Scalable Bayesian seismic wavelet estimation

Guillermina Senn[1]

Joint work with Matt Walker[2], Håkon Tjelmeland[1], and Andrew Holbrook[3]

[1]Norwegian University of Science and Technology (NTNU)
[2]bp · [3]University of California, Los Angeles

SINTEF Industri, Trondheim
May 27, 2025

Scalable Bayesian seismic wavelet estimation

Guillermina Senn[1]

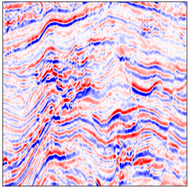
Joint work with Matt Walker[2], Håkon Tjelmeland[1], and Andrew Holbrook[3]

[1]Norwegian University of Science and Technology (NTNU)
[2]bp · [3]University of California, Los Angeles

SINTEF Industri, Trondheim
May 27, 2025

Summary

1. Seismic data



- Reflectivity $\boldsymbol{c}$
- Seismic wavelet $\boldsymbol{w}$
- Observational noise $\boldsymbol{e}$
- Seismic data $\boldsymbol{d}$

2. Model

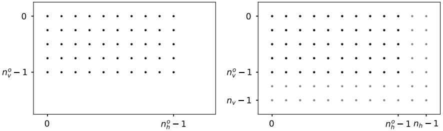
1D convolutional model:

$$\boldsymbol{d}_j = \boldsymbol{c}_j \star \boldsymbol{w} + \boldsymbol{e}_j \quad \text{for column } j.$$

Probabilistic model:

- Gaussian reflectivity prior.
- Gaussian wavelet prior.
- Gaussian likelihood.
- Inverse-Gamma variances.

3. Scalability



4. Estimation

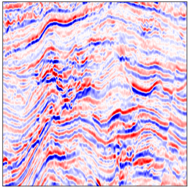
Sample posterior distribution

$$p(\boldsymbol{c}, \boldsymbol{w}, \sigma_c^2, \sigma_w^2, \sigma_d^2 \mid \boldsymbol{d})$$

with Markov chain Monte Carlo (MCMC).

Summary

1. Seismic data



- Reflectivity $\boldsymbol{c}$
- Seismic wavelet $\boldsymbol{w}$
- Observational noise $\boldsymbol{e}$
- Seismic data $\boldsymbol{d}$

2. Model

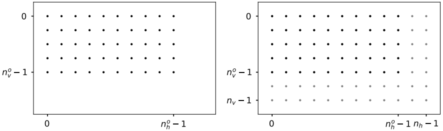
Convolutional model:

$$\boldsymbol{d}_j = \boldsymbol{c}_j \star \boldsymbol{w} + \boldsymbol{e}_j \quad \text{for column } j.$$

Probabilistic model:

- Gaussian reflectivity prior.
- Gaussian wavelet prior.
- Gaussian likelihood.
- Inverse-Gamma variances.

3. Scalability



4. Estimation

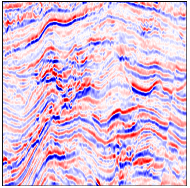
Sample posterior distribution

$$p(\boldsymbol{c}, \boldsymbol{w}, \sigma_c^2, \sigma_w^2, \sigma_d^2 \mid \boldsymbol{d})$$

with Markov chain Monte Carlo (MCMC).

Summary

1. Seismic data



- Reflectivity $\boldsymbol{c}$
- Seismic wavelet $\boldsymbol{w}$
- Observational noise $\boldsymbol{e}$
- Seismic data $\boldsymbol{d}$

2. Model

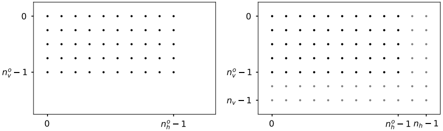
Convolutional model:

$$\boldsymbol{d}_j = \boldsymbol{c}_j \star \boldsymbol{w} + \boldsymbol{e}_j \quad \text{for column } j.$$

Probabilistic model:

- Gaussian reflectivity prior.
- Gaussian wavelet prior.
- Gaussian likelihood.
- Inverse-Gamma variances.

3. Scalability



4. Estimation

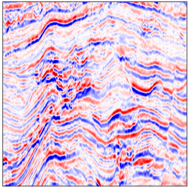
Sample posterior distribution

$$p(\boldsymbol{c}, \boldsymbol{w}, \sigma_c^2, \sigma_w^2, \sigma_d^2 \mid \boldsymbol{d})$$

with Markov chain Monte Carlo (MCMC).

Summary

1. Seismic data



- Reflectivity $\boldsymbol{c}$
- Seismic wavelet $\boldsymbol{w}$
- Observational noise $\boldsymbol{e}$
- Seismic data $\boldsymbol{d}$

2. Model

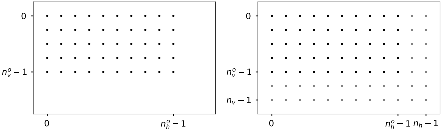
Convolutional model:

$$\boldsymbol{d}_j = \boldsymbol{c}_j \star \boldsymbol{w} + \boldsymbol{e}_j \quad \text{for column } j.$$

Probabilistic model:

- Gaussian reflectivity prior.
- Gaussian wavelet prior.
- Gaussian likelihood.
- Inverse-Gamma variances.

3. Scalability



4. Estimation

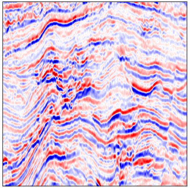
Sample posterior distribution

$$p(\boldsymbol{c}, \boldsymbol{w}, \sigma_c^2, \sigma_w^2, \sigma_d^2 \mid \boldsymbol{d})$$

with Markov chain Monte Carlo (MCMC).

Summary

1. Seismic data



- Reflectivity $\boldsymbol{c}$
- Seismic wavelet $\boldsymbol{w}$
- Observational noise $\boldsymbol{e}$
- Seismic data $\boldsymbol{d}$

2. Model

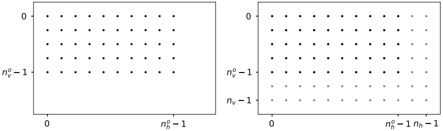
Convolutional model:

$$\boldsymbol{d}_j = \boldsymbol{c}_j \star \boldsymbol{w} + \boldsymbol{e}_j \quad \text{for column } j.$$

Probabilistic model:

- Gaussian reflectivity prior.
- Gaussian wavelet prior.
- Gaussian likelihood.
- Inverse-Gamma variances.

3. Scalability



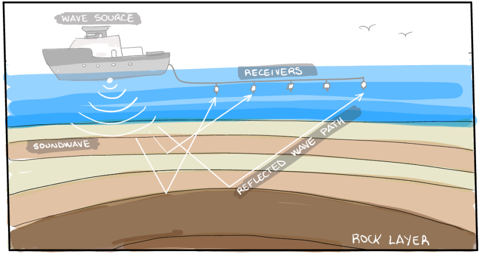
4. Estimation

Sample posterior distribution

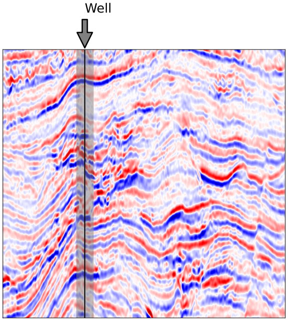
$$p(\boldsymbol{c}, \boldsymbol{w}, \sigma_c^2, \sigma_w^2, \sigma_d^2 \mid \boldsymbol{d})$$

with Markov chain Monte Carlo (MCMC).

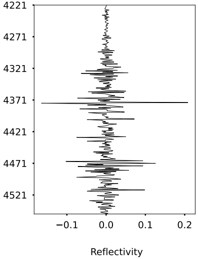
Seismic data acquisition and processing



(a) Data acquisition.

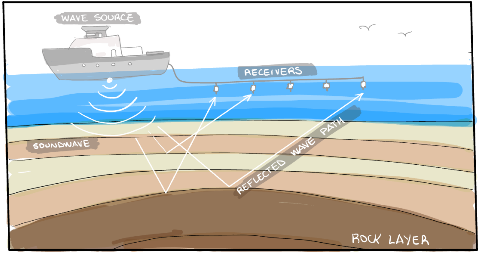


(b) Seismic data D

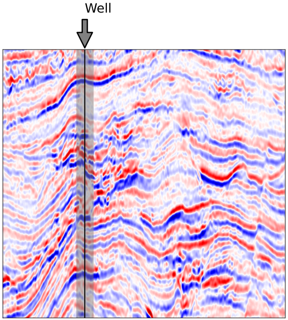


(c) Well log data c_{well}.

