Signal analysis using sparse representation and proximal optimization methods

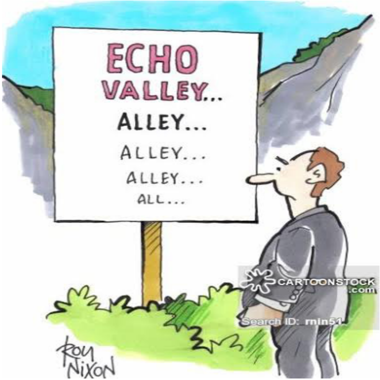
Mai Quyen PHAM

GIPSA-Lab, 11 rue des mathématiques, 38402 Saint Martin d'Hères

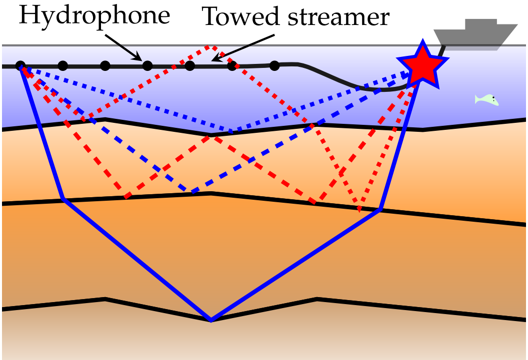
04 November 2016

Contents

(Multiple + noise) removal

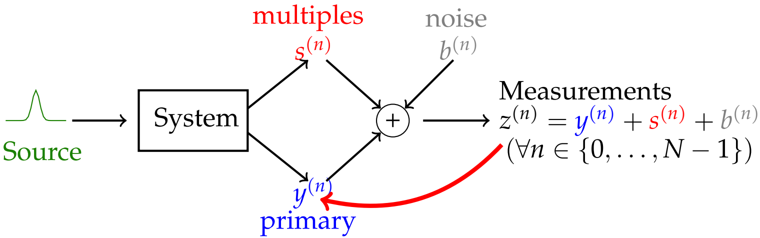


Seismic multiple reflection



Solid blue: primaries; dashed red: multiple reflection disturbances.

(Multiple + noise) removal strategies



Which strategy for restoring the primary signal $y^{(n)}$ corrupted by the unknown multiples $s^{(n)}$, plus noise $b^{(n)}$?

- Methodology for primary/multiple adaptive separation based on approximate templates
- Variational approach
- Proximal methods to solve the resulting optimization problem

Multi-model

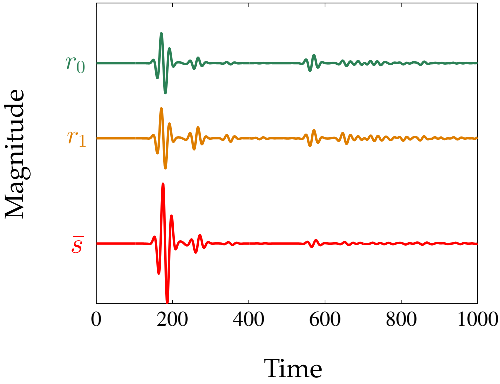
- J models $r_j^{(n)}$ are known (available)
- Imperfect in time, amplitude and frequency
- **Assumption**: models linked to $\bar{s}^{(n)}$ throughout time varying filters (FIR)

$$\bar{s}^{(n)} = \sum_{j=0}^{J-1} \sum_{p=p'}^{p'+P_j-1} \bar{h}_j^{(n)}(p) r_j^{(n-p)}$$

where

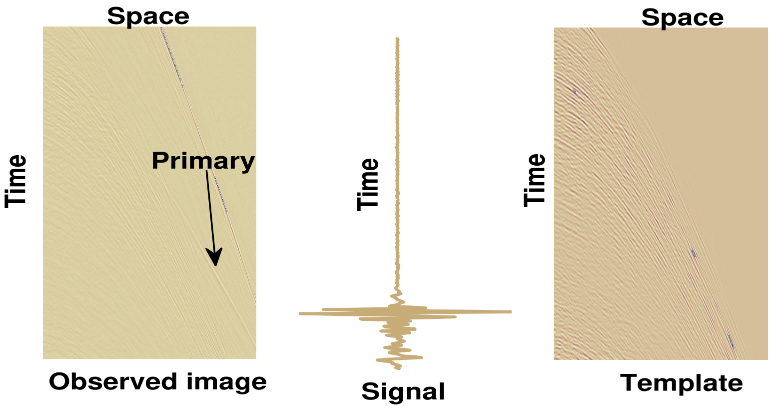
- $\bar{h}_j^{(n)}$: unknown impulse reponse of the filter corresponding to model j and time n (P_j tap coefficients)
- $p' \in \{-P_j + 1, \ldots, 0\}$
- New definition: $P = \sum_{j=0}^{J-1} P_j$.

Template



First model, Second model, Multiple

Template


Space
Time
Primary
Observed image
Time
Signal
Space
Time
Template

Problem reformulation

$$\underbrace{z}_{\text{observed signal}} = \mathbf{R} \underbrace{\bar{\mathbf{h}}}_{\text{filter}} + \underbrace{\bar{y}}_{\text{primary}} + \underbrace{b}_{\text{noise}}$$

where

- $\bar{s} = \sum_{j=0}^{J-1} R_j \bar{h}_j = \mathbf{R}\bar{\mathbf{h}} = \left[\bar{s}^{(0)}, \; \cdots, \; \bar{s}^{(N-1)}\right]^\top$
- $\mathbf{R} = [R_0 \cdots R_{J-1}]$, R_j is a block diagonal matrix
- $\bar{\mathbf{h}} = \left[\bar{h}_0^\top \cdots \bar{h}_{J-1}^\top\right]^\top$
- $\bar{h}_j^{(n)} = \Big[\bar{h}_j^{(0)}(p') \; \cdots \; \bar{h}_j^{(0)}(p' + P_j - 1) \; \cdots \; \bar{h}_j^{(N-1)}(p') \; \cdots \; \bar{h}_j^{(N-1)}(p' + P_j - 1)\Big]^\top$

Estimation of y

Assumption: $\bar{y}$ is a realization of a random vector Y, whose probability density is given by:

$$(\forall y \in \mathbb{R}^N) \qquad f_Y(y) \propto \exp(-\varphi(Fy))$$

$F \in \mathbb{R}^{K \times N}$: linear operator.
φ is chosen separable:

$$\big(\forall x = (x_k)_{1 \le k \le K} \in \mathbb{R}^K\big) \qquad \varphi(x) = \sum_{k=1}^{K} \varphi_k(x_k)$$

where, for all $k \in \{1, \dots, K\}$, $\varphi_k \colon \mathbb{R} \to \,]-\infty, +\infty]$.

Estimation : filter $\mathbf{h}$ and noise b

- **Assumption**: $\bar{\mathbf{h}}$ is a realization of a random vector H, whose probability density can be expressed as:
  $(\forall \mathbf{h} \in \mathbb{R}^{NP})\ f_H(\mathbf{h}) \propto \exp(-\rho(\mathbf{h}))$
  H is independent of Y.
- **Assumption**: b is a realization of a random vector B, of probability density:

$$(\forall b \in \mathbb{R}^N) \qquad f_B(b) \propto \exp(-\psi(b))$$

B is assumed to be independent from Y and H

Estimation : filter $\mathbf{h}$ and noise b

- **Assumption**: $\bar{\mathbf{h}}$ is a realization of a random vector H, whose probability density can be expressed as:
  $(\forall \mathbf{h} \in \mathbb{R}^{NP})\ f_H(\mathbf{h}) \propto \exp(-\rho(\mathbf{h}))$
  H is independent of Y.
- **Assumption**: b is a realization of a random vector B, of probability density:

$$(\forall b \in \mathbb{R}^N) \qquad f_B(b) \propto \exp(-\psi(b))$$

B is assumed to be independent from Y and H

MAP estimation of $(y, \mathbf{h})$

$$\underset{y \in \mathbb{R}^N, \mathbf{h} \in \mathbb{R}^{NP}}{\text{minimize}} \ \underbrace{\psi(z - \mathbf{R}\mathbf{h} - y)}_{\text{fidelity: linked to noise}} + \underbrace{\varphi(Fy)}_{\textit{a priori} \text{ on the signal}} + \underbrace{\rho(\mathbf{h})}_{\textit{a priori} \text{ on the filters}}$$

