A debiasing approach to microseismic

Shashin Sharan, Rajiv Kumar, Rongrong Wang and Felix J. Herrmann

SEG 2018
Motivation

Unconventional Reservoir Schematic
Motivation

Unconventional Reservoir Schematic
Motivation

Unconventional Reservoir Schematic
Motivation

Unconventional Reservoir Schematic
Motivation

Unconventional Reservoir Schematic
Motivation

Unconventional Reservoir Schematic

Objectives

- detection of microseismic events in space and time
- estimation of source time function
Proposed method w/ sparsity promotion

Unconventional Reservoir Schematic

Assumptions

- localized in space
Proposed method w/ sparsity promotion

Unconventional Reservoir Schematic

Assumptions
- localized in space
- finite energy along time
Proposed method w/ sparsity promotion

\[ \text{minimize} \quad \|Q\|_{2,1} \]
\[ \text{subject to} \quad \|\mathcal{F}[m](Q) - d\|_2 \leq \epsilon \]

Source wavefield

Forward modeling operator

Noise level

Slowness square

Observed data

\[ Q \in \mathbb{R}^{n_x \times n_t} \]
\[ n_x: \text{ number of grid points} \]
\[ n_t: \text{ number of time samples} \]

[Van Den Berg et al.,'08]
Proposed method w/ sparsity promotion

\[
\begin{align*}
\text{minimize} & \quad \| Q \|_{2,1} \\
\text{subject to} & \quad \| \mathcal{F}[m](Q) - d \|_2 \leq \epsilon
\end{align*}
\]

Source wavefield

Forward modeling operator

Slowness square

Observe data

Noise level

\( Q \in \mathbb{R}^{n_x \times n_t} \)

\( n_x \): number of grid points

\( n_t \): number of time samples

Similar to classic Basis pursuit denoising (BPDN)

[Van Den Berg et al.,’08]
Solving w/ Linearized Bregman

\[
\begin{align*}
\min_Q & \quad \|Q\|_{2,1} + \frac{1}{2\mu} \|Q\|_F^2 \\
\text{subject to} & \quad \|\mathcal{F}[m](Q) - d\|_2 \leq \epsilon
\end{align*}
\]

*where \(\|\cdot\|_F\) is the Frobenius norm

- Recent successful application to seismic imaging problem
- Three-step algorithm simple to implement
- Choice of \(\mu\) controls the trade off between sparsity and the Frobenius norm
- \(\mu \uparrow \infty\) corresponds to solving original BPDN problem
Linearized Bregman algorithm

1. Data $d$, slowness square $m$  /**Input
2. for $k = 0, 1, \cdots$
3. \begin{equation}
V_k = \mathcal{F}^\top[m](\Pi_\epsilon(\mathcal{F}[m](Q_k) - d))\end{equation}  /**adjoint solve
4. \begin{equation}
Z_{k+1} = Z_k - t_k V_k\end{equation}  /**auxiliary variable update
5. \begin{equation}
Q_{k+1} = \text{Prox}_{\mu \ell_2,1}(Z_{k+1})\end{equation}  /**sparsity promotion
6. end
7. $I(x) = \sum_t |Q(x, t)|$  /**Intensity plot

* $\Pi_\epsilon(r) = \max\{0, 1 - \frac{r}{\|r\|}\}.(r)$ the projection on to $\ell_2$ norm ball
Linearized Bregman algorithm

1. **Data** $d$, **slowness square** $m$  //Input
2. **for** $k = 0, 1, \cdots$
3.   \[ V_k = F^T[m](\Pi_\epsilon(F[m](Q_k) - d)) \] //adjoint solve
4.   \[ Z_{k+1} = Z_k - t_k V_k \] //auxiliary variable update
5.   \[ Q_{k+1} = \text{Prox}_{\mu \ell_2,1}(Z_{k+1}) \] //sparsity promotion
6. **end**
7. **I(x) = \sum_t |Q(x, t)|** //Intensity plot

* $\Pi_\epsilon(r) = \max\{0, 1 - \frac{\epsilon}{||r||}\}(r)$ the projection on to $\ell_2$ norm ball

*where $t_k = \frac{||F[m](Q_k) - d||^2}{||F[m](F[m](Q_k) - d)||^2}$ is the dynamic step length
Linearized Bregman algorithm

