DM Optimization
Piecewise constant denoising

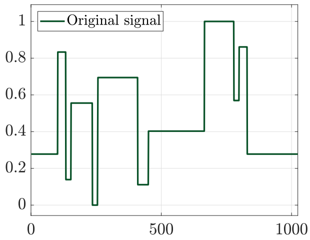
Barbara Pascal, Nelly Pustelnik

CNRS, Laboratoire de Physique de l'ENS de Lyon, Univ. Lyon 1
nelly.pustelnik@ens-lyon.fr

barbara.pascal@ens-lyon.fr

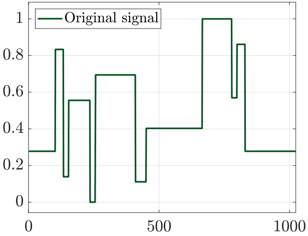
Piecewise noisy signal

Ground truth

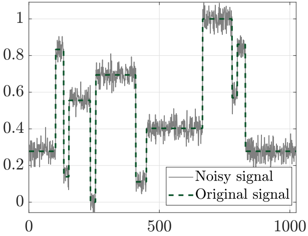


Piecewise noisy signal

Ground truth



Gaussian noise with $\sigma = 0.04$



Purpose: recover the true signal with *sharp* transitions

Denoising by functional minimization

Regularized scheme *D: differential operator, $\|\cdot\|_p$: ℓ_p-norm*

$$\widehat{x}_\lambda = \underset{x\in\mathbb{R}^N}{\arg\min}\ \frac{1}{2}\|x-y\|_2^2 + \lambda\|Dx\|_p^p$$

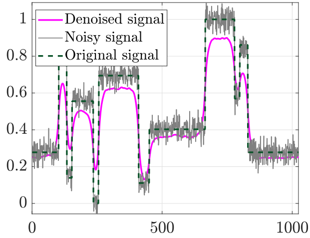
Denoising by functional minimization

Regularized scheme — *D: differential operator*, $\|\cdot\|_p$: ℓ_p*-norm*

$$\widehat{x}_\lambda = \arg\min_{x\in\mathbb{R}^N} \frac{1}{2}\|x-y\|_2^2 + \lambda\|Dx\|_p^p$$

Tikhonov regularizer $\|Lx\|_2^2$

Smooth: gradient descent



✗ fuzzy transitions

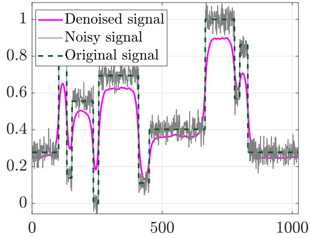
Denoising by functional minimization

Regularized scheme — D: *differential operator*, $\|\cdot\|_p$: ℓ_p-*norm*

$$\widehat{x}_\lambda = \arg\min_{x\in\mathbb{R}^N} \frac{1}{2}\|x-y\|_2^2 + \lambda\|Dx\|_p^p$$

Tikhonov regularizer $\|Lx\|_2^2$

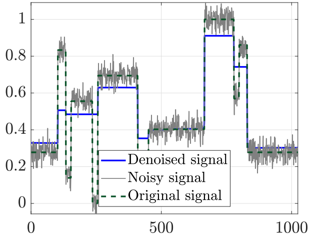
Smooth: gradient descent



✗ fuzzy transitions

Total Variation $\|Lx\|_1$

Nonsmooth: proximal algorithm



✓ sharp transitions

Formulation of the problem

Piecewise denoising

$$\widehat{x}_\lambda = \underset{x \in \mathbb{R}^N}{\arg\min}\ \frac{1}{2}\|x - y\|_2^2 + \lambda \|Lx\|_1$$

Formulation of the problem

Piecewise denoising

$$\widehat{x}_\lambda = \underset{x \in \mathbb{R}^N}{\arg\min}\ \frac{1}{2}\|x - y\|_2^2 + \lambda \|Lx\|_1$$

- *Smooth* data-fidelity $\quad f(x) = \frac{1}{2}\|x - y\|_2^2$

Formulation of the problem

Piecewise denoising

$$\widehat{x}_\lambda = \underset{x \in \mathbb{R}^N}{\arg\min}\ \frac{1}{2}\|x - y\|_2^2 + \lambda \|Lx\|_1$$

