Solving total variation denoising using proximal gradient with ADMM

A simple illustration

Andersen Ang

Department of Combinatorics and Optimization, University of Waterloo, Canada

ms$\boldsymbol{x}$ang@uwaterloo.ca, where $\boldsymbol{x} = \lfloor \pi \rfloor$

Homepage: angms.science

First draft: June 17, 2022 Last update: June 21, 2022

A simple 1-dimensional Total Variation denoising problem

$$\mathcal{P} \;:\; \operatorname*{argmin}_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{D}\boldsymbol{x}\|_1.$$

- $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is a blurring matrix
- $\boldsymbol{b} \in \mathbb{R}^{n}$ is the observed blurred noisy signal
- $\boldsymbol{x} \in \mathbb{R}^{n}$ is the optimization problem
- $\|\boldsymbol{D}\boldsymbol{x}\|_1$ is the Total Variation of $\boldsymbol{x}$
- $\boldsymbol{D} \in \{-1, 0, 1\}^{n-1\times n}$ is the first-order difference

$$\boldsymbol{D} = \begin{bmatrix} -1 & 1 & & \\ & -1 & 1 & \\ & & & \ddots \end{bmatrix}$$

- $\lambda \geq 0$ is a regularization parameter

Proximal gradient method

$$\mathcal{P} \; : \; \underset{\boldsymbol{x}}{\operatorname{argmin}} \; \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2+\lambda\|\boldsymbol{D}\boldsymbol{x}\|_1$$

- Consider solving $\mathcal{P}$ by the proximal gradient method.
- For the optimization problem in the form $\underset{\boldsymbol{x}}{\operatorname{argmin}} \; f(\boldsymbol{x})+g(\boldsymbol{x})$, a proximal gradient update is

$$\boldsymbol{x}_{k+1}=\operatorname{prox}_{\frac{1}{L}g}\Big(\boldsymbol{x}_k-\frac{1}{L}\nabla f(\boldsymbol{x}_k)\Big),$$

where L is the Lipschitz constant of ∇f and $\operatorname{prox}_{\alpha g}$ is the proximal operator of the function $\alpha \cdot g$.

- The proximal gradient update is

$$\boldsymbol{x}_{k+1}=\operatorname{prox}_{\frac{\lambda}{L}\|\boldsymbol{D}\,\cdot\,\|_1}\Big(\boldsymbol{x}_k-\frac{\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{x}_k-\boldsymbol{A}^\top\boldsymbol{b}}{L}\Big), \qquad L=\|\boldsymbol{A}^\top\boldsymbol{A}\|_2. \tag{1}$$

- That is, we just need to repeat (1), we can solve $\mathcal{P}$ eventually.

Proximal gradient subproblem

$$\mathcal{P} \ : \ \operatorname*{argmin}_{\boldsymbol{x}} \tfrac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{D}\boldsymbol{x}\|_1$$

$$\operatorname{prox}_{\frac{\lambda}{L}\|\boldsymbol{D}\cdot\|_1}(\boldsymbol{z}), \qquad \boldsymbol{z} = \boldsymbol{x}_k - \frac{\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{A}^\top\boldsymbol{b}}{L}.$$

- The proximal operator is equivalent to solving the following optimization problem

$$\mathcal{Q} \ : \ \boldsymbol{u}^* = \operatorname*{argmin}_{\boldsymbol{u}} \frac{\lambda}{L}\|\boldsymbol{D}\boldsymbol{u}\|_1 + \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{z}\|_2^2.$$

- Unfortunately, such proximal operator has no closed-form solution, because $\boldsymbol{D}$ "inter-locks" the variables in $\boldsymbol{x}$ together.
- $\mathcal{Q}$ is a (strongly-)convex optimization problem, solution exists, is unique and global.
- $\mathcal{Q}$ can be solved by many methods, here we discuss solving $\mathcal{Q}$ by ADMM.

