A full backtracking strategy for a generalised FISTA algorithm

Luca Calatroni

Centre de Mathematiqués Appliquées (CMAP), École Polytechnique CNRS

joint work with: A. Chambolle.

Journées annuelles 2017

GDR MOA-MIA

17-20 October 2017

Bordeaux, France

Table of contents

Introduction

Composite optimisation

Let $\mathcal{X}$ a Hilbert space with norm $\|\cdot\|$. We want to solve:

$$\min_{x\in\mathcal{X}} \; \{F(x) := f(x) + g(x)\}$$

- f is **smooth**: differentiable, convex with Lipschitz gradient

$$\|\nabla f(y) - \nabla f(x)\| \leq L_f \|y - x\|, \qquad \text{for any } x, y \in \mathcal{X}.$$

- g is convex, l.s.c., **non-smooth**. Example: ℓ^1, Total Variation...

[1] Combettes, Ways, '05, Nesterov, '13...

Composite optimisation

Let $\mathcal{X}$ a Hilbert space with norm $\|\cdot\|$. We want to solve:

$$\min_{x\in\mathcal{X}} \; \{F(x) := f(x) + g(x)\}$$

- f is **smooth**: differentiable, convex with Lipschitz gradient

$$\|\nabla f(y) - \nabla f(x)\| \leq L_f \|y - x\|, \qquad \text{for any } x, y \in \mathcal{X}.$$

- g is convex, l.s.c., **non-smooth**. Example: ℓ^1, Total Variation...

Composite optimisation problem

Forward-Backward splitting[1].

- forward gradient descent step in f;
- backward implicit gradient descent step in g.

Basic algorithm: take $x_0 \in \mathcal{X}$, fix $\tau > 0$ and for $k \geq 0$ do:

$$x_{k+1} = \text{prox}_{\tau g}\left(x_k - \tau\nabla f(x_k)\right) =: T_\tau x_k.$$

[1] Combettes, Ways, '05, Nesterov, '13...

Composite optimisation

Let $\mathcal{X}$ a Hilbert space with norm $\|\cdot\|$. We want to solve:

$$\min_{x\in\mathcal{X}} \; \{F(x) := f(x) + g(x)\}$$

- f is **smooth**: differentiable, convex with Lipschitz gradient

$$\|\nabla f(y) - \nabla f(x)\| \leq L_f \|y - x\|, \qquad \text{for any } x, y \in \mathcal{X}.$$

- g is convex, l.s.c., **non-smooth**. Example: ℓ^1, Total Variation...

Composite optimisation problem

Forward-Backward splitting[1].

- forward gradient descent step in f;
- backward implicit gradient descent step in g.

Basic algorithm: take $x_0 \in \mathcal{X}$, fix $\tau > 0$ and for $k \geq 0$ do:

$$x_{k+1} = \text{prox}_{\tau g}\left(x_k - \tau \nabla f(x_k)\right) =: T_\tau x_k.$$

Rate of convergence: $O(1/k)$ (non-strongly convex case).

[1] Combettes, Ways, '05, Nesterov, '13...

Accelerated forward-backward, FISTA: references

In Nesterov ’04 and Beck, Teboulle ’09, accelerated $O(1/k^2)$ convergence of FB splitting is achieved by extrapolation.

Further properties:

- **convergence of iterates** (Chambolle, Dossal ’15);
- **monotone** variants (Beck, Teboulle ’09, Tseng ’08, Tao, Boley, Zhang ’15)
- acceleration for **approximated evaluation** of operators (Villa, Salzo, Baldassarre, Verri ’13)
- . . .

Accelerated forward-backward, FISTA: references

In Nesterov '04 and Beck, Teboulle '09, accelerated $O(1/k^2)$ convergence of FB splitting is achieved by extrapolation.

Further properties:

- **convergence of iterates** (Chambolle, Dossal '15);
- **monotone** variants (Beck, Teboulle '09, Tseng '08, Tao, Boley, Zhang '15)
- acceleration for **approximated evaluation** of operators (Villa, Salzo, Baldassarre, Verri '13)
- . . .

Two main questions

1. Can we say more when f and/or g are strongly convex? Linear convergence?
2. Can we let the gradient/proximal parameter "adapt" along the iterations, preserving acceleration?

A strongly convex variant of FISTA (GFISTA)

Let $\mu_f, \mu_g > 0$. Then $\mu = \mu_f + \mu_g$.

For $\tau > 0$ define:

$$q := \frac{\tau\mu}{1 + \tau\mu_g} \in (0, 1).$$

Algorithm 1 GFISTA[2] (no backtracking)

Input: $0 < \tau \leq 1/L_f, x^0 = x^{-1} \in \mathcal{X}$ and let $t_0 \in \mathbb{R}$ s.t. $0 \leq t_0 \leq 1/\sqrt{q}$.
for $k \geq 0$ **do**

$$y^k = x^k + \beta_k(x^k - x^{k-1})$$
$$x^{k+1} = T_\tau y^k = \text{prox}_{\tau g}(y^k - \tau \nabla f(y^k))$$

$$t_{k+1} = \frac{1 - qt_k^2 + \sqrt{(1 - qt_k^2)^2 + 4t_k^2}}{2}$$
$$\beta_k = \frac{t_k - 1}{t_{k+1}} \frac{1 + \tau\mu_g - t_{k+1}\tau\mu}{1 - \tau\mu_f}$$

end for

Remark: $\mu = q = 0 \implies$ standard FISTA.

