

Primal Dual optimization

J. Gilles

October 22, 2024

1 Definitions and notations

In this section, we recall some general definitions and properties from convex optimization and we also fix some notations which will be used throughout this document. We first define the numerical gradient ∇ and divergence div operators such that the adjoint of ∇ is given by $\nabla^* = -\text{div}$. Given an image u of size $M \times N$ where $u_{i,j}$ represents the pixel at coordinates (i, j) , we have

$$\nabla u_{i,j} = (\nabla u_{i,j,1}, \nabla u_{i,j,2}),$$

where $\forall i, j \in D = \llbracket 1, \dots, M \rrbracket \times \llbracket 1, \dots, N \rrbracket$ (we will always use D to denote the image domain),

$$\nabla u_{i,j,1} = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < M \\ 0 & \text{if } i = M \end{cases} \quad \text{and} \quad \nabla u_{i,j,2} = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } j < N \\ 0 & \text{if } j = N \end{cases}. \quad (1)$$

The divergence is defined by, let $p = (p_1, p_2)$

$$\text{div } p_{i,j} = \begin{cases} p_{i,j,1} - p_{i-1,j,1} & \text{if } 1 < i < M \\ p_{i,j,1} & \text{if } i = 1 \\ -p_{i-1,j,1} & \text{if } i = M \end{cases} + \begin{cases} p_{i,j,2} - p_{i,j-1,2} & \text{if } 1 < j < N \\ p_{i,j,2} & \text{if } j = 1 \\ -p_{i,j-1,2} & \text{if } j = N \end{cases} \quad (2)$$

These two operators are implemented by the functions *PD_grad.m* and *PD_div.m*, respectively.

We now recall some tools from convex optimization. Given a functional $F(x)$ and an operator K , the Fenchel transform (or conjugate transform) is defined via the following relationship.

$$\forall x \in X \quad , \quad F(Kx) = \max_{y \in Y} \langle Kx, y \rangle - F^*(y). \quad (3)$$

The proximal operator associated to F is defined by

$$\text{prox}_{\gamma F}(x) = \arg \min_{y \in Y} \frac{1}{2} \|x - y\|_2^2 + \gamma F(y).$$

In practice, the link between the proximal operator of F and the proximal operator of F^* is given by Moreau's identity:

$$x = \text{prox}_{\gamma F^*}(x) + \gamma \text{prox}_{\frac{1}{\gamma} F}\left(\frac{x}{\gamma}\right).$$

2 General primal-dual problem

Primal-dual problems are saddle points problems of the form

$$\min_{x \in X} \max_{y \in Y} \langle Kx, y \rangle + G(x) - F^*(y), \quad (4)$$

where F and G are functionals and K is an operator. Using (3), it is easy to see that this primal-dual problem is equivalent the primal problem

$$\min_{x \in X} F(Kx) + G(x). \quad (5)$$

On the other side, we can rewrite

$$\begin{aligned} \min_{x \in X} \max_{y \in Y} \langle Kx, y \rangle + G(x) - F^*(y) &= \max_{y \in Y} \min_{x \in X} \langle x, K^*y \rangle + G(x) - F^*(y) \\ &= \max_{y \in Y} \left(- \max_{x \in X} (-(\langle x, K^*y \rangle + G(x))) \right) - F^*(y) \\ &= \max_{y \in Y} -G^*(-K^*y) - F^*(y). \end{aligned}$$

Thus the original primal-dual problem is also equivalent to the dual problem

$$\max_{y \in Y} -G^*(-K^*y) - F^*(y).$$

It has been proven that the solution of the primal-dual problem (4) can be achieve by the following algorithm

Algorithm 1 Solution to general primal-dual problems

Initialization: choose $\tau, \sigma > 0, \theta \in \{0, 1\}, x^0 = \tilde{x}^0 \in X, y^0 \in Y$

repeat

$$\begin{aligned} y^{n+1} &= \text{prox}_{\sigma F^*}(y^n + \sigma K \tilde{x}^n) \\ x^{n+1} &= \text{prox}_{\tau G}(x^n - \tau K^* y^{n+1}) \\ \tilde{x}^{n+1} &= x^{n+1} + \theta(x^{n+1} - x^n) \end{aligned}$$

until N iterations or any convergence criteria

In the case $\theta = 1$, convergence is guaranteed if $\tau\sigma\|K\|^2 < 1$ (where $\|K\|$ is the norm operator).