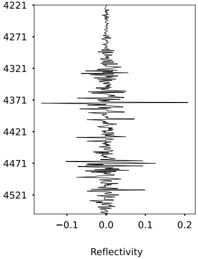
Seismic data acquisition and processing



(a) Data acquisition.

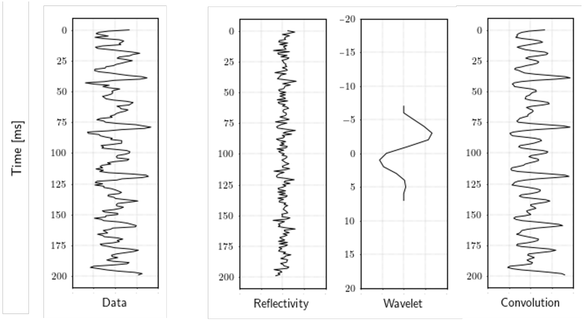


(b) Seismic data $\boldsymbol{D}$



(c) Well log data $\boldsymbol{c}_{\text{well}}$.

Model assumptions



1D convolutional model for column j:

$$\boldsymbol{d}_j = \boldsymbol{w} \star \boldsymbol{c}_j + \boldsymbol{e}_j, \quad j = 1, ..., m$$

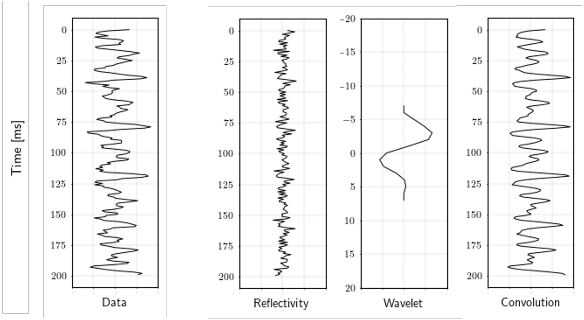
- Denote the data, reflectivity, and noise for all columns on the image by $\boldsymbol{D}$, $\boldsymbol{C}$, and $\boldsymbol{E}$.
- Denote their vectorized versions as $\boldsymbol{d} = \text{vec}(\boldsymbol{D})$, $\boldsymbol{c} = \text{vec}(\boldsymbol{C})$, and $\boldsymbol{e} = \text{vec}(\boldsymbol{E})$.
- The 1D convolutional model for the $n \times m$ image is

$$\boldsymbol{d} = \boldsymbol{W}\boldsymbol{c} + \boldsymbol{e}.$$

Assumptions:

- Same wavelet acts on each column.
- Stationary $\boldsymbol{C}$ and $\boldsymbol{E}$.

Model assumptions



1D convolutional model for column j:

$$\boldsymbol{d}_j = \boldsymbol{w} \star \boldsymbol{c}_j + \boldsymbol{e}_j, \quad j = 1, ..., m$$

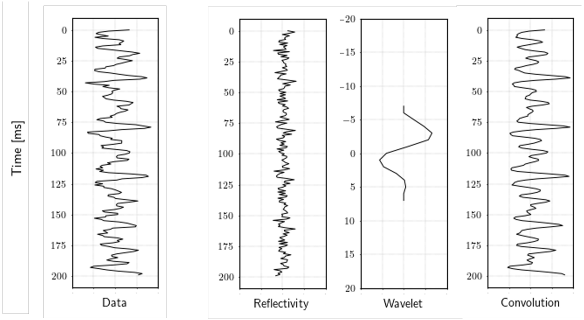
- Denote the data, reflectivity, and noise for all columns on the image by $\boldsymbol{D}$, $\boldsymbol{C}$, and $\boldsymbol{E}$.
- Denote their vectorized versions as $\boldsymbol{d} = \text{vec}(\boldsymbol{D})$, $\boldsymbol{c} = \text{vec}(\boldsymbol{C})$, and $\boldsymbol{e} = \text{vec}(\boldsymbol{E})$.
- The 1D convolutional model for the $n \times m$ image is

$$\boldsymbol{d} = \boldsymbol{W}\boldsymbol{c} + \boldsymbol{e}.$$

Assumptions:

- Same wavelet acts on each column.
- Stationary $\boldsymbol{C}$ and $\boldsymbol{E}$.

Model assumptions



1D convolutional model for column j:

$$\boldsymbol{d}_j = \boldsymbol{w} \star \boldsymbol{c}_j + \boldsymbol{e}_j, \quad j = 1, \ldots, m$$

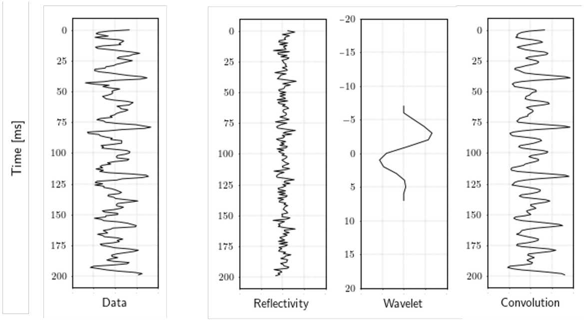
- Denote the data, reflectivity, and noise for all columns on the image by $\boldsymbol{D}$, $\boldsymbol{C}$, and $\boldsymbol{E}$.
- Denote their vectorized versions as $\boldsymbol{d} = \text{vec}(\boldsymbol{D})$, $\boldsymbol{c} = \text{vec}(\boldsymbol{C})$, and $\boldsymbol{e} = \text{vec}(\boldsymbol{E})$.
- The 1D convolutional model for the $n \times m$ image is

$$\boldsymbol{d} = \boldsymbol{W}\boldsymbol{c} + \boldsymbol{e}.$$

Assumptions:

- Same wavelet acts on each column.
- Stationary $\boldsymbol{C}$ and $\boldsymbol{E}$.

Model assumptions



1D convolutional model for column j:

$$\boldsymbol{d}_j = \boldsymbol{w} \star \boldsymbol{c}_j + \boldsymbol{e}_j, \quad j = 1, \ldots, m$$

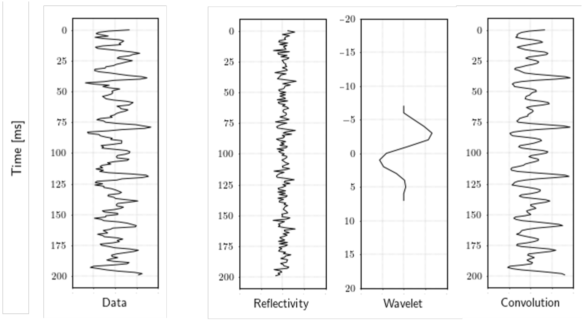
- Denote the data, reflectivity, and noise for all columns on the image by $\boldsymbol{D}$, $\boldsymbol{C}$, and $\boldsymbol{E}$.
- Denote their vectorized versions as $\boldsymbol{d} = \text{vec}(\boldsymbol{D})$, $\boldsymbol{c} = \text{vec}(\boldsymbol{C})$, and $\boldsymbol{e} = \text{vec}(\boldsymbol{E})$.
- The 1D convolutional model for the $n \times m$ image is

$$\boldsymbol{d} = \boldsymbol{W}\boldsymbol{c} + \boldsymbol{e}.$$

Assumptions:

- Same wavelet acts on each column.
- Stationary $\boldsymbol{C}$ and $\boldsymbol{E}$.

Model assumptions



1D convolutional model for column j:

$$\boldsymbol{d}_j = \boldsymbol{w} \star \boldsymbol{c}_j + \boldsymbol{e}_j, \quad j = 1, \ldots, m$$

- Denote the data, reflectivity, and noise for all columns on the image by $\boldsymbol{D}$, $\boldsymbol{C}$, and $\boldsymbol{E}$.
- Denote their vectorized versions as $\boldsymbol{d} = \text{vec}(\boldsymbol{D})$, $\boldsymbol{c} = \text{vec}(\boldsymbol{C})$, and $\boldsymbol{e} = \text{vec}(\boldsymbol{E})$.
- The 1D convolutional model for the $n \times m$ image is

$$\boldsymbol{d} = \boldsymbol{W}\boldsymbol{c} + \boldsymbol{e}.$$

Assumptions:

- Same wavelet acts on each column.
- Stationary $\boldsymbol{C}$ and $\boldsymbol{E}$.

