Deep null space regularization for seismic inverse problems

SAIG Annual meeting

Kristian Torres

January 25, 2022

Content

Content

Introduction

SIGNAL ANALYSIS & IMAGING GROUP

We are interested in solving linear seismic inverse problems of the form

$$\mathbf{d}_\epsilon = \mathbf{L}\mathbf{m} + \epsilon,$$

- $\mathbf{d}_\epsilon \in \mathbb{R}^m$: data vector
- $\mathbf{m} \in \mathbb{R}^n$: earth model or unknown signal
- ϵ: unknown data error (the noise)
- $\mathbf{L} : \mathbb{R}^n \to \mathbb{R}^m$: linear forward operator that maps $\mathbf{m}$ to $\mathbf{d}$

Introduction

- Seismic inversion is severely ill-posed due to a non-trivial null space of the forward operator.
- Many solutions can fit the acquired data equally well.
- $\mathbf{m} = \mathbf{L}^{-1}\mathbf{d}_\epsilon$ is not possible.

Introduction

SIGNAL ANALYSIS & IMAGING GROUP

- Seismic inversion is severely ill-posed due to a non-trivial null space of the forward operator.
- Many solutions can fit the acquired data equally well.
- $\mathbf{m} = \mathbf{L}^{-1}\mathbf{d}_\epsilon$ is not possible.

A *simple* unique solution: $\mathbf{m}^* = \mathbf{L}^\dagger \mathbf{d}_\epsilon$

- Enjoys data consistency: $\mathbf{L}\mathbf{m}^* = \mathbf{d}_\epsilon$
- No assumption about the null space component $\rightarrow$ poor solution for ill-posed problems.
- We can use regularization.

Content

Learned post-processing approach

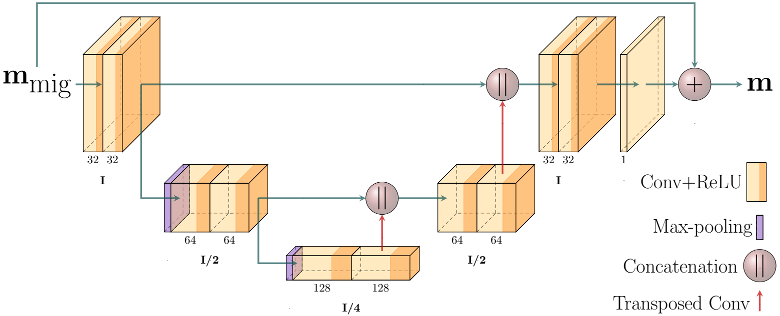
Improve an initial reconstruction $\mathbf{m}^*$ with a model-to-model mapping DNN $\Lambda_\theta(\mathbf{m}^*)$, typically by means of residual architectures (learn a perturbation, don't learn the physics):

$$\Lambda_\theta(\mathbf{m}^*) = (\mathbf{I}_n + \mathbf{N}_\theta)(\mathbf{m}^*)$$

Learned post-processing approach

Improve an initial reconstruction $\mathbf{m}^*$ with a model-to-model mapping DNN $\Lambda_\theta(\mathbf{m}^*)$, typically by means of residual architectures (learn a perturbation, don't learn the physics):

$$\Lambda_\theta(\mathbf{m}^*) = (\mathbf{I}_n + \mathbf{N}_\theta)(\mathbf{m}^*)$$



Learned post-processing approaches generally cannot preserve data consistency.

Let's assume $\mathbf{m}^* = \mathbf{L}^\dagger \mathbf{d}_\epsilon$. Then:

$$\mathbf{L}\Lambda_\theta(\mathbf{m}^*) \neq \mathbf{d}_\epsilon$$

SIGNAL ANALYSIS & IMAGING GROUP

Learned post-processing approaches generally cannot preserve data consistency.
Let's assume $\mathbf{m}^* = \mathbf{L}^\dagger \mathbf{d}_\epsilon$. Then:

$$\mathbf{L}\Lambda_\theta(\mathbf{m}^*) \neq \mathbf{d}_\epsilon$$

Null space networks

- Akin to learned post-processing approach $\Lambda_\theta(\mathbf{m}^*) = (\mathbf{I}_n + \mathbf{N}_\theta)(\mathbf{m}^*)$.
- Residual architecture with a twist: after the last weight layer, incorporate projection onto the null space P_N such that $\mathbf{L}P_N(\mathbf{m}) = \mathbf{L}\mathbf{m}_N = \mathbf{0}$. Then:

$$\Lambda_\theta(\mathbf{m}^*) = (\mathbf{I}_n + P_N \circ \mathbf{N}_\theta)(\mathbf{m}^*) \quad (1)$$

- Preserve data consistency in the sense that

$$\mathbf{L}\Lambda_\theta(\mathbf{m}^*) = \mathbf{L}(\mathbf{I}_n + P_N \circ \mathbf{N}_\theta)(\mathbf{m}^*) = \mathbf{L}\mathbf{L}^\dagger \mathbf{d}_\epsilon + \mathbf{0} = \mathbf{d}_\epsilon \quad (2)$$