♣

Problem to be solved

MAP estimation of $(y, \mathbf{h})$

$$\underset{y\in\mathbb{R}^N,\mathbf{h}\in\mathbb{R}^{NP}}{\text{minimize}} \quad \underbrace{\psi(z - \mathbf{R}\mathbf{h} - y)}_{\text{fidelity: linked to noise}} + \underbrace{\varphi(Fy)}_{\textit{a priori} \text{ on the signal}} + \underbrace{\rho(\mathbf{h})}_{\textit{a priori} \text{ on the filters}}$$

♣

- **Difficulty: Choosing the good regularization parameters**
- **Proposed: Use a constrained minimization problem**

Problem to be solved

$$\underset{y\in\mathbb{R}^N,\mathbf{h}\in\mathbb{R}^{NP}}{\text{minimize}} \ \psi(z - \mathbf{R}\mathbf{h} - y) + \iota_D(Fy) + \iota_C(\mathbf{h})$$

About convex set D

Problem to be solved

$$\underset{y\in\mathbb{R}^N,\mathbf{h}\in\mathbb{R}^{NP}}{\text{minimize}}\ \psi\big(z-\mathbf{Rh}-y\big)+\iota_D(Fy)+\iota_C(\mathbf{h})$$

$$\iota_{\mathrm{D}}(x)=\begin{cases}0 & \text{if } x\in D\\ +\infty & \text{otherwise.}\end{cases}$$

- $F\in\mathbb{R}^{K\times N}$: analysis frame operator
- $\{\mathbb{K}_l \mid l\in\{1,\ldots,\mathcal{L}\}\}\subset\{1,\ldots,K\}$
- $D=D_1\times\cdots\times D_{\mathcal{L}}$ with $D_l=\{(x_k)_{k\in\mathbb{K}_l}\mid \sum_{k\in\mathbb{K}_l}\varphi_\ell(x_k)\le\beta_l\}$, where $\forall l\in\{1,\ldots,\mathcal{L}\}$, $\beta_l\in\,]0,+\infty[$, and $\varphi_l:\mathbb{R}\to[0,+\infty[$ is a lower-semicontinuous convex function.

About convex set C

Problem to be solved

$$\underset{y\in\mathbb{R}^N,\mathbf{h}\in\mathbb{R}^{NP}}{\text{minimize}}\ \psi(z-\mathbf{Rh}-y)+\iota_D(Fy)+\iota_C(\mathbf{h})$$

$$C = C_1 \cap C_2 \cap C_3$$

- $C_1 = \left\{\mathbf{h} \in \mathbb{R}^{PN} \ : \ \rho(\mathbf{h}) = \sum_{j=0}^{J-1} \rho_j(h_j) \leq \tau\right\}$
- $\rho_j(h_j) = \|h_j\|_{\ell_2}^2 = \sum_{n=0}^{N-1} \sum_{p=p'}^{p'+P_j-1} |h_j^{(n)}(p)|^2$

About convex set C

Problem to be solved

$$\underset{y\in\mathbb{R}^N,\mathbf{h}\in\mathbb{R}^{NP}}{\text{minimize}}\ \psi\big(z-\mathbf{R}\mathbf{h}-y\big)+\iota_D(Fy)+\iota_C(\mathbf{h})$$

$$C = C_1 \cap C_2 \cap C_3$$

- $C_1 = \left\{\mathbf{h}\in\mathbb{R}^{PN}\ :\ \rho(\mathbf{h}) = \sum_{j=0}^{J-1}\rho_j(h_j) \le \tau\right\}$
- $\rho_j(h_j) = \|h_j\|_{\ell_2}^2 = \sum_{n=0}^{N-1}\sum_{p=p'}^{p'+P_j-1}|h_j^{(n)}(p)|^2$
- $\rho_j(h_j) = \|h_j\|_{\ell_1} = \sum_{n=0}^{N-1}\sum_{p=p'}^{p'+P_j-1}|h_j^{(n)}(p)|$

About convex set C

Problem to be solved

$$\underset{y\in\mathbb{R}^N,\mathbf{h}\in\mathbb{R}^{NP}}{\text{minimize}}\ \psi(z-\mathbf{Rh}-y)+\iota_D(Fy)+\iota_C(\mathbf{h})$$

$$C = C_1 \cap C_2 \cap C_3$$

- $C_1 = \left\{\mathbf{h}\in\mathbb{R}^{PN} \ : \ \rho(\mathbf{h}) = \sum_{j=0}^{J-1}\rho_j(h_j) \leq \tau\right\}$
- $\rho_j(h_j) = \|h_j\|_{\ell_2}^2 = \sum_{n=0}^{N-1}\sum_{p=p'}^{p'+P_j-1} |h_j^{(n)}(p)|^2$
- $\rho_j(h_j) = \|h_j\|_{\ell_1} = \sum_{n=0}^{N-1}\sum_{p=p'}^{p'+P_j-1} |h_j^{(n)}(p)|$
- $\rho_j(h_j) = \|h_j\|_{\ell_{1,2}} = \sum_{n=0}^{N-1}\left(\sum_{p=p'}^{p'+P_j-1} |h_j^{(n)}(p)|^2\right)^{1/2}$

Hard constraints on the filters $C_2,\ C_3$

Problem to be solved

$$\underset{y\in\mathbb{R}^N,\mathbf{h}\in\mathbb{R}^{NP}}{\text{minimize}}\ \psi\big(z-\mathbf{Rh}-y\big)+\iota_D(Fy)+\iota_C(\mathbf{h})$$

$$C = C_1 \cap C_2 \cap C_3$$

Assumption: slow variations of the filters along time.

$$\big(\forall (j,n,p)\big) \qquad |h_j^{(n+1)}(p) - h_j^{(n)}(p)| \le \varepsilon_{j,p}$$

For computational issues, $h \in C_2 \cap C_3$ where

$$C_2 = \Big\{h \mid \forall p,\ \forall n \in \Big\{0,\dots,\Big\lfloor \frac{N}{2}\Big\rfloor - 1\Big\} \quad \Big|h^{(2n+1)}(p) - h^{(2n)}(p)\Big| \le \varepsilon_p\Big\}$$

$$C_3 = \Big\{h \mid \forall p,\ \forall n \in \Big\{1,\dots,\Big\lfloor \frac{N-1}{2}\Big\rfloor\Big\} \quad \Big|h^{(2n)}(p) - h^{(2n-1)}(p)\Big| \le \varepsilon_p\Big\}$$

Proximity operator

Definition

Let φ be a lower semi-continuous convex function. For all $x \in \mathbb{R}^N$, $\operatorname{prox}_\varphi$ is the unique minimizer of

$$y \mapsto \varphi(y) + \frac{1}{2}\|x - y\|^2$$

♣

Examples:

C a non-empty closed convex subset of $\mathbb{R}^N$.

$$\begin{aligned}\operatorname{prox}_{\iota_C}(x) &= \underset{y \in \mathbb{R}^N}{\operatorname{minimize}}\ \iota_C(y) + \frac{1}{2}\|x - y\|^2 \\ &= \underbrace{\underset{y \in C}{\operatorname{minimize}}\ \|x - y\|^2}_{\Pi_C(x)\text{: projection operator onto } C}\end{aligned}$$

Proximity operator

Definition

Let φ be a lower semi-continuous convex function. For all $x \in \mathbb{R}^N$, prox_φ is the unique minimizer of

$$y \mapsto \varphi(y) + \frac{1}{2}\|x - y\|^2$$

♣

Examples:

C a non-empty closed convex subset of $\mathbb{R}^N$.

$$\begin{aligned}\text{prox}_{\iota_C}(x) &= \underset{y \in \mathbb{R}^N}{\text{minimize}}\ \iota_C(y) + \frac{1}{2}\|x - y\|^2 \\ &= \underbrace{\underset{y \in C}{\text{minimize}}\ \|x - y\|^2}_{\Pi_C(x)\text{: projection operator onto } C}\end{aligned}$$

Proximity operator

Definition

Let φ be a lower semi-continuous convex function. For all $x \in \mathbb{R}^N$, prox_φ is the unique minimizer of