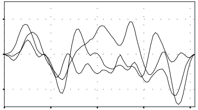
The probabilistic model (1)

- Represent the wavelet in the time domain as the sequence with k elements

$$\boldsymbol{w} = \{w_1, \ldots, w_k\}.$$

- Define the $2 \times k$ constraint matrix $\boldsymbol{A}_w$ that forces the endpoints of $\boldsymbol{w}$ to be zero.
- Model the constrained sequence as a Gaussian process:

$$\boldsymbol{w}|\sigma_w^2 \sim N_k(0, \sigma_w^2 \boldsymbol{R}_w), \quad \boldsymbol{w}^\star = \boldsymbol{w}|\boldsymbol{A}_w \boldsymbol{w} = 0, \quad \rightarrow \boldsymbol{w}^\star|\sigma_w^2 \sim N_k(0, \sigma_w^2 \boldsymbol{R}_w^\star)$$



Samples from (constrained) wavelet prior.

- Similarly, model the constrained reflectivity vector as a Gaussian field:

$$\boldsymbol{c}|\sigma_c^2 \sim N_{nm}(0, \sigma_c^2 \boldsymbol{R}_c), \quad \boldsymbol{c}^\star = \boldsymbol{c}|\boldsymbol{A}_c \boldsymbol{c} = \boldsymbol{c}_{\text{well}} \quad \rightarrow \boldsymbol{c}^\star|\sigma_c^2 \sim N_{nm}(\boldsymbol{\mu}_c^\star, \sigma_c^2 \boldsymbol{R}_c^\star)$$

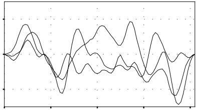
The probabilistic model (1)

- Represent the wavelet in the time domain as the sequence with k elements

$$\boldsymbol{w} = \{w_1, \ldots, w_k\}.$$

- Define the $2 \times k$ constraint matrix $\boldsymbol{A}_w$ that forces the endpoints of $\boldsymbol{w}$ to be zero.
- Model the constrained sequence as a Gaussian process:

$$\boldsymbol{w}|\sigma_w^2 \sim N_k(0, \sigma_w^2 \boldsymbol{R}_w), \quad \boldsymbol{w}^\star = \boldsymbol{w}|\boldsymbol{A}_w \boldsymbol{w} = 0, \quad \rightarrow \boldsymbol{w}^\star|\sigma_w^2 \sim N_k(0, \sigma_w^2 \boldsymbol{R}_w^\star)$$



Samples from (constrained) wavelet prior.

- Similarly, model the constrained reflectivity vector as a Gaussian field:

$$\boldsymbol{c}|\sigma_c^2 \sim N_{nm}(0, \sigma_c^2 \boldsymbol{R}_c), \quad \boldsymbol{c}^\star = \boldsymbol{c}|\boldsymbol{A}_c \boldsymbol{c} = \boldsymbol{c}_{\text{well}} \quad \rightarrow \boldsymbol{c}^\star|\sigma_c^2 \sim N_{nm}(\boldsymbol{\mu}_c^\star, \sigma_c^2 \boldsymbol{R}_c^\star)$$

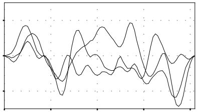
The probabilistic model (1)

- Represent the wavelet in the time domain as the sequence with k elements

$$\boldsymbol{w} = \{w_1, \ldots, w_k\}.$$

- Define the $2 \times k$ constraint matrix $\boldsymbol{A}_w$ that forces the endpoints of $\boldsymbol{w}$ to be zero.
- Model the constrained sequence as a Gaussian process:

$$\boldsymbol{w}|\sigma_w^2 \sim N_k(0, \sigma_w^2 \boldsymbol{R}_w), \quad \boldsymbol{w}^\star = \boldsymbol{w}|\boldsymbol{A}_w \boldsymbol{w} = 0, \quad \rightarrow \boldsymbol{w}^\star|\sigma_w^2 \sim N_k(0, \sigma_w^2 \boldsymbol{R}_w^\star)$$



Samples from (constrained) wavelet prior.

- Similarly, model the constrained reflectivity vector as a Gaussian field:

$$\boldsymbol{c}|\sigma_c^2 \sim N_{nm}(0, \sigma_c^2 \boldsymbol{R}_c), \quad \boldsymbol{c}^\star = \boldsymbol{c}|\boldsymbol{A}_c \boldsymbol{c} = \boldsymbol{c}_{\text{well}} \quad \rightarrow \boldsymbol{c}^\star|\sigma_c^2 \sim N_{nm}(\boldsymbol{\mu}_c^\star, \sigma_c^2 \boldsymbol{R}_c^\star)$$

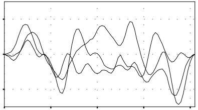
The probabilistic model (1)

- Represent the wavelet in the time domain as the sequence with k elements

$$\boldsymbol{w} = \{w_1, \ldots, w_k\}.$$

- Define the $2 \times k$ constraint matrix $\boldsymbol{A}_w$ that forces the endpoints of $\boldsymbol{w}$ to be zero.
- Model the constrained sequence as a Gaussian process:

$$\boldsymbol{w}|\sigma_w^2 \sim N_k(0, \sigma_w^2 \boldsymbol{R}_w), \quad \boldsymbol{w}^\star = \boldsymbol{w}|\boldsymbol{A}_w \boldsymbol{w} = 0, \quad \rightarrow \boldsymbol{w}^\star|\sigma_w^2 \sim N_k(0, \sigma_w^2 \boldsymbol{R}_w^\star)$$



Samples from (constrained) wavelet prior.

- Similarly, model the constrained reflectivity vector as a Gaussian field:

$$\boldsymbol{c}|\sigma_c^2 \sim N_{nm}(0, \sigma_c^2 \boldsymbol{R}_c), \quad \boldsymbol{c}^\star = \boldsymbol{c}|\boldsymbol{A}_c \boldsymbol{c} = \boldsymbol{c}_{\text{well}} \quad \rightarrow \boldsymbol{c}^\star|\sigma_c^2 \sim N_{nm}(\boldsymbol{\mu}_c^\star, \sigma_c^2 \boldsymbol{R}_c^\star)$$

The probabilistic model (2)

- Assume Gaussian observational noise and obtain the Gaussian likelihood

$$\boldsymbol{d}|\boldsymbol{c},\boldsymbol{w},\sigma_c^2,\sigma_w^2,\zeta \sim N_{nm}(\boldsymbol{W}\boldsymbol{c},\sigma_d^2\boldsymbol{R}_d), \quad \sigma_d^2 \propto \sigma_c^2\sigma_w^2\zeta, \quad \zeta^{-1} = \text{SNR}$$

- Model the spatial correlations as the product of the correlations in each direction (separability):

$$\boldsymbol{R}_d = \boldsymbol{R}_{d,h} \otimes \boldsymbol{R}_{d,v}, \quad \boldsymbol{R}_c = \boldsymbol{R}_{c,h} \otimes \boldsymbol{R}_{c,v}.$$

- Assign inverse-Gamma hyperpriors to the marginal variance parameters

$$\sigma_c^2 \sim IG(\alpha_c,\beta_c)$$
$$\sigma_w^2 \sim IG(\alpha_w,\beta_w)$$
$$\zeta \sim IG(\alpha_\zeta,\beta_\zeta)$$

The stochastic model represented by a DAG.

The probabilistic model (2)

- Assume Gaussian observational noise and obtain the Gaussian likelihood

$$\boldsymbol{d} | \boldsymbol{c}, \boldsymbol{w}, \sigma_c^2, \sigma_w^2, \zeta \sim N_{nm}(\boldsymbol{W}\boldsymbol{c}, \sigma_d^2 \boldsymbol{R}_d), \quad \sigma_d^2 \propto \sigma_c^2 \sigma_w^2 \zeta, \quad \zeta^{-1} = \text{SNR}$$

- Model the spatial correlations as the product of the correlations in each direction (separability):

$$\boldsymbol{R}_d = \boldsymbol{R}_{d,h} \otimes \boldsymbol{R}_{d,v}, \quad \boldsymbol{R}_c = \boldsymbol{R}_{c,h} \otimes \boldsymbol{R}_{c,v}.$$

- Assign inverse-Gamma hyperpriors to the marginal variance parameters

$$\sigma_c^2 \sim IG(\alpha_c, \beta_c)$$
$$\sigma_w^2 \sim IG(\alpha_w, \beta_w)$$
$$\zeta \sim IG(\alpha_\zeta, \beta_\zeta)$$

The stochastic model represented by a DAG.

