Optimization reading notes

Fall-Winter 2024

1 Background

Image restoration (IR) problems can be formulated as inverse problems of the form

$$x^* \in \arg\min_x f(x) + \lambda g(x) \tag{1}$$

where f is a term measuring the fidelity to a degraded observation y, and g is a regularization term weighted by a parameter $\lambda \geq 0$. Generally, the degradation of a clean image $\hat{x}$ can be modeled by a linear operation $y = A\hat{x} + \xi$, where A is a degradation matrix and ξ a white Gaussian noise. In this context, the maximum a posteriori (MAP) derivation relates the data-fidelity term to the likelihood $f(x) = -\log p(y|x) = \frac{1}{2\sigma^2}||Ax - y||^2$, while the regularization term is related to the chosen prior.

Regularization is crucial since it tackles the ill-posedness of the IR task by bringing a priori knowledge on the solution. A lot of research has been dedicated to designing accurate priors g. Among the most classical priors, one can single out total variation [Rudin et al., 1992], wavelet sparsity [Mallat, 2009] or patch-based Gaussian mixtures [Zoran and Weiss, 2011]. Designing a relevant prior g is a difficult task and recent approaches rather apply deep learning techniques to directly learn a prior from a database of clean images.

Generally, the problem (1) does not have a closed-form solution, and an optimization algorithm is required. First-order proximal splitting algorithms [Combettes and Pesquet, 2011] operate individually on f and g via the proximity operator

$$\text{Prox}_f(x) = \arg\min_z \frac{1}{2}||x - z||^2 + f(z). \tag{2}$$

Among them, half-quadratic splitting (HQS) alternately applies the proximal operators of f and g. Proximal methods are particularly useful when either f or g is nonsmooth.

2 Optimization Methods

We consider the following convex optimization problem:

$$\min_{x \in \mathbb{R}^n} f(x) + h(x), \tag{3}$$

where $f(x) := \frac{1}{2}\|Ax - b\|^2$, $h(x)$ is the convex regularization term.

ISTA

$$x_{k+1} = \text{Prox}_{\frac{1}{L}h}\left(x_k - \frac{1}{L}\nabla f(x_k)\right),$$

where:

- x_k is the current solution.
- $\nabla f(x_k)$ is the gradient of the smooth function $f(x)$.
- $\text{Prox}_{\frac{1}{L}h}$ is the Proximal mapping of the non-smooth part $h(x)$, defined as:

$$\text{Prox}_{\frac{1}{L}h}(v) = \arg\min_x \left\{ h(x) + \frac{L}{2}\|x - v\|^2 \right\}.$$

- $L > 0$ is the Lipschitz constant of the gradient of the smooth function $f(x)$.

Initialization: The initial point is given as x_0.

Convergence Rate:

$$f(x_k) + h(x_k) - f(x_\star) - h(x_\star) \leq \frac{L\|x_0 - x_\star\|^2}{2k}.$$

FISTA

$$\begin{aligned} y_{i+1} &= \text{Prox}_{\frac{1}{L}h}\left(x_i - \frac{1}{L}\nabla f(x_i)\right) \\ x_{i+1} &= y_{i+1} + \frac{\theta_i - 1}{\theta_{i+1}}(y_{i+1} - y_i) \end{aligned}$$

with starting point $x_0 = y_0$, with $f \in \mathcal{F}_{0,L}$ and $h \in \mathcal{F}_{0,\infty}$, and with the sequence θ_i satisfying $\theta_0 = 1$, $\theta_i = {(1+\sqrt{1+4\theta_{i-1}^2})}/{2}$ for $i \in [1 : N-1]$. Its convergence rate is

$$f(y_N) + h(y_N) - f(x_\star) - h(x_\star) \leq \frac{L\|x_0 - x_\star\|^2}{2\theta_{N-1}^2} \leq \frac{2L\|x_0 - x_\star\|^2}{(N+1)^2}.$$

OGM

$$\begin{aligned} y_{i+1} &= x_i - \frac{1}{L}\nabla f(x_i) \\ x_{i+1} &= y_{i+1} + \frac{\theta_i - 1}{\theta_{i+1}}(y_{i+1} - y_i) + \frac{\theta_i}{\theta_{i+1}}(y_{i+1} - x_i) \end{aligned}$$

with starting point $x_0 = y_0$, with $f \in \mathcal{F}_{0,L}$, and with the sequence θ_i satisfying $\theta_0 = 1$, $\theta_i = {(1+\sqrt{1+4\theta_{i-1}^2})}/{2}$ for $i \in [1 : N-1]$, and $\theta_N = {(1+\sqrt{1+8\theta_{N-1}^2})}/{2}$. Its convergence rate is

$$f(x_N) - f(x_\star) \leq \frac{L\|x_0 - x_\star\|^2}{2\theta_N^2} \leq \frac{L\|x_0 - x_\star\|^2}{(N+1)^2}.$$