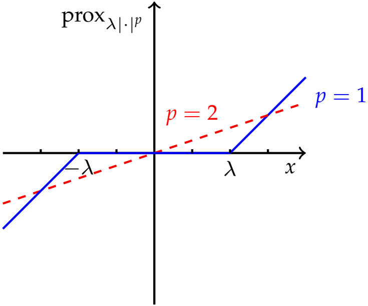
$$y \mapsto \varphi(y) + \frac{1}{2}\|x - y\|^2$$

Examples:

$(\forall x \in \mathbb{R})$

$$a)\ \text{prox}_{\lambda|\cdot|^2}(x) = \underbrace{\frac{1}{1+2\lambda}x}_{\text{“Wiener” filter}}$$

$$b)\ \text{prox}_{\lambda|\cdot|}(x) = \underbrace{\text{sign}(x)\max(|x| - \lambda, 0)}_{\text{shrinkage operator}}$$



Proposed algorithm

Problem to be solved

$$\underset{y\in\mathbb{R}^N,\mathbf{h}\in\mathbb{R}^{NP}}{\text{minimize}} \ \Psi\left(\begin{bmatrix} y \\ \mathbf{h}\end{bmatrix}\right) + \iota_D(Fy) + \iota_C(\mathbf{h})$$

- $\Psi : \mathbb{R}^{N+NP} \to \mathbb{R} : \begin{bmatrix} y \\ \mathbf{h}\end{bmatrix} \mapsto \psi(z - \mathbf{R}\mathbf{h} - y)$ is convex and differentiable with μ-Lipschitzian gradient ($\mu \in]0, +\infty[$) i.e.

$$\left(\forall (u,v) \in \mathbb{R}^{2(N+NP)}\right) \quad \|\nabla\Psi(u) - \nabla\Psi(v)\| \leq \mu\|u-v\|,$$

- $(\forall i \in \mathbb{N})$, $\gamma^{[i]} \in [\epsilon, \frac{1-\epsilon}{\beta}]$ where

$$\beta = \mu + \sqrt{\|F\|^2 + 3} \text{ and } \epsilon \in]0, \frac{1}{\beta+1}[$$

$\rightsquigarrow$ **Use the M+LFBF algorithm [Combettes and Pesquet, 2012]**

Algorithm M+LFBF [Combettes and Pesquet, 2012]

Set $y^{[0]} \in \mathbb{R}^N, \mathbf{h}^{[0]} \in \mathbb{R}^{NP}, v^{[0]} \in \mathbb{R}^K, u^{[0]} \in \mathbb{R}^{NP}$

for $i = 0, 1, \dots$ **do**

Gradient computation

$$\begin{bmatrix} s_1^{[i]} \\ t_1^{[i]} \end{bmatrix} = \begin{bmatrix} y^{[i]} \\ \mathbf{h}^{[i]} \end{bmatrix} - \gamma^{[i]} \left(\nabla \Psi \left(\begin{bmatrix} y^{[i]} \\ \mathbf{h}^{[i]} \end{bmatrix} \right) + \begin{bmatrix} F^* v^{[i]} \\ u^{[i]} \end{bmatrix} \right)$$

Projection computation

$$s_2^{[i]} = v^{[i]} + \gamma^{[i]} F y^{[i]}, \quad w_1^{[i]} = s_2^{[i]} - \gamma^{[i]} \Pi_D((\gamma^{[i]})^{-1} s_2^{[i]})$$

$$t_2^{[i]} = u^{[i]} + \gamma^{[i]} \mathbf{h}^{[i]}, \quad w_2^{[i]} = t_2^{[i]} - \gamma^{[i]} \Pi_C((\gamma^{[i]})^{-1} t_2^{[i]})$$

Averaging

$$q_1^{[i]} = w_1^{[i]} + \gamma^{[i]} F s_1^{[i]}, \quad v^{[i+1]} = v^{[i]} - s_2^{[i]} + q_1^{[i]}$$

$$q_2^{[i]} = w_2^{[i]} + \gamma^{[i]} t_1^{[i]}, \quad u^{[i+1]} = u^{[i]} - t_2^{[i]} + q_2^{[i]}$$

Update

$$\begin{bmatrix} y^{[i+1]} \\ \mathbf{h}^{[i+1]} \end{bmatrix} = \begin{bmatrix} y^{[i]} \\ \mathbf{h}^{[i]} \end{bmatrix} - \gamma^{[i]} \left(\nabla \Psi \left(\begin{bmatrix} s_1^{[i]} \\ t_1^{[i]} \end{bmatrix} \right) + \begin{bmatrix} F^* w_1^{[i]} \\ w_2^{[i]} \end{bmatrix} \right)$$

end for

Algorithm M+LFBF [Combettes and Pesquet, 2012]

Set $y^{[0]} \in \mathbb{R}^N$, $\mathbf{h}^{[0]} \in \mathbb{R}^{NP}$, $v^{[0]} \in \mathbb{R}^K$, $u^{[0]} \in \mathbb{R}^{NP}$

for $i = 0, 1, \ldots$ **do**

Gradient computation

$$\begin{bmatrix} s_1^{[i]} \\ t_1^{[i]} \end{bmatrix} = \begin{bmatrix} y^{[i]} \\ \mathbf{h}^{[i]} \end{bmatrix} - \gamma^{[i]} \left(\nabla\Psi \left(\begin{bmatrix} y^{[i]} \\ \mathbf{h}^{[i]} \end{bmatrix} \right) + \begin{bmatrix} F^* v^{[i]} \\ u^{[i]} \end{bmatrix} \right)$$

Projection computation

$$s_2^{[i]} = v^{[i]} + \gamma^{[i]} F y^{[i]}, \quad w_1^{[i]} = s_2^{[i]} - \gamma^{[i]} \Pi_D((\gamma^{[i]})^{-1} s_2^{[i]})$$

$$t_2^{[i]} = u^{[i]} + \gamma^{[i]} \mathbf{h}^{[i]}, \quad w_2^{[i]} = t_2^{[i]} - \gamma^{[i]} \Pi_C((\gamma^{[i]})^{-1} t_2^{[i]})$$

Averaging

$$q_1^{[i]} = w_1^{[i]} + \gamma^{[i]} F s_1^{[i]}, \quad v^{[i+1]} = v^{[i]} - s_2^{[i]} + q_1^{[i]}$$

$$q_2^{[i]} = w_2^{[i]} + \gamma^{[i]} t_1^{[i]}, \quad u^{[i+1]} = u^{[i]} - t_2^{[i]} + q_2^{[i]}$$

Update

$$\begin{bmatrix} y^{[i+1]} \\ \mathbf{h}^{[i+1]} \end{bmatrix} = \begin{bmatrix} y^{[i]} \\ \mathbf{h}^{[i]} \end{bmatrix} - \gamma^{[i]} \left(\nabla\Psi \left(\begin{bmatrix} s_1^{[i]} \\ t_1^{[i]} \end{bmatrix} \right) + \begin{bmatrix} F^* w_1^{[i]} \\ w_2^{[i]} \end{bmatrix} \right)$$

end for

Algorithm M+LFBF [Combettes and Pesquet, 2012]

Set $y^{[0]} \in \mathbb{R}^N, \mathbf{h}^{[0]} \in \mathbb{R}^{NP}, v^{[0]} \in \mathbb{R}^K, u^{[0]} \in \mathbb{R}^{NP}$

for $i = 0, 1, \ldots$ **do**

Gradient computation

$$\begin{bmatrix} s_1^{[i]} \\ t_1^{[i]} \end{bmatrix} = \begin{bmatrix} y^{[i]} \\ \mathbf{h}^{[i]} \end{bmatrix} - \gamma^{[i]} \left(\nabla \Psi \left(\begin{bmatrix} y^{[i]} \\ \mathbf{h}^{[i]} \end{bmatrix} \right) + \begin{bmatrix} F^* v^{[i]} \\ u^{[i]} \end{bmatrix} \right)$$

Projection computation

$$s_2^{[i]} = v^{[i]} + \gamma^{[i]} F y^{[i]}, \quad w_1^{[i]} = s_2^{[i]} - \gamma^{[i]} \Pi_D((\gamma^{[i]})^{-1} s_2^{[i]})$$
$$t_2^{[i]} = u^{[i]} + \gamma^{[i]} \mathbf{h}^{[i]}, \quad w_2^{[i]} = t_2^{[i]} - \gamma^{[i]} \Pi_C((\gamma^{[i]})^{-1} t_2^{[i]})$$