The probabilistic model (2)

- Assume Gaussian observational noise and obtain the Gaussian likelihood

$$\boldsymbol{d}|\boldsymbol{c},\boldsymbol{w},\sigma_c^2,\sigma_w^2,\zeta \sim N_{nm}(\boldsymbol{W}\boldsymbol{c},\sigma_d^2\boldsymbol{R}_d), \quad \sigma_d^2 \propto \sigma_c^2\sigma_w^2\zeta, \quad \zeta^{-1} = \text{SNR}$$

- Model the spatial correlations as the product of the correlations in each direction (separability):

$$\boldsymbol{R}_d = \boldsymbol{R}_{d,h} \otimes \boldsymbol{R}_{d,v}, \quad \boldsymbol{R}_c = \boldsymbol{R}_{c,h} \otimes \boldsymbol{R}_{c,v}.$$

- Assign inverse-Gamma hyperpriors to the marginal variance parameters

$$\sigma_c^2 \sim IG(\alpha_c,\beta_c)$$
$$\sigma_w^2 \sim IG(\alpha_w,\beta_w)$$
$$\zeta \sim IG(\alpha_\zeta,\beta_\zeta)$$

The stochastic model represented by a DAG.

The probabilistic model (2)

- Assume Gaussian observational noise and obtain the Gaussian likelihood

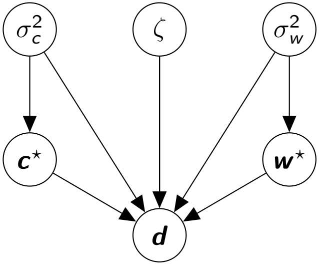
$$\boldsymbol{d}|\boldsymbol{c},\boldsymbol{w},\sigma_c^2,\sigma_w^2,\zeta \sim N_{nm}(\boldsymbol{Wc},\sigma_d^2\boldsymbol{R}_d), \quad \sigma_d^2 \propto \sigma_c^2\sigma_w^2\zeta, \quad \zeta^{-1} = \text{SNR}$$

- Model the spatial correlations as the product of the correlations in each direction (separability):

$$\boldsymbol{R}_d = \boldsymbol{R}_{d,h} \otimes \boldsymbol{R}_{d,v}, \quad \boldsymbol{R}_c = \boldsymbol{R}_{c,h} \otimes \boldsymbol{R}_{c,v}.$$

- Assign inverse-Gamma hyperpriors to the marginal variance parameters

$$\sigma_c^2 \sim IG(\alpha_c,\beta_c)$$
$$\sigma_w^2 \sim IG(\alpha_w,\beta_w)$$
$$\zeta \sim IG(\alpha_\zeta,\beta_\zeta)$$



The stochastic model represented by a DAG.

Posterior distribution of the model

- Denote: data $\boldsymbol{y} = (\boldsymbol{d}, \boldsymbol{c}_{well})$ and unknown parameters $\boldsymbol{\theta} = (\boldsymbol{w}^\star, \boldsymbol{c}^\star, \sigma_w^2, \sigma_c^2, \zeta)$.
- Use Bayes' rule

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{p(\boldsymbol{\theta}, \boldsymbol{y})}{p(\boldsymbol{y})} \propto p(\boldsymbol{y}, \boldsymbol{\theta})$$

to find the joint distribution

$$p(\boldsymbol{c}^\star, \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta|\boldsymbol{d}) = p(\boldsymbol{d}|\boldsymbol{c}^\star, \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta)p(\boldsymbol{c}^\star|\sigma_c^2)p(\boldsymbol{w}^\star|\sigma_w^2)p(\sigma_c^2)p(\sigma_w^2)p(\zeta).$$

- Conjugate distributions in the model (Gaussian-Gaussian and Gaussian-inverse-Gamma) let us write a Gibbs sampler.

Posterior distribution of the model

- Denote: data $\boldsymbol{y} = (\boldsymbol{d}, \boldsymbol{c}_{well})$ and unknown parameters $\boldsymbol{\theta} = (\boldsymbol{w}^\star, \boldsymbol{c}^\star, \sigma_w^2, \sigma_c^2, \zeta)$.
- Use Bayes' rule

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{p(\boldsymbol{\theta}, \boldsymbol{y})}{p(\boldsymbol{y})} \propto p(\boldsymbol{y}, \boldsymbol{\theta})$$

  to find the joint distribution

$$p(\boldsymbol{c}^\star, \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta|\boldsymbol{d}) = p(\boldsymbol{d}|\boldsymbol{c}^\star, \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta)p(\boldsymbol{c}^\star|\sigma_c^2)p(\boldsymbol{w}^\star|\sigma_w^2)p(\sigma_c^2)p(\sigma_w^2)p(\zeta).$$

- Conjugate distributions in the model (Gaussian-Gaussian and Gaussian-inverse-Gamma) let us write a Gibbs sampler.

Posterior distribution of the model

- Denote: data $\boldsymbol{y} = (\boldsymbol{d}, \boldsymbol{c}_{well})$ and unknown parameters $\boldsymbol{\theta} = (\boldsymbol{w}^\star, \boldsymbol{c}^\star, \sigma_w^2, \sigma_c^2, \zeta)$.
- Use Bayes' rule

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{p(\boldsymbol{\theta}, \boldsymbol{y})}{p(\boldsymbol{y})} \propto p(\boldsymbol{y}, \boldsymbol{\theta})$$

  to find the joint distribution

$$p(\boldsymbol{c}^\star, \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta|\boldsymbol{d}) = p(\boldsymbol{d}|\boldsymbol{c}^\star, \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta)p(\boldsymbol{c}^\star|\sigma_c^2)p(\boldsymbol{w}^\star|\sigma_w^2)p(\sigma_c^2)p(\sigma_w^2)p(\zeta).$$

- Conjugate distributions in the model (Gaussian-Gaussian and Gaussian-inverse-Gamma) let us write a Gibbs sampler.

Gibbs sampler for joint reflectivity and wavelet estimation

Algorithm Gibbs sampler

Input: Seismic data $\boldsymbol{d}$, well log $\boldsymbol{c}_{\text{well}}$, priors, initial values
Output: Samples from posterior $p(\boldsymbol{c}^*, \boldsymbol{w}^*, \sigma_c^2, \sigma_w^2, \zeta \mid \boldsymbol{d})$
Initialize $\boldsymbol{c}^{(0)}, \boldsymbol{w}^{(0)}, (\sigma_c^2)^{(0)}, (\sigma_w^2)^{(0)}, \zeta^{(0)}$
for $t = 1, 2, \dots, T$ **do**
- **Update reflectivity:** Sample $\boldsymbol{c}^{(t)} \sim p(\boldsymbol{c}^* \mid \boldsymbol{w}^{(t-1)}, (\sigma_c^2)^{(t-1)}, (\sigma_w^2)^{(t-1)}, \zeta^{(t-1)}, \boldsymbol{d})$
- **Update wavelet:** Sample $\boldsymbol{w}^{(t)} \sim p(\boldsymbol{w}^* \mid \boldsymbol{c}^{(t)}, (\sigma_c^2)^{(t-1)}, (\sigma_w^2)^{(t-1)}, \zeta^{(t-1)}, \boldsymbol{d})$
- **Update reflectivity variance:** Sample $(\sigma_c^2)^{(t)} \sim p(\sigma_c^2 \mid \boldsymbol{c}^{(t)}, \boldsymbol{w}^{(t)}, (\sigma_w^2)^{(t-1)}, \zeta^{(t-1)}, \boldsymbol{d})$
- **Update wavelet variance:** Sample $(\sigma_w^2)^{(t)} \sim p(\sigma_w^2 \mid \boldsymbol{c}^{(t)}, \boldsymbol{w}^{(t)}, (\sigma_c^2)^{(t)}, \zeta^{(t-1)}, \boldsymbol{d})$
- **Update SNR:** Sample $\zeta^{(t)} \sim p(\zeta \mid \boldsymbol{c}^{(t)}, \boldsymbol{w}^{(t)}, (\sigma_c^2)^{(t)}, (\sigma_w^2)^{(t)}, \boldsymbol{d})$

Return: $\{\boldsymbol{c}^{(t)}, \boldsymbol{w}^{(t)}, (\sigma_c^2)^{(t)}, (\sigma_w^2)^{(t)}, \zeta^{(t)}\}_{t=B+1}^{T}$ after burn-in B

The problem is that the full conditionals $p(\theta_j|\boldsymbol{\theta}_{-j}, \boldsymbol{d})$ involve operations with computational complexity $\mathcal{O}(l^3)$, $l = nm$.

