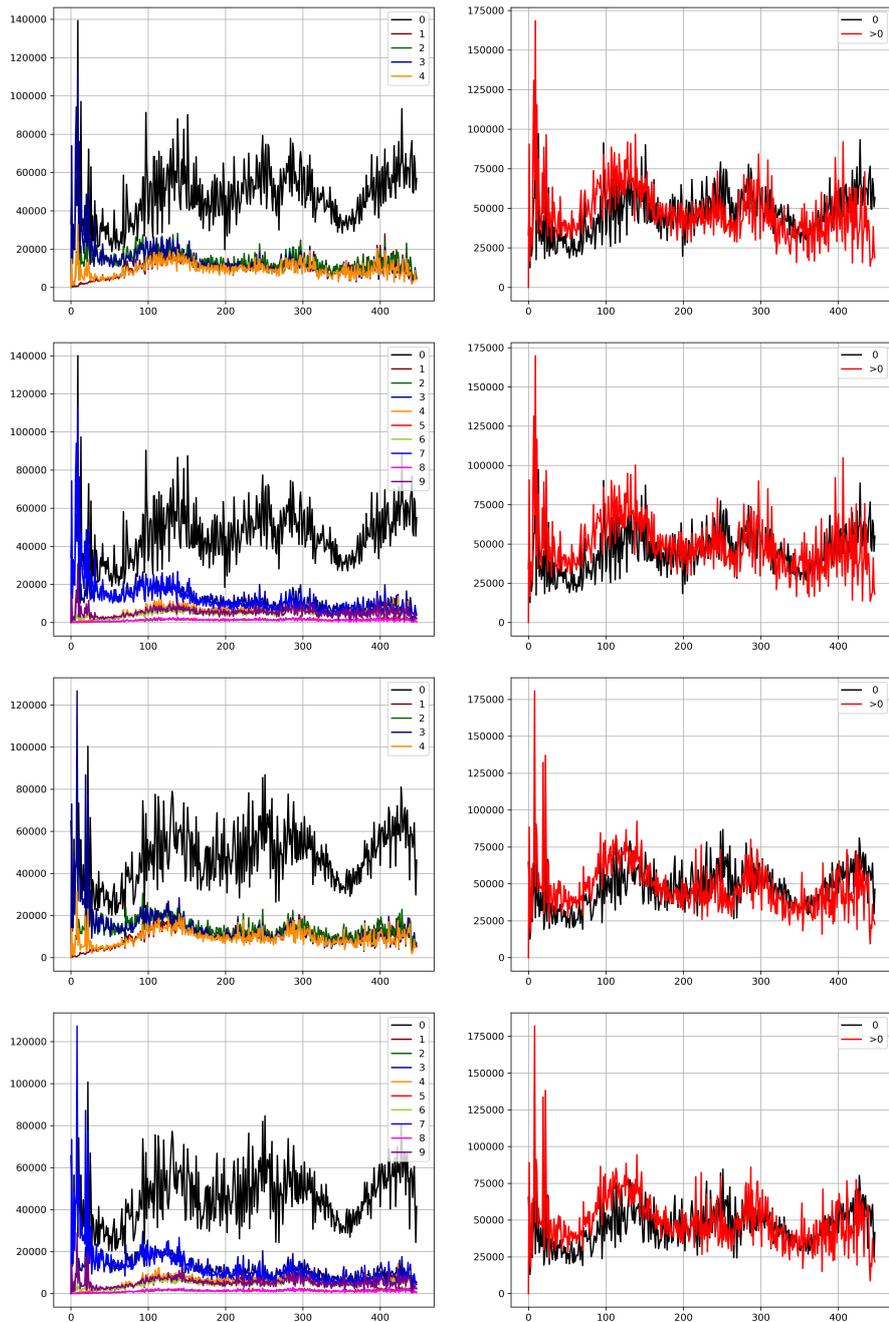


Figura 6.11: Ilustração do uso dos coeficientes do dicionário ao longo da profundidade usando **4000** traços de entrada para o treinamento. Eixo horizontal: profundidade da amostra. Eixo vertical: contagem dos coeficientes utilizados em cada profundidade. À esquerda, uso do dicionário treinado e aumentado, distinguindo cada trecho do dicionário utilizado. À direita, comparação de uso entre o trecho original (com legenda “0”) e trecho aumentado do dicionário (com legenda “>0”). Duas primeiras linhas: parâmetro de regularidade de supertraços 0,125. Duas últimas linhas: parâmetro de regularidade 2,0. Primeira e terceira linhas: dicionário aumentado para 5 vezes. Segunda e quarta linhas: dicionário aumentado em 10 vezes.

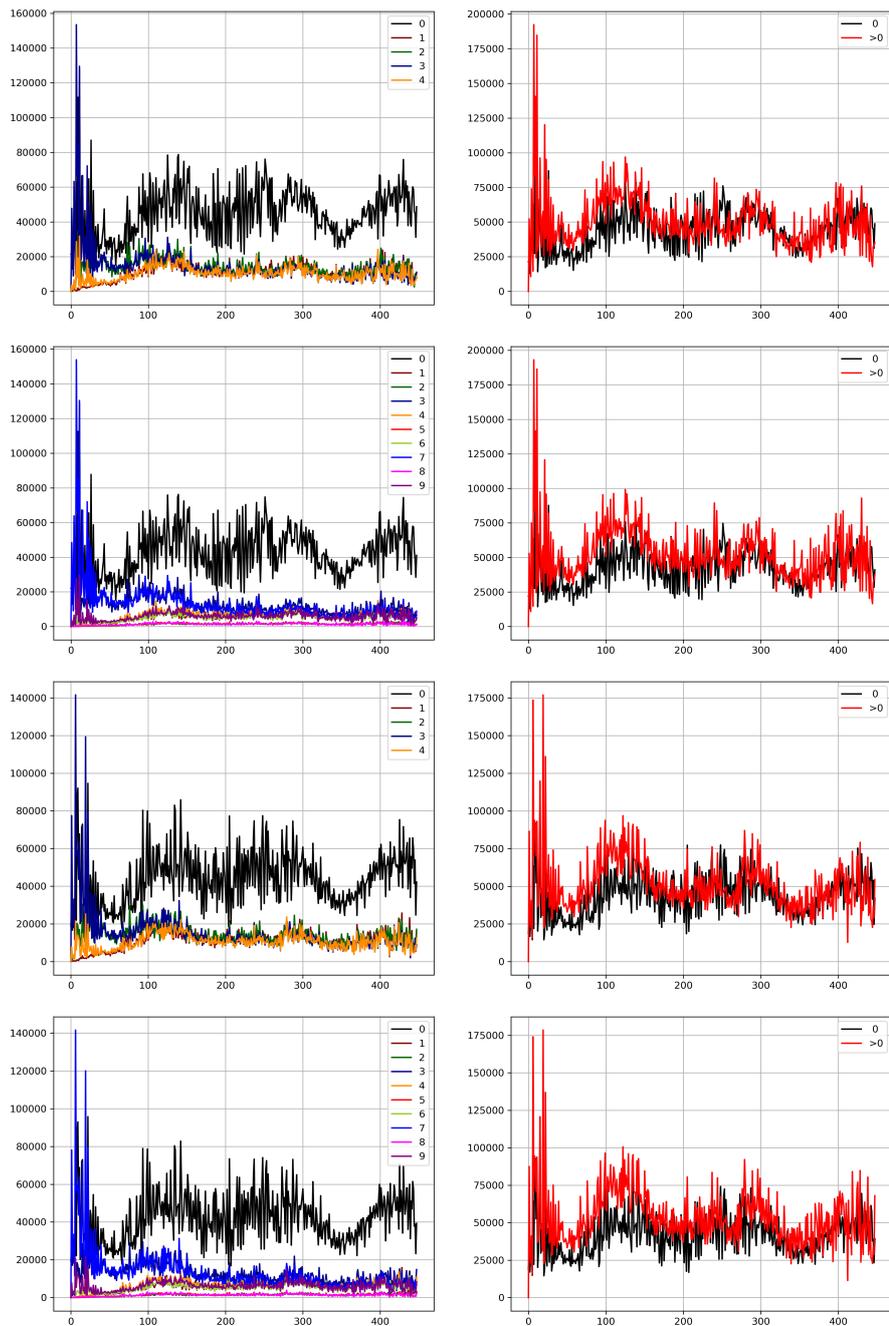


Figura 6.12: Ilustração do uso dos coeficientes do dicionário ao longo da profundidade usando **8000** traços de entrada para o treinamento. Eixo horizontal: profundidade da amostra. Eixo vertical: contagem dos coeficientes utilizados em cada profundidade. À esquerda, uso do dicionário treinado e aumentado, distinguindo cada trecho do dicionário utilizado. À direita, comparação de uso entre o trecho original (com legenda “0”) e trecho aumentado do dicionário (com legenda “>0”). Duas primeiras linhas: parâmetro de regularidade de supertraços 0,125. Duas últimas linhas: parâmetro de regularidade 2,0. Primeira e terceira linhas: dicionário aumentado para 5 vezes. Segunda e quarta linhas: dicionário aumentado em 10 vezes.