Averaging

$$q_1^{[i]} = w_1^{[i]} + \gamma^{[i]} F s_1^{[i]}, \quad v^{[i+1]} = v^{[i]} - s_2^{[i]} + q_1^{[i]}$$
$$q_2^{[i]} = w_2^{[i]} + \gamma^{[i]} t_1^{[i]}, \quad u^{[i+1]} = u^{[i]} - t_2^{[i]} + q_2^{[i]}$$

Update

$$\begin{bmatrix} y^{[i+1]} \\ \mathbf{h}^{[i+1]} \end{bmatrix} = \begin{bmatrix} y^{[i]} \\ \mathbf{h}^{[i]} \end{bmatrix} - \gamma^{[i]} \left(\nabla \Psi \left(\begin{bmatrix} s_1^{[i]} \\ t_1^{[i]} \end{bmatrix} \right) + \begin{bmatrix} F^* w_1^{[i]} \\ w_2^{[i]} \end{bmatrix} \right)$$

end for

Algorithm M+LFBF [Combettes and Pesquet, 2012]

Set $y^{[0]} \in \mathbb{R}^N, \mathbf{h}^{[0]} \in \mathbb{R}^{NP}, v^{[0]} \in \mathbb{R}^K, u^{[0]} \in \mathbb{R}^{NP}$

for $i = 0, 1, \ldots$ **do**

Gradient computation

$$\begin{bmatrix} s_1^{[i]} \\ t_1^{[i]} \end{bmatrix} = \begin{bmatrix} y^{[i]} \\ \mathbf{h}^{[i]} \end{bmatrix} - \gamma^{[i]} \left(\nabla \Psi \left(\begin{bmatrix} y^{[i]} \\ \mathbf{h}^{[i]} \end{bmatrix} \right) + \begin{bmatrix} F^* v^{[i]} \\ u^{[i]} \end{bmatrix} \right)$$

Projection computation

$$s_2^{[i]} = v^{[i]} + \gamma^{[i]} F y^{[i]}, \quad w_1^{[i]} = s_2^{[i]} - \gamma^{[i]} \Pi_D((\gamma^{[i]})^{-1} s_2^{[i]})$$

$$t_2^{[i]} = u^{[i]} + \gamma^{[i]} \mathbf{h}^{[i]}, \quad w_2^{[i]} = t_2^{[i]} - \gamma^{[i]} \Pi_C((\gamma^{[i]})^{-1} t_2^{[i]})$$

Averaging

$$q_1^{[i]} = w_1^{[i]} + \gamma^{[i]} F s_1^{[i]}, \quad v^{[i+1]} = v^{[i]} - s_2^{[i]} + q_1^{[i]}$$

$$q_2^{[i]} = w_2^{[i]} + \gamma^{[i]} t_1^{[i]}, \quad u^{[i+1]} = u^{[i]} - t_2^{[i]} + q_2^{[i]}$$

Update

$$\begin{bmatrix} y^{[i+1]} \\ \mathbf{h}^{[i+1]} \end{bmatrix} = \begin{bmatrix} y^{[i]} \\ \mathbf{h}^{[i]} \end{bmatrix} - \gamma^{[i]} \left(\nabla \Psi \left(\begin{bmatrix} s_1^{[i]} \\ t_1^{[i]} \end{bmatrix} \right) + \begin{bmatrix} F^* w_1^{[i]} \\ w_2^{[i]} \end{bmatrix} \right)$$

end for

Algorithm M+LFBF [Combettes and Pesquet, 2012]

Set $y^{[0]} \in \mathbb{R}^N, \mathbf{h}^{[0]} \in \mathbb{R}^{NP}, v^{[0]} \in \mathbb{R}^K, u^{[0]} \in \mathbb{R}^{NP}$

for $i = 0, 1, \ldots$ **do**

Gradient computation

$$\begin{bmatrix} s_1^{[i]} \\ t_1^{[i]} \end{bmatrix} = \begin{bmatrix} y^{[i]} \\ \mathbf{h}^{[i]} \end{bmatrix} - \gamma^{[i]} \left(\nabla\Psi \left(\begin{bmatrix} y^{[i]} \\ \mathbf{h}^{[i]} \end{bmatrix} \right) + \begin{bmatrix} F^* v^{[i]} \\ u^{[i]} \end{bmatrix} \right)$$

Projection computation

$$s_2^{[i]} = v^{[i]} + \gamma^{[i]} F y^{[i]}, \quad w_1^{[i]} = s_2^{[i]} - \gamma^{[i]} \Pi_D((\gamma^{[i]})^{-1} s_2^{[i]})$$

$$t_2^{[i]} = u^{[i]} + \gamma^{[i]} \mathbf{h}^{[i]}, \quad w_2^{[i]} = t_2^{[i]} - \gamma^{[i]} \Pi_C((\gamma^{[i]})^{-1} t_2^{[i]})$$

Averaging

$$q_1^{[i]} = w_1^{[i]} + \gamma^{[i]} F s_1^{[i]}, \quad v^{[i+1]} = v^{[i]} - s_2^{[i]} + q_1^{[i]}$$

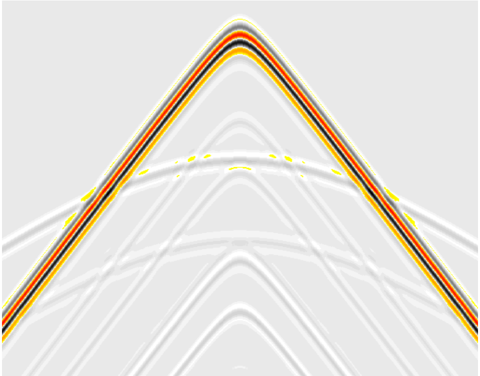
$$q_2^{[i]} = w_2^{[i]} + \gamma^{[i]} t_1^{[i]}, \quad u^{[i+1]} = u^{[i]} - t_2^{[i]} + q_2^{[i]}$$

Update

$$\begin{bmatrix} y^{[i+1]} \\ \mathbf{h}^{[i+1]} \end{bmatrix} = \begin{bmatrix} y^{[i]} \\ \mathbf{h}^{[i]} \end{bmatrix} - \gamma^{[i]} \left(\nabla\Psi \left(\begin{bmatrix} s_1^{[i]} \\ t_1^{[i]} \end{bmatrix} \right) + \begin{bmatrix} F^* w_1^{[i]} \\ w_2^{[i]} \end{bmatrix} \right)$$