OptISTA

$$
\begin{aligned}
y_{i+1} &= \operatorname{Prox}_{\frac{\gamma_i}{L}h}\left(y_i - \frac{\gamma_i}{L}\nabla f(x_i)\right)\\
z_{i+1} &= x_i + \frac{1}{\gamma_i}\left(y_{i+1} - y_i\right)\\
x_{i+1} &= z_{i+1} + \frac{\theta_i - 1}{\theta_{i+1}}\left(z_{i+1} - z_i\right) + \frac{\theta_i}{\theta_{i+1}}\left(z_{i+1} - x_i\right)
\end{aligned}
$$

with starting point $x_0 = y_0 = z_0$, with $f \in \mathcal{F}_{0,L}$ and $h \in \mathcal{F}_{0,\infty}$, with the sequence θ_i satisfying $\theta_0 = 1$, $\theta_i = \left(1+\sqrt{1+4\theta_{i-1}^2}\right)/2$ for $i \in [1 : N-1]$, and $\theta_N = \left(1+\sqrt{1+8\theta_{N-1}^2}\right)/2$, and with $\gamma_i = 2\theta_i\left(\theta_N^2-2\theta_i+\theta_i\right)/\theta_N^2 > 0$ for $i \in [0 : N-1]$. Its convergence rate is

$$
f(y_N) + h(y_N) - f(x_\star) - h(x_\star) \leq \frac{L\|x_0 - x_\star\|^2}{2(\theta_N^2 - 1)} \leq \frac{L\|x_0 - x_\star\|^2}{(N+1)^2}.
$$

Proof of convergence rate

Recently, to improve the convergence rate of the FISTA algorithm, Jang et al. Jang et al. [2023] proposed the optimal iterative shrinkage thresholding algorithm (OptISTA), which is defined as:

$$
\begin{aligned}
y_{k+1} =& \operatorname{Prox}_{\gamma_k \frac{1}{L}h}(y_k - \gamma_k \frac{1}{L}\nabla f(x_k)),\\
z_{k+1} =& x_k + \frac{1}{\gamma_k}(y_{k+1} - y_k),\\
x_{k+1} =& z_{k+1} + \frac{\theta_k - 1}{\theta_{k+1}}(z_{k+1} - z_k) + \frac{\theta_k}{\theta_{k+1}}(z_{k+1} - x_k),
\end{aligned}
\tag{4}
$$

where $x_0 = y_0 = z_0$, $\theta_0 = 1$, $\theta_k = \frac{\left(1+\sqrt{1+4\theta_{k-1}^2}\right)}{2}$ for $k \in [1 : N-1]$, and $\theta_k = \frac{\left(1+\sqrt{1+8\theta_{N-1}^2}\right)}{2}$, and $\gamma_k = \frac{2\theta_k}{\theta_N^2}(\theta_N^2 - 2\theta_k^2 + \theta_k)$ for $k \in [0 : N-1]$. Its convergence rate is

$$
f(y_N) + h(y_N) - f(x_\star) - h(x_\star) \leq \frac{L\|x_0 - x_\star\|^2}{2(\theta_N^2 - 1)} \leq \frac{L\|x_0 - x_\star\|^2}{(N+1)^2}.
$$

But the proof of OptISTA is particularly complicated. The proof idea is as follows:

Step 1. Rewrite the OptISTA as the following equivalent form:

$$
\begin{aligned}
y_{k+1} &= \operatorname{Prox}_{\frac{\gamma_i}{L}h}\left(y_i - \frac{\gamma_i}{L}\nabla f(x_i)\right)\\
z_{k+1} &= x_i + \frac{1}{\gamma_i}\left(y_{k+1} - y_i\right)\\
w_{k+1} &= w_k - \frac{2\theta_k}{L}\nabla f(x_k) - \frac{2\theta_k}{L}h'(y_{k+1})\\
x_{k+1} &= \left(1 - \frac{1}{\theta_{k+1}}\right)z_{k+1} + \frac{1}{\theta_{k+1}}w_{k+1}
\end{aligned}
\tag{OptISTA-A}
$$

for $i = 1, \ldots, N-1$, where $w_0 = x_0$ and $h'(y_{k+1}) = \frac{L}{\gamma_k}\left(y_k - \frac{\gamma_k}{L}\nabla f(x_k) - y_{k+1}\right) \in \partial h(y_{k+1})$.