3 Rudin-Osher-Fatemi, $TV - L^2$, algorithm

The Rudin-Osher-Fatemi (ROF) model consists in solving the following problem

$$\arg \min_{f \in X} \|\nabla f\|_1 + \frac{\lambda}{2} \|f - g\|_2^2,$$

where g is an input (noisy) image and λ a regularization parameter. If we denote $K = \nabla$, $F(x) = \|x\|_1$ and $G(x) = \frac{\lambda}{2} \|x - g\|_2^2$ then we have a primal problem of the form (5). Let denote $Y = X \times X$, we have

$$F^*(p) = \delta_P(p) = \begin{cases} 0 & \text{if } \|p\|_\infty \leq 1 \\ +\infty & \text{otherwise} \end{cases},$$

where $p : Y \rightarrow \mathbb{R}$ and

$$\|p\|_\infty = \max_{i,j} |p_{i,j}|,$$

where

$$\forall i, j, |p_{i,j}| = \sqrt{p_{i,j,1}^2 + p_{i,j,2}^2}.$$

Therefore,

$$\text{prox}_{\sigma F^*}(y) = \arg \min_{z \in Y} \frac{1}{2} \|y - z\|_2^2 + \sigma \delta_P(z).$$

Clearly δ_P removes all points outside the unit ball with respect to the $\|\cdot\|_\infty$ thus this minimization problem corresponds to project y on the unit L^2 -ball. This is given by the following projector (this operator is implemented in the `PD_projl2.m` function):

$$\text{prox}_{\sigma F^*}(y)_{i,j} = \frac{y_{i,j}}{\max(1, |y_{i,j}|)}.$$

On the other hand,

$$\text{prox}_{\tau G}(x) = \arg \min_{z \in X} \frac{1}{2} \|x - z\|_2^2 + \frac{\tau\lambda}{2} \|z - g\|_2^2.$$

By variational calculus, we get

$$-(x - z) + \tau\lambda(z - g) = 0 \Leftrightarrow z = \frac{x + \tau\lambda g}{1 + \tau\lambda}.$$

Therefore (this operator is implemented in the `PD_proxl2square.m` function),

$$\text{prox}_{\tau G}(x) = \frac{x + \tau\lambda g}{1 + \tau\lambda}.$$

Finally the primal-dual algorithm to solve the ROF model is given by (and is implemented in `PD_ROF`)

Algorithm 2 Rudin-Osher-Fatemi algorithm.

Initialization: choose $\tau, \sigma, \lambda > 0, \theta \in \{0, 1\}, f^0 = \tilde{f}^0 = g, p^0 = \nabla g$

repeat

$$p^{n+1} = \text{prox}_{\sigma F^*}(p^n + \sigma \nabla \tilde{f}^n)$$

$$f^{n+1} = \text{prox}_{\tau G}(f^n + \tau \text{div } p^{n+1})$$

$$\tilde{f}^{n+1} = f^{n+1} + \theta(f^{n+1} - f^n)$$

until N iterations or any convergence criteria

An example is given in the following figure with $\theta = 1, \tau = 0.01, \lambda = 18, N = 100$, and since here $\|\nabla\| < \frac{1}{8}$, we choose $\sigma = 0.01 + \frac{1}{8\tau}$.



Figure 1: Result of the ROF algorithm (input on left, result on right)

4 $TV - L^1$ algorithm

We want to solve the model

$$\arg \min_{f \in X} \|\nabla f\|_1 + \lambda \|f - g\|_1,$$

where g is the input image and λ a regularization parameter. If we denote $K = \nabla$, $F(x) = \|x\|_1$ and $G(x) = \lambda \|x - g\|_1$ then we have a primal problem of the form (5). The proximal operator for F^* is the same as in section 3. We only need to find $\text{prox}_{\tau G}$. We have

$$\text{prox}_{\tau G}(x) = \arg \min_{z \in X} \frac{1}{2} \|x - z\|_2^2 + \tau \lambda \|z - g\|_1 = \arg \min_{z \in X} \|z - g\|_1 + \frac{1}{2\tau\lambda} \|x - z\|_2^2.$$

If we set $w = z - g$, we get

$$\text{prox}_{\tau G}(x) = g + \arg \min_{w \in X} \|w\|_1 + \frac{1}{2\tau\lambda} \|w - (x - g)\|_2^2.$$