- *Smooth* data-fidelity $\quad f(x) = \frac{1}{2}\|x - y\|_2^2$
- *Non-smooth* regularizer $\quad g_\lambda(Lx) = \lambda\|Lx\|_1, \quad$ with $g_\lambda(z) = \lambda\|z\|_1$

Formulation of the problem

Piecewise denoising

$$\widehat{x}_\lambda = \underset{x \in \mathbb{R}^N}{\arg\min} \; \frac{1}{2}\|x - y\|_2^2 + \lambda \|Lx\|_1$$

- *Smooth* data-fidelity $\quad f(x) = \frac{1}{2}\|x - y\|_2^2$
- *Non-smooth* regularizer $\quad g_\lambda(Lx) = \lambda \|Lx\|_1, \quad$ with $g_\lambda(z) = \lambda\|z\|_1$

General form:

$$\widehat{x} = \underset{x \in \mathbb{R}^N}{\arg\min} \; f(x) + g(Lx)$$

f smooth, g nonsmooth.

Why turning to the *dual* problem?

Forward-backward algorithm $(\forall n \in \mathbb{N}) \quad \lambda_n = 1$

$$x_{n+1} = \operatorname{prox}_{\gamma g \circ L} \left(x_n - \gamma \nabla f(x_n) \right)$$

Why turning to the *dual* problem?

Forward-backward algorithm $(\forall n \in \mathbb{N}) \quad \lambda_n = 1$

$$x_{n+1} = \mathrm{prox}_{\gamma g \circ L}\left(x_n - \gamma \nabla f(x_n)\right)$$

- Need to compute $\mathrm{prox}_{\gamma g \circ L}$

Why turning to the *dual* problem?

***Forward-backward* algorithm** $(\forall n \in \mathbb{N}) \quad \lambda_n = 1$

$$x_{n+1} = \operatorname{prox}_{\gamma g \circ L}\left(x_n - \gamma \nabla f(x_n)\right)$$

- Need to compute $\operatorname{prox}_{\gamma g \circ L}$

$$x_{\text{prox}} = \operatorname{prox}_{\gamma g \circ L}(u) = \underset{x \in \mathbb{R}^N}{\arg\min}\ \frac{1}{2}\|x - u\|_2^2 + \gamma g(Lx)$$

Why turning to the *dual* problem?

Forward-backward **algorithm** $(\forall n \in \mathbb{N}) \quad \lambda_n = 1$

$$x_{n+1} = \mathrm{prox}_{\gamma g \circ L}\left(x_n - \gamma \nabla f(x_n)\right)$$

- Need to compute $\mathrm{prox}_{\gamma g \circ L}$

$$x_{\mathrm{prox}} = \mathrm{prox}_{\gamma g \circ L}(u) = \underset{x \in \mathbb{R}^N}{\arg\min}\ \frac{1}{2}\|x - u\|_2^2 + \gamma g(Lx)$$

Fermat's rule: $0 \in x_{\mathrm{prox}} - u + \gamma L^* \partial g(L x_{\mathrm{prox}})$

Why turning to the *dual* problem?

Forward-backward algorithm $(\forall n \in \mathbb{N}) \quad \lambda_n = 1$

$$x_{n+1} = \mathrm{prox}_{\gamma g \circ L}\left(x_n - \gamma \nabla f(x_n)\right)$$

- Need to compute $\mathrm{prox}_{\gamma g \circ L}$

$$x_{\mathrm{prox}} = \mathrm{prox}_{\gamma g \circ L}(u) = \underset{x \in \mathbb{R}^N}{\arg\min}\ \frac{1}{2}\|x - u\|_2^2 + \gamma g(Lx)$$

Fermat's rule: $0 \in x_{\mathrm{prox}} - u + \gamma L^* \partial g(L x_{\mathrm{prox}}) \qquad x_{\mathrm{prox}} = \ldots?$