1. Data \( d \), slowness square \( m \)  //Input
2. \textbf{for} \( k = 0, 1, \cdots \)
3. \( V_k = \mathcal{F}[m](\Pi_\epsilon(\mathcal{F}[m](Q_k) - d)) \)  //adjoint solve
4. \( Z_{k+1} = Z_k - t_k V_k \)  //auxiliary variable update
5. \( Q_{k+1} = \text{Prox}_{\mu \ell_2,1}(Z_{k+1}) \)  //sparsity promotion
6. \textbf{end}
7. \( I(x) = \sum_t |Q(x, t)| \)  //Intensity plot

\* \( \Pi_\epsilon(r) = \max\{0, 1 - \frac{\epsilon}{||r||}\}.(r) \) the projection on to \( \ell_2 \) norm ball

\*where \( t_k = \frac{1}{\|\mathcal{F}[m](\mathcal{F}[m](Q_k) - d)\|^2} \) is the dynamic step length

\* Prox\( \mu \ell_{2,1}(C) := \arg\min_B \|B\|_{2,1} + \frac{1}{2\mu}\|C - B\|_F^2 \) is the proximal mapping of the \( \ell_{2,1} \) norm
Linearized Bregman algorithm

1. Data $d$, slowness square $m$  //Input
2. for $k = 0, 1, \cdots$
3. $V_k = \mathcal{F}^\top[m](\Pi_\epsilon(\mathcal{F}[m](Q_k) - d))$  //adjoint solve
4. $Z_{k+1} = Z_k - t_k V_k$  //auxiliary variable update
5. $Q_{k+1} = \text{Prox}_{\mu \ell_2,1}(Z_{k+1})$  //sparsify promotion
6. end
7. $I(x) = \sum_t |Q(x, t)|$  //Intensity plot

* $\Pi_\epsilon(r) = \max\{0, 1 - \frac{\epsilon}{\|r\|}\}(r)$ the projection on to $\ell_2$ norm ball

* where $t_k = \frac{\|\mathcal{F}[m](Q_k) - d\|^2}{\|\mathcal{F}[m](\mathcal{F}[m](Q_k) - d)\|^2}$ is the dynamic step length

* $\text{Prox}_{\mu \ell_2,1}(C) := \arg \min_B \|B\|_{\ell_2,1} + \frac{1}{2\mu} \|C - B\|^2_F$ is the proximal mapping of the $\ell_{2,1}$ norm

- **Source location**: estimated as outlier in intensity plot
- **Source-time function**: temporal variation of wavefield at estimated source location
\[ V_1 = \mathcal{F}^\dagger [\mathbf{m}] (\mathcal{H}_\epsilon (\mathcal{F}[\mathbf{m]}(Q_0) - \mathbf{d})) \]
\[ \mathbf{V}_1 = \mathbf{F}^\dagger [\mathbf{m}] (\mathbf{H} (\mathbf{F} [\mathbf{m}][Q_0] - \mathbf{d})) \]

**Adjoint solve**
\[ V_1 = \mathcal{F}^\dagger [m](\Pi_\epsilon(\mathcal{F}[m](Q_0) - d)) \]
$$V_1 = \mathcal{F}^\dagger [m] \Pi \epsilon(\mathcal{F}[m](Q_0) - d))$$

**Adjoint solve**

**Auxiliary variable update**

$$Z_1 = Z_0 - t_1 V_1$$
\[ V_1 = \mathcal{F}^\dagger [m] (\Pi \epsilon (\mathcal{F}[m](Q_0) - d)) \]

**Adjoint solve**

\[ Z_1 = Z_0 - t_1 V_1 \]

**Auxiliary variable update**

\[ Q_1 = \text{Prox}_{\mu \epsilon_{2,1}} (Z_1) \]

**Sparsity promotion**
\[ \mathbf{V}_1 = \mathcal{F}^\dagger \left[ \mathbf{m} \left( \Pi_{\epsilon} \left( \mathcal{F} \left[ \mathbf{m} (\mathbf{Q}_0) - \mathbf{d} \right] \right) \right) \right] \]

**Adjoint solve**

\[ \mathbf{Z}_1 = \mathbf{Z}_0 - t_1 \mathbf{V}_1 \]

**Auxiliary variable update**

\[ \mathbf{Q}_1 = \text{Prox}_{\mu \ell_{2,1}} (\mathbf{Z}_1) \]

**Sparsity promotion**

\[ \mathbf{I}(\mathbf{x}) = \sum_t | \mathbf{Q}_1(\mathbf{x}, t) | \]
**Source-time function**

- The source-time function illustrates the intensity or amplitude of the source over time.