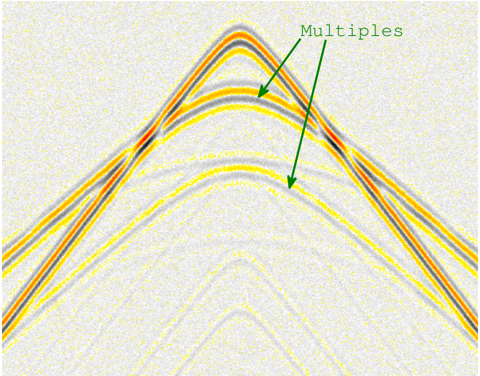
end for

Synthetic data



Primary: $\overline{y}$
- size 512×512

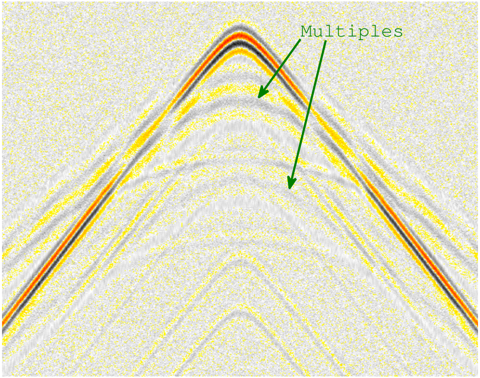
Synthetic data



Primary: $\overline{y}$
- size 512×512

Observed image: z
- Noise: $\sigma = 0.08$
- SNR = 1.13 dB
- SSIM = 0.16

Synthetic data



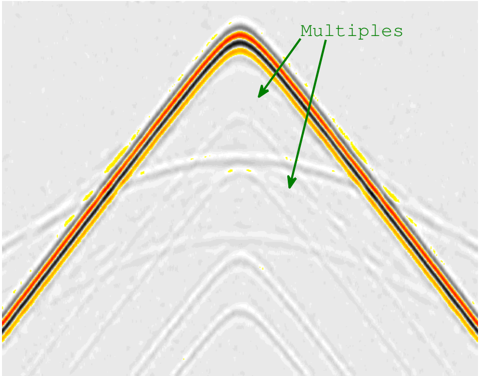
Primary: $\overline{y}$
- size 512×512

Observed image: z
- Noise: $\sigma = 0.08$
- SNR = 1.13 dB
- SSIM = 0.16

Reconstructed image by

[Ventosa *et al.*, 2012]
- SNR = 2.38 dB
- SSIM = 0.13

Synthetic data



Primary: $\overline{y}$
- size 512×512

Observed image: z
- Noise: $\sigma = 0.08$
- SNR = 1.13 dB
- SSIM = 0.16

Reconstructed image by

[Ventosa *et al.*, 2012]
- SNR = 2.38 dB
- SSIM = 0.13

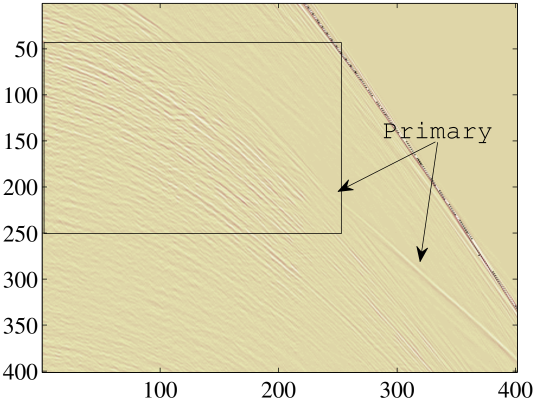
Our method
- SNR = 17.00 dB
- SSIM = 0.74

Summary

		σ	0.04	0.08	0.16
$\bar{y} - z$		$\ell_1(\times 10^{-2})$	3.88	6.89	13.09
[Ventosa *et al.*, 2012]		$\ell_1(\times 10^{-2})$	5.38	7.87	13.36
$\rho_j = \ell_2$	OR-basis	$\ell_1(\times 10^{-2})$	1.53	2.27	3.34
	SI frame$^{(*)}$	$\ell_1(\times 10^{-2})$	1.19	1.69	2.42
	M-band	$\ell_1(\times 10^{-2})$	1.07	1.41	1.96
$\rho_j = \ell_1$	OR-basis	$\ell_1(\times 10^{-2})$	1.66	2.33	3.37
	SI frame$^{(*)}$	$\ell_1(\times 10^{-2})$	1.23	1.70	2.39
	M-band	$\ell_1(\times 10^{-2})$	1.14	1.47	2.00
$\rho_j = \ell_{1,2}$	OR-basis	$\ell_1(\times 10^{-2})$	1.51	2.25	3.32
	SI frame$^{(*)}$	$\ell_1(\times 10^{-2})$	1.10	1.58	2.32
	M-band	$\ell_1(\times 10^{-2})$	0.95	1.31	1.87

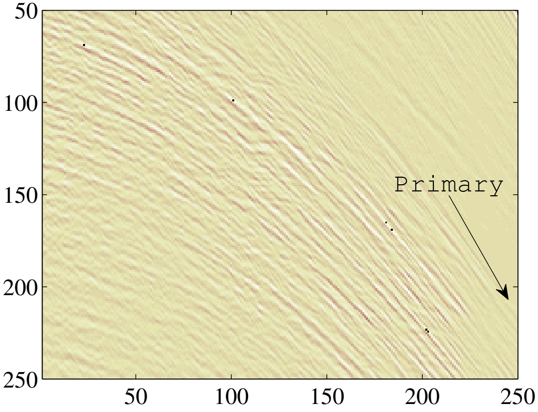
Comparison of the estimated primaries with the 2D proposed version$^{(*)}$ in using three different 2D wavelet transforms, over three noise levels, and three a priori functions $\rho_j \in \{\ell_2, \ell_1, \ell_{1,2}\}$, with [Ventosa *et al.*, 2012]

Real data



Seismic data with
a partially appearing primary
- size 400 $\times$ 400

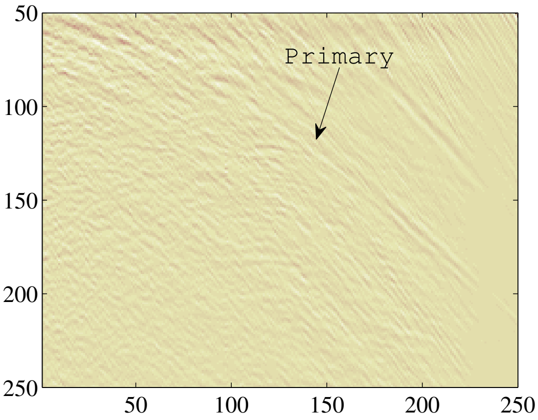
Real data



Seismic data with
a partially appearing primary
- size 400×400

cropped of recorded
seismic data: z
- size 256×256

Real data



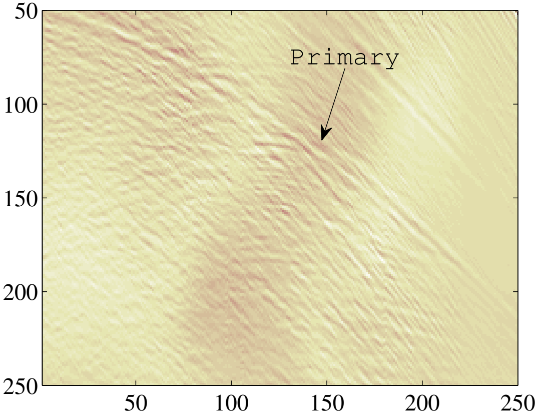
Seismic data with a partially appearing primary
- size 400×400

cropped of recorded seismic data: z
- size 256×256

Reconstructed image by

[Ventosa *et al.*, 2012]

Real data



Seismic data with
a partially appearing primary
- size 400 × 400

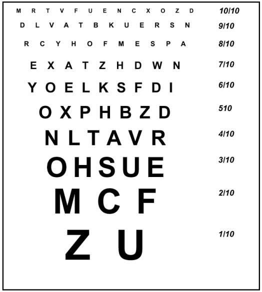
cropped of recorded
seismic data: z
- size 256 × 256

Reconstructed image by

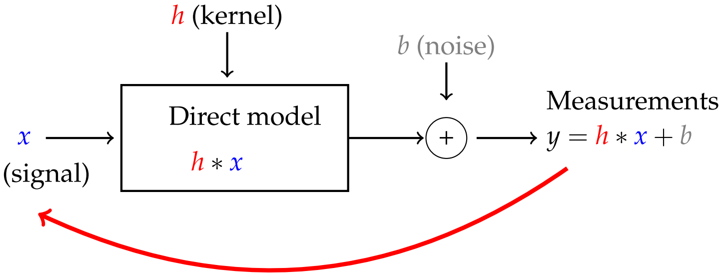
[Ventosa *et al.*, 2012]

Our method

Blind deconvolution



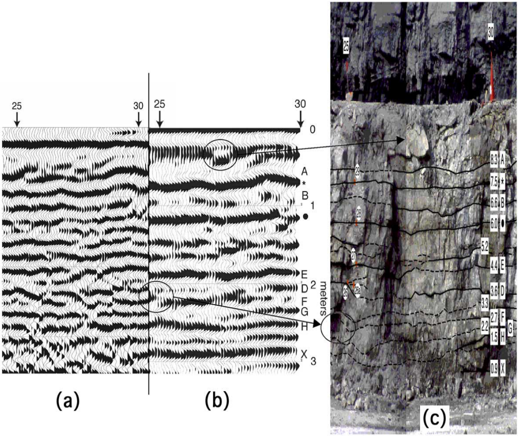
Degradation model



Which strategy for restoring signal x corrupted by a unknown kernel h, plus noise b?