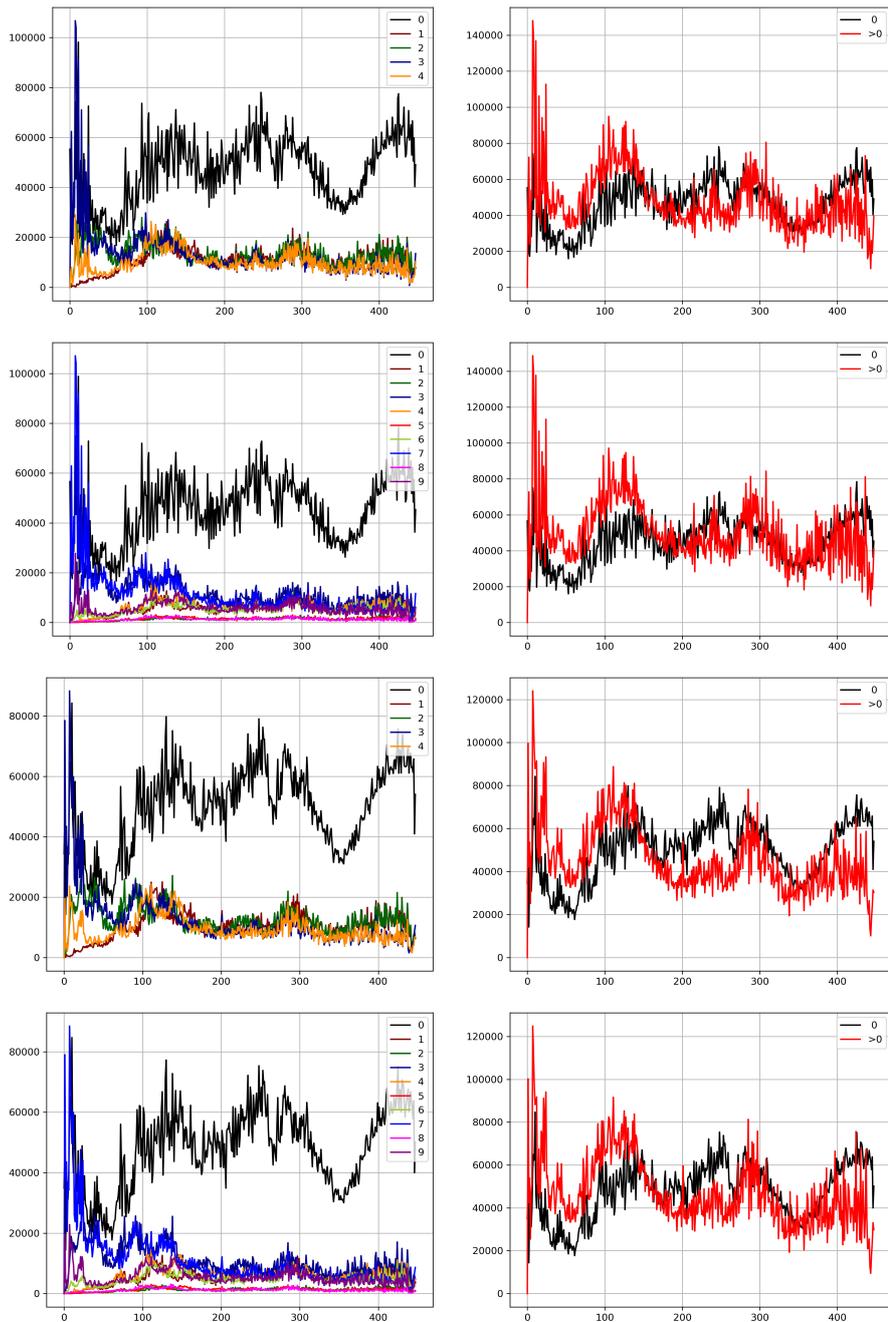


Figura 6.13: Ilustração do uso dos coeficientes do dicionário ao longo da profundidade usando **16000** traços de entrada para o treinamento. Eixo horizontal: profundidade da amostra. Eixo vertical: contagem dos coeficientes utilizados em cada profundidade. À esquerda, uso do dicionário treinado e aumentado, distinguindo cada trecho do dicionário utilizado. À direita, comparação de uso entre o trecho original (com legenda “0”) e trecho aumentado do dicionário (com legenda “>0”). Duas primeiras linhas: parâmetro de regularidade de supertraços 0,125. Duas últimas linhas: parâmetro de regularidade 2,0. Primeira e terceira linhas: dicionário aumentado para 5 vezes. Segunda e quarta linhas: dicionário aumentado em 10 vezes.

Na tentativa de reconstruir o dado original de amplitudes, o algoritmo de representação esparsa acaba por obter uma combinação linear esparsa de diversas ondaletas, da parte do dicionário original ou da parte aumentada pela aplicação de filtro da terra e, portanto, há o efetivo uso de mais de uma ondaleta para melhor representar o dado original. O conjunto das posições (os índices das colunas do dicionário, a menos da aplicação de uma operação de módulo – resto da divisão) e dos valores (multiplicadores das colunas do dicionário) corresponde, no modelo aqui estudado, a um conteúdo hipotético de refletividades.

A Figuras 6.14, 6.15, 6.16 e 6.17 ilustram resultados de inversão de refletividade numa *inline* considerando o dicionário original, a versão aumentada para 5 vezes e a versão aumentada para 10 vezes. Embora a ampliação do dicionário tenha trazido um melhor resultado numérico de reconstrução do dado original, a partir destas figuras, percebe-se que, aparentemente, os eventos de refletividade estão melhor delineados ao nos utilizarmos do dicionário original sem ampliação, ou seja, sem o aumento artificial via aplicações do filtro da terra. Percebe-se, também, que o algoritmo de representação esparsa utilizado durante o treinamento e repetido durante esta fase de recodificação do dado original precisa ser mudado porque se observa que, nos eventos mais profundos, mais à esquerda das imagens, há alternância entre valores de pico e não pico, o que sugere que pode-se chegar a um melhor resultado de refletividade se usarmos uma versão convencional de um algoritmo de representação esparsa como, por exemplo, OMP ou FISTA.

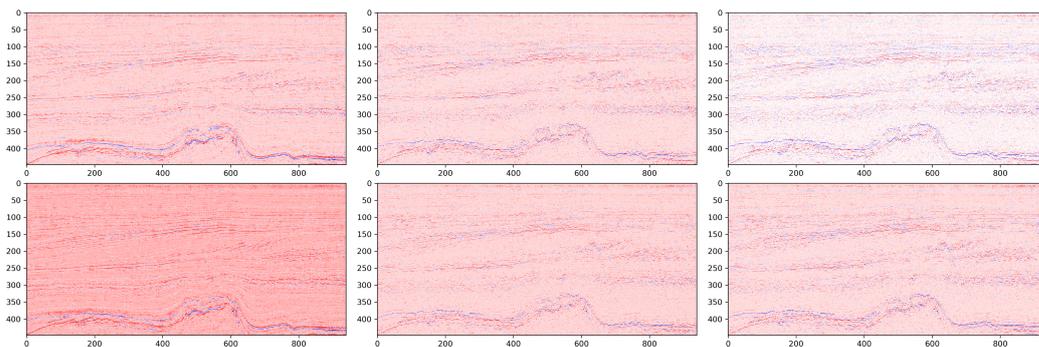


Figura 6.14: Ilustração da inversão da *inline* 284 do dado F3R ao usar dicionários de múltiplos tamanhos treinado usando **2000** amostras. À esquerda, utilizado dicionário treinado e ajustado. Ao centro, utilizado dicionário aumentado para 5 vezes. À direita, utilizado dicionário aumentado para 10 vezes. Primeira linha: parâmetro de regularidade de supertraços 0,125. Segunda linha: parâmetro de regularidade de supertraços 2,0.

O capítulo seguinte traz algumas conclusões e cita um conjunto de oportunidades que podem servir para trabalhos futuros.

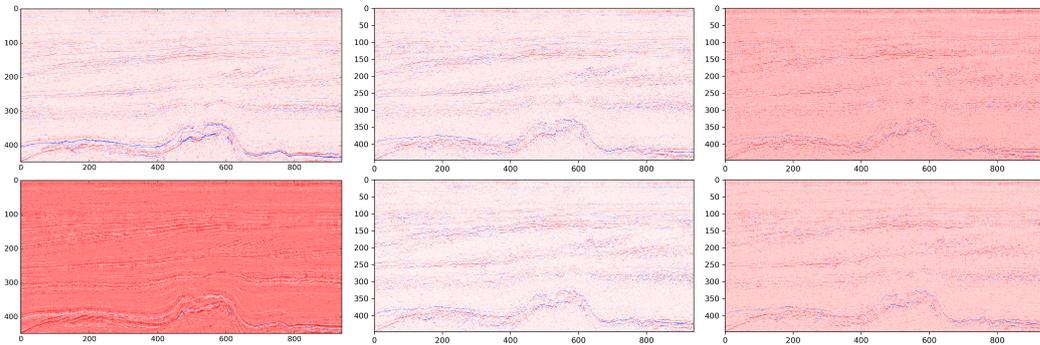


Figura 6.15: Ilustração da inversão da *inline* 284 do dado F3R ao usar dicionários de múltiplos tamanhos treinado usando **4000** amostras. À esquerda, utilizado dicionário treinado e ajustado. Ao centro, utilizado dicionário aumentado para 5 vezes. À direita, utilizado dicionário aumentado para 10 vezes. Primeira linha: parâmetro de regularidade de supertraços 0,125. Segunda linha: parâmetro de regularidade de supertraços 2,0.

PUC-Rio - Certificação Digital N° 1421617/CA

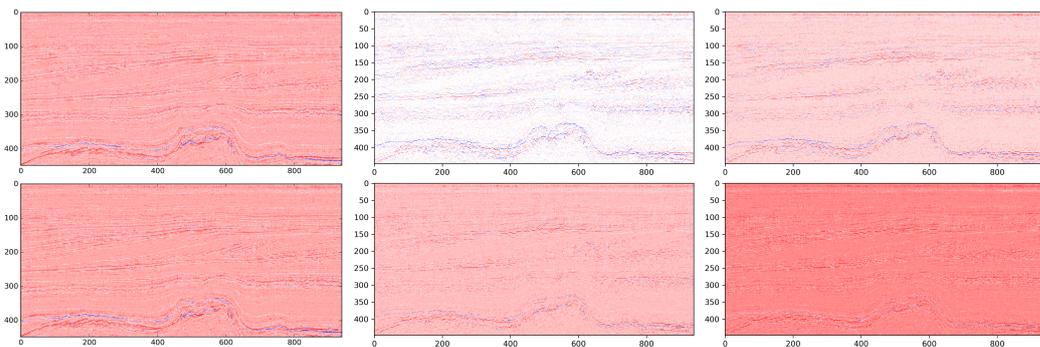


Figura 6.16: Ilustração da inversão da *inline* 284 do dado F3R ao usar dicionários de múltiplos tamanhos treinado usando **8000** amostras. À esquerda, utilizado dicionário treinado e ajustado. Ao centro, utilizado dicionário aumentado para 5 vezes. À direita, utilizado dicionário aumentado para 10 vezes. Primeira linha: parâmetro de regularidade de supertraços 0,125. Segunda linha: parâmetro de regularidade de supertraços 2,0.