Formulation of the *dual* problem

Primal problem

$$\widehat{x} = \arg\min_{x \in \mathbb{R}^N} f(x) + g(Lx)$$

Formulation of the *dual* problem

Primal problem

$$\widehat{x} = \underset{x \in \mathbb{R}^N}{\arg\min}\ f(x) + g(Lx)$$

Dual problem

$$\widehat{u} = \underset{u \in \mathbb{R}^N}{\arg\min}\ f^*(-L^*u) + g^*(u)$$

Formulation of the *dual* problem

Primal problem

$$\widehat{x} = \arg\min_{x\in\mathbb{R}^N} f(x) + g(Lx)$$

Dual problem

$$\widehat{u} = \arg\min_{u\in\mathbb{R}^N} f^*(-L^*u) + g^*(u)$$

Piecewise denoising

- $f^*(z) = \frac{1}{2}\|y+z\|_2^2 \underbrace{- \frac{1}{2}\|y\|_2^2}_{\text{constant}}$
- $g_\lambda^*(u) = \iota_{\|\cdot\|_\infty \leq \lambda}(u)$

$$\widehat{u}_\lambda \in \operatorname*{Argmin}_{u\in\mathbb{R}^N} \frac{1}{2}\|y - L^*u\|_2^2 + \iota_{\|\cdot\|_\infty \leq \lambda}(u)$$

Link between the *primal* and the *dual* solutions

$-L^*\widehat{u} \in \partial f(\widehat{x})$ and $L\widehat{x} \in \partial g^*(\widehat{u})$

Link between the *primal* and the *dual* solutions

$$-L^*\widehat{u} \in \partial f(\widehat{x}),\ \partial f(\widehat{x}) = \{\nabla f(\widehat{x})\} \qquad \text{and} \qquad L\widehat{x} \in \partial g^*(\widehat{u})$$

Piecewise denoising

Gradient of the data-fidelity term: $\nabla f(\widehat{x}) = x - y$

- $-L^*\widehat{u}_\lambda = \widehat{x}_\lambda - y \quad \Longleftrightarrow \quad \widehat{x}_\lambda = y - L^*\widehat{u}_\lambda$

Proximity operators

- ℓ_1 norm

$$x_{\mathrm{prox}} = \mathrm{prox}_{\lambda\|\cdot\|_1}(x)$$
$$\iff$$
$$(\forall i \in \{1,\ldots,N\}) \quad x_{\mathrm{prox}}^{(i)} = \max\left\{0, 1 - \frac{\lambda}{|x^{(i)}|}\right\} x^{(i)}$$

- quadratic data-fidelity term

$$x_{\mathrm{prox}} = \mathrm{prox}_{\frac{\gamma}{2}\|\cdot - y\|_2^2}(x) \iff x_{\mathrm{prox}} = \frac{x + \gamma y}{1+\gamma}$$

Proximity operators

- squared norm composed with a linear operator

$$x_{\mathrm{prox}} = \mathrm{prox}_{\frac{\gamma}{2}\|y - L^* \cdot\|_2^2}(x) \iff x_{\mathrm{prox}} = \left(\mathrm{Id} + \gamma L L^*\right)^{-1}(x + \gamma L y)$$

Implementation of *dual forward-backward* algorithm

$$\widehat{u}_\lambda \in \underset{u \in \mathbb{R}^N}{\operatorname{Argmin}} \ \underbrace{\frac{1}{2}\|y - L^* u\|_2^2}_{:= h(u)} + \iota_{\|\cdot\|_\infty \leq \lambda}(u)$$

- $h \in \Gamma_0(\mathbb{R}^N)$ differentiable, with gradient $\nabla h(u) = L\left(L^* u - y\right)$

Implementation of *dual forward-backward* algorithm

$$\widehat{u}_\lambda \in \operatorname*{Argmin}_{u \in \mathbb{R}^N} \underbrace{\frac{1}{2}\|y - L^* u\|_2^2}_{:=h(u)} + \iota_{\|\cdot\|_\infty \leq \lambda}(u)$$

- $h \in \Gamma_0(\mathbb{R}^N)$ differentiable, with gradient $\nabla h(u) = L\left(L^* u - y\right)$

  $\nabla h(u)$ is ν-Lipschitzian, $\nu = \|L\|^2$

Implementation of *dual forward-backward* algorithm

$$\widehat{u}_\lambda \in \underset{u \in \mathbb{R}^N}{\operatorname{Argmin}} \ \underbrace{\frac{1}{2}\|y - L^* u\|_2^2}_{:= h(u)} + \iota_{\|\cdot\|_\infty \leq \lambda}(u)$$