It known that the solution of the right hand side minimization problem is given by the shrink operator (the operations are done pixelwise, we omit the subscript (i, j) to simplify the notations):

$$\begin{aligned} w = \text{sign}(x - g) \max(|x - g| - \lambda\tau, 0) &= \begin{cases} \max(x - g - \lambda\tau, 0) & \text{if } x - g > 0 \\ -\max(g - x - \lambda\tau, 0) & \text{if } x - g < 0 \\ 0 & \text{if } x - g = 0 \end{cases} \\ &= \begin{cases} x - g - \lambda\tau & \text{if } x - g > \lambda\tau \\ 0 & \text{if } \lambda\tau \geq x - g > 0 \\ -(g - x - \lambda\tau) & \text{if } x - g < -\lambda\tau \\ 0 & \text{if } -\lambda\tau \leq x - g < 0 \\ 0 & \text{if } x - g = 0 \end{cases} \\ &= \begin{cases} x - g - \lambda\tau & \text{if } x - g > \lambda\tau \\ x - g + \lambda\tau & \text{if } x - g < -\lambda\tau \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

Therefore (the implementation is in *PD_proxl1.m*),

$$\text{prox}_{\tau G}(x) = g + w = \begin{cases} x - \lambda\tau & \text{if } x - g > \lambda\tau \\ x + \lambda\tau & \text{if } x - g < -\lambda\tau \\ g & \text{otherwise} \end{cases}.$$

Finally the primal-dual algorithm to solve the TV-L1 model is given by (and is implemented in *PD_TVL1*)

Algorithm 3 $TV - L^1$ algorithm.

Initialization: choose $\tau, \sigma, \lambda > 0, \theta \in \{0, 1\}, f^0 = \tilde{f}^0 = g, p^0 = \nabla g$

repeat

$$p^{n+1} = \text{prox}_{\sigma F^*}(p^n + \sigma \nabla \tilde{f}^n)$$

$$f^{n+1} = \text{prox}_{\tau G}(f^n + \tau \text{div } p^{n+1})$$

$$\tilde{f}^{n+1} = f^{n+1} + \theta(f^{n+1} - f^n)$$

until N iterations or any convergence criteria

An example is given in the following figure with $\theta = 1, \tau = 0.01, \lambda = 1.5, N = 100$, and since here $\|\nabla\| < \frac{1}{8}$, we choose $\sigma = 0.01 + \frac{1}{8\tau}$.



Figure 2: Result of the $TV - L^1$ algorithm (input on left, result on right)

5 $TV - L^2$ inpainting

The inpainting problem corresponds to the case when some pixels in the image g are missing. We assume known the set, denoted I , of coordinates of the missing pixels. The $TV - L^2$ inpainting problem can be formulated as

$$\arg \min_{f \in X} \|\nabla f\|_1 + \frac{\lambda}{2} \sum_{(i,j) \in D \setminus I} (f_{i,j} - g_{i,j})^2,$$

where g is the input image and λ a regularization parameter (large values of λ correspond to the inpainting problem while small values to a denoising case). If we denote $K = \nabla$, $F(x) = \|x\|_1$ and $G(x) = \frac{\lambda}{2} \sum_{(i,j) \in D \setminus I} (x_{i,j} - g_{i,j})^2$ then we have a primal problem of the form (5). The proximal operator for F^* is the same as in section 3. We only need to find $\text{prox}_{\tau G}$. We have

$$\begin{aligned} \text{prox}_{\tau G}(x) &= \arg \min_{z \in X} \frac{1}{2} \|x - z\|_2^2 + \frac{\lambda \tau}{2} \sum_{(i,j) \in D \setminus I} (z_{i,j} - g_{i,j})^2 \\ &= \arg \min_{z \in X} \frac{1}{2} \sum_{(i,j) \in D} (x_{i,j} - z_{i,j})^2 + \frac{\lambda \tau}{2} \sum_{(i,j) \in D \setminus I} (z_{i,j} - g_{i,j})^2 \end{aligned}$$

Then if we decouple the problem for $(i,j) \in D \setminus I$ and $(i,j) \in I$, we get

- if $(i,j) \in D \setminus I$, taking the derivative with respect to $z_{i,j}$, we get

$$\begin{aligned} -(x_{i,j} - z_{i,j}) + \lambda \tau (z_{i,j} - g_{i,j}) &= 0 \\ \Leftrightarrow z_{i,j} &= \frac{x_{i,j} + \lambda \tau g_{i,j}}{1 + \lambda \tau}. \end{aligned}$$

- if $(i,j) \in I$, again taking the derivative with respect to $z_{i,j}$, we get

$$-(x_{i,j} - z_{i,j}) = 0 \Leftrightarrow z_{i,j} = x_{i,j}.$$

Therefore,

$$\text{prox}_{\tau G}(x)_{i,j} = \begin{cases} x_{i,j} & \text{if } (i,j) \in I \\ \frac{x_{i,j} + \lambda \tau g_{i,j}}{1 + \lambda \tau} & \text{if } (i,j) \in D \setminus I \end{cases}.$$