Gibbs sampler for joint reflectivity and wavelet estimation

Algorithm Gibbs sampler

Input: Seismic data $\boldsymbol{d}$, well log $\boldsymbol{c}_{\text{well}}$, priors, initial values
Output: Samples from posterior $p(\boldsymbol{c}^*, \boldsymbol{w}^*, \sigma_c^2, \sigma_w^2, \zeta \mid \boldsymbol{d})$
Initialize $\boldsymbol{c}^{(0)}, \boldsymbol{w}^{(0)}, (\sigma_c^2)^{(0)}, (\sigma_w^2)^{(0)}, \zeta^{(0)}$
for $t = 1, 2, \ldots, T$ **do**
- **Update reflectivity:** Sample $\boldsymbol{c}^{(t)} \sim p(\boldsymbol{c}^* \mid \boldsymbol{w}^{(t-1)}, (\sigma_c^2)^{(t-1)}, (\sigma_w^2)^{(t-1)}, \zeta^{(t-1)}, \boldsymbol{d})$
- **Update wavelet:** Sample $\boldsymbol{w}^{(t)} \sim p(\boldsymbol{w}^* \mid \boldsymbol{c}^{(t)}, (\sigma_c^2)^{(t-1)}, (\sigma_w^2)^{(t-1)}, \zeta^{(t-1)}, \boldsymbol{d})$
- **Update reflectivity variance:** Sample $(\sigma_c^2)^{(t)} \sim p(\sigma_c^2 \mid \boldsymbol{c}^{(t)}, \boldsymbol{w}^{(t)}, (\sigma_w^2)^{(t-1)}, \zeta^{(t-1)}, \boldsymbol{d})$
- **Update wavelet variance:** Sample $(\sigma_w^2)^{(t)} \sim p(\sigma_w^2 \mid \boldsymbol{c}^{(t)}, \boldsymbol{w}^{(t)}, (\sigma_c^2)^{(t)}, \zeta^{(t-1)}, \boldsymbol{d})$
- **Update SNR:** Sample $\zeta^{(t)} \sim p(\zeta \mid \boldsymbol{c}^{(t)}, \boldsymbol{w}^{(t)}, (\sigma_c^2)^{(t)}, (\sigma_w^2)^{(t)}, \boldsymbol{d})$

Return: $\{\boldsymbol{c}^{(t)}, \boldsymbol{w}^{(t)}, (\sigma_c^2)^{(t)}, (\sigma_w^2)^{(t)}, \zeta^{(t)}\}_{t=B+1}^{T}$ after burn-in B

The problem is that the full conditionals $p(\theta_j|\boldsymbol{\theta}_{-j}, \boldsymbol{d})$ involve operations with computational complexity $\mathcal{O}(l^3)$, $l = nm$.

A possible solution

- Matrix multiplication, inversion, and decomposition for arbitrary ss matrices scale $\sim$ cubic with s.
- For matrices with special structures we have efficient multiplication/inversion algorithms.
- For example, take the circulant matrix

$$\underline{\boldsymbol{R}} = \mathrm{circ}(\boldsymbol{r}) = \begin{pmatrix} r_0 & r_1 & r_2 & \cdots & r_{s-1} \\ r_{s-1} & r_0 & r_1 & \cdots & r_{s-2} \\ r_{s-2} & r_{s-1} & r_0 & \cdots & r_{s-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_1 & r_2 & r_3 & \cdots & r_0 \end{pmatrix}, \quad \boldsymbol{r} = (r_0, \cdots, r_{s-1})^T.$$

  - We can represent it by its base $\boldsymbol{r}$.
  - We can find its eigenvalues with the DFT in $\mathcal{O}(s \log(s))$.
  - Multiplication/inversion/decomposition are eigenvalue-based operations.

Can we make our correlation matrices circulant?

A possible solution

- Matrix multiplication, inversion, and decomposition for arbitrary *ss* matrices scale $\sim$ cubic with s.
- For matrices with special structures we have efficient multiplication/inversion algorithms.
- For example, take the circulant matrix

$$\underline{\boldsymbol{R}} = \text{circ}(\boldsymbol{r}) = \begin{pmatrix} r_0 & r_1 & r_2 & \cdots & r_{s-1} \\ r_{s-1} & r_0 & r_1 & \cdots & r_{s-2} \\ r_{s-2} & r_{s-1} & r_0 & \cdots & r_{s-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_1 & r_2 & r_3 & \cdots & r_0 \end{pmatrix}, \quad \boldsymbol{r} = (r_0, \cdots, r_{s-1})^T.$$

  - We can represent it by its base $\boldsymbol{r}$.
  - We can find its eigenvalues with the DFT in $\mathcal{O}(s \log(s))$.
  - Multiplication/inversion/decomposition are eigenvalue-based operations.

Can we make our correlation matrices circulant?

A possible solution

- Matrix multiplication, inversion, and decomposition for arbitrary *ss* matrices scale $\sim$ cubic with s.
- For matrices with special structures we have efficient multiplication/inversion algorithms.
- For example, take the circulant matrix

$$\underline{\boldsymbol{R}} = \text{circ}(\boldsymbol{r}) = \begin{pmatrix} r_0 & r_1 & r_2 & \cdots & r_{s-1} \\ r_{s-1} & r_0 & r_1 & \cdots & r_{s-2} \\ r_{s-2} & r_{s-1} & r_0 & \cdots & r_{s-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_1 & r_2 & r_3 & \cdots & r_0 \end{pmatrix}, \quad \boldsymbol{r} = (r_0, \cdots, r_{s-1})^T.$$

  - We can represent it by its base $\boldsymbol{r}$.
  - We can find its eigenvalues with the DFT in $\mathcal{O}(s \log(s))$.
  - Multiplication/inversion/decomposition are eigenvalue-based operations.

Can we make our correlation matrices circulant?

A possible solution

- Matrix multiplication, inversion, and decomposition for arbitrary ss matrices scale $\sim$ cubic with s.
- For matrices with special structures we have efficient multiplication/inversion algorithms.
- For example, take the circulant matrix

$$\underline{\boldsymbol{R}} = \text{circ}(\boldsymbol{r}) = \begin{pmatrix} r_0 & r_1 & r_2 & \cdots & r_{s-1} \\ r_{s-1} & r_0 & r_1 & \cdots & r_{s-2} \\ r_{s-2} & r_{s-1} & r_0 & \cdots & r_{s-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_1 & r_2 & r_3 & \cdots & r_0 \end{pmatrix}, \quad \boldsymbol{r} = (r_0, \cdots, r_{s-1})^T.$$

  - We can represent it by its base $\boldsymbol{r}$.
  - We can find its eigenvalues with the DFT in $\mathcal{O}(s\log(s))$.
  - Multiplication/inversion/decomposition are eigenvalue-based operations.

Can we make our correlation matrices circulant?

A possible solution

- Matrix multiplication, inversion, and decomposition for arbitrary *ss* matrices scale $\sim$ cubic with s.
- For matrices with special structures we have efficient multiplication/inversion algorithms.
- For example, take the circulant matrix

$$\underline{\boldsymbol{R}} = \operatorname{circ}(\boldsymbol{r}) = \begin{pmatrix} r_0 & r_1 & r_2 & \cdots & r_{s-1} \\ r_{s-1} & r_0 & r_1 & \cdots & r_{s-2} \\ r_{s-2} & r_{s-1} & r_0 & \cdots & r_{s-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_1 & r_2 & r_3 & \cdots & r_0 \end{pmatrix}, \quad \boldsymbol{r} = (r_0, \cdots, r_{s-1})^T.$$

  - We can represent it by its base $\boldsymbol{r}$.
  - We can find its eigenvalues with the DFT in $\mathcal{O}(s \log(s))$.
  - Multiplication/inversion/decomposition are eigenvalue-based operations.

Can we make our correlation matrices circulant?

A possible solution

- Matrix multiplication, inversion, and decomposition for arbitrary *ss* matrices scale $\sim$ cubic with s.
- For matrices with special structures we have efficient multiplication/inversion algorithms.
- For example, take the circulant matrix

$$\underline{\boldsymbol{R}} = \mathrm{circ}(\boldsymbol{r}) = \begin{pmatrix} r_0 & r_1 & r_2 & \cdots & r_{s-1} \\ r_{s-1} & r_0 & r_1 & \cdots & r_{s-2} \\ r_{s-2} & r_{s-1} & r_0 & \cdots & r_{s-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_1 & r_2 & r_3 & \cdots & r_0 \end{pmatrix}, \quad \boldsymbol{r} = (r_0, \cdots, r_{s-1})^T.$$

  - We can represent it by its base $\boldsymbol{r}$.
  - We can find its eigenvalues with the DFT in $\mathcal{O}(s \log(s))$.
  - Multiplication/inversion/decomposition are eigenvalue-based operations.