- $h \in \Gamma_0(\mathbb{R}^N)$ differentiable, with gradient $\nabla h(u) = L\,(L^* u - y)$
  $\nabla h(u)$ is ν-Lipschitzian, $\nu = \|L\|^2$
- choose $\gamma \in]0, 2/\nu[$,

Implementation of *dual forward-backward* algorithm

$$\widehat{u}_\lambda \in \operatorname*{Argmin}_{u \in \mathbb{R}^N} \underbrace{\frac{1}{2}\|y - L^* u\|_2^2}_{:=h(u)} + \iota_{\|\cdot\|_\infty \leq \lambda}(u)$$

- $h \in \Gamma_0(\mathbb{R}^N)$ differentiable, with gradient $\nabla h(u) = L\,(L^* u - y)$

  $\nabla h(u)$ is ν-Lipschitzian, $\nu = \|L\|^2$
- choose $\gamma \in]0, 2/\nu[$, set $\delta = \min\{1, (\gamma\nu)^{-1}\} + 1/2$

Implementation of *dual forward-backward* algorithm

$$\widehat{u}_\lambda \in \operatorname*{Argmin}_{u \in \mathbb{R}^N} \underbrace{\frac{1}{2}\|y - L^* u\|_2^2}_{:=h(u)} + \iota_{\|\cdot\|_\infty \leq \lambda}(u)$$

- $h \in \Gamma_0(\mathbb{R}^N)$ differentiable, with gradient $\nabla h(u) = L\,(L^* u - y)$
  $\nabla h(u)$ is ν-Lipschitzian, $\nu = \|L\|^2$
- choose $\gamma \in]0, 2/\nu[$, set $\delta = \min\{1, (\gamma\nu)^{-1}\} + 1/2$
  e.g.: $\gamma = 1.99/\|L\|^2$

Implementation of *dual forward-backward* algorithm

$$\widehat{u}_\lambda \in \underset{u \in \mathbb{R}^N}{\mathrm{Argmin}} \; \underbrace{\frac{1}{2}\|y - L^* u\|_2^2}_{:=h(u)} + \iota_{\|\cdot\|_\infty \leq \lambda}(u)$$

- $h \in \Gamma_0(\mathbb{R}^N)$ differentiable, with gradient $\nabla h(u) = L\,(L^* u - y)$
  $\nabla h(u)$ is ν-Lipschitzian, $\nu = \|L\|^2$
- choose $\gamma \in]0, 2/\nu[$, set $\delta = \min\{1, (\gamma\nu)^{-1}\} + 1/2$
  e.g.: $\gamma = 1.99/\|L\|^2$
- choose $(\lambda_n)_{n\in\mathbb{N}}$ such that $\lambda_n \in [0, \delta[$ and $\sum_{n\in\mathbb{N}} \lambda_n(\delta - \lambda_n) = \infty$

Implementation of *dual forward-backward* algorithm

$$\widehat{u}_\lambda \in \underset{u \in \mathbb{R}^N}{\mathrm{Argmin}} \; \underbrace{\frac{1}{2}\|y - L^* u\|_2^2}_{:=h(u)} + \iota_{\|\cdot\|_\infty \leq \lambda}(u)$$

- $h \in \Gamma_0(\mathbb{R}^N)$ differentiable, with gradient $\nabla h(u) = L\,(L^* u - y)$
  $\nabla h(u)$ is ν-Lipschitzian, $\nu = \|L\|^2$
- choose $\gamma \in]0, 2/\nu[$, set $\delta = \min\{1, (\gamma\nu)^{-1}\} + 1/2$
  e.g.: $\gamma = 1.99/\|L\|^2$
- choose $(\lambda_n)_{n\in\mathbb{N}}$ such that $\lambda_n \in [0, \delta[$ and $\sum_{n\in\mathbb{N}} \lambda_n(\delta - \lambda_n) = \infty$
  e.g.: $\delta = 1.0025$, $(\forall n \in \mathbb{N}) \quad \lambda_n = 1$

Implementation of *dual forward-backward* algorithm

$$\widehat{u}_\lambda \in \operatorname*{Argmin}_{u\in\mathbb{R}^N} \underbrace{\frac{1}{2}\|y - L^* u\|_2^2}_{:=h(u)} + \iota_{\|\cdot\|_\infty \leq \lambda}(u)$$