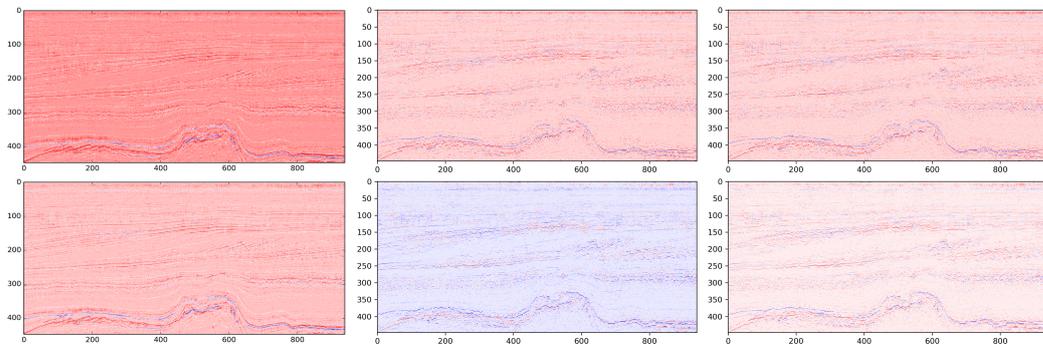


Figura 6.17: Ilustração da inversão da *inline* 284 do dado F3R ao usar dicionários de múltiplos tamanhos treinado usando **16000** amostras. À esquerda, utilizado dicionário treinado e ajustado. Ao centro, utilizado dicionário aumentado para 5 vezes. À direita, utilizado dicionário aumentado para 10 vezes. Primeira linha: parâmetro de regularidade de supertraços 0,125. Segunda linha: parâmetro de regularidade de supertraços 2,0.

7

Conclusões e Trabalhos Futuros

7.1

Conclusões

De modo amplo, neste trabalho há o propósito de, usando modelagem esparsa, realizar a separação do conteúdo de amplitudes sísmicas em duas partes principais: uma parte ondulatória, estacionária ou não; e outra parte formada por picos esparsos de refletividades. Ao realizar esta tese, pode-se chegar a algumas conclusões, listadas a seguir.

Lobbes. O método desenvolvido para se alcançar uma representação esparsa, que corresponde a um conjunto de picos de refletividade extraídos de um dado de amplitude sísmica com base num dicionário, derivado do método em Wang et al. (2016), provou-se útil numericamente. Os experimentos mostrados revelaram que o dicionário gerado com o auxílio deste algoritmo como arcabouço de recuperação esparsa tem a propriedade de gerar uma representação esparsa de qualidade superior em relação aos outros métodos empregados.

SGK com regularização. O método aqui detalhado e algebricamente derivado a partir do método SGK (Sahoo e Makur, 2013; Dumitrescu e Irofti, 2018) foi capaz de gerar átomos do dicionário inicial com a característica de banda limitada. O método parece simples de utilizar, uma vez que se pode gerar uma única matriz de penalização, tomada e definida coluna a coluna, seguindo a regra de penalizar elementos longe da diagonal.

Filtro da terra. O aumento do dicionário inicial usando a aplicação de distorções via filtro da terra se mostrou útil, uma vez que se percebeu, através dos experimentos realizados, que os novos átomos gerados foram efetivamente usados e, em alguns trechos, mais usados do que os átomos do dicionário original. Isso pode indicar que uma única ondaleta determinada a priori poderia não ser capaz de bem codificar o dado completamente.

SLICT - Supetraços. A extensão implementada do algoritmo SLIC para atender o caso de traços de amplitude sísmica mostra-se interessante para criar grupos de traços próximos com características distintas de sua vizinhança, além de servir de arcabouço para subamostragem do próprio dado.

Aproximação via representação esparsa. Alguns argumentos corroboram a observação de que a busca de uma representação esparsa é uma aproximação:

- O treinamento do dicionário é um processo de descoberta de um modelo numérico que depende da quantidade e da representatividade das amostras de entrada;
- A quantidade de átomos num dicionário pode interferir na quantidade de informação nele retida: quanto maior a quantidade, mais informação retida;
- O método de treinamento de dicionário pode interferir no seu conteúdo resultante;
- Diferentes métodos de representação esparsa embutidos no procedimento de treinamento de dicionário podem selecionar átomos distintos em proporções distintas para codificar um sinal de entrada;
- A quantidade de iterações utilizada na fase de treinamento de dicionário pode interferir na qualidade do dado codificado quando se utiliza tal dicionário;
- A quantidade final de coeficientes utilizados para representar um dado de entrada altera a qualidade da representação final do dado original.

7.2

Trabalhos Futuros

Dentre diversas possibilidades em aberto, não abordadas nesta tese, mas que podem merecer atenção a posteriori, levando em conta custo e viabilidade, podemos citar:

Formas dos átomos. Em Fomel (2007) faz-se uso de uma regularização de forma de curvas. Parece uma oportunidade fazer uso desta técnica em conjunto com o algoritmo SGK com regularização para a geração de grupos de átomos com múltiplos formatos, significando variados espectros de frequência. A largura do dicionário final gerado para gerar a representação esparsa original também interfere na qualidade do dado reconstruído. A pergunta que fica é até que ponto é importante aumentar esta largura ou qual seria uma largura mínima para que se tenha uma representação esparsa cuja reconstrução posterior atenda requisitos de qualidade de reconstrução (em dB) estabelecidos previamente.

Particionamento de traços. Em Beckouche e Ma (2014), os autores parecem ter deixado a oportunidade de aplicar o mesmo método para segmentar o dado sísmico em camadas e, a partir de tais camadas, seria possível

treinar mais de um dicionário, um para cada camada, na intenção de que tais dicionários pudessem melhor codificar o dado de amplitude. A forma dos átomos derivada deste tipo de particionamento deveria levar em conta o propósito. Se for para comprimir, blocos 3D podem ser interessantes. Se for para ressaltar características como os picos de refletividade, trechos de traços podem ser úteis.

Um fator de qualidade Q ótimo. Em Sui e Ma (2019) é feita a proposta do fator de qualidade Q que parametriza a geração de uma matriz que modela dispersão e atenuação. Ao aplicar tal matriz em uma outra com estrutura de Toeplitz, espera-se melhor recodificar um conteúdo de amplitudes sísmicas. É uma oportunidade fazer uso do esquema de otimização. Já que é capaz de extrair um único valor de fator de qualidade Q , poderia se avaliar o poder de representação de um dicionário gerado diretamente contra as abordagens aqui expostas. As semelhanças e diferenças entre o trabalho em Sui e Ma (2019) e a tese aqui apresentada reforçam a importância de ambos, além de ressaltar a oportunidade no maior aprofundamento em ambos.

Pesquisa exploratória do uso dos supertraços. A própria medida de dissimilaridade entre traços, mostrada nas imagens da Figura 4.4, pode ajudar, por exemplo, na determinação da densidade de amostras no dado de amplitudes. O custo computacional envolvido na métrica de dissimilaridade usada (distância euclidiana) é baixa. Ao utilizar uma outra métrica mais cara como, por exemplo, alguma que use correlação ou casamento de padrões, é possível que o resultado do agrupamento mude e, ao mesmo tempo, forneça grupos cuja similaridade interna seja mais alta, uma vez que se possa considerar um deslocamento de índices entre dois traços sendo comparados como feito, por exemplo, em Manay et al. (2006) ou em Fernandes (2009).

Coerência Mútua. Pode-se avaliar a **coerência mútua** dos diversos dicionários gerados (Rish e Grabarnik, 2015; Hastie, Tibshirani e Wainwright, 2015; Elad, 2010; Foucart e Rauhut, 2013). A coerência mútua é uma importante propriedade teórica usada em amostragem comprimida que indica a qualidade de um dicionário e permite avaliar a unicidade de soluções esparsas. A medida é tomada como sendo o produto escalar máximo entre distintas colunas de um dicionário, após ajustar cada coluna para que tenham norma euclidiana igual a 1. Matrizes ortogonais têm coerência mútua igual a zero e dicionários sobrecompletos têm, necessariamente, coerência mútua estritamente positiva.

Inicialização do dicionário antes do treinamento. Como foi mencionado no Capítulo 5, inicializamos o dicionário como uma matriz identidade antes de realizar seu treinamento. Experiências com inicializações de nature-

zas distintas, incluindo ondaletas, submetidas às deformações do filtro da terra ou não, podem gerar, eventualmente, resultados de picos de refletividade mais bem definidos, já que o próprio processo de treinamento de dicionário não impõe qualquer filtro ao seu conteúdo, e pode acabar incorporando ruído ou informação de refletores.

