Curvelet-Based Primary-Multiple Separation from a Bayesian Perspective

Rayan Saab

February 20, 2008

Co-authors

- Deli Wang (Jilin University)
- Felix Herrmann (Earth and Ocean Sciences, Univesity of British Columbia)
- Ozgur Yilmaz (Mathematics Department, UBC)

Introduction

- We introduce a new primary-multiple separation scheme that
  1. utilizes the sparsity of primaries and multiples in the curvelet domain, and
  2. uses both seismic data and prediction of multiples (e.g. from SRME)
- The algorithm can be derived from a Bayesian formulation that assumes
  1. a sparsity enforcing Laplacian prior distribution,
  2. an assumption of Gaussian noise and errors.
- The algorithm uses soft-thresholding operations, no matrix inversions, makes great progress and almost converges in only a few iterations (for this type of problems).

Introduction

- We introduce a new primary-multiple separation scheme that
  1. utilizes the sparsity of primaries and multiples in the curvelet domain, and
  2. uses both seismic data and prediction of multiples (e.g. from SRME)
- The algorithm can be derived from a Bayesian formulation that assumes
  1. a sparsity enforcing Laplacian prior distribution,
  2. an assumption of Gaussian noise and errors.
- The algorithm uses soft-thresholding operations, no matrix inversions, makes great progress and almost converges in only a few iterations (for this type of problems).

Introduction

- We introduce a new primary-multiple separation scheme that
  1. utilizes the sparsity of primaries and multiples in the curvelet domain, and
  2. uses both seismic data and prediction of multiples (e.g. from SRME)
- The algorithm can be derived from a Bayesian formulation that assumes
  1. a sparsity enforcing Laplacian prior distribution,
  2. an assumption of Gaussian noise and errors.
- The algorithm uses soft-thresholding operations, no matrix inversions, makes great progress and almost converges in only a few iterations (for this type of problems).

Problem and Scope

- Suppose that we have
  1. Seismic data:
  $$\mathbf{b} = \mathbf{s}_1 + \mathbf{s}_2 + \mathbf{n}$$
  composed of the true primaries $(\mathbf{s}_1)$, multiples $(\mathbf{s}_2)$, noise $(\mathbf{n})$
  2. Predictions of the multiples (e.g. from SRME or other methods):
  $$\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$$
  which we assume are not perfect, so $\mathbf{n}_2$ represents (SRME) prediction error, residual noise, ...
- Our objective is to recover the original primaries $\mathbf{s}_1$ and multiples $\mathbf{s}_2$.
- Note that we can generalize the model and algorithm, to account for higher order multiples.

Problem and Scope

- Suppose that we have
  1. Seismic data:
  $$\mathbf{b} = \mathbf{s}_1 + \mathbf{s}_2 + \mathbf{n}$$
  composed of the true primaries $(\mathbf{s}_1)$, multiples $(\mathbf{s}_2)$, noise $(\mathbf{n})$
  2. Predictions of the multiples (e.g. from SRME or other methods):
  $$\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$$
  which we assume are not perfect, so $\mathbf{n}_2$ represents (SRME) prediction error, residual noise, ...
- Our objective is to recover the original primaries s_1 and multiples s_2.
- Note that we can generalize the model and algorithm, to account for higher order multiples.

Problem and Scope

- Suppose that we have
  1. Seismic data:
  $$\mathbf{b} = \mathbf{s}_1 + \mathbf{s}_2 + \mathbf{n}$$
  composed of the true primaries $(\mathbf{s}_1)$, multiples $(\mathbf{s}_2)$, noise $(\mathbf{n})$
  2. Predictions of the multiples (e.g. from SRME or other methods):
  $$\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$$
  which we assume are not perfect, so $\mathbf{n}_2$ represents (SRME) prediction error, residual noise, ...
- Our objective is to recover the original primaries $\mathbf{s}_1$ and multiples $\mathbf{s}_2$.
- Note that we can generalize the model and algorithm, to account for higher order multiples.