**Adjoint solve**

- The adjoint solve is represented by the equation:
  
  \[ V_1 = F^\dagger[m](\|\epsilon[F[m](Q_0) - d]) \]

**Auxiliary variable update**

- The auxiliary variable update is given by:
  
  \[ Z_1 = Z_0 - t_1 V_1 \]

**Sparsity promotion**

- Sparsity promotion is calculated as:
  
  \[ Q_1 = \text{Prox}_{\mu \ell_2,1}(Z_1) \]
Linearized Bregman algorithm

1. **Data** $d$, slowness square $m$  //Input
2. **for** $k = 0, 1, \cdots$
3. \[ V_k = F^\top [m](\Pi_\epsilon(F[m](Q_k) - d)) \]  //adjoint solve
4. \[ Z_{k+1} = Z_k - t_k V_k \]  //auxiliary variable update
5. \[ Q_{k+1} = \text{Prox}_{\mu \ell_2,1}(Z_{k+1}) \]  //sparsity promotion
6. **end**
7. \[ I(x) = \sum_t |Q(x, t)| \]  //Intensity plot

* $\Pi_\epsilon(r) = \max\{0, 1 - \frac{\epsilon}{\|r\|}\}(r)$ the projection on to $\ell_2$ norm ball
Linearized Bregman algorithm

1. Data \( d \), slowness square \( m \)  //Input
2. for \( k = 0, 1, \cdots \)
3. \[ V_k = \mathcal{F}^\top[m](\Pi_\varepsilon(\mathcal{F}[m](Q_k) - d)) \]  //adjoint solve
4. \[ Z_{k+1} = Z_k - t_k V_k \]  //auxiliary variable update
5. \[ Q_{k+1} = \text{Prox}_{\mu \ell_2,1}(Z_{k+1}) \]  //sparsity promotion
6. end
7. \[ I(x) = \sum_t |Q(x, t)| \]  //Intensity plot

* \( \Pi_\varepsilon(r) = \max\{0, 1 - \frac{r}{\|r\|}\} \) is the projection on to \( \ell_2 \) norm ball

*where \( t_k = \frac{\|\mathcal{F}[m](Q_k) - d\|^2}{\|\mathcal{F}[m](\mathcal{F}[m](Q_k) - d)\|^2} \) is the dynamic step length
Linearized Bregman algorithm

1. Data d, slowness square m  //Input
2. for \( k = 0, 1, \ldots \)
3. \( V_k = \mathcal{F}^\top \{m\} (\Pi_\ell (\mathcal{F} \{m\} (Q_k) - d)) \)  //adjoint solve
4. \( Z_{k+1} = Z_k - t_k V_k \)  //auxiliary variable update
5. \( Q_{k+1} = \text{Prox}_{\mu \ell_2,1} (Z_{k+1}) \)  //sparsity promotion
6. end
7. \( I(x) = \sum_t |Q(x, t)| \)  //Intensity plot

* \( \Pi_\ell (r) = \max\{0, 1 - \frac{r}{\|r\|}\} \cdot (r) \) the projection on to \( \ell_2 \) norm ball

*where \( t_k = \frac{\|\mathcal{F} \{m\} (Q_k) - d\|^2}{\|\mathcal{F} \{m\} (Q_k) - d\|^2} \) is the dynamic step length

* \( \text{Prox}_{\mu \ell_2,1} (C) := \arg\min_B \|B\|_{2,1} + \frac{1}{2\mu} \|C - B\|_F^2 \) is the proximal mapping of the \( \ell_{2,1} \) norm
Linearized Bregman algorithm

1. **Data** $d$, **slowness square** $m$  
   //Input
2. **for** $k = 0, 1, \cdots$
3. $V_k = \mathcal{F}^\top[m](\Pi_\varepsilon(\mathcal{F}[m](Q_k) - d))$  
   //adjoint solve
4. $Z_{k+1} = Z_k - t_k V_k$  
   //auxiliary variable update
5. $Q_{k+1} = \text{Prox}_{\mu \ell_2,1}(Z_{k+1})$  
   //sparsity promotion
6. **end**
7. $I(x) = \sum_t |Q(x, t)|$  
   //Intensity plot