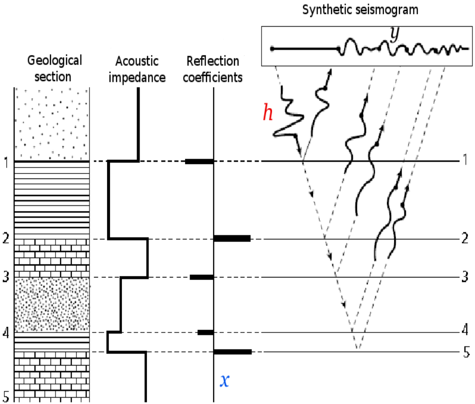
- Majorize-Minimize approach
- Block Coordinate Variable Metric Forward-Backward algorithm
- Smooth approximation of the ℓ_1/ℓ_2 function $\rightsquigarrow$ Efficient for sparse blind deconvolution problems

Seismic blind deconvolution



An enlarged portion from the 25 to 30 meter locations on the bench
(http://www.kgs.ku.edu/Geophysics/OFR/2004/OFR04_41/)

Seismic blind deconvolution



Reflection seismograms showing primary reflection only
[Al-Sadi, 1980]

Smoothed ℓ_1/ℓ_2 sparsity measures

Common assumption: x has a sparse representation

Question: Which sparsity measure should be used?

Theoretically: ℓ_0 measure [Donoho *et al.*, 1995]

Usually: ℓ_1 measure

We propose a smoothed ℓ_1/ℓ_2 sparsity measures

$$\varphi(x) = \log\left(\frac{\ell_{1,\alpha}(x) + \beta}{\ell_{2,\eta}(x)}\right)$$

where $(\alpha, \beta, \eta) \in]0, +\infty[^3$, and

- $\ell_{1,\alpha}(x) = \sum_{n=1}^{N}\left(\sqrt{x_n^2 + \alpha^2} - \alpha\right)$
- $\ell_{2,\eta}(x) = \sqrt{\sum_{n=1}^{N} x_n^2 + \eta^2}$

Smoothed ℓ_1/ℓ_2 sparsity measures

Common assumption: x has a sparse representation

Question: Which sparsity measure should be used?

Theoretically: ℓ_0 measure [Donoho *et al.*, 1995]

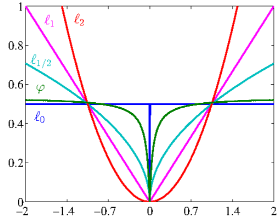
Usually: ℓ_1 measure

We propose a smoothed ℓ_1/ℓ_2 sparsity measures

Convex regularization

Blind image recovery (Alternating proximal algorithm) [Bolte *et al.*, 2010]

Blind video restoration [Abboud *et al.*, 2014]



Non-convex regularization

Image blind deconvolution [Krishnan, 2011]

SOOT algorithm [Repetti *et al.*, 2015]

New convergence proof

Observation model

$$\underbrace{y}_{\text{observed signal}} = \underbrace{\overline{h}}_{\text{original blur}} * \underbrace{\overline{x}}_{\text{reflectivity}} + \underbrace{b}_{\text{noise}}$$

where,

- $y \in \mathbb{R}^N$ observed signal
- $\overline{x} \in \mathbb{R}^N$ unknown sparse original seismic signal (reflectivity)
- $\overline{h} \in \mathbb{R}^S$ unknown original blur kernel
- $b \in \mathbb{R}^N$ additive noise: realization of a zero-mean white Gaussian noise with variance σ^2

Optimization problem

Find

$$(\widehat{x}, \widehat{h}) \in \underset{(x,h)\in\mathbb{R}^{N+S}}{\operatorname{Argmin}} \; \left(G(x,h) = F(x,h) + R(x,h)\right)$$

where

- $F : \mathbb{R}^{N+S} \to \mathbb{R}$ is differentiable, and has a L-Lipschitz gradient on $\operatorname{dom} R$.
- $R : \mathbb{R}^{N+S} \to]-\infty, +\infty]$ is proper, lower semicontinuous.
- G is coercive, i.e. $\lim_{\|z\|\to+\infty} G(z) = +\infty$, and is non necessarily convex.

Majorize-Minimize Algorithm

Find

$$\hat{z} \in \underset{z \in \mathbb{R}^{N+S}}{\text{Argmin}}\ R(z) + F(z)$$

MM Algorithm

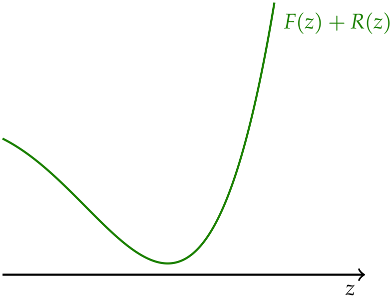
$$(\forall k \in \mathbb{N}) \quad z_{k+1} = \underset{z \in \mathbb{R}^{N+S}}{\text{Argmin}}\ R(z) + Q(z, z_k)$$

Quadratic majorant

$Q(z, \widetilde{z})$: quadratic majorant of the function F at $\widetilde{z}$ i.e.

$$Q(z,\widetilde{z}) = F(\widetilde{z}) + (z-\widetilde{z})^\top \nabla F(\widetilde{z}) + \frac{1}{2}(z-\widetilde{z})^\top A(\widetilde{z})(z-\widetilde{z})$$

where, $\forall z \in \mathbb{R}^{N+S}$, $F(z) \leq Q(z,\widetilde{z})$ and $F(\widetilde{z}) = Q(\widetilde{z},\widetilde{z})$



Majorize-Minimize Algorithm

Find

$$\hat{z} \in \underset{z \in \mathbb{R}^{N+S}}{\mathrm{Argmin}}\ R(z) + F(z)$$

MM Algorithm

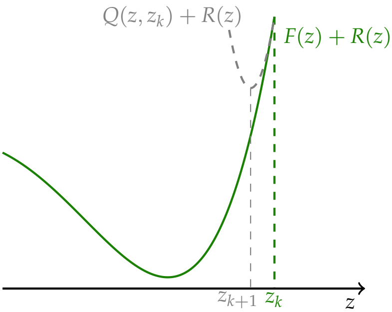
$$(\forall k \in \mathbb{N}) \quad z_{k+1} = \underset{z \in \mathbb{R}^{N+S}}{\mathrm{Argmin}}\ R(z) + Q(z, z_k)$$

Quadratic majorant

$Q(z, \widetilde{z})$: quadratic majorant of the function F at $\widetilde{z}$ i.e.

$$Q(z, \widetilde{z}) = F(\widetilde{z}) + (z - \widetilde{z})^\top \nabla F(\widetilde{z}) + \frac{1}{2}(z - \widetilde{z})^\top \gamma (z - \widetilde{z})$$

where, $\forall z \in \mathbb{R}^{N+S}$, $F(z) \leq Q(z, \widetilde{z})$ and $F(\widetilde{z}) = Q(\widetilde{z}, \widetilde{z})$



Majorize-Minimize Algorithm

Find

$$\hat{z} \in \underset{z\in\mathbb{R}^{N+S}}{\text{Argmin}}\ R(z) + F(z)$$

MM Algorithm

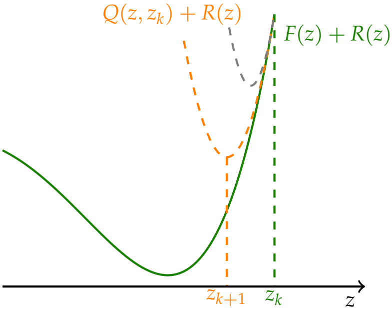
$$(\forall k \in \mathbb{N}) \quad z_{k+1} = \underset{z\in\mathbb{R}^{N+S}}{\text{Argmin}}\ R(z) + Q(z, z_k)$$

Quadratic majorant

$Q(z, \tilde{z})$: quadratic majorant of the function F at $\tilde{z}$ i.e.

$$Q(z,\tilde{z}) = F(\tilde{z}) + (z-\tilde{z})^\top \nabla F(\tilde{z}) + \frac{1}{2}(z-\tilde{z})^\top A(\tilde{z})(z-\tilde{z})$$

where, $\forall z \in \mathbb{R}^{N+S}$, $F(z) \leq Q(z,\tilde{z})$ and $F(\tilde{z}) = Q(\tilde{z},\tilde{z})$



Majorize-Minimize Algorithm

Find

$$\hat{z} \in \underset{z \in \mathbb{R}^{N+S}}{\operatorname{Argmin}}\, R(z) + F(z)$$

MM Algorithm

$$(\forall k \in \mathbb{N}) \quad z_{k+1} = \underset{z \in \mathbb{R}^{N+S}}{\operatorname{Argmin}}\, R(z) + Q(z, z_k)$$