Can we make our correlation matrices circulant?

A possible solution

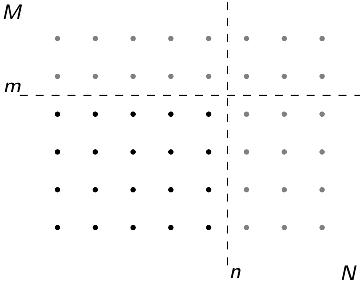
- Matrix multiplication, inversion, and decomposition for arbitrary *ss* matrices scale $\sim$ cubic with s.
- For matrices with special structures we have efficient multiplication/inversion algorithms.
- For example, take the circulant matrix

$$\underline{\boldsymbol{R}} = \mathrm{circ}(\boldsymbol{r}) = \begin{pmatrix} r_0 & r_1 & r_2 & \cdots & r_{s-1} \\ r_{s-1} & r_0 & r_1 & \cdots & r_{s-2} \\ r_{s-2} & r_{s-1} & r_0 & \cdots & r_{s-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_1 & r_2 & r_3 & \cdots & r_0 \end{pmatrix}, \quad \boldsymbol{r} = (r_0, \cdots, r_{s-1})^T.$$

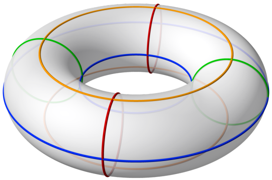
  - We can represent it by its base $\boldsymbol{r}$.
  - We can find its eigenvalues with the DFT in $\mathcal{O}(s\log(s))$.
  - Multiplication/inversion/decomposition are eigenvalue-based operations.

Can we make our correlation matrices circulant?

Scalability



(a) The extended lattice.



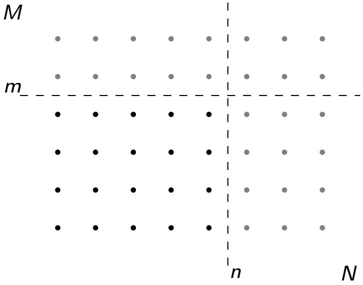
(b) The cyclic lattice.

- Separability on the cyclic lattice gives circulant or block-circulant with circulant blocks correlation matrices:

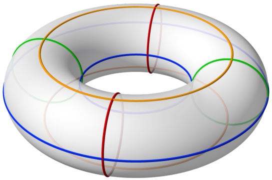
$$\underline{\boldsymbol{R}}_{w,v}, \quad \underline{\underline{\boldsymbol{R}}}_{c} = \underline{\boldsymbol{R}}_{c,h} \otimes \underline{\boldsymbol{R}}_{c,v}, \quad \underline{\underline{\boldsymbol{R}}}_{d} = \underline{\boldsymbol{R}}_{d,h} \otimes \underline{\boldsymbol{R}}_{d,v},$$

- Convolutional matrix $\underline{\underline{\boldsymbol{W}}}$ is BCCB.

Scalability



(a) The extended lattice.



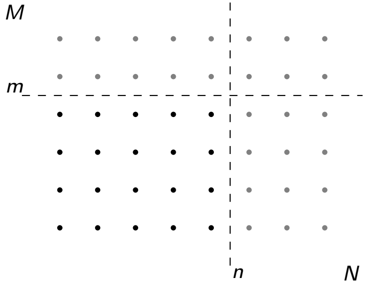
(b) The cyclic lattice.

- Separability on the cyclic lattice gives circulant or block-circulant with circulant blocks correlation matrices:

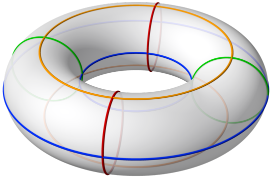
$$\underline{\boldsymbol{R}}_{w,v}, \quad \underline{\underline{\boldsymbol{R}}}_c = \underline{\boldsymbol{R}}_{c,h} \otimes \underline{\boldsymbol{R}}_{c,v}, \quad \underline{\underline{\boldsymbol{R}}}_d = \underline{\boldsymbol{R}}_{d,h} \otimes \underline{\boldsymbol{R}}_{d,v},$$

- Convolutional matrix $\underline{\underline{\boldsymbol{W}}}$ is BCCB.

Scalability



(a) The extended lattice.



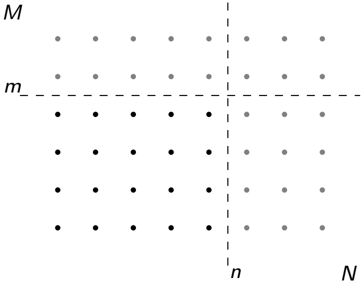
(b) The cyclic lattice.

- Separability on the cyclic lattice gives circulant or block-circulant with circulant blocks correlation matrices:

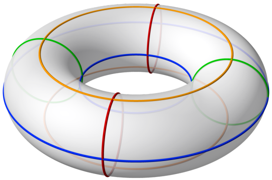
$$\underline{R}_{w,v}, \quad \underline{\underline{R}}_c = \underline{R}_{c,h} \otimes \underline{R}_{c,v}, \quad \underline{\underline{R}}_d = \underline{R}_{d,h} \otimes \underline{R}_{d,v},$$

- Convolutional matrix $\underline{\underline{W}}$ is BCCB.

Scalability



(a) The extended lattice.



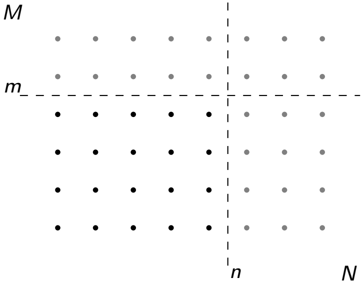
(b) The cyclic lattice.

- Separability on the cyclic lattice gives circulant or block-circulant with circulant blocks correlation matrices:

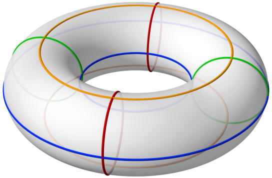
$$\underline{\boldsymbol{R}}_{w,v}, \quad \underline{\underline{\boldsymbol{R}}}_{c} = \underline{\boldsymbol{R}}_{c,h} \otimes \underline{\boldsymbol{R}}_{c,v}, \quad \underline{\underline{\boldsymbol{R}}}_{d} = \underline{\boldsymbol{R}}_{d,h} \otimes \underline{\boldsymbol{R}}_{d,v},$$

- Convolutional matrix $\underline{\underline{\boldsymbol{W}}}$ is BCCB.

Scalability



(a) The extended lattice.



(b) The cyclic lattice.

- Separability on the cyclic lattice gives circulant or block-circulant with circulant blocks correlation matrices:

$$\underline{\boldsymbol{R}}_{w,v}, \quad \underline{\underline{\boldsymbol{R}}}_{c} = \underline{\boldsymbol{R}}_{c,h} \otimes \underline{\boldsymbol{R}}_{c,v}, \quad \underline{\underline{\boldsymbol{R}}}_{d} = \underline{\boldsymbol{R}}_{d,h} \otimes \underline{\boldsymbol{R}}_{d,v},$$

- Convolutional matrix $\underline{\underline{\boldsymbol{W}}}$ is BCCB.

Gibbs sampling on the cyclic lattice

Algorithm Gibbs sampler on the cyclic lattice

Input: $\cdots$
Output: Samples from posterior $p(\boldsymbol{c}^*, \boldsymbol{w}^*, \sigma_c^2, \sigma_w^2, \zeta, \boldsymbol{d}_{\text{aux}} \mid \boldsymbol{d})$
Initialize $\cdots$
for $t = 1, 2, \ldots, T$ **do**
- **Update reflectivity**
- **Update wavelet**
- **Update reflectivity variance**
- **Update wavelet variance**
- **Update SNR**
- **Update auxiliary seismic data $\boldsymbol{d}_{\text{aux}}$**

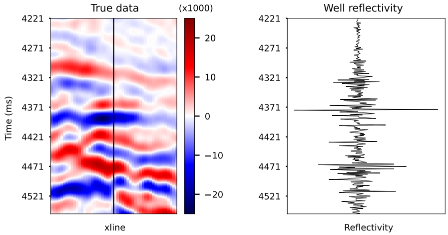
Return: $\{\boldsymbol{c}^{(t)}, \boldsymbol{w}^{(t)}, (\sigma_c^2)^{(t)}, (\sigma_w^2)^{(t)}, \zeta^{(t)}, \boldsymbol{d}_{\text{aux}}^{(t)}\}_{t=B+1}^{T}$ after burn-in B

Some advantages:

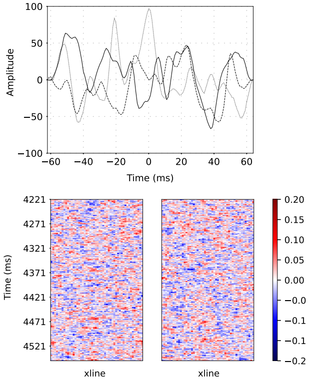
- Store only bases of $\underline{\boldsymbol{R}}_{\cdot,h}$, $\underline{\boldsymbol{R}}_{\cdot,v}$, $\underline{\underline{\boldsymbol{R}}}$.
- Use DFT to compute full conditional parameters and sample from Gaussians in $\mathcal{O}(S\log(S))$, $S = NM$.