SIGNAL ANALYSIS & IMAGING GROUP

Null space networks solution

$$\mathbf{m}^*_{NS} = \mathbf{L}^\dagger \mathbf{d}_\epsilon + P_N(\mathbf{N}_\theta(\mathbf{L}^\dagger \mathbf{d}_\epsilon))$$

(train $\mathbf{N}_\theta$ by minimizing error between $\mathbf{m}$ and $\mathbf{m}^*_{NS}$)

SIGNAL ANALYSIS & IMAGING GROUP

Null space networks solution

$$\mathbf{m}^*_{NS} = \mathbf{L}^\dagger \mathbf{d}_\epsilon + P_N(\mathbf{N}_\theta(\mathbf{L}^\dagger \mathbf{d}_\epsilon))$$

(train $\mathbf{N}_\theta$ by minimizing error between $\mathbf{m}$ and $\mathbf{m}^*_{NS}$)

- Enjoys global data consistency, i.e. $\mathbf{L}\mathbf{m}^*_{NS} = \mathbf{d}_\epsilon$
- Only works for the noise-free case ($\epsilon = 0$):noise may limit the ability to predict the null space component from noisy measurements.
- Only denoises in the null space (no denoising capability in the range component $\mathcal{R}(\mathbf{L}^\dagger)$)

SIGNAL ANALYSIS & IMAGING GROUP

Null space networks solution

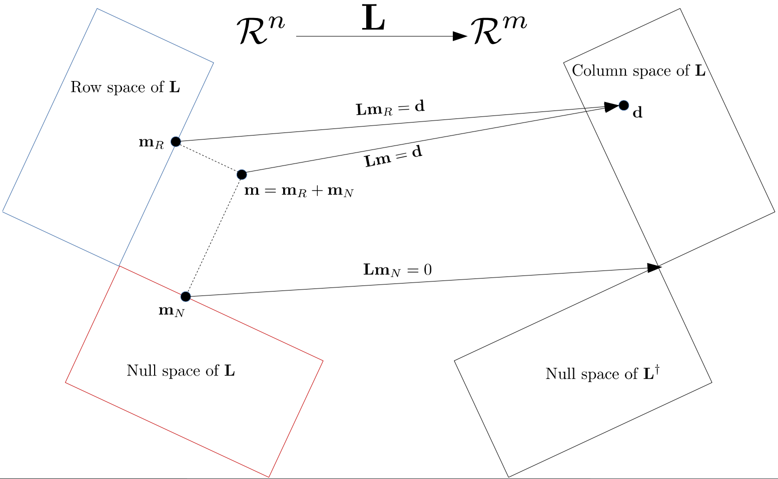
$$\mathbf{m}^*_{NS} = \mathbf{L}^\dagger \mathbf{d}_\epsilon + P_N(\mathbf{N}_\theta(\mathbf{L}^\dagger \mathbf{d}_\epsilon))$$

(train $\mathbf{N}_\theta$ by minimizing error between $\mathbf{m}$ and $\mathbf{m}^*_{NS}$)

- Enjoys global data consistency, i.e. $\mathbf{L}\mathbf{m}^*_{NS} = \mathbf{d}_\epsilon$
- Only works for the noise-free case ($\epsilon = 0$):noise may limit the ability to predict the null space component from noisy measurements.
- Only denoises in the null space (no denoising capability in the range component $\mathcal{R}(\mathbf{L}^\dagger)$)
- Deep Decomposition Learning: extends null space learning by attaching a complementary network to act as a denoiser on the range of the pseudoinverse.