Definition

Let $\widehat{z} \in \mathbb{R}^{N+S}$. The proximity operator of R relative to the metric induced by a SPD matrix $A \in \mathbb{R}^{(S+N)\times(S+N)}$ is defined by

$$\operatorname{prox}_{A,R}(\widehat{z}) = \underset{z \in \mathbb{R}^{N+S}}{\operatorname{minimize}}\, R(z) + \frac{1}{2}(z - \widehat{z})^\top A (z - \widehat{z})$$

Proposed criterion

Criterion

$$\underset{(x,h)\in\mathbb{R}^{N+S}}{\text{minimize}} \; \big(G(x,h) = F(x,h) + R(x,h)\big)$$

- $$F(x,h) = \underbrace{\rho(x,h)}_{\text{data fidelity term}} + \underbrace{\lambda\varphi(x)}_{\text{regularization term}}$$
- $\rho(x,h) = \frac{1}{2}\|h * x - y\|^2$
- $R(x,h) = R_1(x) + R_2(h)$
- $R_1(x) = \iota_{[x_{min},x_{max}]^N}(x)$ with $(x_{min}, x_{max}) \in]0,+\infty[^2$
- $R_2(h) = \iota_C(h)$
  with $C = \{h \in [h_{min}, h_{max}]^S \mid \|h\| \leq \delta\}$, for $(h_{min}, h_{max}, \delta) \in]0,+\infty[^3$

Quadratic majorants

Proposition

For every $(x, h) \in \mathbb{R}^N \times \mathbb{R}^S$, let

$$A_1(x,h) = \left(L_1(h) + \frac{9\lambda}{8\eta^2}\right) \mathrm{I}_N + \frac{\lambda}{\ell_{1,\alpha}(x) + \beta} A_{\ell_{1,\alpha}}(x),$$
$$A_2(x,h) = L_2(x)\, \mathrm{I}_S,$$

where

$$A_{\ell_{1,\alpha}}(x) = \mathrm{Diag}\left(\left(({x_n}^2 + \alpha^2)^{-1/2}\right)_{1 \le n \le N}\right),$$

and $L_1(h)$ (resp. $L_2(x)$) is a Lipschitz constant for $\nabla_1 \rho(\cdot, h)$ (resp. $\nabla_2 \rho(x, \cdot)$). Then, $A_1(x,h)$ (resp. $A_2(x,h)$) satisfies the majoration condition for $F(\cdot, h)$ at x (resp. $F(x, \cdot)$ at h).

♣

SOOT algorithm

For every $k \in \mathbb{N}$, let $J_k \in \mathbb{N}^*$, $I_k \in \mathbb{N}^*$ and let $(\gamma_x^{k,j})_{0 \le j \le J_k - 1}$ and $(\gamma_h^{k,i})_{0 \le i \le I_k - 1}$ be positive sequences. Initialize with $x^0 \in \operatorname{dom} R_1$ and $h^0 \in \operatorname{dom} R_2$.

Iterations:

For $k = 0, 1, \ldots$

$$
\left\lfloor
\begin{array}{l}
x^{k,0} = x^k, \; h^{k,0} = h^k, \\
\text{For } j = 0, \ldots, J_k - 1 \\
\left\lfloor
\begin{array}{l}
\widetilde{x}^{k,j} = x^{k,j} - \gamma_x^{k,j} A_1(x^{k,j}, h^k)^{-1} \nabla_1 F(x^{k,j}, h^k), \\
x^{k,j+1} = \operatorname{prox}_{(\gamma_x^{k,j})^{-1} A_1(x^{k,j}, h^k), R_1} \left(\widetilde{x}^{k,j} \right),
\end{array}
\right. \\
x^{k+1} = x^{k,J_k}. \\
\text{For } i = 0, \ldots, I_k - 1 \\
\left\lfloor
\begin{array}{l}
\widetilde{h}^{k,i} = h^{k,i} - \gamma_h^{k,i} A_2(x^{k+1}, h^{k,i})^{-1} \nabla_2 F(x^{k+1}, h^{k,i}), \\
h^{k,i+1} = \operatorname{prox}_{(\gamma_h^{k,i})^{-1} A_2(x^{k+1}, h^{k,i}), R_2} \left(\widetilde{h}^{k,i} \right),
\end{array}
\right. \\
h^{k+1} = h^{k,I_k}.
\end{array}
\right.
$$

SOOT algorithm

For every $k \in \mathbb{N}$, let $J_k \in \mathbb{N}^*$, $I_k \in \mathbb{N}^*$ and let $(\gamma_x^{k,j})_{0 \leq j \leq J_k - 1}$ and $(\gamma_h^{k,i})_{0 \leq i \leq I_k - 1}$ be positive sequences. Initialize with $x^0 \in \operatorname{dom} R_1$ and $h^0 \in \operatorname{dom} R_2$.

Iterations:

For $k = 0, 1, \ldots$

$$
\left\lfloor
\begin{array}{l}
x^{k,0} = x^k, \;\; h^{k,0} = h^k, \\
\text{For } j = 0, \ldots, J_k - 1 \\
\left\lfloor
\begin{array}{l}
\widetilde{x}^{k,j} = x^{k,j} - \gamma_x^{k,j} A_1(x^{k,j}, h^k)^{-1} \nabla_1 F(x^{k,j}, h^k), \\
x^{k,j+1} = \operatorname{prox}_{(\gamma_x^{k,j})^{-1} A_1(x^{k,j}, h^k), R_1} \left(\widetilde{x}^{k,j} \right),
\end{array}
\right. \\
x^{k+1} = x^{k,J_k}. \\
\text{For } i = 0, \ldots, I_k - 1 \\
\left\lfloor
\begin{array}{l}
\widetilde{h}^{k,i} = h^{k,i} - \gamma_h^{k,i} A_2(x^{k+1}, h^{k,i})^{-1} \nabla_2 F(x^{k+1}, h^{k,i}), \\
h^{k,i+1} = \operatorname{prox}_{(\gamma_h^{k,i})^{-1} A_2(x^{k+1}, h^{k,i}), R_2} \left(\widetilde{h}^{k,i} \right),
\end{array}
\right. \\
h^{k+1} = h^{k,I_k}.
\end{array}
\right.
$$

SOOT algorithm

For every $k \in \mathbb{N}$, let $J_k \in \mathbb{N}^*$, $I_k \in \mathbb{N}^*$ and let $(\gamma_x^{k,j})_{0 \le j \le J_k - 1}$ and $(\gamma_h^{k,i})_{0 \le i \le I_k - 1}$ be positive sequences. Initialize with $x^0 \in \operatorname{dom} R_1$ and $h^0 \in \operatorname{dom} R_2$.

Iterations:

For $k = 0, 1, \ldots$

$$
\left\lfloor
\begin{array}{l}
x^{k,0} = x^k, \; h^{k,0} = h^k, \\
\text{For } j = 0, \ldots, J_k - 1 \\
\left\lfloor
\begin{array}{l}
\widetilde{x}^{k,j} = x^{k,j} - \gamma_x^{k,j} A_1(x^{k,j}, h^k)^{-1} \nabla_1 F(x^{k,j}, h^k), \\
x^{k,j+1} = \operatorname{prox}_{(\gamma_x^{k,j})^{-1} A_1(x^{k,j}, h^k), R_1} \left(\widetilde{x}^{k,j} \right),
\end{array}
\right. \\
x^{k+1} = x^{k,J_k}. \\
\text{For } i = 0, \ldots, I_k - 1 \\
\left\lfloor
\begin{array}{l}
\widetilde{h}^{k,i} = h^{k,i} - \gamma_h^{k,i} A_2(x^{k+1}, h^{k,i})^{-1} \nabla_2 F(x^{k+1}, h^{k,i}), \\
h^{k,i+1} = \operatorname{prox}_{(\gamma_h^{k,i})^{-1} A_2(x^{k+1}, h^{k,i}), R_2} \left(\widetilde{h}^{k,i} \right),
\end{array}
\right. \\
h^{k+1} = h^{k,I_k}.
\end{array}
\right.
$$

Convergence results

Theorem

Let $(x^k)_{k\in\mathbb{N}}$ and $(h^k)_{k\in\mathbb{N}}$ be sequences generated by SOOT Algorithm.