Step 2. Define the Lyapunov sequence $\{\mathcal{U}_k\}_{k\in[-1:N]}$. Explicit form of the sequence is quite cumbersome, therefore we introduce $k=-1,N$ cases only.

$$
\begin{aligned}
\mathcal{U}_N &= f(x_N) - f(x_\star) + h(y_N) - h(x_\star) \\
&+ \frac{L}{2\theta_N^2}\left\| w_N - x_\star + \frac{1}{L}\nabla f(x_\star) + \frac{2\theta_{N-1}}{L}h'(y_N) - \frac{\theta_N}{L}\nabla f(x_N) - \frac{2\tilde{\theta}_{N-1}}{L}h'(y_N)\right\|^2 \\
&+ \frac{L}{2\theta_N^2(\theta_N^2-1)}\left\| x_0 - x_\star - \frac{\theta_N^2-1}{L}\nabla f(x_\star) - \sum_{i=0}^{N-1}\frac{2\tilde{\theta}_k}{L}h'(y_{k+1})\right\|^2 \\
&+ \sum_{i\neq j, i,j\in[1:N]}\frac{\tilde{\theta}_{i-1}\tilde{\theta}_{j-1}}{L\theta_N^2(\theta_N^2-1)}\|h'(y_k)-h'(y_j)\|^2 + \sum_{i=1}^{N-1}\frac{\tilde{\theta}_{i-1}^2}{L\theta_N^2}\|h'(y_k)-h'(y_{k+1})\|^2,
\end{aligned}
$$

$$
\mathcal{U}_{-1} = \frac{L\|x_0 - x_\star\|^2}{2(\theta_N^2-1)}.
$$

Then, they show $\mathcal{U}_N \leq \mathcal{U}_{N-1} \leq \cdots \leq \mathcal{U}_1 \leq \mathcal{U}_0 \leq \mathcal{U}_{-1}$ to get

$$
f(x_N) - f(x_\star) + h(y_N) - h(x_\star) \leq \mathcal{U}_N \leq \cdots \leq \mathcal{U}_{-1} = \frac{L\|x_0 - x_\star\|^2}{2(\theta_N^2-1)}.
$$

Finally, use the fact $x_N = y_N$ to conclude that

$$
f(y_N) + h(y_N) - f(x_\star) - h(x_\star) \leq \frac{L\|x_0 - x_\star\|^2}{2(\theta_N^2-1)} \leq \frac{L\|x_0 - x_\star\|^2}{(N+1)^2}.
$$

Details

The Lyapunov sequence $\{\mathcal{U}_k\}$ in OptISTA is particularly complex and is not constructed directly, but rather two sequences $\{\mathcal{F}_k\}$ and $\{\mathcal{H}_k\}$ that satisfy

$$
\mathcal{U}_k = \mathcal{F}_k + \mathcal{H}_k, \quad k \in [-1:N],
$$

where

- $k = N$

$$
\mathcal{F}_N = f(x_N) - f(x_\star) + \frac{L}{2\theta_N^2}\left\| w_N - x_\star + \frac{1}{L}\nabla f(x_\star) + \frac{2\theta_{N-1}}{L}h'(y_N) - \frac{\theta_N}{L}\nabla f(x_N) - \frac{2\tilde{\theta}_{N-1}}{L}h'(y_N)\right\|^2,
$$

- $k \in [-1 : N-1]$

$$
\begin{aligned}
\mathcal{F}_k =& \frac{2\theta_k^2}{\theta_N^2}\left(f(x_k) - f(x_\star)\right) + \frac{L}{2\theta_N^2}\left\| w_{k+1} - x_\star + \frac{1}{L}\nabla f(x_\star) + \frac{2\theta_k}{L}h'(y_{k+1})\right\|^2 \\
&- \left(\frac{1}{2L} - \frac{\theta_k^2}{L\theta_N^2}\right)\|\nabla f(x_\star)\|^2 - \frac{\theta_k^2}{L\theta_N^2}\|\nabla f(x_k)\|^2,
\end{aligned}
$$

and $\{\mathcal{H}_k\}_{k\in[-1:N]}$ to be

- $k = N$

$$\begin{aligned}\mathcal{H}_N &= h(y_N) - h(x_\star)\\ &+ \frac{L}{2\theta_N^2(\theta_N^2-1)}\left\|x_0 - x_\star - \frac{\theta_N^2-1}{L}\nabla f(x_\star) - \sum_{i=0}^{N-1}\frac{2\tilde{\theta}_i}{L}h'(y_{i+1})\right\|^2\\ &+ \sum_{i\neq j, i,j\in[1:N]}\frac{\tilde{\theta}_{i-1}\tilde{\theta}_{j-1}}{L\theta_N^2(\theta_N^2-1)}\|h'(y_i)-h'(y_j)\|^2 + \sum_{i=1}^{N-1}\frac{\tilde{\theta}_{i-1}^2}{L\theta_N^2}\|h'(y_i)-h'(y_{i+1})\|^2,\end{aligned}$$

  where

$$\tilde{\theta}_i = \begin{cases}\theta_i & \text{if } i\in[0:N-2],\\ \frac{2\theta_{N-1}+\theta_{N-1}}{2} & \text{if } i = N-1.\end{cases}$$