Problem and Scope

- Suppose that we have
  1. Seismic data:
  $$\mathbf{b} = \mathbf{s}_1 + \mathbf{s}_2 + \mathbf{n}$$
  composed of the true primaries $(\mathbf{s}_1)$, multiples $(\mathbf{s}_2)$, noise $(\mathbf{n})$
  2. Predictions of the multiples (e.g. from SRME or other methods):
  $$\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$$
  which we assume are not perfect, so $\mathbf{n}_2$ represents (SRME) prediction error, residual noise, ...
- Our objective is to recover the original primaries $\mathbf{s}_1$ and multiples $\mathbf{s}_2$.
- Note that we can generalize the model and algorithm, to account for higher order multiples.

Sparsity

What is Sparsity ?

- A signal is
  said to be "sparse" if most of its values are zero, or almost zero.

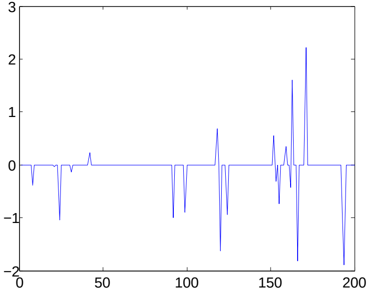


Figure: An Example of a Sparse Signal

Sparsity

What is Sparsity ?

- A signal is
  said to be "sparse" if most of its values are zero, or almost zero.
- If a signal $\mathbf{s}$ is not sparse, sometimes we can find a representation $\mathbf{s} = \mathbf{A}\mathbf{x}$ where $\mathbf{x}$ is sparse.
- Primaries and multiples are sparse in the curvelet domain.
- In other words, a seismic signal can be represented as $\mathbf{s} = \mathbf{A}\mathbf{x}$ where
  - $\mathbf{A} = \mathbf{C}^{\mathbf{H}}$ is the synthesis curvelet operator and
  - $\mathbf{x}$ is the vector of curvelet coefficients.

Curvelets

- Curvelets are localized 'little plane-waves' that are oscillatory in one direction and smooth in the other direction(s).
- They are multiscale and multi-directional.
- Curvelets have an anisotropic shape – they obey the so-called parabolic scaling relationship, yielding a width $\propto$ length^2 for the support of curvelets.
- Very good for detecting wavefronts

Curvelets

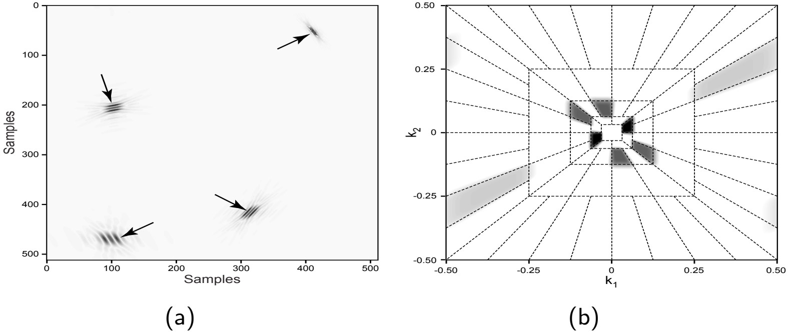


Figure: Curvelet examples. **(a)-(b)** spatial and frequency representation of four different curvelets in the spatial domain at three different scales and in the Fourier domain.

Seismic Primary Multiple Separation

- Here, $\mathbf{s}_1$ are the primaries and $\mathbf{s}_2$ are the multiples. We want to separate them.
- Recall:
  1. seismic data: $\mathbf{b} = \mathbf{s}_1 + \mathbf{s}_2 + \mathbf{n}$
  2. predictions of multiples: $\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$
  3. equivalently $\mathbf{b}_1 = \mathbf{s}_1 + \mathbf{n}_1$ and $\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$
- $\mathbf{s}_1$ and $\mathbf{s}_2$ are sparse in the curvelet domain. $\mathbf{A}$ is the inverse curvelet transform; it is overcomplete, i.e., a frame.
- $\mathbf{s}_1 = \mathbf{A}\mathbf{x}_1$ and $\mathbf{s}_2 = \mathbf{A}\mathbf{x}_2$