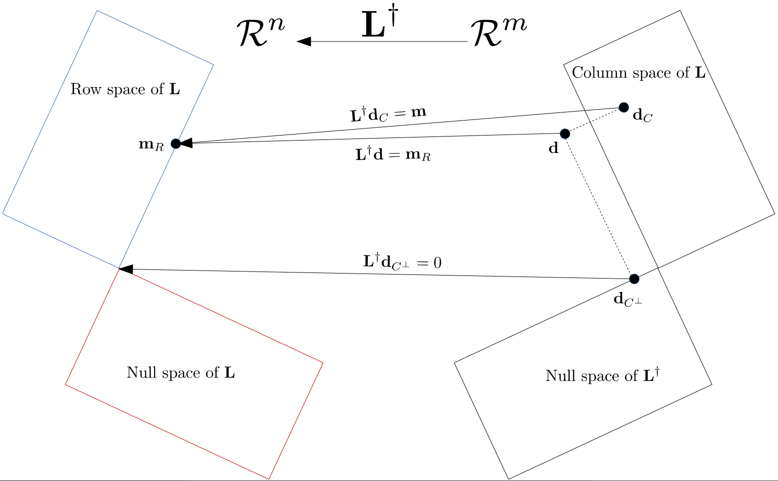
Range - Null space decomposition

$$\mathbf{m} = \mathbf{m}_R + \mathbf{m}_N = P_R(\mathbf{m}) + P_N(\mathbf{m})$$



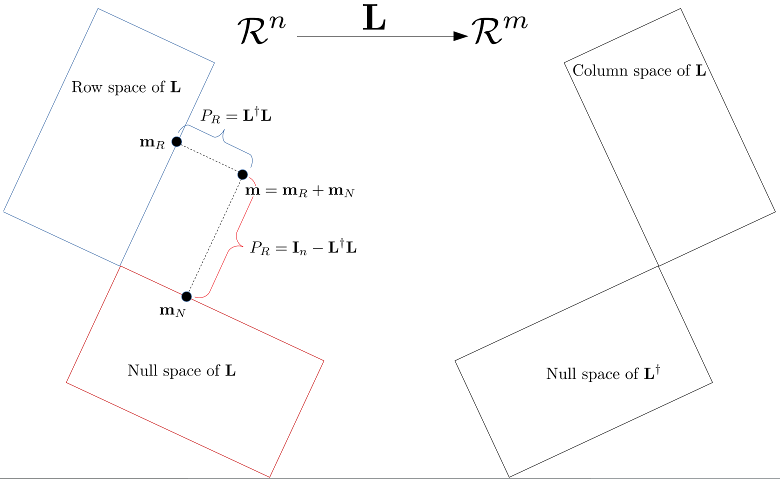
Range - Null space decomposition

$$\mathbf{m} = \mathbf{m}_R + \mathbf{m}_N = P_R(\mathbf{m}) + P_N(\mathbf{m})$$



Range - Null space decomposition

$$\mathbf{m} = \mathbf{m}_R + \mathbf{m}_N = P_R(\mathbf{m}) + P_N(\mathbf{m})$$



Range - Null space decomposition

SIGNAL ANALYSIS & IMAGING GROUP

$$\mathbf{m} = \mathbf{m}_R + \mathbf{m}_N = P_R(\mathbf{m}) + P_N(\mathbf{m})$$

By definition, these two components satisfy, respectively,

$$\mathbf{m}_R = \mathbf{L}^\dagger \mathbf{d}_\epsilon = \mathbf{L}^\dagger \mathbf{L}\mathbf{m} + \mathbf{L}^\dagger \epsilon,$$

and

$$\mathbf{L}\mathbf{m}_N = 0.$$

The two orthogonal projections are defined as:

$$P_R = \mathbf{L}^\dagger \mathbf{L},$$

and

$$P_N = \mathbf{I}_n - \mathbf{L}^\dagger \mathbf{L}.$$

$$\boxed{\mathbf{d}_\epsilon = \mathbf{L}\mathbf{m} + \epsilon}$$

Based on this fragmentation, we can express the ideal reconstruction as

$$\mathbf{m}^* = \mathbf{L}^\dagger \mathbf{d}_\epsilon - \mathbf{L}^\dagger \epsilon + \mathbf{m}_N.$$

Deep decomposition learning attempts to solve above equation with a trained estimator $\Lambda : \mathbb{R}^m \to \mathbb{R}^n$ defined as

$$\Lambda(\mathbf{d}_\epsilon; \theta_1, \theta_2) = \mathbf{L}^\dagger \mathbf{d}_\epsilon + P_R \circ \mathbf{F}_{\theta_1} \circ \mathbf{L}^\dagger \mathbf{d}_\epsilon + P_N \circ \mathbf{N}_{\theta_2} \circ (\mathbf{L}^\dagger \mathbf{d}_\epsilon + P_R \circ \mathbf{F}_{\theta_1} \circ \mathbf{L}^\dagger \mathbf{d}_\epsilon),$$

Deep decomposition learning

SIGNAL ANALYSIS & IMAGING GROUP

$$\mathbf{d}_\epsilon = \mathbf{Lm} + \epsilon$$

Based on this fragmentation of the model, we can express the ideal reconstruction as

$$\mathbf{m}^* = \mathbf{L}^\dagger \mathbf{d}_\epsilon - \mathbf{L}^\dagger \epsilon + \mathbf{m}_N.$$

Deep decomposition learning attempts to solve above equation with a trained estimator $\Lambda : \mathbb{R}^m \to \mathbb{R}^n$ defined as

$$\Lambda(\mathbf{d}_\epsilon; \theta_1, \theta_2) = \mathbf{L}^\dagger \mathbf{d}_\epsilon + P_R \circ \mathbf{F}_{\theta_1} \circ \mathbf{L}^\dagger \mathbf{d}_\epsilon + P_N \circ \mathbf{N}_{\theta_2} \circ (\mathbf{L}^\dagger \mathbf{d}_\epsilon + P_R \circ \mathbf{F}_{\theta_1} \circ \mathbf{L}^\dagger \mathbf{d}_\epsilon),$$

$$\Lambda(\mathbf{d}_\epsilon; \theta) = (\mathbf{I} + P_N \circ \mathbf{N}_\theta)(\mathbf{L}^\dagger \mathbf{d}_\epsilon) \to \text{Standard null space network}$$