Referências bibliográficas

- ACHANTA, R.; SHAJI, A.; SMITH, K.; LUCCHI, A.; FUA, P. ; SÜSSTRUNK, S.. **Slic superpixels compared to state-of-the-art superpixel methods.** IEEE transactions on pattern analysis and machine intelligence, 34(11):2274–2282, 2012.
- AGARWAL, P. K.; HAR-PELED, S. ; VARADARAJAN, K. R.. **Geometric approximation via coresets.** Combinatorial and computational geometry, 52:1–30, 2005.
- AHARON, M.; ELAD, M. ; BRUCKSTEIN, A.. **K-svd: An algorithm for designing overcomplete dictionaries for sparse representation.** IEEE Transactions on signal processing, 54(11):4311–4322, 2006.
- AKIYAMA, K.; ALBERDI, A.; ALEF, W.; ASADA, K.; AZULAY, R.; BACZKO, A.-K.; BALL, D.; BALOKOVIĆ, M.; BARRETT, J.; BINTLEY, D. ; OTHERS. **First m87 event horizon telescope results. iv. imaging the central supermassive black hole.** The Astrophysical Journal Letters, 875(1):L4, 2019.
- AMINZADEH, F.; DASGUPTA, S. N.. **Fundamentals of petroleum geophysics.** Developments in Petroleum Science, 60:37–92, 2013. Geophysics for Petroleum Engineers.
- BECK, A.; TEOULLE, M.. **A fast iterative shrinkage-thresholding algorithm for linear inverse problems.** SIAM journal on imaging sciences, 2(1):183–202, 2009.
- BECKOUCHE, S.; MA, J.. **Simultaneous dictionary learning and denoising for seismic data.** Geophysics, 79(3):A27–A31, 2014.
- BOSSMANN, F.; MA, J.. **Asymmetric chirplet transform for sparse representation of seismic data.** Geophysics, 80(6):WD89–WD100, 2015.
- BURTSCHER, M.; RATANAWORABHAN, P.. **Fpc: A high-speed compressor for double-precision floating-point data.** IEEE Transactions on Computers, 58(1):18–31, 2009.

- CHEN, C.; HUANG, J.. **Exploiting the wavelet structure in compressed sensing mri**. *Magnetic resonance imaging*, 32(10):1377–1389, 2014.
- CHENG, H.. **Sparse representation, modeling and learning in visual recognition**. Springer, 2016.
- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L. ; STEIN, C.. **Introduction to Algorithms**. MIT press Cambridge, 2 ed., 2001.
- DUMITRESCU, B.; IROFTI, P.. **Dictionary Learning Algorithms and Applications**. Springer International Publishing AG, 2018.
- ELAD, M.. **Sparse and Redundant Representations. From Theory to Applications in Signal and Image Processing**. Springer-Verlag New York, 1 ed., 2010.
- ELAD, M.. **Sparse and redundant representation modeling — what next?** *IEEE Signal Processing Letters*, 19(12):922–928, 2012.
- ENGAN, K.; AASE, S. O. ; HUSOY, J. H.. **Method of optimal directions for frame design**. In: 1999 IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING. PROCEEDINGS. ICASSP99 (CAT. NO. 99CH36258), vol. 5, p. 2443–2446. IEEE, 1999.
- FERNANDES, R. C.. **Registro de sísmica 3d a dados de poços**. Master's thesis, PUC-Rio, 2009.
- FERNANDES, R.; LOPES, H. ; GATTASS, M.. **Lobbes: An algorithm for sparse-spike deconvolution**. *IEEE Geoscience and Remote Sensing Letters*, 14(12):2240–2244, 2017.
- FIGUEIREDO, A. M.. **Mapeamento de Eventos Sísmicos baseado em Algoritmos de Agrupamento de Dados**. PhD thesis, Pontifícia Universidade Católica do Rio de Janeiro, 2015.
- FOMEL, S.. **Shaping regularization in geophysical-estimation problems**. *Geophysics*, 72(2):R29–R36, 2007.
- FOUCART, S.; RAUHUT, H.. **A Mathematical Introduction to Compressive Sensing**. Birkhäuser Basel (Springer), 2013.
- FOUT, N.; MA, K.-L.. **An adaptive prediction-based approach to lossless compression of floating-point volume data**. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2295–2304, 2012.

- GOBBETTI, E.; GUITIÁN, J. A. I. ; MARTON, F.. **Covra: A compression-domain output-sensitive volume rendering architecture based on a sparse representation of voxel blocks**. In: COMPUTER GRAPHICS FORUM, vol. 31, p. 1315–1324. Wiley Online Library, 2012.
- GOLUB, G. H.; VAN LOAN, C. F.. **Matrix computations**. Johns Hopkins University Press, 4 ed., 2013.
- GONZALES, R.; WOODS, R.. **Digital Image Processing**. Pearson, 3 ed., 2007.
- HAN, J.; PEI, J. ; KAMBER, M.. **Data Mining: Concepts and Techniques**. Elsevier, 3 ed., 2012.
- HASTIE, T.; TIBSHIRANI, R. ; WAINWRIGHT, M.. **Statistical learning with sparsity: the lasso and generalizations**. CRC press, 2015.
- HERRMANN, F. J.. **Randomized sampling and sparsity: Getting more information from fewer samples**. *Geophysics*, 75(6):WB173–WB187, 2010.
- HINO, H.. **Introduction to Sparse Modeling**, 2014. Disponível em: http://www.cs.tsukuba.ac.jp/~hinohide/MIRU_Sparse.pdf. Data de acesso: 09/07/2017.
- HOEFLING, H.. **A path algorithm for the fused lasso signal approximator**. *Journal of Computational and Graphical Statistics*, 19(4):984–1006, 2010.
- HONMA, M.; AKIYAMA, K.; UEMURA, M. ; IKEDA, S.. **Super-resolution imaging with radio interferometry using sparse modeling**. *Publications of the Astronomical Society of Japan*, 66(5):95, 2014.
- HOSKING, J. R.. **Moments or l moments? an example comparing two measures of distributional shape**. *The American Statistician*, 46(3):186–189, 1992.
- IZMAILOV, A.; SOLODOV, M.. **Otimização volume 2**. Rio de Janeiro: IMPA, 1 ed., 2007.
- KLEINBERG, J.; TARDOS, E.. **Algorithm Design**. Pearson Education, Inc., 2006.

- LI, H.; TUO, X.; SHEN, T.; HENDERSON, M. J.; COURTOIS, J. ; YAN, M.. **An improved lossless group compression algorithm for seismic data in seg-y and miniseed file formats.** *Computers & Geosciences*, 100:41–45, 2017.
- LINDSTROM, P.; ISENBURG, M.. **Fast and efficient compression of floating-point data.** *IEEE transactions on visualization and computer graphics*, 12(5):1245–1250, 2006.
- LIU, G.; FOMEL, S. ; CHEN, X.. **Time-frequency analysis of seismic data using local attributes.** *Geophysics*, 76(6):P23–P34, 2011.
- LIU, J.; YUAN, L. ; YE, J.. **An efficient algorithm for a class of fused lasso problems.** In: PROCEEDINGS OF THE 16TH ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, p. 323–332. ACM, 2010.
- LUSTIG, M.; DONOHO, D. ; PAULY, J. M.. **Sparse mri: The application of compressed sensing for rapid mr imaging.** *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.
- MAIRAL, J.; BACH, F.; PONCE, J. ; SAPIRO, G.. **Online dictionary learning for sparse coding.** In: PROCEEDINGS OF THE 26TH ANNUAL INTERNATIONAL CONFERENCE ON MACHINE LEARNING, p. 689–696. ACM, 2009.
- MAIRAL, J.; BACH, F.; PONCE, J. ; SAPIRO, G.. **Online dictionary learning for sparse coding.** In: PROCEEDINGS OF THE 26TH ANNUAL INTERNATIONAL CONFERENCE ON MACHINE LEARNING, p. 689–696. ACM, 2009.
- MANAY, S.; CREMERS, D.; HONG, B.-W.; YEZZI, A. J. ; SOATTO, S.. **Integral invariants for shape matching.** *IEEE Transactions on pattern analysis and machine intelligence*, 28(10):1602–1618, 2006.
- MARTIN, G. S.; WILEY, R. ; MARFURT, K. J.. **Marmousi2: An elastic upgrade for marmousi.** *The Leading Edge*, 25(2):156–166, 2006.
- MAX, J.. **Quantizing for minimum distortion.** *IRE Transactions on Information Theory*, 6(1):7–12, 1960.
- NATARAJAN, B. K.. **Sparse approximate solutions to linear systems.** *SIAM journal on computing*, 24(2):227–234, 1995.

- OLIVEIRA, A. S.. **Compensation for q-losses revisited-a more stable approach using svd.** In: 9TH INTERNATIONAL CONGRESS OF THE BRAZILIAN GEOPHYSICAL SOCIETY, 2005.
- ONAJITE, E.. **Seismic Data Analysis Techniques in Hydrocarbon Exploration.** Elsevier, 2014.
- PEREIRA-DIAS, B.; BULCÃO, A.; SOARES FILHO, D. M.; SANTOS, L. A.; DIAS, R. D. M.; LOUREIRO, F. P. ; DUARTE, F. D. S.. **Least-squares migration in the image domain with sparsity constraints: An approach for super-resolution in depth imaging.** In: 15º CONGRESSO DA SOCIEDADE BRASILEIRA DE GEOFÍSICA, Rio de Janeiro, Brasil, 2017. Sociedade Brasileira de Geofísica.
- PETERSEN, K. B.; PEDERSEN, M. S.. **The matrix cookbook**, nov 2012. Version 20121115.
- RISH, I.; GRABARNIK, G. Y.. **Sparse Modeling.** CRC Press, 2015.
- RODRÍGUEZ, M. B.; GOBBETTI, E.; IGLESIAS GUITIÁN, J.; MAKHINYA, M.; MARTON, F.; PAJAROLA, R. ; SUTER, S. K.. **State-of-the-art in compressed gpu-based direct volume rendering.** In: COMPUTER GRAPHICS FORUM, vol. 33, p. 77–100. Wiley Online Library, 2014.
- SAHOO, S. K.; MAKUR, A.. **Dictionary training for sparse representation as generalization of k-means clustering.** IEEE Signal Processing Letters, 20(6):587–590, 2013.
- SALOMON, D.. **Data Compression: The Complete Reference.** Springer Science & Business Media, 4 ed., 2007.
- SALVADOR, S.; CHAN, P.. **Toward accurate dynamic time warping in linear time and space.** Intelligent Data Analysis, 11(5):561–580, 2007.
- SAMMUT, C.; WEBB, G. I.. **Encyclopedia of Machine Learning.** Springer Science & Business Media, 2011.
- SAYOOD, K.. **Introduction to data compression.** Morgan Kaufmann, 4 ed., 2012.
- SUI, Y.; MA, J.. **A nonstationary sparse spike deconvolution with anelastic attenuation.** Geophysics, 84(2):R221–R234, 2019.
- THEODORIDIS, S.; KOUTROUMBAS, K.. **Pattern Recognition, Fourth Edition.** Academic Press, Inc., Orlando, FL, USA, 4th ed., 2008.