- $k \in [1 : N-1]$

$$\begin{aligned}\mathcal{H}_k &= \sum_{i,j\in\{\star,1,\ldots,k\}}\tau_{i,j}\left(h(y_j)-h(y_i)\right)\\ &+ \frac{L}{2\theta_N^2(\theta_N^2-1)}\left\|x_0 - x_\star - \frac{\theta_N^2-1}{L}\nabla f(x_\star) - \sum_{i=0}^{k-1}\frac{2\theta_i}{L}h'(y_{i+1})\right\|^2\\ &+ \sum_{i\neq j, i,j\in[1:k]}\frac{\theta_{i-1}\theta_{j-1}}{L\theta_N^2(\theta_N^2-1)}\|h'(y_i)-h'(y_j)\|^2 + \sum_{i=1}^{k-1}\frac{\theta_{i-1}^2}{L\theta_N^2}\|h'(y_i)-h'(y_{i+1})\|^2\\ &+ \frac{2\theta_{k-1}^2}{L\theta_N^2}\langle\nabla f(x_k), h'(y_k)\rangle + \sum_{i=1}^{k}\sum_{\ell=k}^{N-1}\frac{2\tilde{\theta}_\ell\theta_{i-1}}{L\theta_N^2(\theta_N^2-1)}\|h'(y_i)\|^2 + \frac{\theta_{k-1}^2}{L\theta_N^2}\|h'(y_k)\|^2,\end{aligned}$$

- $k = 0, -1$

$$\mathcal{H}_0 = \mathcal{H}_{-1} = \frac{L}{2\theta_N^2(\theta_N^2-1)}\left\|x_0 - x_\star - \frac{\theta_N^2-1}{L}\nabla f(x_\star)\right\|^2,$$

Then the following holds.

$$\mathcal{F}_{N-1}-\mathcal{F}_N \geq \frac{2\tilde{\theta}_{N-1}}{\theta_N^2}\left\langle w_N - x_\star + \frac{1}{L}\nabla f(x_\star), h'(y_N)\right\rangle + \frac{\tilde{\theta}_{N-1}(4\theta_{N-1}-2\tilde{\theta}_{N-1})}{L\theta_N^2}\|h'(y_N)\|^2$$

$$\mathcal{F}_k-\mathcal{F}_{k+1} \geq \frac{2\theta_k}{\theta_N^2}\left\langle w_{k+1} - x_\star + \frac{1}{L}\nabla f(x_\star), h'(y_{k+1})\right\rangle + \frac{2\theta_k^2}{L\theta_N^2}\|h'(y_{k+1})\|^2 + \frac{2\theta_k^2}{L\theta_N^2}\langle h'(y_{k+1}), \nabla f(x_{k+1})\rangle.\ k\in[-1:N-2]$$

and

$$\begin{aligned}\mathcal{H}_{N-1}-\mathcal{H}_N &= \sum_{i=1}^{N-1}\tau_{i,N}\left(h(y_i)-h(y_N)\right) + \tau_{\star,N}\left(h(x_\star)-h(y_N)\right) + \tau_{N,N-1}\left(h(y_N)-h(y_{N-1})\right)\\ &+ \frac{2\tilde{\theta}_{N-1}}{\theta_N^2(\theta_N^2-1)}\left\langle x_0 - x_\star - \frac{\theta_N^2-1}{L}\nabla f(x_\star), h'(y_N)\right\rangle - \frac{\tilde{\theta}_{N-1}+\theta_{N-2}^2}{L\theta_N^2}\|h'(y_N)\|^2\\ &+ \frac{2\theta_{N-2}^2}{L\theta_N^2}\langle h'(y_{N-1}), h'(y_N) + \nabla f(x_{N-1})\rangle.\end{aligned}$$

$$
\begin{aligned}
\mathcal{H}_k - \mathcal{H}_{k+1} = & \sum_{i=1}^{k} \tau_{i,k+1} \left(h(y_i) - h(y_{k+1})\right) + \tau_{\star,k+1} \left(h(x_\star) - h(y_{k+1})\right) + \tau_{k+1,k} \left(h(y_{k+1}) - h(y_k)\right) \\
& + \frac{2\theta_k}{\theta_N^2(\theta_N^2 - 1)} \left\langle x_0 - x_\star - \frac{\theta_N^2 - 1}{L} \nabla f(x_\star), h'(y_{k+1}) \right\rangle - \frac{2\theta_k^2}{L\theta_N^2} \|h'(y_{k+1})\|^2 \\
& + \frac{2\theta_{k-1}^2}{L\theta_N^2} \langle h'(y_k), h'(y_{k+1}) + \nabla f(x_k)\rangle - \frac{2\theta_k^2}{L\theta_N^2} \langle h'(y_{k+1}), \nabla f(x_{k+1})\rangle. \quad k \in [0 : N-2],
\end{aligned}
$$