Moreover, if we denote the inpainting mask

$$M_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in D \setminus I \\ 0 & \text{if } (i,j) \in I \end{cases},$$

the previous expression can be rewritten as (the implementation is in *PD_proxtvl2inpainting.m*)

$$\text{prox}_{\tau G}(x)_{i,j} = \frac{x_{i,j} + \lambda \tau M_{i,j} g_{i,j}}{1 + \lambda \tau M_{i,j}}.$$

Finally the primal-dual algorithm to solve the $TV - L^2$ inpainting model is given by (and is implemented in *PD_TVL2inpainting*)

Algorithm 4 $TV - L^2$ inpainting algorithm.

Initialization: choose $\tau, \sigma, \lambda > 0, \theta \in \{0, 1\}, f^0 = \tilde{f}^0 = g, p^0 = \nabla g$

repeat

$$p^{n+1} = \text{prox}_{\sigma F^*}(p^n + \sigma \nabla \tilde{f}^n)$$

$$f^{n+1} = \text{prox}_{\tau G}(f^n + \tau \text{div } p^{n+1})$$

$$\tilde{f}^{n+1} = f^{n+1} + \theta(f^{n+1} - f^n)$$

until N iterations or any convergence criteria

An example is given in the following figure with $\theta = 1, \tau = 0.01, \lambda = 10000, N = 100$, and since here $\|\nabla\| < \frac{1}{8}$, we choose $\sigma = 0.01 + \frac{1}{8\tau}$.



Figure 3: Result of the $TV - L^2$ inpainting algorithm (input on left, result on right)

6 $TV - L^2$ multi-observations

Here, we assume that we have a sequence of input frames $\{g_k\}_{k=1}^{N_g}$, we want solve the problem

$$\arg \min_{f \in X} \|\nabla f\|_1 + \frac{\lambda}{2} \sum_{k=1}^{N_g} \|f - g_k\|_2^2.$$

If we denote $K = \nabla$, $F(x) = \|x\|_1$ and $G(x) = \frac{\lambda}{2} \sum_{k=1}^{N_g} \|x - g_k\|_2^2$ then we have a primal problem of the form (5). The proximal operator for F^* is the same as in section 3. We only need to find $\text{prox}_{\tau G}$. We have

$$\text{prox}_{\tau G}(x) = \arg \min_{z \in X} \frac{1}{2} \|x - z\|_2^2 + \frac{\lambda\tau}{2} \sum_{k=1}^{N_g} \|z - g_k\|_2^2.$$

thus

$$\begin{aligned} -(x - z) + \lambda\tau \sum_{k=1}^{N_g} (z - g_k) &= 0 \\ \Leftrightarrow (1 + N_g \lambda\tau)z &= x + \lambda\tau \sum_{k=1}^{N_g} g_k. \end{aligned}$$

Therefore, (the implementation is in *PD_proxtvl2multi.m*)

$$\text{prox}_{\tau G}(x) = \frac{x + \lambda\tau \sum_{k=1}^{N_g} g_k}{1 + N_g \lambda\tau}.$$

Finally the primal-dual algorithm to solve the $TV - L^2$ multi-observations model is given by (and is implemented in *PD_TVL2multi*)

Algorithm 5 $TV - L^2$ multi-observations algorithm.

Initialization: choose $\tau, \sigma, \lambda > 0, \theta \in \{0, 1\}, f^0 = \tilde{f}^0 = g_1, p^0 = \nabla g_1$

repeat

$$\begin{aligned} p^{n+1} &= \text{prox}_{\sigma F^*}(p^n + \sigma \nabla \tilde{f}^n) \\ f^{n+1} &= \text{prox}_{\tau G}(f^n + \tau \text{div } p^{n+1}) \\ \tilde{f}^{n+1} &= f^{n+1} + \theta(f^{n+1} - f^n) \end{aligned}$$

until N iterations or any convergence criteria

An example is given in the following figure with $\theta = 1, \tau = 0.01, \lambda = 2, N = 100, N_g = 20$, and since here $\|\nabla\| < \frac{1}{8}$, we choose $\sigma = 0.01 + \frac{1}{8\tau}$.



Figure 4: Result of the $TV - L^2$ multi-observations algorithm ($N_g = 20$, input g_1 on left, result on right)