Seismic Primary Multiple Separation

- Here, $\mathbf{s}_1$ are the primaries and $\mathbf{s}_2$ are the multiples. We want to separate them.
- Recall:
  1. seismic data: $\mathbf{b} = \mathbf{s}_1 + \mathbf{s}_2 + \mathbf{n}$
  2. predictions of multiples: $\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$
  3. equivalently $\mathbf{b}_1 = \mathbf{s}_1 + \mathbf{n}_1$ and $\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$
- $\mathbf{s}_1$ and $\mathbf{s}_2$ are sparse in the curvelet domain. $\mathbf{A}$ is the inverse curvelet transform; it is overcomplete, i.e., a frame.
- $\mathbf{s}_1 = \mathbf{A}\mathbf{x}_1$ and $\mathbf{s}_2 = \mathbf{A}\mathbf{x}_2$

Seismic Primary Multiple Separation

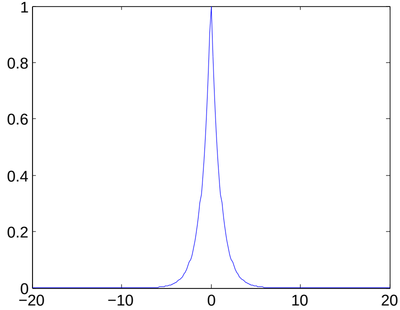
- Here, $\mathbf{s}_1$ are the primaries and $\mathbf{s}_2$ are the multiples. We want to separate them.
- Recall:
  1. seismic data: $\mathbf{b} = \mathbf{s}_1 + \mathbf{s}_2 + \mathbf{n}$
  2. predictions of multiples: $\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$
  3. equivalently $\mathbf{b}_1 = \mathbf{s}_1 + \mathbf{n}_1$ and $\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$
- $\mathbf{s}_1$ and $\mathbf{s}_2$ are sparse in the curvelet domain. $\mathbf{A}$ is the inverse curvelet transform; it is overcomplete, i.e., a frame.
- $\mathbf{s}_1 = \mathbf{A}\mathbf{x}_1$ and $\mathbf{s}_2 = \mathbf{A}\mathbf{x}_2$

Seismic Primary Multiple Separation

- Here, $\mathbf{s}_1$ are the primaries and $\mathbf{s}_2$ are the multiples. We want to separate them.
- Recall:
  1. seismic data: $\mathbf{b} = \mathbf{s}_1 + \mathbf{s}_2 + \mathbf{n}$
  2. predictions of multiples: $\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$
  3. equivalently $\mathbf{b}_1 = \mathbf{s}_1 + \mathbf{n}_1$ and $\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$
- $\mathbf{s}_1$ and $\mathbf{s}_2$ are sparse in the curvelet domain. $\mathbf{A}$ is the inverse curvelet transform; it is overcomplete, i.e., a frame.
- $\mathbf{s}_1 = \mathbf{A}\mathbf{x}_1$ and $\mathbf{s}_2 = \mathbf{A}\mathbf{x}_2$

Sparsity Enforcing Bayesian Prior

- We know that primaries and multiples are sparse in curvelets, and we want to use that knowledge.
- A good sparsity enforcing prior distribution is the Laplacian (Cauchy) distribution $p(x) = ce^{-a|x|}$



Sparsity Enforcing Bayesian Prior

- We know that primaries and multiples are sparse in curvelets, and we want to use that knowledge.
- A good sparsity enforcing prior distribution is the Laplacian (Cauchy) distribution $p(x) = ce^{-a|x|}$
- We also have predictions of the multiples (and primaries), so we use a weighted laplacian prior instead.
- $p(\mathbf{x}_1) = ce^{-\mathbf{w}_1|\mathbf{x}_1|}$
- $p(\mathbf{x}_2) = ce^{-\mathbf{w}_2|\mathbf{x}_2|}$
- In other words we make it unlikely that the curvelet coefficients of the primaries are high where there are high coefficients for the multiples and vice versa.