Prox-grad subproblem in ADMM

$$\mathcal{P} \ : \ \operatorname*{argmin}_{\boldsymbol{x}} \tfrac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{D}\boldsymbol{x}\|_1$$

- Goal: given $\boldsymbol{z}, \lambda, L, \boldsymbol{D}$, solve

$$\mathcal{Q} \ : \ \boldsymbol{u}^* = \operatorname*{argmin}_{\boldsymbol{u}} \frac{\lambda}{L}\|\boldsymbol{D}\boldsymbol{u}\|_1 + \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{z}\|_2^2.$$

- To solve $\mathcal{Q}$ by ADMM, first we convert $\mathcal{Q}$ to the "ADMM form". Introduce new a variable $\boldsymbol{v} = \boldsymbol{D}\boldsymbol{u}$ to split the objective function of $\mathcal{Q}$

$$\mathcal{Q}' \ : \ [\boldsymbol{u}^*, \boldsymbol{v}^*] = \operatorname*{argmin}_{\boldsymbol{u},\boldsymbol{v}} \underbrace{\frac{\lambda}{L}\|\boldsymbol{v}\|_1}_{\text{only } \boldsymbol{v}} + \underbrace{\frac{1}{2}\|\boldsymbol{u} - \boldsymbol{z}\|_2^2}_{\text{only } \boldsymbol{u}} \quad \text{s.t.} \quad \boldsymbol{v} = \boldsymbol{D}\boldsymbol{u}.$$

$\mathcal{Q}'$ has two variables $\boldsymbol{v}$ and $\boldsymbol{u}$ and they are connected by the constraint $\boldsymbol{v} = \boldsymbol{D}\boldsymbol{u}$.

Augmented Lagrangian

$$\mathcal{Q}' \; : \; \underset{\boldsymbol{u},\boldsymbol{v}}{\operatorname{argmin}} \; \frac{\lambda}{L}\|\boldsymbol{v}\|_1 + \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{z}\|_2^2 \quad \text{s.t.} \quad \boldsymbol{v} = \boldsymbol{D}\boldsymbol{u}.$$

- Given a problem in the form

$$\underset{\boldsymbol{u},\boldsymbol{v}}{\operatorname{argmin}} \; f(\boldsymbol{u}) + g(\boldsymbol{v}) \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{u} + \boldsymbol{B}\boldsymbol{v} = \boldsymbol{c},$$

  the augmented Lagrangian function is

$$\mathcal{L}_\rho(\boldsymbol{u},\boldsymbol{v},\boldsymbol{\gamma}) = \underbrace{f(\boldsymbol{u}) + g(\boldsymbol{v}) + \langle \boldsymbol{\gamma}, \boldsymbol{A}\boldsymbol{u} + \boldsymbol{B}\boldsymbol{v} - \boldsymbol{c}\rangle}_{\text{Lagrangian}} + \underbrace{\frac{\rho}{2}\|\boldsymbol{A}\boldsymbol{u} + \boldsymbol{B}\boldsymbol{v} - \boldsymbol{c}\|_2^2}_{\text{quadratic penalty term}}$$

  where $\boldsymbol{\gamma}$ is the Lagrange multiplier and $\rho \geq 0$ is the ADMM parameter.

- The augmented Lagrangian for $\mathcal{Q}'$ is

$$\mathcal{L}_\rho(\boldsymbol{u},\boldsymbol{v},\boldsymbol{\gamma}) = \frac{\lambda}{L}\|\boldsymbol{v}\|_1 + \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{z}\|_2^2 + \langle \boldsymbol{\gamma}, \boldsymbol{D}\boldsymbol{u} - \boldsymbol{v}\rangle + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{u} - \boldsymbol{v}\|_2^2.$$

ADMM algorithm

$$\mathcal{Q}' \;:\; \operatorname*{argmin}_{\boldsymbol{u},\boldsymbol{v}} \tfrac{\lambda}{L}\|\boldsymbol{v}\|_1 + \tfrac{1}{2}\|\boldsymbol{u}-\boldsymbol{z}\|_2^2 \quad \text{s.t.} \quad \boldsymbol{v} = \boldsymbol{D}\boldsymbol{u}.$$

$$\mathcal{L}_\rho(\boldsymbol{u},\boldsymbol{v},\boldsymbol{\gamma}) = \frac{\lambda}{L}\|\boldsymbol{v}\|_1 + \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{z}\|_2^2 + \langle \boldsymbol{\gamma}, \boldsymbol{D}\boldsymbol{u}-\boldsymbol{v}\rangle + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{u}-\boldsymbol{v}\|_2^2.$$