1. There exists $(\underline{\nu},\overline{\nu}) \in]0,+\infty[^2$ such that, for all $k \in \mathbb{N}$,

$$(\forall j \in \{0,\dots,J_k-1\}) \quad \underline{\nu}\,\mathrm{I}_N \preceq A_1(x^{k,j},h^k) \preceq \overline{\nu}\,\mathrm{I}_N,$$
$$(\forall i \in \{0,\dots,I_k-1\}) \quad \underline{\nu}\,\mathrm{I}_S \preceq A_2(x^{k+1},h^{k,i}) \preceq \overline{\nu}\,\mathrm{I}_S.$$

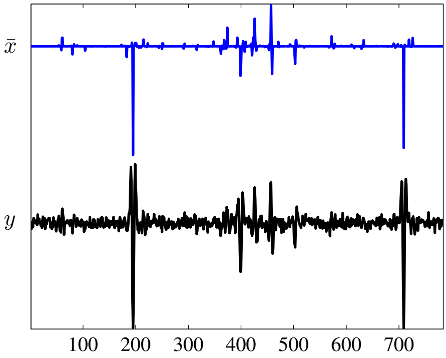
2. Set $(\underline{\gamma},\overline{\gamma}) \in]0,+\infty[^2$, for all $k \in \mathbb{N}$, $j \in \{0,\dots,J_k-1\}$ and $i \in \{0,\dots,I_k-1\}$, $(\gamma_x^{k,j},\gamma_h^{k,i}) \in [\underline{\gamma}, 2-\overline{\gamma}]^2$
3. R_1, R_2 are the semi-algebraic functions.

⇝ $(x^k,h^k)_{k\in\mathbb{N}}$ converges to a critical point $(\widehat{x},\widehat{h})$ of G.
⇝ $\big(G(x^k,h^k)\big)_{k\in\mathbb{N}}$ is a nonincreasing sequence converging to $G(\widehat{x},\widehat{h})$.

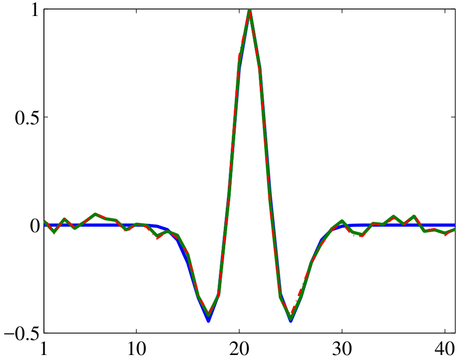
♣

Results (1D)

Reflectivity ($N = 784$)
Observed signal



Results (1D)

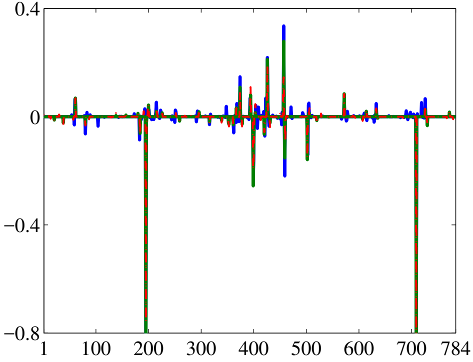


Reflectivity ($N = 784$)
Observed signal

Blur ($S = 41$)

- Original
- [Krishnan, 2011]: SNR = 19.12 dB
  Time = 41 s.
- SOOT: SNR = 20.98 dB
  Time = 14 s.

Results (1D)



Reflectivity ($N = 784$)
Observed signal

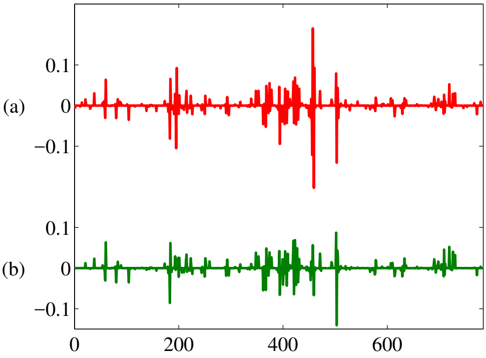
Blur ($S = 41$)

- Original
- [Krishnan, 2011]: SNR = 19.12 dB
  Time = 41 s.
- SOOT: SNR = 20.98 dB
  Time = 14 s.

Reflectivity ($N = 784$)

- Original
- [Krishnan, 2011]: SNR = 8.60 dB
- SOOT: SNR = 11.44 dB

Results (1D)



Reflectivity ($N = 784$)
Observed signal

Blur ($S = 41$)

- Original
- [Krishnan, 2011]: SNR = 19.12 dB
  Time = 41 s.
- SOOT: SNR = 20.98 dB
  Time = 14 s.

Reflectivity ($N = 784$)

- Original
- [Krishnan, 2011]: SNR = 8.60 dB
- SOOT: SNR = 11.44 dB

Errors

- [Krishnan, 2011]: $\ell_1 = 5.5 \times 10^{-3}$
- SOOT: $\ell_1 = 4.2 \times 10^{-3}$

Summary

Noise level (σ)			0.01	0.02	0.03
Observation error		$\ell_2(\times 10^{-2})$	7.14	7.35	7.68
		$\ell_1(\times 10^{-2})$	2.85	3.44	4.09
Signal error	[Krishnan, 2011]	$\ell_2(\times 10^{-2})$	1.23	1.66	1.84
		$\ell_1(\times 10^{-3})$	3.79	4.69	5.30
	SOOT algorithm	$\ell_2(\times 10^{-2})$	1.09	1.63	1.83
		$\ell_1(\times 10^{-3})$	3.42	4.30	4.85
Kernel error	[Krishnan, 2011]	$\ell_2(\times 10^{-2})$	1.88	2.51	3.21
		$\ell_1(\times 10^{-2})$	1.44	1.96	2.53
	SOOT algorithm	$\ell_2(\times 10^{-2})$	1.62	2.26	2.93
		$\ell_1(\times 10^{-2})$	1.22	1.77	2.31
Time (s.)	[Krishnan, 2011]		106	61	56
	SOOT algorithm		56	22	18

Comparison between [Krishnan, 2011] and SOOT algorithm for $\bar{x}$ and $\bar{h}$ estimates (Intel(R) Xeon(R) CPU E5-2609 v2@2.5GHz using Matlab 8).

Image blind deconvolution

Original image
- size 300×300

Image blind deconvolution

Original image
- size 300×300

Observed image
- SNR = 13.90 dB
- SSIM = 0.92

Image blind deconvolution

Original image
- size 300 × 300

Observed image
- SNR = 13.90 dB
- SSIM = 0.92

Reconstructed image by

[Krishnan, 2011]
- SNR = 15.19 dB
- SSIM = 0.94
- Time = 2 min.

Image blind deconvolution

Original image
- size 300×300

Observed image
- SNR = 13.90 dB
- SSIM = 0.92

Reconstructed image by

[Krishnan, 2011]
- SNR = 15.19 dB
- SSIM = 0.94
- Time = 2 min.

SOOT
- SNR = 16.06 dB
- SSIM = 0.94
- Time = 7 s.

Image blind deconvolution

Observed image
- size 512 × 512
- SNR = 19.66 dB
- SSIM = 0.93

Image blind deconvolution

Observed image
- size 512 × 512
- SNR = 19.66 dB
- SSIM = 0.93

Reconstructed image by

[Krishnan, 2011]
- SNR = 20.96 dB
- SSIM = 0.95
- Time = 7 min.

Image blind deconvolution

Observed image
- size 512×512
- SNR = 19.66 dB
- SSIM = 0.93

Reconstructed image by

[Krishnan, 2011]
- SNR = 20.96 dB
- SSIM = 0.95
- Time = 7 min.

SOOT
- SNR = 22.80 dB
- SSIM = 0.97
- Time = 45 s.

Contributions:

- A generic methodology to impose sparsity and regularity properties through constrained adaptive filtering in a transformed domain.
- Versatility of the proposed optimization framework which permits different strategies for sparse modeling, and adaptation constraints.
- Smooth parametric approximations to the ℓ_1/ℓ_2 norm ratio.
- Convergence results both on iterates and function values.
- Blocks updated according to a flexible quasi-cyclic rule.
- Acceleration of the convergence thanks to the choice of matrices $\left(A_{j\ell}(x_\ell)\right)_{\ell\in\mathbb{N}}$ based on MM principle.
- Application to sparse blind deconvolution.
- Results demonstrated on seismic reflectivity, acoustic source localization, biological/medical image retoration.

Software:

- SOOT-Blind deconvolution: (http://lc.cx/soot)