MAP estimator

We want to find the curvelet coefficients of the primaries and multiples ($\mathbf{x}_1$ and $\mathbf{x}_2$) knowing that

- $\mathbf{b}_1 = \mathbf{s}_1 + \mathbf{n}_1$ and $\mathbf{b}_2 = \mathbf{s}_2 + \mathbf{n}_2$
- Maximize $P(\mathbf{x}_1, \mathbf{x}_2 | \mathbf{b}_1, \mathbf{b}_2)$
- This leads to the following formulation

$$\begin{aligned} & \arg\max_{\mathbf{x}_1,\mathbf{x}_2} P(\mathbf{x}_1, \mathbf{x}_2 | \mathbf{b}_1, \mathbf{b}_2) = \arg\max_{\mathbf{x}_1,\mathbf{x}_2} P(\mathbf{x}_1, \mathbf{x}_2) P(\mathbf{n}) P(\mathbf{n}_2) \\ = \quad & \arg\max_{\mathbf{x}_1,\mathbf{x}_2} -\Bigg(\alpha_1 \|\mathbf{x}_1\|_{1,\mathbf{w}_1} + \alpha_2 \|\mathbf{x}_2\|_{1,\mathbf{w}_1} + \frac{\|\mathbf{A}\mathbf{x}_2 - \mathbf{b}_2\|_2^2}{\sigma_2^2} \\ & \quad + \frac{\|\mathbf{A}(\mathbf{x}_1 + \mathbf{x}_2) - (\mathbf{b}_1 + \mathbf{b}_2)\|_2^2}{\sigma^2} \Bigg) \\ = \quad & \arg\min_{\mathbf{x}_1,\mathbf{x}_2} f(\mathbf{x}_1, \mathbf{x}_2) \end{aligned}$$

- Here $\|\mathbf{x}_i\|_{1,\mathbf{w}_i} = \sum_\mu |w_{i,\mu} x_{i,\mu}|$, $\mu \in \mathcal{M}$

Selection of Weights

- If the weights are selected independently, then the Bayesian interpretation is valid, and the corresponding priors are as in the previous slide.
- In practice, we select the weights as $\mathbf{w}_1 = \lambda_1 \mathbf{A^H b_2}$ and $\mathbf{w}_2 = \lambda_2 \mathbf{A^H b_1}$.
- This choice still corresponds to the same posterior distribution (with the new choice of weights) but with a different prior that is not explicitly formulated.

Three Key Components

Objective Function

$$f(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{x}_1\|_{1,\mathbf{w}_1} + \|\mathbf{x}_2\|_{1,\mathbf{w}_2} + \|\mathbf{A}\mathbf{x}_2 - \mathbf{b}_2\|_2^2 + \eta\|\mathbf{A}(\mathbf{x}_1 + \mathbf{x}_2) - (\mathbf{b}_1 + \mathbf{b}_2)\|_2^2$$

Three Key Components

Objective Function

$$f(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{x}_1\|_{1,\mathbf{w}_1} + \|\mathbf{x}_2\|_{1,\mathbf{w}_2} + \|\mathbf{A}\mathbf{x}_2 - \mathbf{b}_2\|_2^2 + \eta\|\mathbf{A}(\mathbf{x}_1 + \mathbf{x}_2) - (\mathbf{b}_1 + \mathbf{b}_2)\|_2^2$$

Bayesian Interpretation

Minimizing $f(\mathbf{x}_1, \mathbf{x}_2)$ is equivalent to finding the MAP estimator assuming that the coefficients of the sources follow independent weighted Laplacian prior and noise (error) is Gaussian.

Three Key Components

Objective Function

$f(\mathbf{x}_1, \mathbf{x}_2) =$
$\|\mathbf{x}_1\|_{1,\mathbf{w}_1} + \|\mathbf{x}_2\|_{1,\mathbf{w}_2} + \|\mathbf{A}\mathbf{x}_2 - \mathbf{b}_2\|_2^2 + \eta\|\mathbf{A}(\mathbf{x}_1 + \mathbf{x}_2) - (\mathbf{b}_1 + \mathbf{b}_2)\|_2^2$

Bayesian Interpretation

Minimizing $f(\mathbf{x}_1, \mathbf{x}_2)$ is equivalent to finding the MAP estimator assuming that the coefficients of the sources follow independent weighted Laplacian prior and noise (error) is Gaussian.