* $\Pi_\varepsilon(r) = \max\{0, 1 - \frac{\varepsilon}{\|r\|}\} \cdot r$ the projection on to $\ell_2$ norm ball

*where $t_k = \frac{\|\mathcal{F}[m](Q_k) - d\|^2}{\|\mathcal{F}[m](\mathcal{F}[m](Q_k) - d)\|^2}$ is the dynamic step length

* $\text{Prox}_{\mu \ell_2,1}(C) := \arg\min_B \|B\|_{2,1} + \frac{1}{2\mu} \|C - B\|_F^2$ is the proximal mapping of the $\ell_{2,1}$ norm

- **Source location**: estimated as outlier in intensity plot
- **Source-time function**: temporal variation of wavefield at estimated source location
Linearized Bregman algorithm

1. **Data d, slowness square m**  //Input
2. **for** $k = 0, 1, \cdots$
3. \[ V_k = \mathcal{F}^\top [m] (\Pi_\epsilon (\mathcal{F}[m](Q_k) - d)) \]  //adjoint solve
4. \[ Z_{k+1} = Z_k - t_k V_k \]  //auxiliary variable update
5. \[ Q_{k+1} = \text{Prox}_{\mu \ell_{2,1}} (Z_{k+1}) \]  //sparsity promotion
6. **end**
7. \[ I(x) = \sum_t |Q(x, t)| \]  //Intensity plot

* $\Pi_\epsilon (r) = \max\{0, 1 - \frac{\epsilon}{\|r\|}\} \cdot (r)$ the projection on to $\ell_2$ norm ball

* where $t_k = \frac{\|\mathcal{F}[m](Q_k) - d\|^2}{\|\mathcal{F}^\top[m](\mathcal{F}[m](Q_k) - d)\|^2}$ is the dynamic step length

* Prox$_{\mu \ell_{2,1}} (C) := \arg \min_B \|B\|_{2,1} + \frac{1}{2\mu} \|C - B\|_F^2$ is the proximal mapping of the $\ell_{2,1}$ norm

- **Source location**: estimated as outlier in intensity plot
- **Source-time function**: temporal variation of wavefield at estimated source location
High level of noise

\[ \Pi_\varepsilon(r) = \max\{0, 1 - \frac{\varepsilon}{\|r\|}\} \cdot (r) \]

* \( r = \mathcal{F}[m](Q) - d \) is the data residual

\[ \varepsilon \gg \|r\| \text{ for very noisy data} \]

\[ \Pi_\varepsilon(r) = 0 \]
Linearized Bregman algorithm with extreme noise

1. Data $d$, slowness square $m$ //Input
2. for $k = 0, 1, \ldots$
3. $V_k = \mathcal{F}^\top[m](\Pi_\epsilon(\mathcal{F}[m](Q_k) - d))$ //adjoint solve
4. $Z_{k+1} = Z_k - \ell_k V_k$ //auxiliary variable update
5. $Q_{k+1} = \text{Prox}_{\mu \ell_{2,1}}(Z_{k+1})$ //sparsity promotion
6. end
7. $I(x) = \sum_t |Q(x, t)|$ //Intensity plot

No updates
Numerical Experiment: Extreme noise

Modeling information:

- **Model size**: 3.15 km x 1.08 km
- **Grid spacing**: 5 m
- **Total number of sources**: 5
- **Peak frequency**: 25 Hz & 30 Hz
- **Receiver spacing**: 10m
- **Receiver depth**: 20m
- **Sampling interval**: 0.5 ms
- **Recording length**: 1 s
- **Free surface**: No
- **Amplitude ratio of sources**: 2:1
Noisy Data

SNR: -7.3 dB
Estimated source location w/o denoising
Properties of noise in microseismic data

Amplitude of ambient noise is similar or higher than the amplitude of signal

Frequency range of noise is similar to the frequency range of signal

This makes signal and noise separation difficult

Eventually causes problem in detecting microseismic sources

[Forghani-Arani et al., '12; St-Onge et al., '11]
Road ahead

Noise is a big problem in microseismic data

Transform based methods which discriminate signal components based on directionality and scale can be useful in separating microseismic signal and noise
Curvelet transform

[Candes et al., ’00; Herrmann et al., ’08]
**Curvelet transform**

- **Curvelet transform**: multi-scale and multi-directional transform
- Maps seismic data into angular wedges of different scales in f-k domain
- Better separation of signal and noise in transform domain
Curvlet based denoising steps