- TIAN, X.; LI, S.. **Seismic signal compression using nonparametric bayesian dictionary learning via clustering**. *Algorithms*, 10(2):65, 2017.
- TKALCIC, M.; TASIC, J. F.. **Colour spaces: perceptual, historical and applicational background**. In: Zajc, B.; Tkalcic, M., editors, *THE IEEE REGION 8 EUROCON – COMPUTER AS A TOOL*, vol. A.
- VAN DEN BERGH, M.; BOIX, X.; ROIG, G. ; VAN GOOL, L.. **Seeds: Super-pixels extracted via energy-driven sampling**. *International Journal of Computer Vision*, 111(3):298–314, 2015.
- VELIS, D. R.. **Stochastic sparse-spike deconvolution**. *Geophysics*, 73(1):R1–R9, 2008.
- VERA RODRIGUEZ, I.; SACCHI, M. ; GU, Y. J.. **A compressive sensing framework for seismic source parameter estimation**. *Geophysical Journal International*, 191(3):1226–1236, 2012.
- VILLARREAL, O. P.; LEÓN, K.; ESPINOSA, D.; AGUDELO, W. ; ARGUELLO, H.. **Compressive sensing seismic acquisition by using regular sampling in an orthogonal grid**. In: 2017 IEEE 7TH INTERNATIONAL WORKSHOP ON COMPUTATIONAL ADVANCES IN MULTI-SENSOR ADAPTIVE PROCESSING (CAMSAP), p. 1–5. IEEE, 2017.
- WANG, L.; ZHAO, Q.; GAO, J.; XU, Z.; FEHLER, M. ; JIANG, X.. **Seismic sparse-spike deconvolution via toeplitz-sparse matrix factorization**. *Geophysics*, 81(3):V169–V182, 2016.
- WAZLAWICK, R.. **Metodologia de pesquisa para ciência da computação**. Elsevier Brasil, 2009.
- YILMAZ, O.. **Seismic Data Processing**, vol. 2. Society of Exploration Geophysicists, 1987.
- ZHANG, R.. **Time-variant wavelet extraction with spectral decomposition for seismic inversion**. In: SEG TECHNICAL PROGRAM EXPANDED ABSTRACTS 2016, p. 3751–3756. Society of Exploration Geophysicists, 2016.
- ZHOU, Y.; CHEN, W. ; GAO, J.. **Separation of seismic blended data by sparse inversion over dictionary learning**. *Journal of Applied Geophysics*, 106:146–153, 2014.
- ZHOU, Y.; CHEN, W. ; GAO, J.. **Separation of seismic blended data by sparse inversion over dictionary learning**. *Journal of Applied Geophysics*, 106:146–153, 2014.

ZHOU, Y.; CHEN, W.; GAO, J. ; PASCAL, F.. **Seismic deblending by sparse inversion over dictionary learning**. In: SEG TECHNICAL PROGRAM EXPANDED ABSTRACTS 2013, p. 273–278. Society of Exploration Geophysicists, 2013.

ZHU, L.; LIU, E. ; MCCLELLAN, J. H.. **Seismic data denoising through multiscale and sparsity-promoting dictionary learning**. *Geophysics*, 80(6):WD45–WD57, 2015.

A

Lobbès

Eis uma versão em Português do resumo do trabalho anexado a seguir.

Lobbès: Um Algoritmo para Deconvolução por Picos Esparsos.

Este documento propõe um algoritmo para resolver o problema de deconvolução por picos esparsos, chamado Lobbès (Seleção de parâmetro por busca binária baseado no Lasso). Ele melhora o algoritmo iterativo de achatamento e limiarização para fatoração esparsa de uma matriz de Toeplitz realizando três passos para encontrar um parâmetro de regularização adequado: 1) um procedimento de normalização sobre o dado de entrada; um passo de busca binária baseado no operador Lasso; e 3) a eliminação picos consecutivos similar a uma supressão de não máximos. Tal parâmetro nos permite encontrar uma solução com a esparsidade especificada. Comparamos nossos resultados contra o algoritmo original e com o conhecido método guloso indutor de soluções esparsas Orthogonal Matching Pursuit. Em relação ao estado da arte, os resultados demonstram que o Lobbès gera melhores resultados: melhor razão sinal-ruído do sinal reconstruído e melhor resultado para picos de refletividade. Também derivamos uma nova forma de medir a qualidade da deconvolução.

Lobbes: An Algorithm for Sparse-Spike Deconvolution

Rodrigo Fernandes, Hélio Lopes¹, and Marcelo Gattass

Abstract—This letter proposes an algorithm for solving the sparse-spike deconvolution problem, named *Lobbes* (Lasso-based binary search for parameter selection). It improves the fast iterative shrinkage and threshold algorithm for Toeplitz-sparse matrix factorization by performing three steps to find a suitable regularization parameter: 1) a normalization procedure over the input data; 2) a binary search step based on the least absolute shrinkage and selection operator; and 3) the elimination of consecutive peaks similar to non-maximum suppression. Such parameter allows us to find a solution with a specified sparsity. We compare our results against the original algorithm and with the known sparse-inducing greedy approach of orthogonal matching pursuit. Relative to state-of-the-art, results demonstrate that *Lobbes* generates better results: better signal-to-noise ratio of the reconstructed signal and better result for reflectivity peaks. We also derive a new way to measure the quality of the deconvolution.

Index Terms—Deconvolution, matching pursuit algorithms, sparse matrices, statistical learning.

I. INTRODUCTION

THE problem of sparse-spike deconvolution (SSD) can be viewed as an inverse problem in which one wants to determine: 1) the number; 2) the position; 3) the magnitude; and 4) the phase of sparse reflectivity spikes that, when convolved with an estimated seismic wavelet, can accurately approximate the original noisy seismic signal of different amplitudes. The main objective of SSD methods is to increase the bandwidth content from band-limited seismic amplitude observations [1], enabling an enhancement in the quality of a seismic image, and finally facilitating interpretability. SSD is a possible strategy to undo the fusion (or deblending, or unmixing) between a wavelet and a reflectivity profile. In SSD, we assume that the subsurface structure is a set of planar layers, and the seismic amplitude signal can be modeled and summarized by $s_t = w_t * r_t + n_t$, $t = 1, 2, \dots, m$, where w_t is the seismic wavelet, r_t is the reflectivity profile, n_t is an additive noise, and there are m different traces from which we want to extract the respective reflectivity profile. Usually, some constraints are imposed to the SSD problem in order to shrink the solution space. As an example of constraint, we could

cite the number of reflectivity spikes per seismic trace. The better the placement of reflectivity peaks, the better we could describe and understand the variations of subsurface geology. As said by Velis in [1]: “This is especially important because the search for subtle hydrocarbons traps has become a major task in today’s seismic exploration.”

On the one hand, the computation of reflectivity peaks by SSD could be interpreted as a special case of sparse coding. In this case, the dictionary of atoms could have the special structure of a Toeplitz matrix, which embodies a wavelet and serves as a convolution operator. On the other hand, the estimation of a candidate wavelet, starting from a fixed reflectivity information, could be viewed as a particular case of dictionary learning, in which a single atom is computed when disassembling the Toeplitz matrix.

In order to clarify what we mean by *sparse*, a vector \mathbf{v} is called s -sparse if at most s entries of \mathbf{v} are nonzero. Methods based on sparsity and sparsity-inducing strategies, such as least absolute shrinkage and selection operator (Lasso) [2] and orthogonal matching pursuit (OMP) [3], have been receiving a lot of attention. Lasso have been used, e.g., to fill the gaps of satellite images [4]; to select bands of hyperspectral images for target detection [5]; to separate hyperspectral images into its constituent materials and its abundancies [6]; to resolve the inverse problem of seismic sparse-layer reflectivity discovery [7]; and to achieve a sparse representation of data to enhance classification [8]. OMP have been used, e.g., to separate blended seismic images [9] and to remove noise from seismic images in a dictionary learning heuristic [10].

This letter proposes an algorithm, named *Lobbes* (Lasso-based binary search for parameter selection), for solving the SSD problem, which improves the fast iterative shrinkage and threshold algorithm (FISTA) for Toeplitz-sparse for matrix factorization (TSMF) algorithm presented in [11]. Our algorithm makes this improvement by performing a normalization procedure of the input data, followed by a binary search and a postprocessing peak elimination and accumulation to find a suitable regularization parameter. Such parameter allows us to find a solution with a specified sparsity. We compare our results against the ones generated by: 1) the original algorithm [11]; 2) three modifications of the original algorithm; and 3) the known sparse-inducing greedy approach of OMP. We demonstrate that *Lobbes* gives better reflectivity profiles when observing the peaks, good signal-to-noise ratio (SNR) of the reconstructed signal, and superior measurement of quality of the deconvolution.