Separation Algorithm

$$\begin{aligned}
\mathbf{x_1^{n+1}} &= \mathbf{S}_{\frac{\mathbf{w_1}}{\mathbf{2}\eta}}\left[\mathbf{A}^T\mathbf{b}_2 - \mathbf{A}^T\mathbf{A}\mathbf{x_2^n} + \mathbf{A}^T\mathbf{b}_1 - \mathbf{A}^T\mathbf{A}\mathbf{x_1^n} + \mathbf{x_1^n}\right] \\
\mathbf{x_2^{n+1}} &= \mathbf{S}_{\frac{\mathbf{w_2}}{\mathbf{2(1+}\eta\mathbf{)}}}\left[\mathbf{A}^T\mathbf{b}_2 - \mathbf{A}^T\mathbf{A}\mathbf{x_2^n} + \mathbf{x_2^n} + \frac{\eta}{\eta+1}\left(\mathbf{A}^T\mathbf{b}_1 - \mathbf{A}^T\mathbf{A}\mathbf{x_1^n}\right)\right]
\end{aligned}$$

Iterative Thresholding

- Thus our algorithm can be described as

$$\begin{aligned}
\mathbf{x_1^{n+1}} &= \mathbf{S}_{\frac{\mathbf{w_1}}{\mathbf{2}\eta}} \left[\mathbf{A}^T\mathbf{b}_2 - \mathbf{A}^T\mathbf{A}\mathbf{x_2^n} + \mathbf{A}^T\mathbf{b}_1 - \mathbf{A}^T\mathbf{A}\mathbf{x_1^n} + \mathbf{x_1^n}\right] \\
\mathbf{x_2^{n+1}} &= \mathbf{S}_{\frac{\mathbf{w_2}}{\mathbf{2(1+}\eta)}} \left[\mathbf{A}^T\mathbf{b}_2 - \mathbf{A}^T\mathbf{A}\mathbf{x_2^n} + \mathbf{x_2^n} + \tfrac{\eta}{\eta+1}\left(\mathbf{A}^T\mathbf{b}_1 - \mathbf{A}^T\mathbf{A}\mathbf{x_1^n}\right)\right]
\end{aligned}$$

- Here $\mathbf{S}_\alpha$ is the soft-thresholding operator acting *elementwise* as

$$S_{\alpha_\mu}(v_\mu) = \mathrm{sgn}(v_\mu) \cdot \max(0, |v_\mu| - |\alpha_\mu|).$$

- Theorem: The algorithm converges to the minimizer of the objective function.
- Proof: Similar to work of Daubechies04.

Description of Parameters

- The parameters λ_1, λ_2 and η control the tradeoff between the sparsity of the curvelet coefficients (primaries and multiples) and how well we fit both the predicted multiples and the total data. How ?
  1. As we increase λ_1 (or λ_2) we are forcing the estimated curvelet coefficients to be more sparse, allowing for better separation of primaries from multiples. On the other hand, we may introduce artifacts.
  2. As we increase η, we are putting more weight on the total data fit, and less on the predicted multiples.
- While this describes a general trend, in practice the algorithm is robust to parameter choice (within reason).

Description of Parameters

- The parameters λ_1, λ_2 and η control the tradeoff between the sparsity of the curvelet coefficients (primaries and multiples) and how well we fit both the predicted multiples and the total data. How ?
  1. As we increase λ_1 (or λ_2) we are forcing the estimated curvelet coefficients to be more sparse, allowing for better separation of primaries from multiples. On the other hand, we may introduce artifacts.
  2. As we increase η, we are putting more weight on the total data fit, and less on the predicted multiples.
- While this describes a general trend, in practice the algorithm is robust to parameter choice (within reason).

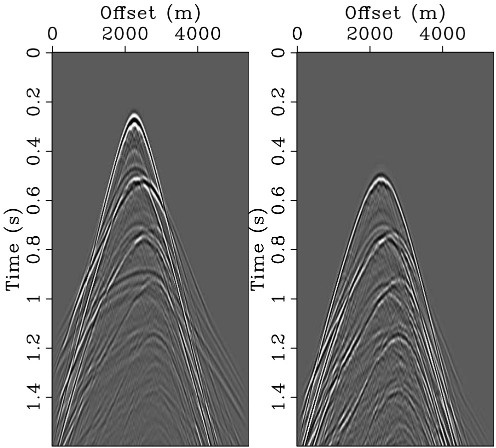
Description of Parameters

- The parameters λ_1, λ_2 and η control the tradeoff between the sparsity of the curvelet coefficients (primaries and multiples) and how well we fit both the predicted multiples and the total data. How ?
  1. As we increase λ_1 (or λ_2) we are forcing the estimated curvelet coefficients to be more sparse, allowing for better separation of primaries from multiples. On the other hand, we may introduce artifacts.
  2. As we increase η, we are putting more weight on the total data fit, and less on the predicted multiples.
- While this describes a general trend, in practice the algorithm is robust to parameter choice (within reason).