Then

$$
\begin{aligned}
\mathcal{U}_{N-1} - \mathcal{U}_N &= \mathcal{F}_{N-1} - \mathcal{F}_N + \mathcal{H}_{N-1} - \mathcal{H}_N \geq 0, \\
\mathcal{U}_k - \mathcal{U}_{k+1} &= \mathcal{F}_k - \mathcal{F}_{k+1} + \mathcal{H}_k - \mathcal{H}_{k+1} \geq 0, \quad k \in [0 : N-2], \\
\mathcal{U}_{-1} - \mathcal{U}_0 &= \mathcal{F}_{-1} - \mathcal{F}_0 + \mathcal{H}_{-1} - \mathcal{H}_0 \geq 0.
\end{aligned}
$$

and

$$
\begin{aligned}
\mathcal{U}_{-1} &= \mathcal{F}_{-1} + \mathcal{H}_{-1} \\
&= \frac{L}{2(\theta_N^2 - 1)} \|x_0 - x_\star\|^2.
\end{aligned}
$$

3 Gradient Step Plug-and-Play

3.1 Gradient Step Denoiser

In order to keep tractability of a minimization problem, [Romano et al., 2017] proposed, with regularization by denoising (RED), an explicit prior g that exploits a given generic denoiser D in the form $g(x) = \frac{1}{2}\langle x, x - D(x)\rangle$. With strong assumptions on the denoiser (in particular a symmetric Jacobian assumption), they show that it verifies

$$
\nabla_x g(x) = x - D(x). \tag{5}
$$

We propose to plug a denoising operator D_σ that takes the form of a gradient descent step

$$
D_\sigma = \operatorname{Id} - \nabla g_\sigma, \tag{6}
$$

with $g_\sigma : \mathbb{R}^n \to \mathbb{R}$.

In order to keep the strength of state-of-the-art unconstrained denoisers, we rather use

$$
g_\sigma(x) = \frac{1}{2} ||x - N_\sigma(x)||^2, \tag{7}
$$

$$
\text{which leads to } D_\sigma(x) = x - \nabla g_\sigma(x) = N_\sigma(x) + J_{N_\sigma}(x)^T (x - N_\sigma(x)), \tag{8}
$$

where $N_\sigma : \mathbb{R}^n \to \mathbb{R}^n$ is parameterized by a neural network and $J_{N_\sigma}(x)$ is the Jacobian of N_σ at point x.

We train the denoiser D_σ for Gaussian noise by minimizing the MSE loss function

$$\mathcal{L}(D_\sigma) = \mathbb{E}_{x\sim p, \xi_\sigma \sim \mathcal{N}(0,\sigma^2 I)}[||D_\sigma(x+\xi_\sigma) - x||^2], \tag{9}$$
$$\text{or } \mathcal{L}(g_\sigma) = \mathbb{E}_{x\sim p, \xi_\sigma \sim \mathcal{N}(0,\sigma^2 I)}[||\nabla g_\sigma(x+\xi_\sigma) - \xi_\sigma||^2], \tag{10}$$

when written in terms of g_σ using equation (6).

3.2 GS-Pnp

The standard PnP-HQS operator is $T_{\text{PnP-HQS}} = D_\sigma \circ \text{Prox}_{\tau f}$, *i.e.* $(\text{Id} - \nabla g_\sigma) \circ \text{Prox}_{\tau f}$ when using the GS denoiser as D_σ.

For convergence analysis, we wish to fit the proximal gradient descent (PGD) algorithm. We thus propose to switch the proximal and gradient steps and to relax the denoising step with a parameter $\lambda \geq 0$. Our PnP algorithm with GS denoiser (GS-PnP) then writes

$$x_{k+1} = T^{\tau,\lambda}_{\text{GS-PnP}}(x_k) \text{ with } \begin{array}{rcl} T^{\tau,\lambda}_{\text{GS-PnP}} & = & \text{Prox}_{\tau f} \circ(\tau\lambda D_\sigma + (1-\tau\lambda)\,\text{Id}), \\ & = & \text{Prox}_{\tau f} \circ(\text{Id} - \tau\lambda\nabla g_\sigma). \end{array} \tag{11}$$