- The ADMM iteration for Problem $\mathcal{Q}'$

1. $\boldsymbol{u}_{k+1} = \operatorname*{argmin}_{\boldsymbol{u}} \mathcal{L}_\rho(\boldsymbol{u},\boldsymbol{v}_k,\boldsymbol{\gamma}_k)$ — Primal descent on $\boldsymbol{u}$
2. $\boldsymbol{v}_{k+1} = \operatorname*{argmin}_{\boldsymbol{v}} \mathcal{L}_\rho(\boldsymbol{u}_{k+1},\boldsymbol{v},\boldsymbol{\gamma}_k)$ — Primal descent on $\boldsymbol{v}$
3. $\boldsymbol{\gamma}_{k+1} = \boldsymbol{\gamma}_k + \rho(\boldsymbol{D}\boldsymbol{u}_{k+1} - \boldsymbol{v}_{k+1})$ — Dual ascent on $\boldsymbol{\gamma}$

Iteration on $\boldsymbol{u}$

$$\boxed{\mathcal{L}_\rho(\boldsymbol{u},\boldsymbol{v},\boldsymbol{\gamma}) = \frac{\lambda}{L}\|\boldsymbol{v}\|_1 + \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{z}\|_2^2 + \langle\boldsymbol{\gamma}, \boldsymbol{D}\boldsymbol{u}-\boldsymbol{v}\rangle + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{u}-\boldsymbol{v}\|_2^2}$$

$$\boldsymbol{u}_{k+1} = \underset{\boldsymbol{u}}{\operatorname{argmin}} \ \underbrace{\frac{1}{2}\|\boldsymbol{u}-\boldsymbol{z}\|_2^2 + \langle\boldsymbol{\gamma}_k, \boldsymbol{D}\boldsymbol{u}\rangle + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{u}-\boldsymbol{v}_k\|_2^2}_{f}$$

- $f(\boldsymbol{u})$ is differentiable, we can solve this problem using 1st-order optimality condition:

$$\begin{aligned} & \nabla_{\boldsymbol{u}} f(\boldsymbol{u}_{k+1}) = \mathbf{0} \\ \Longleftrightarrow \quad & \boldsymbol{u}_{k+1} - \boldsymbol{z} + \boldsymbol{D}^\top \utilde{\boldsymbol{\gamma}_k} + \rho(\boldsymbol{D}^\top\boldsymbol{D}\boldsymbol{u}_{k+1} - \boldsymbol{D}^\top \underline{\boldsymbol{v}_k}) = \mathbf{0} \\ \Longleftrightarrow \quad & \boldsymbol{u}_{k+1} = \left(\boldsymbol{I} + \rho\boldsymbol{D}^\top\boldsymbol{D}\right)^{-1}\left[\boldsymbol{z} + \boldsymbol{D}^\top\left(\rho\underline{\boldsymbol{v}_k} - \utilde{\boldsymbol{\gamma}_k}\right)\right] \end{aligned}$$

- In terms of implementation, the matrix $\boldsymbol{I} + \rho\boldsymbol{D}^\top\boldsymbol{D}$ is positive definite, hence we can perform Cholesky factorization to speedup the computation of $(\boldsymbol{I} + \rho\boldsymbol{D}^\top\boldsymbol{D})^{-1}$.

Iteration on $\boldsymbol{v}$

$$\mathcal{L}_\rho(\boldsymbol{u}, \boldsymbol{v}, \gamma) = \frac{\lambda}{L}\|\boldsymbol{v}\|_1 + \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{z}\|_2^2 + \langle \gamma, \boldsymbol{D}\boldsymbol{u} - \boldsymbol{v} \rangle + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{u} - \boldsymbol{v}\|_2^2$$

$$\boldsymbol{v}_{k+1} = \underset{\boldsymbol{v}}{\operatorname{argmin}} \frac{\lambda}{L}\|\boldsymbol{v}\|_1 + \langle \boldsymbol{\gamma}_k, -\boldsymbol{v} \rangle + \frac{\rho}{2}\|\boldsymbol{v} - \boldsymbol{D}\boldsymbol{u}_{k+1}\|_2^2$$