Section II explains briefly the Lasso approach as an option for selecting the important features/dimensions of candidate

Manuscript received April 7, 2017; revised August 24, 2017; accepted September 27, 2017. Date of publication October 16, 2017; date of current version December 4, 2017. This work was supported in part by Petróleo Brasileiro S.A. (Petrobras), in part by PUC-Rio, in part by CAPES, and in part by CNPq. (Corresponding author: Hélio Lopes.)

The authors are with the Department of Informatics, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro 22430-060, Brazil (e-mail: rcfernandes@inf.puc-rio.br; helio@inf.puc-rio.br; mgattass@tecgraf.puc-rio.br).

Color versions of one or more of the figures in this letter are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/LGRS.2017.2758899

1545-598X © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

results of reflectivity profile. Section III describes the FISTA for TSMF (F TSMF) algorithm and our main contribution, the Lobbes algorithm. Section V shows the results of a set of experiments comparing the mentioned approaches, and explains the measurements we took. Finally, Section VI brings some concluding remarks.

II. LASSO ALGORITHM

A very simple formulation of an inverse problem can be written as a linear system of equations $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A} \in \mathbb{R}^{m \times n}$ is the “design” or measurements matrix, meaning that we have m signals, each one of size n ; $\mathbf{b} \in \mathbb{R}^{m \times 1}$ is the vector containing observations and $\mathbf{x} \in \mathbb{R}^{n \times 1}$ is the vector representing coefficient matrix. The solution vector \mathbf{x} encodes the linear combination of columns of \mathbf{A} that results in \mathbf{b} . In a noiseless setting, without constraints on the coefficients matrix \mathbf{x} , and \mathbf{A} having few columns, i.e., low-dimensional examples, one could use the normal equation to find the solution according to: $\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$. By considering noise, one could impose constraints over the coefficients. Regression with penalization or regularization, e.g., the Tikhonov regularization, is a possible choice to alleviate excessive fitting to train examples and to achieve a better generalization performance [12]. The Tikhonov regression does not guarantee the sparse solution desired, even though it is a possible choice when solving SSD [1], [13].

Starting from the assumption that a limited number of reflectivity spikes are to be considered, the usage of a sparse-luving approach, e.g., the Lasso [2], seems essential. Equation (1) describes the Lasso’s optimization problem

$$\min_{\mathbf{x}} \|\mathbf{b} - \mathbf{Ax}\|_2^2 + \lambda \|\mathbf{x}\|_1 \quad (1)$$

where λ is the regularization parameter responsible for the linkage of values of \mathbf{x} toward zero and $\|\mathbf{x}\|_p$ is the ℓ_p norm $\mathbf{x} \in \mathbb{R}^n$: $\|\mathbf{x}\|_p = (\sum_{i=1}^n |x_i|^p)^{1/p}$.

When dealing with multidimensional data, the range on each coordinate could influence the convergence of the first-order-based optimization algorithms such as the gradient descent or the coordinate descent. Normalization strategies could be employed to force, or at least approximate, the same range on each coordinate of the data. An interesting procedure is described in [2], which suggests a particular way to normalize each column in the input data, and enables the computation of the λ_{\max} , the maximum value of the regularization parameter beyond which any coefficient of the \mathbf{x} vector in (1) is set to zero.

Notice that the first term of (1) corresponds to the ordinary least squares (OLS) error, and the second part is the Lagrangian formulation that includes a constraint over the solution vector \mathbf{x} . The first term forces the solution \mathbf{x} to be as close as possible to the observed data, and the second one controls the complexity and the generalization capacity of the model. The Lasso can be considered a framework for ℓ_1 regularization, and there are variations and implementations available [14]. Coordinate descent, least angle regression, and proximal methods are known strategies employed in its solution.

We could think that the problem modeled in (1) is a relaxed version of the problem of finding the best subset of coefficients, modeled as follows:

$$\min_{\mathbf{x}} \|\mathbf{b} - \mathbf{Ax}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{x}\|_0 \leq K \quad (2)$$

where K is the desired sparsity, i.e., the number of nonzero coefficients in the solution vector \mathbf{x} , a set of size n , and the $\|\mathbf{x}\|_0^0$ means the pseudo-norm ℓ_0 of \mathbf{x} : $\|\mathbf{x}\|_0^0 = \lim_{p \rightarrow 0} \|\mathbf{x}\|_p^p = \sum_{i=1}^n 1, x_i \neq 0$.

This problem is in the class of NP-Hard problems, preventing us of applying such an approach to the problem of SSD, in which a signal or trace could have some hundreds to thousands samples.

Note that in (1) and (2), the selected (nonzero) coefficients of the solution do not necessarily have the best value minimizing the OLS error. After selection, we could apply OLS only to these coefficients to achieve the best fit to the data.

In a recent work [15], authors employed Adaptive Lasso [16] to extract reflectivity. They also observed that Adaptive Lasso could perform worse than Lasso when the wavelet is poorly defined; and this is our case. We also argue that performing Adaptive Lasso to extract reflectivity could be a good choice if wavelet is smooth. Furthermore, choosing the value of the regularization parameter is still mandatory.

III. LOBBES ALGORITHM

In this section, we propose an improvement based on the work by Wang *et al.* [11]. They proposed a modification over a more sophisticated version of the FISTA algorithm [17], which is a solver for the Lasso formulation. They named this algorithm as FISTA for TSMF. Let K denotes the maximum number of reflectivity peaks for a trace, and let it be multiplied by a constant between 1.5 and 2. After an optimization step of a proximal method, they sort the absolute value of the coefficients of the partial solution \mathbf{x} . After sorting, half of the value that is in the $(K + 1)$ th position is used as the threshold/shrinkage/regularization parameter λ . Their algorithm eliminates several coefficients of \mathbf{x} in a few iterations, and its convergence was proven. The formulation in [11] divides the SSD in a two-step approach, similar to coordinate descent [2], [14]. The general formulation for the cost function is expressed as

$$J = \min_{\alpha, \mathbf{R}} \frac{1}{2} \|\mathbf{Y} - \mathbf{AR}\|_2^2 + \Psi(\alpha) + \lambda \sum_{j=1}^m \|\mathbf{r}_j\|_1. \quad (3)$$

In the quadratic error term of (3), \mathbf{Y} captures the seismic amplitude signals, one trace in each column; \mathbf{A} is a Toeplitz matrix that embodies the estimate wavelet; \mathbf{R} encodes the reflectivity profiles, one trace in each column. $\Psi(\alpha)$ is a penalty function depending exclusively on the wavelet, and is given by: $\Psi(\alpha) = \beta_1 \|\alpha\|_1 + \beta_f \sum_{k=1}^{w-1} \|a_{k+1} - a_k\| + \beta_2 \|\alpha\|_2$, where β_1 , β_f , and β_2 are the weights to the parcels corresponding to the ℓ_1 norm, the fusion cost [18] and the ℓ_2 norm, respectively. For now, for space reasons, we omit the details of this formulation.

Equation (3) is solved in an iterative procedure that is divided in two steps. The first step attempts to find a reflectivity profile, given an initial candidate wavelet. Keeping the wavelet fixed, (3) is simplified to

$$J_{\mathbf{R}} = \min_{\mathbf{R}} \frac{1}{2} \|\mathbf{Y} - \mathbf{A}\mathbf{R}\|_2^2 + \lambda \sum_{j=1}^m \|\mathbf{r}_j\|_1. \quad (4)$$

Equation (4) can be broken into pieces, one for each trace, decomposing it to $J_{\mathbf{r}_j} = \min_{\mathbf{r}_j} 1/2 \|\mathbf{y}_j - \mathbf{A}\mathbf{r}_j\|_2^2 + \lambda \|\mathbf{r}_j\|_1$, which can be solved by existing methods for the Lasso formulation [2], [14], [17]. The second step keeps the reflectivity matrix \mathbf{R} constant after the result obtained in the first step. This leads to other simplification of (3)

$$J_{\alpha} = \min_{\alpha} \frac{1}{2} \|\mathbf{Y} - \mathbf{A}\mathbf{R}\|_2^2 + \Psi(\alpha). \quad (5)$$

After some algebraic manipulation, (5) becomes $J_{\alpha} = \min_{\alpha} (1/2) \|\text{vec}(\mathbf{Y}) - \text{vec}(\mathbf{R})\alpha\|_2^2 + \Psi(\alpha)$, where the **vec** operator joins each column of a matrix into a single column vector. This problem could also be solved by available software for the fused Lasso [19].

In our algorithm, we change the way that the reflectivity profile is computed. Instead of solving (4) directly as done in [11], we first guess a good direct result for the reflectivity peaks. This change can be implemented by the use of an OMP algorithm, and this step only will be used to compare the algorithms in Section V. This choice seems adequate since we have a range of acceptable amount of peaks. Second, we apply Lobbes to solve (4) by searching for an adequate value of λ parameter corresponding to each value in the range. We encapsulate our implementation inside the outer algorithm described in [11]. We partially reproduced their result to serve as a baseline. From that point, we could argue that both the proposed approaches seem adequate to solve SSD.

When using OMP, we pass a desired sparsity K and the algorithm returns a coefficients vector \mathbf{r}_j with K nonzeros. Using a number much higher than the ground truth number peaks could lead to false peaks in \mathbf{r}_j , nevertheless the SNR¹ of the reconstructed amplitude could be higher. When using Lobbes, once given a desired sparsity, we iterate from zero to λ_{\max} each time we run Lasso, indirectly trying to achieve the required value of sparsity. Each resulting λ associated with the required sparsity serves as a warm restart parameter to the Lasso procedure in the next iteration. If a specific sparsity is not achievable, we consider the lowest sparsity higher than the required as a valid result.