- $h \in \Gamma_0(\mathbb{R}^N)$ differentiable, with gradient $\nabla h(u) = L\,(L^* u - y)$
  $\nabla h(u)$ is ν-Lipschitzian, $\nu = \|L\|^2$
- choose $\gamma \in]0, 2/\nu[$, set $\delta = \min\{1, (\gamma\nu)^{-1}\} + 1/2$
  e.g.: $\gamma = 1.99/\|L\|^2$
- choose $(\lambda_n)_{n\in\mathbb{N}}$ such that $\lambda_n \in [0, \delta[$ and $\sum_{n\in\mathbb{N}} \lambda_n(\delta - \lambda_n) = \infty$
  e.g.: $\delta = 1.0025$, $(\forall n \in \mathbb{N}) \quad \lambda_n = 1$

$$u_{n+1} = \operatorname{prox}_{\gamma\iota_{\|\cdot\|_\infty \leq \lambda}}\left(u_n - \gamma\nabla h(u_n)\right)$$

Implementation of *dual forward-backward* algorithm

$$\widehat{u}_\lambda \in \operatorname*{Argmin}_{u \in \mathbb{R}^N} \underbrace{\frac{1}{2}\|y - L^* u\|_2^2}_{:=h(u)} + \iota_{\|\cdot\|_\infty \leq \lambda}(u)$$

$$\begin{aligned} w_{n+1} &= u_n - \gamma L(L^* u_n - y) \\ u_{n+1} &= w_{n+1} - \gamma \mathrm{prox}_{\lambda/\gamma \|\cdot\|_1}(w_{n+1}/\gamma) = \mathrm{Proj}_{\|\cdot\|_\infty \leq \lambda}(w_{n+1}) \\ x_{n+1} &= y - L^* u_{n+1} \end{aligned}$$

Implementation of *dual forward-backward* algorithm

$$\widehat{u}_\lambda \in \underset{u\in\mathbb{R}^N}{\mathrm{Argmin}} \ \underbrace{\frac{1}{2}\|y - L^* u\|_2^2}_{:=h(u)} + \iota_{\|\cdot\|_\infty \leq \lambda}(u)$$

$$w_{n+1} = u_n - \gamma L(L^* u_n - y)$$
$$u_{n+1} = w_{n+1} - \gamma \mathrm{prox}_{\lambda/\gamma \|\cdot\|_1}(w_{n+1}/\gamma) = \mathrm{Proj}_{\|\cdot\|_\infty \leq \lambda}(w_{n+1})$$
$$x_{n+1} = y - L^* u_{n+1}$$

Measure convergence:

- *dual* functional $\mathcal{D}_n = 1/2\|y - L^* u_n\|_2^2 - 1/2\|y\|_2^2 + \iota_{\|\cdot\|_\infty \leq \lambda}(u_n)$

Implementation of *dual forward-backward* algorithm

$$\widehat{u}_\lambda \in \underset{u\in\mathbb{R}^N}{\mathrm{Argmin}} \underbrace{\frac{1}{2}\|y - L^* u\|_2^2}_{:=h(u)} + \iota_{\|\cdot\|_\infty \leq \lambda}(u)$$

$$w_{n+1} = u_n - \gamma L(L^* u_n - y)$$
$$u_{n+1} = w_{n+1} - \gamma \mathrm{prox}_{\lambda/\gamma\|\cdot\|_1}(w_{n+1}/\gamma) = \mathrm{Proj}_{\|\cdot\|_\infty\leq\lambda}(w_{n+1})$$
$$x_{n+1} = y - L^* u_{n+1}$$

Measure convergence:

- *dual* functional $\mathcal{D}_n = 1/2\|y - L^* u_n\|_2^2 - 1/2\|y\|_2^2 + \iota_{\|\cdot\|_\infty\leq\lambda}(u_n)$

Implementation of *dual forward-backward* algorithm

$$\widehat{u}_\lambda \in \operatorname*{Argmin}_{u \in \mathbb{R}^N} \underbrace{\frac{1}{2}\|y - L^* u\|_2^2}_{:=h(u)} + \iota_{\|\cdot\|_\infty \leq \lambda}(u)$$

$$\begin{aligned} w_{n+1} &= u_n - \gamma L(L^* u_n - y) \\ u_{n+1} &= w_{n+1} - \gamma \operatorname{prox}_{\lambda/\gamma \|\cdot\|_1}(w_{n+1}/\gamma) = \operatorname{Proj}_{\|\cdot\|_\infty \leq \lambda}(w_{n+1}) \\ x_{n+1} &= y - L^* u_{n+1} \end{aligned}$$