Deep decomposition learning

SIGNAL ANALYSIS & IMAGING GROUP

Substituting $\mathbf{L}^{\dagger}$ by a regularized initial approximation $\mathbf{L}_k^{\dagger}$ such that:

$$\mathbf{m}^*_{\text{TSVD}} = \mathbf{L}_k^{\dagger}\mathbf{d}_\epsilon = \sum_{i=1}^{k} \frac{\mathbf{u}_i^T \mathbf{d}}{\sigma_i} \mathbf{v}_i$$

we can train the estimator $\Lambda(\mathbf{d}_\epsilon^i; \theta_1, \theta_2)$ as:

$$\arg\min_{\theta_1,\theta_2} \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{m}^i - \Lambda(\mathbf{d}_\epsilon^i; \theta_1, \theta_2)||_2^2 + \lambda_1 \sum_{i=1}^{N} ||\mathbf{L}\mathbf{F}_{\theta_1}(\mathbf{L}_k^{\dagger}\mathbf{d}_\epsilon^i) - \epsilon^i||_2^2 + \lambda_2||\theta_2||_2^2,$$

Substituting $\mathbf{L}^\dagger$ by a regularized initial approximation $\mathbf{L}_k^\dagger$ such that:

$$\mathbf{m}^*_{\mathrm{TSVD}} = \mathbf{L}_k^\dagger \mathbf{d}_\epsilon = \sum_{i=1}^{k} \frac{\mathbf{u}_i^T \mathbf{d}}{\sigma_i} \mathbf{v}_i$$

we can train the estimator $\Lambda(\mathbf{d}_\epsilon^i; \theta_1, \theta_2)$ as:

$$\arg\min_{\theta_1,\theta_2} \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{m}^i - \Lambda(\mathbf{d}_\epsilon^i; \theta_1, \theta_2)||_2^2 + \lambda_1 \sum_{i=1}^{N} ||\mathbf{L}\mathbf{F}_{\theta_1}(\mathbf{L}_k^\dagger \mathbf{d}_\epsilon^i) - \epsilon^i||_2^2 + \lambda_2 ||\theta_2||_2^2,$$

- Supervised training on a synthetic dataset $\mathcal{D} = \{(\mathbf{m}^i, \mathbf{d}_\epsilon^i)\}_{i=1}^N$ using the MSE loss

Deep decomposition learning

SIGNAL ANALYSIS & IMAGING GROUP

Substituting $\mathbf{L}^\dagger$ by a regularized initial approximation $\mathbf{L}_k^\dagger$ such that:

$$\mathbf{m}^*_{\text{TSVD}} = \mathbf{L}_k^\dagger \mathbf{d}_\epsilon = \sum_{i=1}^{k} \frac{\mathbf{u}_i^T \mathbf{d}}{\sigma_i} \mathbf{v}_i$$

we can train the estimator $\Lambda(\mathbf{d}_\epsilon^i; \theta_1, \theta_2)$ as:

$$\arg\min_{\theta_1,\theta_2} \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{m}^i - \Lambda(\mathbf{d}_\epsilon^i; \theta_1, \theta_2)||_2^2 + \lambda_1 \sum_{i=1}^{N} ||\mathbf{L}\mathbf{F}_{\theta_1}(\mathbf{L}_k^\dagger \mathbf{d}_\epsilon^i) - \epsilon^i||_2^2 + \lambda_2 ||\theta_2||_2^2,$$

- Supervised training on a synthetic dataset $\mathcal{D} = \{(\mathbf{m}^i, \mathbf{d}_\epsilon^i)\}_{i=1}^N$ using the MSE loss
- Prevents the denoising component from breaking the data consistency property

Deep decomposition learning

SIGNAL ANALYSIS & IMAGING GROUP

Substituting $\mathbf{L}^\dagger$ by a regularized initial approximation $\mathbf{L}_k^\dagger$ such that:

$$\mathbf{m}^*_{\text{TSVD}} = \mathbf{L}_k^\dagger \mathbf{d}_\epsilon = \sum_{i=1}^{k} \frac{\mathbf{u}_i^T \mathbf{d}}{\sigma_i} \mathbf{v}_i$$