Algorithm 1 shows in high level what steps we take to check many values of the regularization parameter λ and the respective solution. The simple and general purpose Algorithm 1 needs, at first, a normalization step making all columns of the design matrix to have zero mean and variance equals to one. This normalization is described in [2]. Additionally, this function makes observations to have zero mean and computes the maximum value of the regularization parameter λ_{\max}

¹Let \mathbf{Y} a set of n traces of seismic amplitude, each one with m samples, \mathbf{A} the Toeplitz matrix incorporating a candidate wavelet and \mathbf{R} the set of n traces of reflectivity, also with m samples. Then the SNR is $10 \log_{10} (\|\mathbf{Y}\|_2^2 / \|\mathbf{Y} - \mathbf{A}\mathbf{R}\|_2^2)$.

Algorithm 1 Lasso-Based Parameter Selection

```

1: function LOBBES( $\mathbf{A} \in \mathbb{R}^{m \times n}$ ,  $\mathbf{b} \in \mathbb{R}^m$ ,  $K$ ,  $p$ )
2:    $\mathbf{x} \in \mathbb{R}^n \leftarrow \mathbf{0}$ 
3:    $(\hat{\mathbf{A}}, \hat{\mathbf{b}}, \lambda_{\max}) \leftarrow \text{normalize}(\mathbf{A}, \mathbf{b})$ 
4:    $\delta_{\lambda} \leftarrow \lambda_{\max} / (p + 1)$ 
5:   for  $i \leftarrow (0..p)$  do
6:      $\lambda \leftarrow i * \delta$ 
7:     if  $i == 0$  then  $\hat{\mathbf{x}}_{\text{new}} \leftarrow (\hat{\mathbf{A}}^T \hat{\mathbf{A}} + \epsilon \mathbf{I})^{-1} \hat{\mathbf{A}}^T \hat{\mathbf{b}}$ 
8:     else  $\hat{\mathbf{x}}_{\text{new}} \leftarrow \text{spams.fistaFlat}(\hat{\mathbf{A}}, \hat{\mathbf{b}}, \mathbf{x}, \lambda)$ 
9:      $(b_0, \mathbf{x}_{\text{new}}) \leftarrow \text{denormalize}(\hat{\mathbf{x}}_{\text{new}})$ 
10:     $s \leftarrow \text{sparsityOf}(\mathbf{x}_{\text{new}})$ 
11:    if  $s == K$  then return  $\mathbf{x}_{\text{new}}$ 
12:    else if  $s > K$  then  $\lambda_{\text{upper}} \leftarrow \lambda$ ;  $\mathbf{x} \leftarrow \mathbf{x}_{\text{new}}$ 
13:    else  $\lambda_{\text{lower}} \leftarrow \lambda$ ; break for
14:    while  $s != K$  do
15:       $\lambda \leftarrow (\lambda_{\text{lower}} + \lambda_{\text{upper}}) / 2$ 
16:       $\hat{\mathbf{x}}_{\text{new}} \leftarrow \text{spams.fistaFlat}(\hat{\mathbf{A}}, \hat{\mathbf{b}}, \mathbf{x}, \lambda)$ 
17:       $(b_0, \mathbf{x}_{\text{new}}) \leftarrow \text{denormalize}(\hat{\mathbf{x}}_{\text{new}})$ 
18:       $\mathbf{x}_p \leftarrow \text{centerPeaks}(\mathbf{x}_{\text{new}})$ 
19:       $s \leftarrow \text{sparsityOf}(\mathbf{x}_p)$ 
20:      if  $s == K$  then return  $\mathbf{x}_p$ 
21:      else if  $s > K$  then  $\lambda_{\text{upper}} \leftarrow \lambda$ ;  $\mathbf{x} \leftarrow \mathbf{x}_p$ 
22:      else  $\lambda_{\text{lower}} \leftarrow \lambda$ 
23:    if so many iterations then return  $\mathbf{x}$ 

```

capable of flattening the solution of Lasso to zero. Lobbes subdivide the interval from 0 to λ_{\max} into p steps. The first step of the Lobbes' first loop is to find a solution using a ridge regression, where ϵ is a small number (we use 10^{-5}), and this solution corresponds to setting $\lambda = 0$. Given a desired sparsity K , the binary search procedure tries to find a value for λ to achieve that sparsity. The procedure called *centerPeaks* takes a reflectivity solution vector and, for each subset of three contiguous samples, does the following. If the middle element has the maximum absolute value, update its value as the sum of the three values and set its prior and next neighbors to zero. We call this step as elimination and accumulation of peaks. There is a chance that the solution path of parameters does not achieve the desired sparsity, and for this case we approximate the solution to a less sparse one, implemented in the final steps of the *while* loop at the end. We explicitly use the function *fistaFlat* available in the SParse Modeling Software (SPAMS) [20]. For each trace, we update at each iteration and record: 1) the last value of λ ; 2) the list of nonzero indices; and 3) the last reflectivity solution. The last value of λ is to be used in the next iteration for finding a reflectivity profile with the desired sparsity; if it is not, then we progressively enlarge an interval centered on the last λ in order to guarantee that the desired sparsity is achievable within it. The indexes of nonzero parameters serve as an indicator that we could use the last valid solution to restart the procedure of adjustment, that is done by using coordinate descent specifically using those nonzero values of the solution vector for each trace.

We could observe that Lobbes, when approximating to a local minimum, converges more slowly when compared

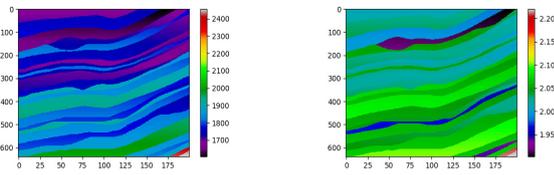


Fig. 1. Chosen slice of the original input data from Marmousi2 model. There are 200 traces of (Left) velocity and (Right) density, each one with 640 samples. Vertical scale is one sample per 1.25 ms, and we interpolate to produce 800 samples per trace.

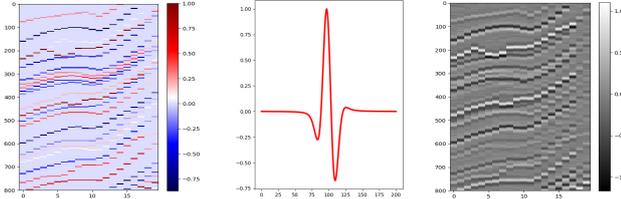


Fig. 2. Synthetic input data. (Left) Generated reflectivities, (Center) ground-truth wavelets, and (Right) generated amplitudes from 20 out of 200 traces of the region in Fig. 1. Additive white noise was added to amplitudes.

with the method by Wang *et al.* [11]. However, Lobbes was able to get better solution of (3). Because of that, we set some stopping criteria: 1) no more than 50 iterations would be performed; 2) no more than five iterations is performed after achieving a local minimum of (3); and 3) stop if the difference between the cost of consecutive iterations is below threshold (10^{-5} in our tests). Also, we record and return the best solution for each set of traces as the one that minimizes (3) after the iterations.

IV. DATA SET DESCRIPTION

We selected approximately the same region studied in [11] from the Marmousi2 model [21]. We used the Pysit library to handle velocity and density data [22]. Fig. 1 shows the subset of traces we applied the algorithms compared in this letter. The set of 200 traces was divided into 10 sets of 20 traces each. Each set of index $q \in [0, 9]$ was composed by traces with indices $10r + q$, $r \in [0, 19]$, creating sets of possibly closely correlated traces, generating a more difficult, and perhaps unusual, set of traces from which one could run blind deconvolution. From the 20 reflectivity traces of each set, we convolve with a ground-truth wavelet, and add some white Gaussian noise, producing a synthetic noisy amplitude data (Fig. 2). For the approaches compared, we used a zero-phase Ricker wavelet (duration of 201 ms, 201 samples) as a first (and wrong) guess. In the next section, we explain our experiments. Differently from [11], we used the density information to generate a harder scenario, since small peaks of impedance contrast appeared instead of larger ones.

V. RESULTS AND DISCUSSION

In order to check the validity of OMP and Lobbes compared with F TSMF for solving SSD, we applied the three approaches initially to a single trace of amplitude, and Fig. 3 shows the results.

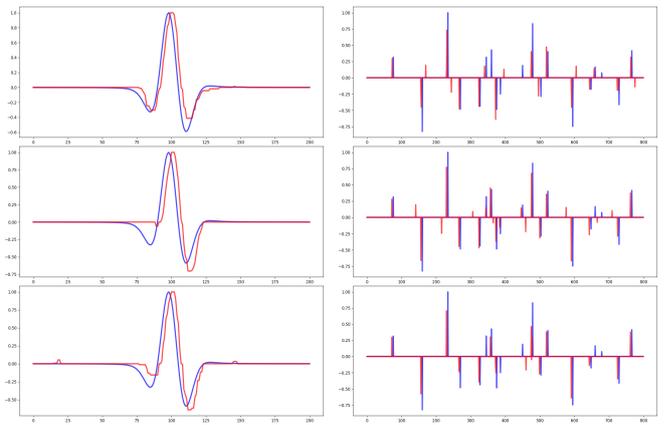


Fig. 3. (Left) Extracted wavelet (red) against the original (blue). (Right) Extracted reflectivity (red) against the original (blue). Task is to select 25 nonzero values in the reflectivity profile. (Top) Result of applying F TSMF (based on [11]). (Middle) Result of applying OMP. (Bottom) Result of applying Lasso with an adequate λ .

To evaluate the quality of the reflectivity, we eliminated the zeros (nonpeaks), divided each trace by the maximum absolute value. We then performed a cosine similarity computation against the original reflectivity (without zeros and normalized) for each trace and recorded the average of the 20 traces. Cosine similarity after normalization was also performed to evaluate the extracted wavelet. From that point, we propose the usage of a new measure, which we named deconvolution quality index (DQI), given by

$$DQI = \frac{\sqrt{\max(0, ACSR(\mathbf{r}_e, \mathbf{r}_o))^2 + \max(0, CSW(\mathbf{w}_e, \mathbf{w}_o))^2}}{\sqrt{2}}$$

where \mathbf{r} and \mathbf{w} denote reflectivity and wavelet, respectively, and subscripts $_e$ and $_o$ are relative to the extracted and to the original data, respectively; ACSR is the average of the cosine similarity for the reflectivity peaks after removing zeros and normalizing; and CSW is the cosine similarity for the wavelet.