Under suitable conditions on f and g_σ, fixed points of the PGD operator $T^{\tau,\lambda}_{\text{GS-PnP}}$ correspond to critical points of a classical objective function in IR problems

$$F(x) = f(x) + \lambda g_\sigma(x). \tag{12}$$

Param.: init. z_0, $\lambda > 0$, $\sigma \geq 0$, $\epsilon > 0$, $\tau_0 > 0$, $K \in \mathbb{N}^*$, $\eta \in (0,1)$, $\gamma \in (0, 1/2)$.
Input : degraded image y.
Output: restored image $\hat{x}$.
$k = 0$; $x_0 = \text{Prox}_{\tau f}(z_0)$; $\tau = \tau_0/\eta$; $\Delta > \epsilon$;
while $k < K$ *and* $\Delta > \epsilon$ **do**
  $z_k = \lambda\tau D_\sigma(x_k) + (1-\lambda\tau)x_k$;
  $x_{k+1} = \text{Prox}_{\tau f}(z_k)$;
  if $F(x_k) - F(x_{k+1}) < \frac{\gamma}{\tau}||x_k - x_{k+1}||^2$;
  then $\tau = \eta\tau$;
  else $\Delta = \frac{F(x_k) - F(x_{k+1})}{F(x_0)}$; $k = k+1$;
end
$\hat{x} = \lambda\tau D_\sigma(x_K) + (1-\lambda\tau)x_K$;

Algorithm 1: Plug-and-Play image restoration

3.3 Covergence

Theorem 1. *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ and $g_\sigma : \mathbb{R}^n \to \mathbb{R}$ be proper lower semicontinous functions with f convex and g_σ differentiable with L-Lipschitz gradient. Let $\lambda > 0$, $F = f + \lambda g_\sigma$ and assume that F is bounded from below. Then, for $\tau < \frac{1}{\lambda L}$, the iterates x_k given by the iterative scheme (11) verify* *(i)* $(F(x_k))$ *is non-increasing and converges.*

(ii) *The residual* $||x_{k+1} - x_k||$ *converges to* 0.

(iii) *All cluster points of the sequence* (x_k) *are stationary points of (12).*

3.4 Comparison

Gradient Step Denoiser

The Gradient Step Denoiser, denoted as D_σ, is defined through a gradient descent step:

$$D_\sigma = \mathrm{Id} - \nabla g_\sigma, \tag{13}$$

where Id is the identity operator and ∇g_σ is the gradient of the function g_σ.

Parameterization: The function $g_\sigma(x)$ is parameterized as:

$$g_\sigma(x) = \frac{1}{2}||x - N_\sigma(x)||^2, \tag{14}$$

where N_σ is a neural network parameterized function, allowing g_σ to be optimized through training.

Training: The denoiser is trained by minimizing the Mean Squared Error (MSE) loss function to handle Gaussian noise.

Gradient Step Plug-and-Play (GS-PnP) Algorithm

The GS-PnP algorithm incorporates the Gradient Step Denoiser in its iterative process:

$$x_{k+1} = T_{\tau,\lambda}^{GS-PnP}(x_k), \tag{15}$$

where

$$T_{\tau,\lambda}^{GS-PnP} = \mathrm{Prox}_{\tau f} \circ (\mathrm{Id} - \tau\lambda\nabla g_\sigma). \tag{16}$$

Convergence: The GS-PnP algorithm provides **theoretical convergence guarantees**, even when the data-fidelity term is not strongly convex.

Performance: The algorithm demonstrates superior or comparable performance to state-of-the-art methods across various image restoration tasks.

Advantages

- **Theoretical Convergence Guarantees**: The GS-PnP algorithm ensures convergence, which is critical for scientific and engineering applications.
- **Performance**: It shows excellent performance in image restoration tasks.
- **Adaptability**: GS-PnP can adapt to different image restoration tasks, including ill-posed inverse problems.
- **Adaptive Step Size Adjustment**: The algorithm dynamically adjusts the step size based on the progress during iterations, enhancing stability and efficiency.

4 Bregman Plug-and-Play

4.1 Bregman denoising prior

Bregman divergence

$$D_h : \mathbb{R}^n \times int\,\mathrm{dom}\,h \to [0, +\infty] : (x, y) \to \begin{cases} h(x) - h(y) - \langle \nabla h(y), x - y \rangle & \text{if } x \in \mathrm{dom}(h) \\ +\infty & \text{otherwise.} \end{cases} \tag{17}$$

4.1.1 Bregman noise model

We consider the following observation noise model, referred to as *Bregman noise*[1],

$$\text{for } x, y \in \mathrm{dom}(h) \times int\,\mathrm{dom}(h) \quad p(y|x) := \exp\left(-\gamma D_h(x, y) + \rho(x)\right). \tag{18}$$

We assume that there is $\gamma > 0$ and a normalizing function $\rho : \mathrm{dom}(h) \to \mathbb{R}$ such that the expression (18) defines a probability measure. For instance, for $h(x) = \frac{1}{2}||x||^2$, $\gamma = \frac{1}{\sigma^2}$ and $\rho = 0$, we retrieve the Gaussian noise model with variance σ^2. For h given by Burg's entropy, $p(y|x)$ corresponds to a multivariate Inverse Gamma ($\mathcal{IG}$) distribution.