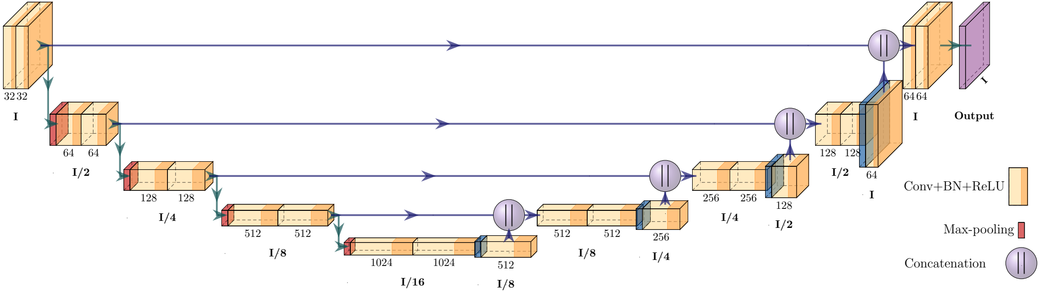
we can train the estimator $\Lambda(\mathbf{d}_\epsilon^i; \theta_1, \theta_2)$ as:

$$\arg\min_{\theta_1,\theta_2} \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{m}^i - \Lambda(\mathbf{d}_\epsilon^i; \theta_1, \theta_2)||_2^2 + \lambda_1 \sum_{i=1}^{N} ||\mathbf{L}\mathbf{F}_{\theta_1}(\mathbf{L}_k^\dagger \mathbf{d}_\epsilon^i) - \epsilon^i||_2^2 + \lambda_2 ||\theta_2||_2^2,$$

- Supervised training on a synthetic dataset $\mathcal{D} = \{(\mathbf{m}^i, \mathbf{d}_\epsilon^i)\}_{i=1}^N$ using the MSE loss
- Prevents the denoising component from breaking the data consistency property
- Provides $\mathbf{N}_{\theta_2}$ with robustness to small perturbations via weight regularization.

$$\Lambda(\mathbf{d}_\epsilon;\theta_1,\theta_2) = \mathbf{L}^\dagger\mathbf{d}_\epsilon + P_R \circ \mathbf{F}_{\theta_1} \circ \mathbf{L}^\dagger\mathbf{d}_\epsilon + P_N \circ \mathbf{N}_{\theta_2} \circ (\mathbf{L}^\dagger\mathbf{d}_\epsilon + P_R \circ \mathbf{F}_{\theta_1} \circ \mathbf{L}^\dagger\mathbf{d}_\epsilon)$$

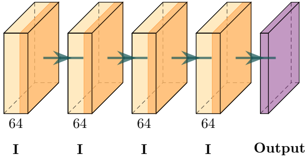
$\mathbf{N}_{\theta_2} =$



Original U-net architecture

$$\Lambda(\mathbf{d}_\epsilon;\theta_1,\theta_2) = \mathbf{L}^\dagger\mathbf{d}_\epsilon + P_R \circ \mathbf{F}_{\theta_1} \circ \mathbf{L}^\dagger\mathbf{d}_\epsilon + P_N \circ \mathbf{N}_{\theta_2} \circ (\mathbf{L}^\dagger\mathbf{d}_\epsilon + P_R \circ \mathbf{F}_{\theta_1} \circ \mathbf{L}^\dagger\mathbf{d}_\epsilon)$$

$$\mathbf{F}_{\theta_1} =$$



Four-layered CNN denoising architecture.

Content

Content

Deconvolution

- Example 1: single-channel deconvolution.
- Example 2: 2D application to a real dataset.

$$s(t) = w(t) * r(t) + \epsilon(t)$$

Deconvolution

- Example 1: single-channel deconvolution.
- Example 2: 2D application to a real dataset.

$$s(t) = w(t) * r(t) + \epsilon(t)$$

$$\mathbf{d}_\epsilon = \mathbf{L}\mathbf{m} + \epsilon$$

Deconvolution

SIGNAL ANALYSIS & IMAGING GROUP

- Example 1: single-channel deconvolution.
- Example 2: 2D application to a real dataset.

$$s(t) = w(t) * r(t) + \epsilon(t)$$

$$\mathbf{d}_\epsilon = \mathbf{L}\mathbf{m} + \epsilon$$

- $\mathbf{L} = \mathbf{U}\Sigma\mathbf{V}^T$
- Initial estimator $\mathbf{L}_k^\dagger = \mathbf{V}_k \Sigma_k^{-1} \mathbf{U}_k^T$
- $\mathbf{m}_{\text{TSVD}} = \sum_{i=1}^{k} \frac{\mathbf{u}_i^T \mathbf{d}}{\sigma_i} \mathbf{v}_i$

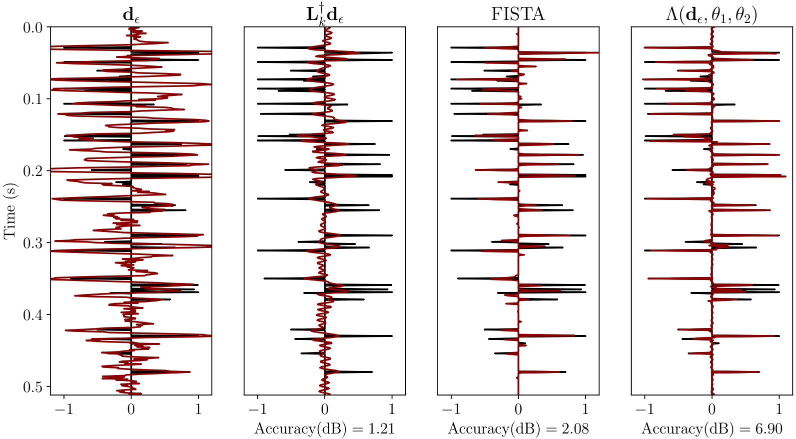
Training details:

- Additive Gaussian noise (SNR = 20%) added to the clean data.
- 5000 randomly generated reflectivity sequences
- 400 epochs of stochastic gradient descent with learning rate of 0.001

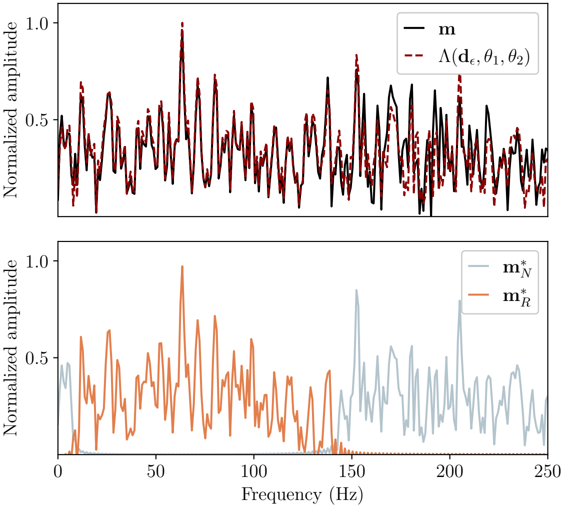
Example 1: single-channel deconvolution

- Test sample

$$\text{Accuracy(dB)} = 10 \times \log_{10} \frac{||\mathbf{m}||_2^2}{||\mathbf{m} - \mathbf{m}^*||_2^2}$$

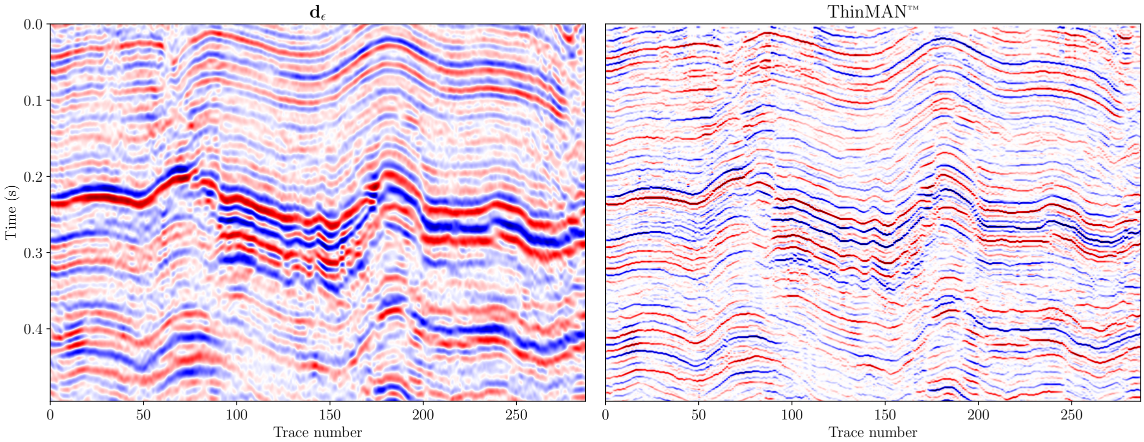


Example 1: single-channel deconvolution



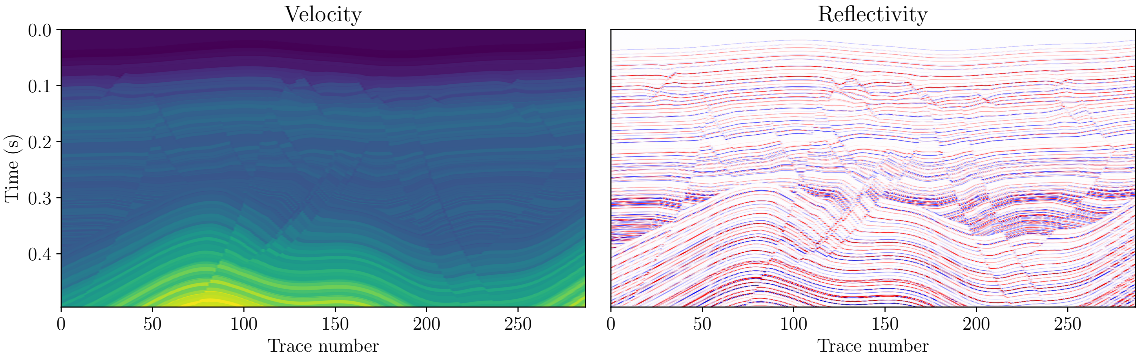
Example 2: Applications to real data

- *Seismic resolution and thin-bed reflectivity inversion* (Chopra et al., 2006)