We performed the following test procedure. Over each set we repeated the deconvolution 10 times, always applying white noise, whose magnitude was 5% of the maximum signal amplitude. As data was synthetically generated, we recorded the amount of peaks for each one, and it was used to compare against the extracted reflectivity and to find the regularization parameter. The deconvolution was performed using F TSMF, OMP, Lobbes, and three variations of FISTA for TSMF: adding normalization and denormalization (F TSMF N), adding the peak elimination and accumulation (F TSMF PA), and adding normalization, denormalization, and peak elimination and accumulation (F TSMF N PA). We analyze the average of cosine similarities for the reflectivity peaks, the CSW, the SNR of the reconstructed signal, and the proposed DQI. We condensed the results for the six methods and inform the average and standard deviation for each measurement in Tables I and II.

Results show that Lobbes, OMP, and the three variations of the original methods give better numerical results than the original one. Despite of that, the original F TSMF showed to have faster convergency than the other methods, denoted by the

TABLE I
COMPARISON OF METHODS, PART I. MEASUREMENTS OF
ACSR, CSW, AND SNR OF THE RECONSTRUCTED SIGNAL

| Results | ACSR | | CSW | | SNR | |
|-------------|-------------|-------------|-------------|-------------|--------------|-------------|
| | μ | σ | μ | σ | μ | σ |
| Lobbes | 0.46 | 0.05 | 0.87 | 0.05 | 54.63 | 0.68 |
| OMP | 0.50 | 0.07 | 0.83 | 0.01 | 55.01 | 0.41 |
| F TSMF | 0.25 | 0.05 | 0.87 | 0.01 | 53.46 | 0.35 |
| F TSMF N | 0.32 | 0.05 | 0.88 | 0.01 | 53.81 | 0.62 |
| F TSMF PA | 0.35 | 0.06 | 0.88 | 0.01 | 53.50 | 0.38 |
| F TSMF N PA | 0.33 | 0.05 | 0.87 | 0.01 | 53.64 | 0.33 |

TABLE II
COMPARISON OF METHODS, PART II. MEASUREMENTS
OF DQI AND THE NUMBER OF ITERATIONS

| Results | DQI | | Iters | |
|-------------|-------------|-------------|-------|----------|
| | μ | σ | μ | σ |
| Lobbes | 0.69 | 0.03 | 27.06 | 8.54 |
| OMP | 0.68 | 0.02 | 15.68 | 4.3 |
| F TSMF | 0.64 | 0.01 | 13.86 | 2.62 |
| F TSMF N | 0.66 | 0.01 | 18.18 | 4.71 |
| F TSMF PA | 0.67 | 0.02 | 15.01 | 2.75 |
| F TSMF N PA | 0.66 | 0.01 | 15.41 | 1.12 |

number of times, the main algorithm is called, expressed in the last columns of Table II. Peak elimination and concentration forced us to consider a greater value of sparsity each time the desired sparsity was not achieved, which forced some more iterations in all algorithms using it. We also employed Adaptive Lasso as used in [15]. In our implementation, Adaptive Lasso was inserted into Lobbes replacing Lasso in order to achieve a desired sparsity by guessing the value of the regularization parameter λ .

VI. CONCLUSION

In this letter, we propose a more natural algorithm (Lobbes) based on machine learning treatment of high-dimensional data with regularization. This algorithm was compared with recently published work [11], which convinced us about the validity of our choices. The proposition of Lobbes and the usage of OMP for comparison were motivated by the observation that TSMF for extracting reflectivity traces and wavelet could be understood as a simplification of sparse coding and dictionary learning and, nevertheless, dimension reduction gives a good result.

From the proof of concept code, we plan to rebuild it using high performance libraries (probably in C++). The source code we produced for this letter is available in a Github public repository as a set of Python scripts: <https://github.com/rgcostafs/PySSD>.

In this letter, the amount of reflectivity peaks remains constant for each trace, and a natural variation is to confine this amount to a predefined range. Cross-validation could be used to determine the best sparsity taking into account some quality criterion. Furthermore, we used a set of traces that were not contiguous in the original data, suggesting some robustness of the entire original approach [11] and, particularly, of our contribution, since those traces are less similar than the neighbors in the original data.

We plan to study an experimental measure of computational efficiency. At the present moment, the heterogeneity of the code produced and used would not allow us to assure a good measurement of memory consumption or CPU time consumption. Also, reflectivity extraction could be run in parallel by using OpenMP for example.

ACKNOWLEDGMENT

The authors would like to thank for valuable suggestions given by the reviewers.

REFERENCES

- [1] D. R. Velis, "Stochastic sparse-spike deconvolution," *Geophysics*, vol. 73, no. 1, pp. R1–R9, 2008.
- [2] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning With Sparsity: The Lasso and Generalizations*, 1st ed. Boca Raton, FL, USA: CRC Press, 2015.
- [3] M. Elad, *Sparse and Redundant Representations. From Theory to Applications in Signal and Image Processing*, 1st ed. New York, NY, USA: Springer-Verlag, 2010.
- [4] S. El Fellah, M. Rziza, and M. El Haziti, "An efficient approach for filling gaps in Landsat 7 satellite images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 1, pp. 62–66, Jan. 2017.
- [5] K. Sun, X. Geng, and L. Ji, "A new sparsity-based band selection method for target detection of hyperspectral image," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 2, pp. 329–333, Feb. 2015.
- [6] J. Bieniarz, E. Aguilerá, X. X. Zhu, R. Müller, and P. Reinartz, "Joint sparsity model for multilook hyperspectral image unmixing," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 4, pp. 696–700, Apr. 2015.
- [7] R. Zhang and J. Castagna, "Seismic sparse-layer reflectivity inversion using basis pursuit decomposition," *Geophysics*, vol. 76, no. 6, pp. R147–R158, 2011.
- [8] C. Li, Y. Ma, X. Mei, C. Liu, and J. Ma, "Hyperspectral image classification with robust sparse representation," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 5, pp. 641–645, May 2016.
- [9] Y. Zhou, W. Chen, J. Gao, and F. Pascal, "Seismic deblending by sparse inversion over dictionary learning," *SEG Tech. Program Expanded Abstracts*, pp. 273–278, 2013.
- [10] S. Beckouche and J. Ma, "Simultaneous dictionary learning and denoising for seismic data," *Geophysics*, vol. 79, no. 3, pp. A27–A31, 2014.
- [11] L. Wang, Q. Zhao, J. Gao, Z. Xu, M. Fehler, and X. Jiang, "Seismic sparse-spike deconvolution via Toeplitz-sparse matrix factorization," *Geophysics*, vol. 81, no. 3, pp. V169–V182, 2016.
- [12] C. Sammut and G. I. Webb, *Encyclopedia of Machine Learning*. New York, NY, USA: Springer, 2011.
- [13] H. Gonzalez, S. Avendaño, and G. Camacho, "A proposal for regularized inversion for an ill-conditioned deconvolution operator," *CTF-Ciencia, Tecnología Futuro*, vol. 5, no. 3, pp. 47–59, 2013.
- [14] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *J. Stat. Softw.*, vol. 33, no. 1, pp. 1–22, 2010.
- [15] Z. Yao, V. Khachatryan, and R. Kolesar, "Practical regularization approach to simultaneous estimation of seismic source wavelet and reflectivity," in *Proc. GeoConvention*, 2017, pp. 042-1–042-4.
- [16] H. Zou, "The adaptive lasso and its oracle properties," *J. Amer. Statist. Assoc.*, vol. 101, no. 476, pp. 1418–1429, Dec. 2006.
- [17] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.
- [18] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *J. Roy. Statist. Soc. B (Statist. Methodol.)*, vol. 67, no. 1, pp. 91–108, 2005.
- [19] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, Mar. 2010.
- [20] J. Mairal, F. Bach, and J. Ponce, "Sparse modeling for image and vision processing," *Found. Trends Comput. Graph. Vis.*, vol. 8, nos. 2–3, pp. 283–285, 2014.
- [21] G. S. Martin, R. Wiley, and K. J. Marfurt, "Marmousi2: An elastic upgrade for Marmousi," *Lead. Edge*, vol. 25, no. 2, pp. 156–166, 2006.
- [22] R. J. Hewett and L. Demanet, "PySIT: Seismic imaging toolbox for Python," Mass. Inst. Technol., Cambridge, MA, USA, Tech. Rep., Mar. 2017.

B

Trechos de Código Fonte

```
import math
import numpy as np
import scipy.signal

def rickerShift(t, freqDom, anguloFase): #{
    a = (np.pi * freqDom * t)
    a *= a
    amp = (1.0 - 2.0*a) * np.exp(-a)
    x = scipy.signal.hilbert(amp)
    x2 = np.cos(anguloFase)*np.real(x) - np.sin(anguloFase)*np.imag(x)
    return x2
#}

def geraRicker(angulo, frequenciaDominante, amostras, duracao): #{
    t = np.linspace(-duracao/2.0, duracao/2.0, amostras)
    rs = rickerShift(t, frequenciaDominante, -angulo*math.pi/180.0)
    rs = rs / np.max(np.abs(rs))
    return rs
#}

import matplotlib.pyplot as plt

if __name__ == "__main__":
    r = geraRicker(30., 30., 1000, 0.2)
    plt.plot(r)
    plt.show()
```

Figura B.1: Código fonte em Python para geração de ondaletas Ricker, convolução com refletividade sintética e exibição gráfica.