Measure convergence:

- *dual* functional $\mathcal{D}_n = 1/2\|y - L^* u_n\|_2^2 - 1/2\|y\|_2^2 + \iota_{\|\cdot\|_\infty \leq \lambda}(u_n)$
- *primal* functional $\mathcal{P}_n = 1/2\|x_n - y\|_2^2 + \lambda\|Lx_n\|_1$

Implementation of *dual forward-backward* algorithm

$$\widehat{u}_\lambda \in \operatorname*{Argmin}_{u \in \mathbb{R}^N} \underbrace{\frac{1}{2}\|y - L^* u\|_2^2}_{:=h(u)} + \iota_{\|\cdot\|_\infty \leq \lambda}(u)$$

$$\begin{aligned} w_{n+1} &= u_n - \gamma L(L^* u_n - y) \\ u_{n+1} &= w_{n+1} - \gamma \mathrm{prox}_{\lambda/\gamma\|\cdot\|_1}(w_{n+1}/\gamma) = \mathrm{Proj}_{\|\cdot\|_\infty \leq \lambda}(w_{n+1}) \\ x_{n+1} &= y - L^* u_{n+1} \end{aligned}$$

Measure convergence:

- *dual* functional $\mathcal{D}_n = 1/2\|y - L^* u_n\|_2^2 - 1/2\|y\|_2^2 + \iota_{\|\cdot\|_\infty \leq \lambda}(u_n)$
- *primal* functional $\mathcal{P}_n = 1/2\|x_n - y\|_2^2 + \lambda\|Lx_n\|_1$
- *duality gap* $\mathcal{G}_n := \mathcal{P}_n - \mathcal{D}_n \rightarrow 0$

Measuring the quality of the solution

SNR (Signal-to-Noise Ratio)

$$\mathrm{PSNR}(\bar{x}, \widehat{x}) := 20 \log_{10} \left(\frac{\|\bar{x}\|_2^2}{\|\widehat{x} - \bar{x}\|_2^2} \right)$$

$\bar{x}$: ground truth, $\widehat{x}$: denoised signal

High SNR indicates good estimation performance.

Interesting questions to investigate

For each *algorithm*

- does-it converge?

  convergence of *primal* and *dual* functionals, *duality gap*, SNR

Interesting questions to investigate

For each *algorithm*

- does-it converge?

  convergence of *primal* and *dual* functionals, *duality gap*, SNR

- influence of the choice of descent stepγ?

  forward-backward algorithm $\gamma = \{0.01, 1.99, 2.01\}/\|L\|^2$

Interesting questions to investigate

For each *algorithm*

- does-it converge?

  convergence of *primal* and *dual* functionals, *duality gap*, SNR

- influence of the choice of descent stepγ?

  forward-backward algorithm $\gamma = \{0.01, 1.99, 2.01\}/\|L\|^2$

Compare algorithms

- number of iterations to reach *same* value of $f(x) + g(Lx)$?

Interesting questions to investigate

For each *algorithm*

- does-it converge?

  convergence of *primal* and *dual* functionals, *duality gap*, SNR

- influence of the choice of descent stepγ?

  forward-backward algorithm $\gamma = \{0.01, 1.99, 2.01\}/\|L\|^2$

Compare algorithms

- number of iterations to reach *same* value of $f(x) + g(Lx)$?
- *fixed* number of iteration, which achieves the lowest objective value?

Interesting questions to investigate

For each *algorithm*

- does-it converge?

  convergence of *primal* and *dual* functionals, *duality gap*, SNR

- influence of the choice of descent step γ?

  forward-backward algorithm $\gamma = \{0.01, 1.99, 2.01\}/\|L\|^2$

Compare algorithms

- number of iterations to reach *same* value of $f(x) + g(Lx)$?
- *fixed* number of iteration, which achieves the lowest objective value?

For *one* given algorithm and choice of parameters

- influence of regularization parameter λ

  which value of λ leads to the largest final SNR?