Maximum-A-Posteriori (MAP) estimator The MAP denoiser selects the mode of the a-posteriori probability distribution $p(x|y)$. Given the prior p_X, it writes

$$\hat{x}_{MAP}(y) = \arg\min_x -\log p(x|y) = \arg\min_x -\log p_X(x) - \log p(y|x) = \mathrm{Prox}^h_{-\frac{1}{\gamma}(\rho + \log p_X)}(y). \tag{19}$$

Posterior mean (MMSE) estimator The MMSE denoiser is the expected value of the posterior probability distribution and the optimal Bayes estimator for the L_2 score. Note that our Bregman noise conditional probability (18) belongs to the regular *exponential family of distributions*

$$p(y|x) = p_0(y) \exp\left(\langle x, T(y) \rangle - \psi(x)\right) \tag{20}$$

with $T(y) = \gamma \nabla h(y)$, $\psi(x) = \gamma h(x) - \rho(x)$ and $p_0(y) = \exp\left(\gamma h(y) - \gamma \langle \nabla h(y), y \rangle\right)$. It is shown in [Efron, 2011] (for $T = \mathrm{Id}$ and generalized in [Kim and Ye, 2021] for $T \neq \mathrm{Id}$) that the corresponding posterior mean estimator verifies a generalized Tweedie formula $\nabla T(y).\hat{x}_{MMSE}(y) = -\nabla \log p_0(y) + \nabla \log p_Y(y)$,

$$\hat{x}_{MMSE}(y) = \mathbb{E}[x|y] = y - \frac{1}{\gamma}(\nabla^2 h(y))^{-1}.\nabla(-\log p_Y)(y). \tag{21}$$

Note that for the Gaussian noise model, we have $h(x) = \frac{1}{2}||x||^2$, $\gamma = 1/\sigma^2$ and (21) falls back to the more classical Tweedie formula of the Gaussian posterior mean denoiser $\hat{x} = y - \sigma^2 \nabla(-\log p_Y)(y)$. Therefore, given an off-the-shelf "Bregman denoiser" $\mathcal{B}_\gamma$ specially devised to remove Bregman noise (18) of level γ, if the denoiser approximates the posterior mean $\mathcal{B}_\gamma(y) \approx \hat{x}_{MMSE}(y)$, then it provides an approximation of the score $-\nabla \log p_Y(y) \approx \gamma \nabla^2 h(y).\,(y - \mathcal{B}_\gamma(y))$.

[1] The Bregman divergence being non-symmetric, the order of the variables (x, y) in D_h is important. Distributions of the form (18) with reverse order in D_h have been characterized in [Banerjee et al., 2005] but this analysis does not apply here.

4.1.2 Bregman Score Denoiser

Based on previous observations, we propose to define a denoiser following the form of the MMSE (21)

$$\mathcal{B}_\gamma(y) = y - (\nabla^2 h(y))^{-1}.\nabla g_\gamma(y), \tag{22}$$

with $g_\gamma : \mathbb{R}^n \to \mathbb{R}$ a nonconvex potential parametrized by a neural network.

4.2 Pnp with Bregman denoising prior

4.2.1 Bregman Proximal Gradient (BPG) algorithm

$$x^{k+1} \in \underset{x\in\mathbb{R}^n}{\arg\min}\{\mathcal{R}(x) + \langle x - x^k, \nabla F(x^k)\rangle + \frac{1}{\tau} D_h(x, x^k)\}. \tag{23}$$

when $\nabla h(x_k) - \tau\nabla F(x_k) \in \mathrm{dom}(h^*)$, the previous iteration can be written as

$$x^{k+1} \in \mathrm{Prox}^h_{\tau\mathcal{R}} \circ \nabla h^*(\nabla h - \tau\nabla F)(x_k). \tag{24}$$

- **Non-smooth Optimization:** The BPG algorithm is suitable for non-smooth optimization problems, not requiring global Lipschitz continuity of the gradient.
- **Bregman Divergence:** It uses Bregman divergence instead of the traditional Euclidean distance, providing a more flexible optimization framework.
- **Iterative Update:** The algorithm progressively approaches the optimal solution through iterative updates as described in equation.