Real application: Gas reservoir in offshore Egypt

We extract a 330 × 50s window from the AVO data for a fixed inline, centered around the well.

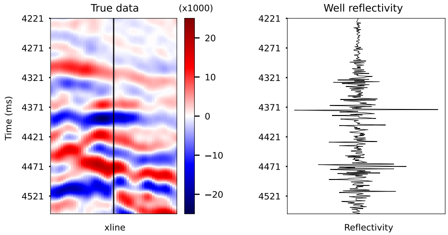


We choose high correlation in the wavelet prior and moderate correlation in the reflectivity prior. Some prior samples:

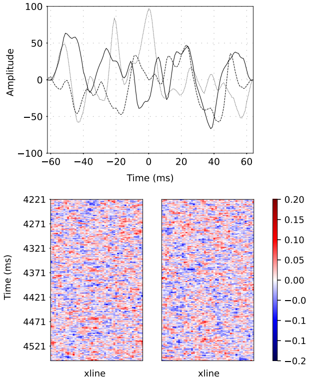


Real application: Gas reservoir in offshore Egypt

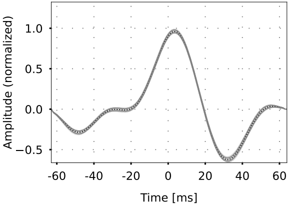
We extract a 330 × 50s window from the AVO data for a fixed inline, centered around the well.



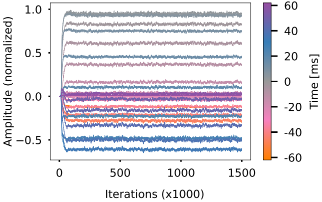
We choose high correlation in the wavelet prior and moderate correlation in the reflectivity prior. Some prior samples:



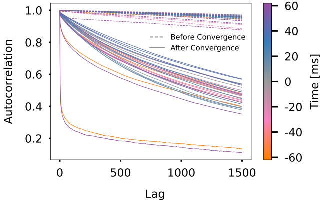
Wavelet posterior exploration



(a) Posterior samples



(b) Traceplots

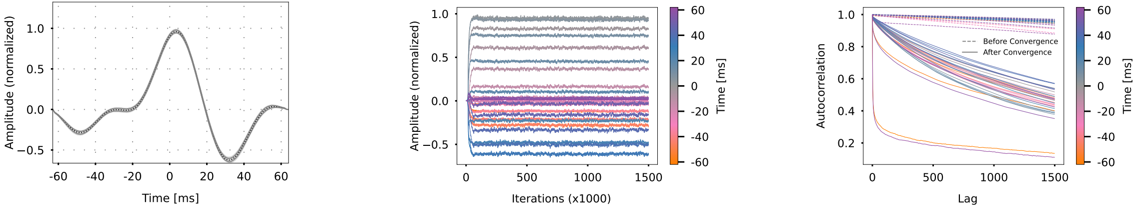


(c) Sample autocorrelation

MCMC sampling results for wavelet posterior distribution.

Question: Can we sample more efficiently from the same posterior distribution?

Wavelet posterior exploration

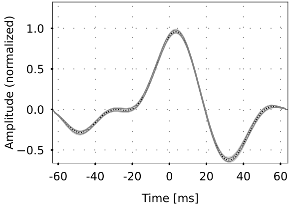


(a) Posterior samples (b) Traceplots (c) Sample autocorrelation

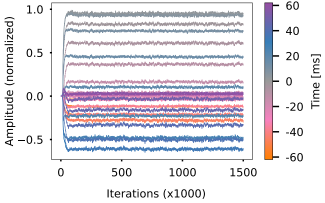
MCMC sampling results for wavelet posterior distribution.

Question: Can we sample more efficiently from the same posterior distribution?

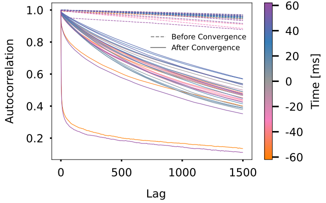
Wavelet posterior exploration



(a) Posterior samples



(b) Traceplots

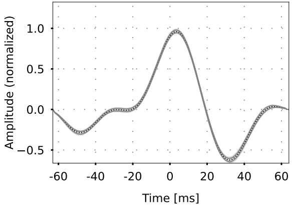


(c) Sample autocorrelation

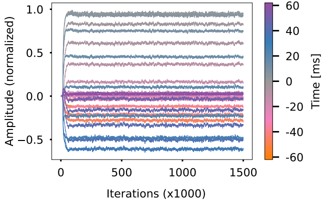
MCMC sampling results for wavelet posterior distribution.

Question: Can we sample more efficiently from the same posterior distribution?

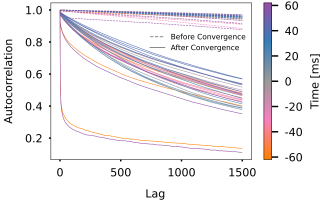
Wavelet posterior exploration



(a) Posterior samples



(b) Traceplots



(c) Sample autocorrelation

MCMC sampling results for wavelet posterior distribution.

Question: Can we sample more efficiently from the same posterior distribution?

Marginal seismic wavelet estimation with Hamiltonian Monte Carlo

- The joint reflectivity and wavelet estimation is mathematically equivalent to sampling from the joint $p(\boldsymbol{w}^\star, \boldsymbol{c}^\star | \boldsymbol{d}, \sigma_c^2, \sigma_c^2, \zeta)$.
- We instead want a marginal wavelet estimation, i.e. to sample from $p(\boldsymbol{w}^\star | \boldsymbol{d}, \sigma_c^2, \sigma_c^2, \zeta)$.
- This new marginal posterior looks like

$$p(\boldsymbol{w}^\star | \boldsymbol{d}, \sigma_c^2, \sigma_w^2, \zeta, \boldsymbol{d}_{\text{aux}}) \propto p(\boldsymbol{w}^\star | \sigma_w^2) p(\boldsymbol{d} | \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta, \boldsymbol{d}_{\text{aux}}),$$

with

$$p(\boldsymbol{d} | \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta, \boldsymbol{d}_{\text{aux}}) = \int p(\boldsymbol{d}, \boldsymbol{c}^\star | \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta) d\boldsymbol{c}^\star \tag{1}$$

- This integral can be done analytically.
- Using HMC might be better because it's a gradient-based algorithm.
- Evaluating gradients and densities at each iteration is $\mathcal{O}(s^{1.5})$ on the cyclic lattice.

Marginal seismic wavelet estimation with Hamiltonian Monte Carlo

- The joint reflectivity and wavelet estimation is mathematically equivalent to sampling from the joint $p(\boldsymbol{w}^\star, \boldsymbol{c}^\star | \boldsymbol{d}, \sigma_c^2, \sigma_c^2, \zeta)$.
- We instead want a marginal wavelet estimation, i.e. to sample from $p(\boldsymbol{w}^\star | \boldsymbol{d}, \sigma_c^2, \sigma_c^2, \zeta)$.
- This new marginal posterior looks like

$$p(\boldsymbol{w}^\star | \boldsymbol{d}, \sigma_c^2, \sigma_w^2, \zeta, \boldsymbol{d}_{\text{aux}}) \propto p(\boldsymbol{w}^\star | \sigma_w^2) p(\boldsymbol{d} | \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta, \boldsymbol{d}_{\text{aux}}),$$

with

$$p(\boldsymbol{d} | \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta, \boldsymbol{d}_{\text{aux}}) = \int p(\boldsymbol{d}, \boldsymbol{c}^\star | \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta) d\boldsymbol{c}^\star \quad (1)$$

- This integral can be done analytically.
- Using HMC might be better because it's a gradient-based algorithm.
- Evaluating gradients and densities at each iteration is $\mathcal{O}(s^{1.5})$ on the cyclic lattice.