- The gradient of the smooth differentiable part is

$$-\boldsymbol{\gamma}_k + \rho(\boldsymbol{v} - \boldsymbol{D}\boldsymbol{u}_{k+1})$$

- Proximal gradient solves this problem: a soft-thresholding step with threshold $\frac{\lambda}{L}$

$$\begin{aligned} \boldsymbol{v}_{k+1} &= \operatorname{prox}_{\frac{\lambda}{L}\|\cdot\|_1}\left(\boldsymbol{v} - \frac{-\boldsymbol{\gamma}_k + \rho(\boldsymbol{v} - \boldsymbol{D}\boldsymbol{u}_{k+1})}{\rho}\right) \\ &= \mathcal{S}_{\frac{\lambda}{L}}\left(\boldsymbol{D}\boldsymbol{u}_{k+1} + \frac{1}{\rho}\boldsymbol{\gamma}_k\right) \end{aligned}$$

ADMM algorithm

- The problem

$$\mathcal{Q}' \ : \ [\boldsymbol{u}^*, \boldsymbol{v}^*] = \underset{\boldsymbol{u},\boldsymbol{v}}{\operatorname{argmin}} \ \frac{\lambda}{L}\|\boldsymbol{v}\|_1 + \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{z}\|_2^2 \quad \text{s.t.} \quad \boldsymbol{v} = \boldsymbol{D}\boldsymbol{u}$$

- The ADMM iterations for $\mathcal{Q}'$

1. $\boldsymbol{u}_{k+1} = \left(\boldsymbol{I} + \rho\boldsymbol{D}^\top\boldsymbol{D}\right)^{-1}\left[\boldsymbol{z} + \boldsymbol{D}^\top\left(\rho\boldsymbol{v}_k - \boldsymbol{\gamma}_k\right)\right]$
2. $\boldsymbol{v}_k = \mathcal{S}_{\frac{\lambda}{L}}\left(\boldsymbol{D}\boldsymbol{u}_{k+1} + \frac{1}{\rho}\gamma_k\right)$
3. $\boldsymbol{\gamma}_{k+1} = \boldsymbol{\gamma}_k + \rho(\boldsymbol{D}\boldsymbol{u}_{k+1} - \boldsymbol{v}_{k+1})$

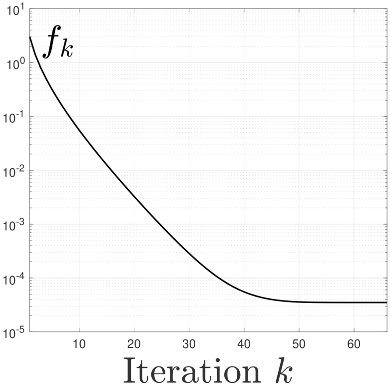
A quick-and-dirty MATLAB code of the ADMM part

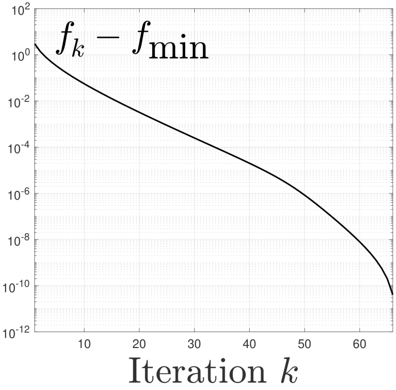
```
function [u,v,f,g] = ADMM_l1TV( z, D, L , lambda, rho)
n     = numel(z);
IrDtD = speye(n) + rho*(D')*D; % this is pd, can do chelsoky to speed up
v     = zeros(size(D,1),1);
gamma = v;
for k = 1 : 200
    u     = IrDtD \ ( z + D'*( rho*v - gamma ));
    v     = soft_thresholding(  D*u + gamma/rho, lambda/L);
    gamma = gamma + rho*( D*u - v);
    f(k)  = lambda/L*norm(v,1) + 1/2*norm(u-z);
    g(k)  = norm(D*u-v);
end
end%EOF

function y = soft_thresholding(x,th)
d = abs(x) - th;
y = sign(x) .* ( (d>0) .* d   );
end
```