4.2.2 Bregman Regularization-by-Denoising (B-RED)

We propose to minimize $F_{\lambda,\gamma} = \lambda f + g_\gamma$ on $\mathrm{dom}(h)$ using the Bregman Gradient Descent algorithm

$$x_{k+1} = \nabla h^*(\nabla h - \tau\nabla F_{\lambda,\gamma})(x_k) \tag{25}$$

which writes in a more general version as the BPG algorithm (23) with $\mathcal{R} = 0$

$$x_{k+1} = \underset{x\in\mathbb{R}^n}{\arg\min}\{\langle x - x_k, \lambda\nabla f(x_k) + \nabla g_\gamma(x_k)\rangle + \frac{1}{\tau} D_h(x, x_k)\}. \tag{26}$$

$$\textbf{(B-RED)} \qquad x^{k+1} \in T_\tau(x_k) = \underset{x\in\mathbb{R}^n}{\arg\min}\{i_C(x) + \langle x - x^k, \nabla F_{\lambda,\gamma}(x^k)\rangle + \frac{1}{\tau} D_h(x, x^k)\}. \tag{27}$$

- **Integration with Denoising:** The B-RED algorithm combines a denoiser with the Bregman framework, utilizing the denoiser as a regularization term.
- **Adaptive Step Size:** It ensures the convergence and effectiveness of the algorithm through an adaptive step size adjustment strategy as given in equation (19).
- **Flexibility:** It can be applied to various denoising priors, including those based on Deep Neural Networks (DNNs).

4.2.3 Bregman Plug-and-Play (B-PnP)

We now consider the equivalent of PnP Proximal Gradient Descent algorithm in the Bregman framework. Given a denoiser $\mathcal{B}_\gamma$ with $\mathrm{Im}(\mathcal{B}_\gamma) \subset \mathrm{dom}(h)$ and $\lambda > 0$ such that $\mathrm{Im}(\nabla h - \lambda \nabla f) \subseteq \mathrm{dom}(\nabla h^*)$, it writes

$$\textbf{(B-PnP)} \qquad x^{k+1} = \mathcal{B}_\gamma \circ \nabla h^*(\nabla h - \lambda \nabla f)(x_k). \tag{28}$$

The algorithm B-PnP (28) then becomes $x^{k+1} \in \mathrm{Prox}^h_{\phi_\gamma} \circ \nabla h^*(\nabla h - \lambda \nabla f)(x_k)$, which writes as a Bregman Proximal Gradient algorithm, with stepsize $\tau = 1$,

$$x^{k+1} \in \underset{x \in \mathbb{R}^n}{\arg\min} \{\phi_\gamma(x) + \langle x - x^k, \lambda \nabla f(x^k) \rangle + D_h(x, x^k)\}. \tag{29}$$

- **Plug-and-Play:** The B-PnP algorithm is a type of Plug-and-Play algorithm, offering great flexibility and a plug-and-play feature.
- **Denoiser as Prior:** It employs the Bregman Score denoiser as the prior within the algorithm, capable of handling more complex image restoration problems.
- **Fixed Step Size:** Unlike B-RED, the B-PnP algorithm uses a fixed step size $\tau = 1$, without an adaptive step size adjustment process.

4.2.4 Comparison

1. The BPG algorithm provides an optimization framework based on Bregman divergence, while B-RED and B-PnP algorithms incorporate denoisers as regularization terms on this basis.
2. Both B-RED and B-PnP use a denoiser, but B-RED ensures convergence through adaptive step size adjustments, whereas B-PnP relies on a fixed step size.

3. B-RED guarantees the convergence of the algorithm through adaptive step size adjustments, while B-PnP requires specific conditions (such as the convexity of $\psi_\gamma \circ \nabla h^*$) to ensure convergence.

4. Due to its plug-and-play nature, the B-PnP algorithm offers higher flexibility when dealing with different image restoration problems.

References

Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, Joydeep Ghosh, and John Lafferty. Clustering with bregman divergences. *Journal of machine learning research*, 6(10), 2005.

P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer, 2011.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

Uijeong Jang, Shuvomoy Das Gupta, and Ernest K. Ryu. Computer-assisted design of accelerated composite optimization methods: OptISTA. *Arxiv: 2305.15704*, 2023.

Kwanyoung Kim and Jong Chul Ye. Noise2score: tweedie's approach to self-supervised image denoising without clean images. *Advances in Neural Information Processing Systems*, 34:864–874, 2021.

Stéphane Mallat. *A Wavelet Tour of Signal Processing, The Sparse Way.* Academic Press, Elsevier, 3rd edition edition, 2009. ISBN 978-0-12-374370-1.

Yaniv Romano, Michael Elad, and Peyman Milanfar. The little engine that could: Regularization by denoising (red). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, 2017.

L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60:259–268, 1992.

Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *2011 International Conference on Computer Vision*, pages 479–486. IEEE, 2011.