1. Noisy Data $d$, forward curvelet transform operator $C$, Threshold parameter $\lambda$ //Input
2. $b = Cd$ //Forward curvelet transform
3. $[sb, \text{idx}] = \text{Sort}(|b|)$ //Sorting in descending order where $\text{idx}$ stores the indices of sorted curvelet coefficients
4. $e_h = \sqrt{\frac{\sum_{i=1}^{h} sb_i^2}{\sum_{i=1}^{h} sb_i^2}}$ //normalized cumulative energy
5. Find the smallest index $p$ such that $e_p \geq \lambda$
6. $S = C^H(\text{idx}(1:p), :)$ //New inverse curvelet transform operator
7. $b_{dn} = (S^H S)^{-1} S^H d$ //Solving the normal equation
8. $d_{dn} = \Re(Sb_{dn})$ //denoising
Noisy Data

SNR: -7.3 dB
Denoised data w/ curvelet based denoising

SNR: 3.5 dB
Difference

![Graph showing the difference between receiver position and time](image)
Estimated source location w/o denoising
Estimated source location after denoising
Estimated source location after denoising
Estimated source location after denoising
Estimated source location after denoising (zoomed)
Estimated source location after denoising (zoomed)
Source-time function comparison
Source-time function comparison
Source-time function comparison

- Correct shape
- Incorrect amplitude
- 40 times amplified
Debiasing
Debiasing

![Graph with coordinates and a matrix]

\[ H = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
\end{bmatrix} \]

*where \( n \) is the number of detected microseismic sources*
Debiasing is possible because

- we are able to detect microseismic sources
- Even from noisy data with very low SNR

\[ \hat{W} = \arg \min_{W \in \mathbb{R}^{n_t \times n}} \| \mathcal{F}(HW^\top) - d \| \]

*where \( H \in n_x \times n \)

*we use noisy data \( d \) to avoid any amplitude error
Source-time function comparison after debiasing
Source-time function comparison after debiasing
Source-time function comparison after debiasing

- Correct shape
- Correct amplitude
Numerical Experiment: BG Compass model

Modeling information:

- **Model size:** 2.20 km x 0.90 km
- **Grid spacing:** 5 m
- **Total number of sources:** 2
- **Peak frequency:** 30 Hz
- **Receiver spacing:** 10 m
- **Receiver depth:** 20 m
- **Sampling interval:** 0.5 ms
- **Recording length:** 1 s
- **Free surface:** No
Numerical Experiment: BG Compass model

Sources are located within half a wavelength with overlapping source-time functions

Modeling information:

- **Model size**: 2.20 km x 0.90 km
- **Grid spacing**: 5 m
- **Total number of sources**: 2
- **Peak frequency**: 30 Hz
- **Receiver spacing**: 10m
- **Receiver depth**: 20m
- **Sampling interval**: 0.5 ms
- **Recording length**: 1 s
- **Free surface**: No
Noisy data

SNR: -8.2 dB
Denoised data w/ curvelet based denoising

SNR: 5.9 dB
Difference
Estimated source location w/o denoising
Estimated source location after denoising
Estimated source location after denoising
Estimated source location after denoising
Estimated source location after denoising (zoomed)
Estimated source location after denoising (zoomed)
Source-time function comparison after debiasing
Source-time function comparison after debiasing

Estimated source-time function after denoising (without debiasing) is scaled by a factor of 4
Conclusions and future work

With debiasing based approach we are able to:

- locate closely spaced microseismic sources and
- estimate the associated source-time function with correct amplitude from data with very low SNR
Conclusions and future work

With debiasing based approach we are able to:

- locate closely spaced microseismic sources and
- estimate the associated source-time function with correct amplitude from data with very low SNR

Proposed approach is computationally cheap as it requires:

- very few forward and backward Curvelet transforms and
- few iterations of accelerated version of linearized Bregman algorithm
Conclusions and future work

With debiasing based approach we are able to:

- locate closely spaced microseismic sources and
- estimate the associated source-time function with correct amplitude from data with very low SNR

Proposed approach is computationally cheap as it requires:

- very few forward and backward Curvelet transforms and
- few iterations of accelerated version of linearized Bregman algorithm

Use PCA based techniques to deal with different kinds of noise
Acknowledgement

This research was carried out as part of the SINBAD project with the support of the member organizations of the SINBAD Consortium.
Thank you !!