Description of Parameters

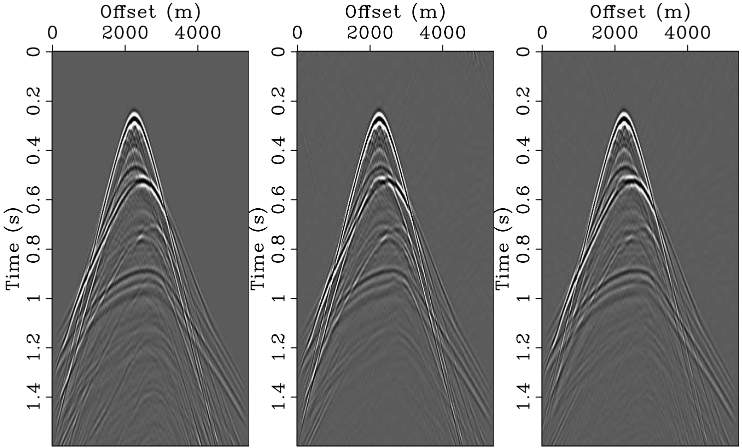
- The parameters λ_1, λ_2 and η control the tradeoff between the sparsity of the curvelet coefficients (primaries and multiples) and how well we fit both the predicted multiples and the total data. How ?
    1. As we increase λ_1 (or λ_2) we are forcing the estimated curvelet coefficients to be more sparse, allowing for better separation of primaries from multiples. On the other hand, we may introduce artifacts.
    2. As we increase η, we are putting more weight on the total data fit, and less on the predicted multiples.
- While this describes a general trend, in practice the algorithm is robust to parameter choice (within reason).

Total Data and Predicted Multiples



Separation Results

- SRME   Single threshold (Herrmann07)   proposed algorithm



Results with Different Parameter Choices

SNR (dB)	$\{\lambda_1^*, \lambda_2^*\}$	$\{2 \cdot \lambda_1^*, \lambda_2^*\}$	$\{\lambda_1^*, 2 \cdot \lambda_2^*\}$	$100 \cdot \{\lambda_1^*, \lambda_2^*\}$
η^*	12.133	11.211	11.455	-
$\frac{1}{2} \cdot \eta^*$	11.356	9.428	11.459	-
$2 \cdot \eta^*$	11.436	12.129	9.924	-
$100 \cdot \eta^*$	-	-	-	10.647

Model Generalization

- The model can be generalized by assuming that we have m observations of n sources:

$$\mathbf{b}_i = \sum_{j=1}^{N} \psi_{ij}\mathbf{s}_j + \mathbf{n}_j, \quad i \in \{1, ..., m\}$$

and n' available predictions of the sources:

$$\mathbf{b}_{i+m} = \mathbf{s}_i + \mathbf{n}_{i+m}, \quad i \in \{1, ..., n'\}$$

- We still assume that the underlying sources are sparse in the transform domain:

$$\mathbf{s}_j = \mathbf{A}\mathbf{x}_j$$

where $\mathbf{s}_j \in \mathbb{R}^{M\times 1}$, $\mathbf{A} \in \mathbb{R}^{M\times N}$ and $\mathbf{x}_j \in \mathbb{R}^{N\times 1}$

Generalized Cost Function and Iterative Algorithm

- Using the same approach as before we can derive a cost function and an optimization algorithm to minimize it.

$$f(\mathbf{x}_1, ..., \mathbf{x}_n) = \sum_{i=1}^{n} \lambda_i \|\mathbf{x}_i\|_{1,\mathbf{w}_i} + \sum_{i=1}^{m+n'} \eta_i \|\mathbf{A} \sum_{j=1}^{n} \psi_{ij} \mathbf{x}_j - \mathbf{b}_i\|_2^2.$$