Marginal seismic wavelet estimation with Hamiltonian Monte Carlo

- The joint reflectivity and wavelet estimation is mathematically equivalent to sampling from the joint $p(\boldsymbol{w}^\star, \boldsymbol{c}^\star | \boldsymbol{d}, \sigma_c^2, \sigma_c^2, \zeta)$.
- We instead want a marginal wavelet estimation, i.e. to sample from $p(\boldsymbol{w}^\star | \boldsymbol{d}, \sigma_c^2, \sigma_c^2, \zeta)$.
- This new marginal posterior looks like

$$p(\boldsymbol{w}^\star | \boldsymbol{d}, \sigma_c^2, \sigma_w^2, \zeta, \boldsymbol{d}_{\text{aux}}) \propto p(\boldsymbol{w}^\star | \sigma_w^2) p(\boldsymbol{d} | \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta, \boldsymbol{d}_{\text{aux}}),$$

with

$$p(\boldsymbol{d} | \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta, \boldsymbol{d}_{\text{aux}}) = \int p(\boldsymbol{d}, \boldsymbol{c}^\star | \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta) d\boldsymbol{c}^\star \tag{1}$$

- This integral can be done analytically.
- Using HMC might be better because it's a gradient-based algorithm.
- Evaluating gradients and densities at each iteration is $\mathcal{O}(s^{1.5})$ on the cyclic lattice.

Marginal seismic wavelet estimation with Hamiltonian Monte Carlo

- The joint reflectivity and wavelet estimation is mathematically equivalent to sampling from the joint $p(\boldsymbol{w}^\star, \boldsymbol{c}^\star | \boldsymbol{d}, \sigma_c^2, \sigma_c^2, \zeta)$.
- We instead want a marginal wavelet estimation, i.e. to sample from $p(\boldsymbol{w}^\star | \boldsymbol{d}, \sigma_c^2, \sigma_c^2, \zeta)$.
- This new marginal posterior looks like

$$p(\boldsymbol{w}^\star | \boldsymbol{d}, \sigma_c^2, \sigma_w^2, \zeta, \boldsymbol{d}_{\text{aux}}) \propto p(\boldsymbol{w}^\star | \sigma_w^2) p(\boldsymbol{d} | \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta, \boldsymbol{d}_{\text{aux}}),$$

with

$$p(\boldsymbol{d} | \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta, \boldsymbol{d}_{\text{aux}}) = \int p(\boldsymbol{d}, \boldsymbol{c}^\star | \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta) d\boldsymbol{c}^\star \tag{1}$$

- This integral can be done analytically.
- Using HMC might be better because it's a gradient-based algorithm.
- Evaluating gradients and densities at each iteration is $\mathcal{O}(s^{1.5})$ on the cyclic lattice.

Marginal seismic wavelet estimation with Hamiltonian Monte Carlo

- The joint reflectivity and wavelet estimation is mathematically equivalent to sampling from the joint $p(\boldsymbol{w}^\star, \boldsymbol{c}^\star | \boldsymbol{d}, \sigma_c^2, \sigma_c^2, \zeta)$.
- We instead want a marginal wavelet estimation, i.e. to sample from $p(\boldsymbol{w}^\star | \boldsymbol{d}, \sigma_c^2, \sigma_c^2, \zeta)$.
- This new marginal posterior looks like

$$p(\boldsymbol{w}^\star | \boldsymbol{d}, \sigma_c^2, \sigma_w^2, \zeta, \boldsymbol{d}_{\text{aux}}) \propto p(\boldsymbol{w}^\star | \sigma_w^2) p(\boldsymbol{d} | \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta, \boldsymbol{d}_{\text{aux}}),$$

with

$$p(\boldsymbol{d} | \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta, \boldsymbol{d}_{\text{aux}}) = \int p(\boldsymbol{d}, \boldsymbol{c}^\star | \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta) d\boldsymbol{c}^\star \tag{1}$$

- This integral can be done analytically.
- Using HMC might be better because it's a gradient-based algorithm.
- Evaluating gradients and densities at each iteration is $\mathcal{O}(s^{1.5})$ on the cyclic lattice.

Marginal seismic wavelet estimation with Hamiltonian Monte Carlo

- The joint reflectivity and wavelet estimation is mathematically equivalent to sampling from the joint $p(\boldsymbol{w}^\star, \boldsymbol{c}^\star | \boldsymbol{d}, \sigma_c^2, \sigma_c^2, \zeta)$.
- We instead want a marginal wavelet estimation, i.e. to sample from $p(\boldsymbol{w}^\star | \boldsymbol{d}, \sigma_c^2, \sigma_c^2, \zeta)$.
- This new marginal posterior looks like

$$p(\boldsymbol{w}^\star | \boldsymbol{d}, \sigma_c^2, \sigma_w^2, \zeta, \boldsymbol{d}_{\text{aux}}) \propto p(\boldsymbol{w}^\star | \sigma_w^2) p(\boldsymbol{d} | \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta, \boldsymbol{d}_{\text{aux}}),$$

with

$$p(\boldsymbol{d} | \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta, \boldsymbol{d}_{\text{aux}}) = \int p(\boldsymbol{d}, \boldsymbol{c}^\star | \boldsymbol{w}^\star, \sigma_c^2, \sigma_w^2, \zeta) d\boldsymbol{c}^\star \tag{1}$$

- This integral can be done analytically.
- Using HMC might be better because it's a gradient-based algorithm.
- Evaluating gradients and densities at each iteration is $\mathcal{O}(s^{1.5})$ on the cyclic lattice.

Conclusions

- We proposed an approach for estimating a seismic wavelet with full UQ.
- Gibbs sampler allows joint wavelet and reflectivity estimation but is slow.
- Collapsed HMC allows efficient marginal wavelet estimation.

References

- Buland, A. and Omre, H. (2003). Bayesian wavelet estimation from seismic and well data. *Geophysics*, 68, 2000–2009.
- Buland, A. and Omre, H. (2003). Joint AVO inversion, wavelet estimation and noise-level estimation using a spatially coupled hierarchical Bayesian model. *Geophysical Prospecting*, 51(6).
- Senn, G., Walker, M. and Tjelmeland, H. (2025). Scalable Bayesian seismic wavelet estimation. *Geophysical Prospecting*, 73(5), 1635–1650.

Conclusions

- We proposed an approach for estimating a seismic wavelet with full UQ.
- Gibbs sampler allows joint wavelet and reflectivity estimation but is slow.
- Collapsed HMC allows efficient marginal wavelet estimation.

References

- Buland, A. and Omre, H. (2003). Bayesian wavelet estimation from seismic and well data. *Geophysics*, 68, 2000–2009.
- Buland, A. and Omre, H. (2003). Joint AVO inversion, wavelet estimation and noise-level estimation using a spatially coupled hierarchical Bayesian model. *Geophysical Prospecting*, 51(6).
- Senn, G., Walker, M. and Tjelmeland, H. (2025). Scalable Bayesian seismic wavelet estimation. *Geophysical Prospecting*, 73(5), 1635–1650.

Conclusions

- We proposed an approach for estimating a seismic wavelet with full UQ.
- Gibbs sampler allows joint wavelet and reflectivity estimation but is slow.
- Collapsed HMC allows efficient marginal wavelet estimation.

References

- Buland, A. and Omre, H. (2003). Bayesian wavelet estimation from seismic and well data. *Geophysics*, 68, 2000–2009.
- Buland, A. and Omre, H. (2003). Joint AVO inversion, wavelet estimation and noise-level estimation using a spatially coupled hierarchical Bayesian model. *Geophysical Prospecting*, 51(6).
- Senn, G., Walker, M. and Tjelmeland, H. (2025). Scalable Bayesian seismic wavelet estimation. *Geophysical Prospecting*, 73(5), 1635–1650.

Conclusions

- We proposed an approach for estimating a seismic wavelet with full UQ.
- Gibbs sampler allows joint wavelet and reflectivity estimation but is slow.
- Collapsed HMC allows efficient marginal wavelet estimation.

References

- Buland, A. and Omre, H. (2003). Bayesian wavelet estimation from seismic and well data. *Geophysics*, 68, 2000–2009.
- Buland, A. and Omre, H. (2003). Joint AVO inversion, wavelet estimation and noise-level estimation using a spatially coupled hierarchical Bayesian model. *Geophysical Prospecting*, 51(6).
- Senn, G., Walker, M. and Tjelmeland, H. (2025). Scalable Bayesian seismic wavelet estimation. *Geophysical Prospecting*, 73(5), 1635–1650.