An example of ADMM convergence

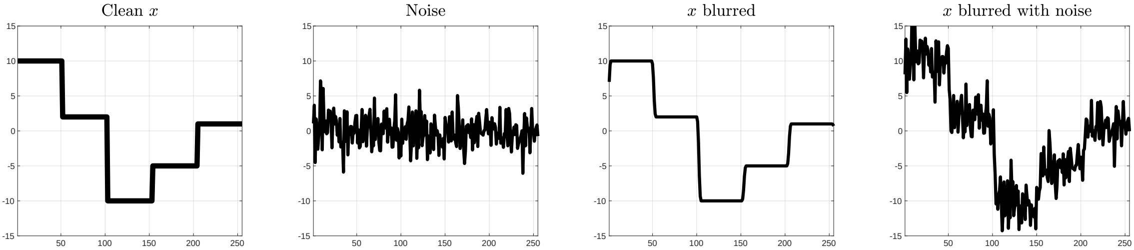
- Using $\rho = 1$ and initialize $\boldsymbol{v}_0, \boldsymbol{\gamma}$ as zero vectors.
- Plotting $f_k = f(\boldsymbol{u}_k, \boldsymbol{v}_k) = \dfrac{\lambda}{L}\|\boldsymbol{v}_k\|_1 + \dfrac{1}{2}\|\boldsymbol{u}_k - \boldsymbol{z}\|_2^2$
- At the end the gap $\|\boldsymbol{D}\boldsymbol{u}_k - \boldsymbol{v}_k\|$ is 10^{-15}.



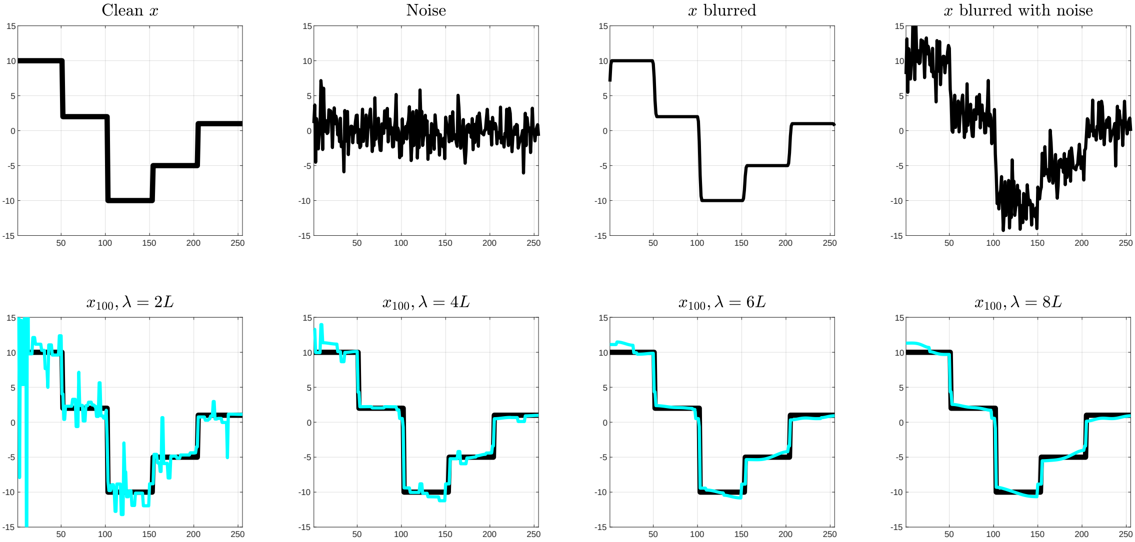


An example of Total Variation deburring denoising

$$\mathcal{P} \ : \ \operatorname*{argmin}_{\boldsymbol{x}} \ \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{D}\boldsymbol{x}\|_1.$$



Results with different λ



Last page - summary

- 1D Total Variation denoising
- Proximal gradient algorithm
- ADMM algorithm for solving the proximal gradient subproblem

End of document