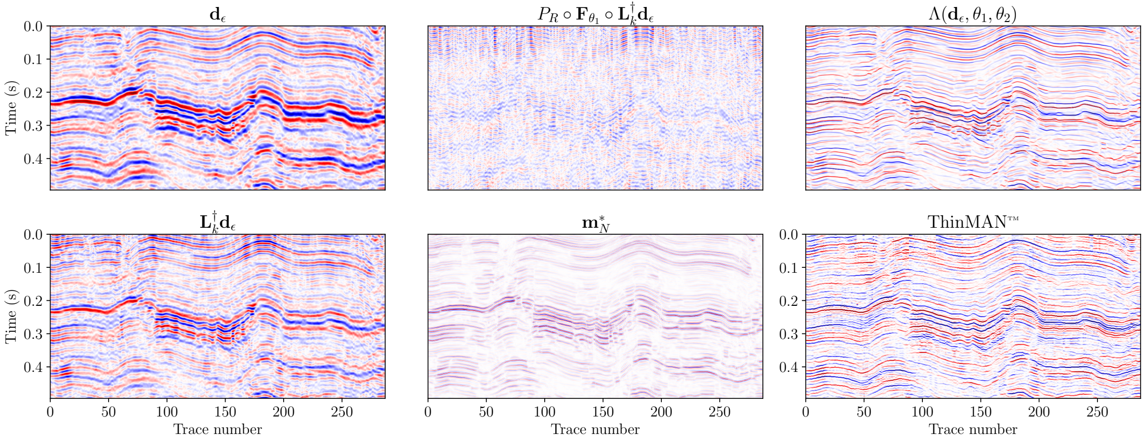


Training details:

- Additive Gaussian noise (SNR = 20%) added to the clean data.
- 5000 randomly generated reflectivity sequences
- 400 epochs of stochastic gradient descent with learning rate of 0.001

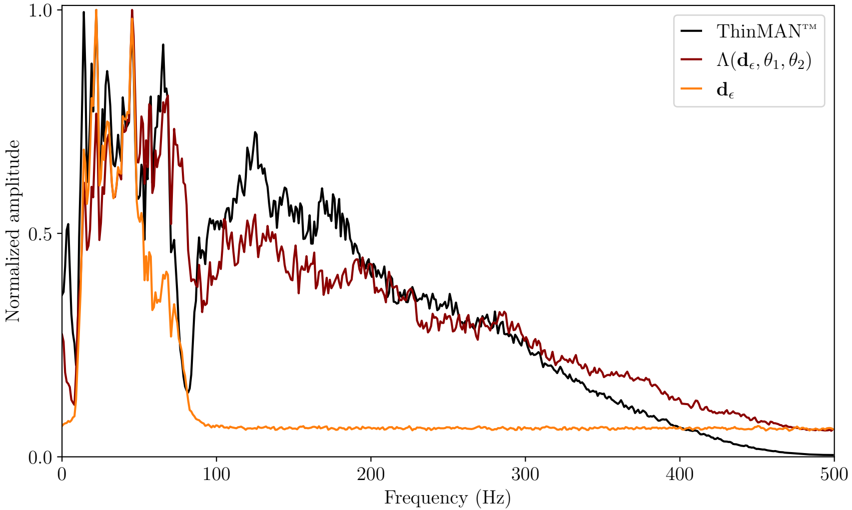


- Results for 2D data:



Example 2: Applications to real data

SIGNAL ANALYSIS & IMAGING GROUP



Content

SIGNAL ANALYSIS & IMAGING GROUP

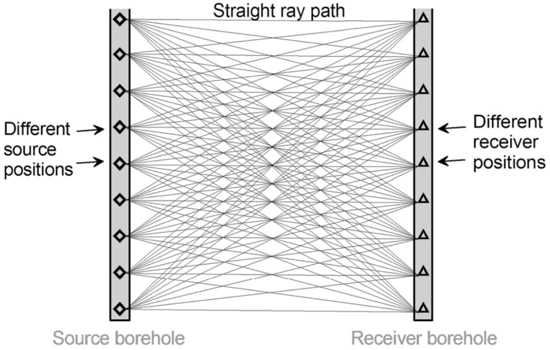
Example 3: Traveltime tomography

Linearized traveltime tomography $\mathbf{d}_\epsilon = \mathbf{L}\mathbf{m} + \epsilon$.

Example 3: Traveltime tomography

Linearized traveltime tomography $\mathbf{d}_\epsilon = \mathbf{L}\mathbf{m} + \epsilon$.

Straight ray tomography does not take into account ray bending but can provide a good quick first velocity model.