- In order to derive the algorithm, define

$$\hat{\mathbf{x}} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{bmatrix}, \hat{\mathbf{w}} = \begin{bmatrix} \lambda_1 \mathbf{w}_1 \\ \vdots \\ \lambda_n \mathbf{w}_n \end{bmatrix}, \hat{\mathbf{b}} = \begin{bmatrix} \sqrt{\eta_1}\mathbf{b}_1 \\ \vdots \\ \sqrt{\eta_{m+n'}}\mathbf{b}_{m+n'} \end{bmatrix},$$

$$\hat{\mathbf{\Psi}} = \begin{bmatrix} \sqrt{\eta_1}\psi_{11} & \cdots & \sqrt{\eta_1}\psi_{1n} \\ \vdots & & \vdots \\ \sqrt{\eta_{m+n'}}\psi_{m+n',1} & \cdots & \sqrt{\eta_{m+n'}}\psi_{m+n',n} \end{bmatrix}.$$

Iterative Algorithm

We still need to define the matrix $\hat{\mathbf{A}} = \mathbf{A} \otimes \hat{\mathbf{\Psi}}$ where $\otimes$ represents the Kroenecker product of two operators. We can now rewrite the cost function as

$$f(\mathbf{x}_1, ..., \mathbf{x}_n) = f(\hat{\mathbf{x}}) = \|\hat{\mathbf{x}}\|_{1,\hat{\mathbf{w}}} + \|\hat{\mathbf{b}} - \hat{\mathbf{A}}\hat{\mathbf{x}}\|_2^2 \quad (1)$$

We can now derive the following recursion to optimize the cost function at iteration $k+1$, with $c = \|\hat{\mathbf{A}}\|_2^2 = \|\hat{\mathbf{\Psi}}\|_2^2 \|\mathbf{A}\|_2^2$.

$$\hat{\mathbf{x}}^{(k+1)} = \mathbf{S}_{\frac{\hat{\mathbf{w}}}{2c}} \left(\frac{1}{c} [\hat{\mathbf{A}}^H \hat{\mathbf{b}} - \hat{\mathbf{A}}^H \hat{\mathbf{A}} \hat{\mathbf{x}}^{(k)}] + \hat{\mathbf{x}}^{(k)} \right) \quad (2)$$

Conclusion

- We introduced a primary-multiple separation algorithm that
  1. utilizes the sparsity of primaries and multiples in the curvelet domain, and
  2. uses both seismic data and prediction of multiples (e.g. from SRME)
- It can be derived from a Bayesian formulation that assumes
  1. a sparsity enforcing Laplacian prior distribution,
  2. noise and errors are Gaussian.
- The algorithm uses soft-thresholding operations, no matrix inversions, converges in only a few iterations (for this type of problems).

Conclusion

- We introduced a primary-multiple separation algorithm that
  1. utilizes the sparsity of primaries and multiples in the curvelet domain, and
  2. uses both seismic data and prediction of multiples (e.g. from SRME)
- It can be derived from a Bayesian formulation that assumes
  1. a sparsity enforcing Laplacian prior distribution,
  2. noise and errors are Gaussian.
- The algorithm uses soft-thresholding operations, no matrix inversions, converges in only a few iterations (for this type of problems).

Conclusion

- We introduced a primary-multiple separation algorithm that
  1. utilizes the sparsity of primaries and multiples in the curvelet domain, and
  2. uses both seismic data and prediction of multiples (e.g. from SRME)
- It can be derived from a Bayesian formulation that assumes
  1. a sparsity enforcing Laplacian prior distribution,
  2. noise and errors are Gaussian.
- The algorithm uses soft-thresholding operations, no matrix inversions, converges in only a few iterations (for this type of problems).

The authors would like to thank

- The authors of CurveLab (Demanet, Ying, Candes, Donoho)
- SLIM team members: S. Ross Ross, H. Modzelewski, and C. Brown for SLIMpy (slim.eos.ubc.ca/SLIMpy)

This presentation was carried out as part of the SINBAD project with financial support, secured through ITF, from the following organizations: BG, BP, Chevron, ExxonMobil, and Shell. SINBAD is part of the collaborative research and development (CRD) grant number 334810-05 funded by the Natural Science and Engineering Research Council (NSERC).