Example 3: Traveltime tomography

SIGNAL ANALYSIS & IMAGING GROUP

Acquisition setup

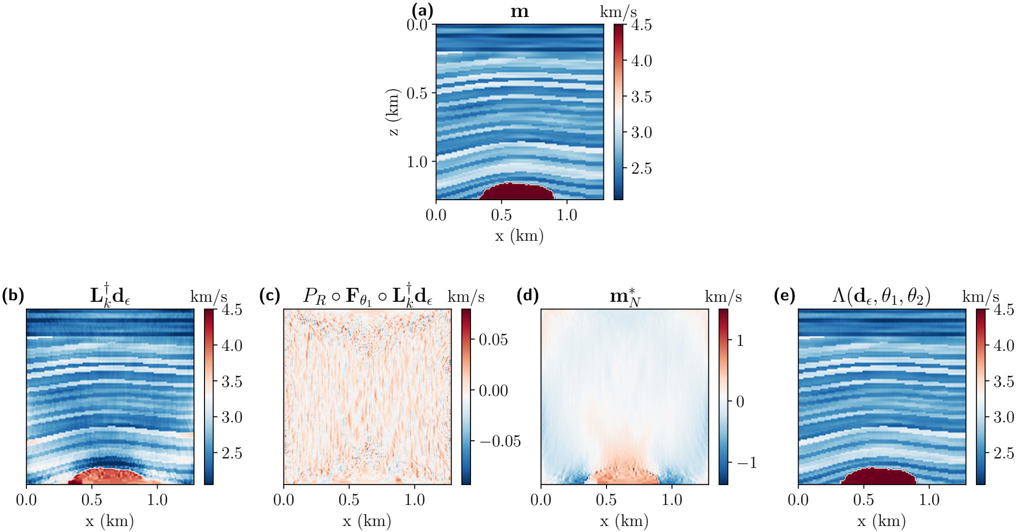
- Transmission experiment: 128 sources and receivers on the right and left boundaries of the domain, respectively.
- $\mathbf{m}$ is discretized in 128×128 cells with 10 m grid spacing.

Training details:

- Additive Gaussian noise (SNR = 20%) added to the clean data.
- 1000 randomly generated training samples (slowness). 250 wuth salt bodies.
- 400 epochs of stochastic gradient descent with learning rate of 0.001

Example 3: Traveltime tomography

- Test model



Content

Advantages

- Data consistency (the reconstruction is consistent with the measurement)
- Interpretable ML: deep learning is only used for inferring lost information.
- Physics-engaged: components of the solution are obtained by pseudoinverse and orthogonal projections.
- Unlike traditional algorithms, this approach does not make any prior explicit assumption on the solution.

Disadvantages

- Still a supervised approach (it learns from ground-truth models)
- Requires easy access to projections (examples where we can explicitly compute the pseudoinverse).

Conclusions

SIGNAL ANALYSIS & IMAGING GROUP

With the numerical applications we showed that:

- Learned null space regularization adds reasonable estimates from the null space while naturally enforcing that the high-resolution prediction is consistent with the low-resolution input.

- Implementing a deep decomposition architecture with TSVD helped produce clean inputs for the efficient training of the null space network.

Future work

SIGNAL ANALYSIS & IMAGING GROUP

Extension to bigger problems:

- Main ingredient in null space networks is access to the projection operators $\mathcal{P}_r$ and $\mathcal{P}_n$
- Explicitly computing $\mathbf{L}^\dagger$ is prohibitive

Future work

SIGNAL ANALYSIS & IMAGING GROUP

Extension to bigger problems:

- Main ingredient in null space networks is access to the projection operators $\mathcal{P}_r$ and $\mathcal{P}_n$
- Explicitly computing $\mathbf{L}^\dagger$ is prohibitive

Improve training of denoising component

- Adapt to specific types of seismic noise/artifacts

Future work

SIGNAL ANALYSIS & IMAGING GROUP

Extension to bigger problems:

- Main ingredient in null space networks is access to the projection operators $\mathcal{P}_r$ and $\mathcal{P}_n$
- Explicitly computing $\mathbf{L}^\dagger$ is prohibitive

Improve training of denoising component

- Adapt to specific types of seismic noise/artifacts

Learning without labels?

- Unsupervised and Semi-supervised learning

Future work

SIGNAL ANALYSIS & IMAGING GROUP

Extension to bigger problems:

- Main ingredient in null space networks is access to the projection operators $\mathcal{P}_r$ and $\mathcal{P}_n$
- Explicitly computing $\mathbf{L}^\dagger$ is prohibitive

Improve training of denoising component

- Adapt to specific types of seismic noise/artifacts

Learning without labels?

- Unsupervised and Semi-supervised learning

Uncertainty quantification

- Null space shutters (Deal and Nolet, 1996)

Acknowledgements

- I would like to thank the sponsors of SAIG for supporitng our work.
- I would also like to thank you all for attending my talk.

Deep null space regularization for seismic inverse problems

SAIG Annual meeting

Kristian Torres

January 25, 2022