[2]Chambolle, Pock '16

GFISTA: acceleration results

Theorem [Chambolle, Pock '16]

Let $\tau \leq 1/L_f$ and $0 \leq t_0\sqrt{q} \leq 1$. Then, the sequence (x^k) of iterates of GFISTA satisfies

$$F(x^k) - F(x^*) \leq r_k(q) \left(t_0^2(F(x^0) - F(x^*)) + \frac{1 + \tau\mu_g}{2} \|x - x^*\|^2 \right),$$

where x^* is a minimiser of F and:

$$r_k(q) = \min \left\{ \frac{4}{(k+1)^2}, (1 + \sqrt{q})(1 - \sqrt{q})^k, \frac{(1 - \sqrt{q})^k}{t_0^2} \right\}.$$

Note: for $\mu = q = 0$, $t_0 = 0$ this is the standard FISTA convergence result.

GFISTA: acceleration results

Theorem [Chambolle, Pock '16]

Let $\tau \leq 1/L_f$ and $0 \leq t_0\sqrt{q} \leq 1$. Then, the sequence (x^k) of iterates of GFISTA satisfies

$$F(x^k) - F(x^*) \leq r_k(q) \left(t_0^2 (F(x^0) - F(x^*)) + \frac{1+\tau\mu_g}{2} \|x - x^*\|^2 \right),$$

where x^* is a minimiser of F and:

$$r_k(q) = \min \left\{ \frac{4}{(k+1)^2}, (1+\sqrt{q})(1-\sqrt{q})^k, \frac{(1-\sqrt{q})^k}{t_0^2} \right\}.$$

Note: for $\mu = q = 0$, $t_0 = 0$ this is the standard FISTA convergence result.

Question: What if an estimate of L_f is *not* available? Backtracking!

Backtracking of FISTA:

- Armijo-type backtracking (Beck, Teboulle, '09): $\tau_{k+1} \leq \tau_k$ at every k.
- Full backtracking (Scheinberg, Goldfarb, Bai, '14): larger steps in "flat" areas!

GFISTA with backtracking

Backtracking strategy and Bregman distance

Backtracking strategies check if for every $x \in \mathcal{X}$:

$$F(\hat{x}) + (1+\tau\mu_g)\frac{\|x-\hat{x}\|^2}{2\tau} + \left(\frac{\|\hat{x}-\bar{x}\|^2}{2} - D_f(\hat{x},\bar{x})\right) \leq F(x) + (1-\tau\mu_f)\frac{\|x-\bar{x}\|^2}{2\tau},$$

where

$$D_f(\hat{x},\bar{x}) := f(\hat{x}) - f(\bar{x}) - \langle \nabla f(\bar{x}), \bar{x}-\hat{x}\rangle$$

is the *Bregman distance* of f between $\hat{x} = T_\tau \bar{x}$ and $\bar{x} \in \mathcal{X}$.

Backtracking strategy and Bregman distance

Backtracking strategies check if for every $x \in \mathcal{X}$:

$$F(\hat{x}) + (1 + \tau\mu_g)\frac{\|x - \hat{x}\|^2}{2\tau} + \left(\frac{\|\hat{x} - \bar{x}\|^2}{2} - D_f(\hat{x}, \bar{x})\right) \leq F(x) + (1 - \tau\mu_f)\frac{\|x - \bar{x}\|^2}{2\tau},$$

where

$$D_f(\hat{x}, \bar{x}) := f(\hat{x}) - f(\bar{x}) - \langle \nabla f(\bar{x}), \bar{x} - \hat{x} \rangle$$

is the *Bregman distance* of f between $\hat{x} = T_\tau \bar{x}$ and $\bar{x} \in \mathcal{X}$.

Constant steps

Such condition is verified as long as:

$$D_f(\hat{x}, \bar{x}) \leq \frac{\|\hat{x} - \bar{x}\|^2}{2\tau}, \qquad (*)$$

which is always true if $\boldsymbol{\tau \leq 1/L_f}$ with L_f known.

However, one can alternatively check (*) along the iterations. This corresponds to compute a **local** Lipschitz Constant Estimate (LCE).

GFISTA with backtracking: Algorithm

For any $k \geq 0$ we let $\tau = \tau_k$ and define:

$$\tau_k' = \frac{\tau_k}{1 - \tau_k \mu_f}, \qquad q_k = \frac{\mu \tau_k}{1 + \tau_k \mu_g}.$$

Update rule for extrapolation: for any $k \geq 0$ set

$$t_{k+1} = \frac{1 - \frac{q_{k+1}}{1-q_{k+1}} \frac{\tau_k'}{\tau_{k+1}'} t_k^2 + \sqrt{\left(\frac{q_{k+1}}{1-q_{k+1}} \frac{\tau_k'}{\tau_{k+1}'} t_k^2 - 1\right)^2 + 4 \frac{\tau_k'}{\tau_{k+1}'} \frac{t_k^2}{1-q_{k+1}}}}{2}$$

GFISTA with backtracking: algorithm

Algorithm 2 GFISTA with backtracking

Input: μ_f, μ_g, $\tau_0 > 0$, q_0, $\rho \in (0,1)$, $x^0 = x^{-1} \in \mathcal{X}$ and $t_0 \in \mathbb{R}$ s.t. $0 \leq t_0 \leq 1/\sqrt{q_0}$.
for $k \geq 0$ **do**

$$y^k = x^k + \beta_k(x^k - x^{k-1}).$$

Set $i_{bt} = 0$;
  if too close to LCE **then**
    while Backtracking condition (*) is **not** verified & $i_{bt} \leq i_{max}$ **do**
      keep/reduce step-size: $\tau_{k+1} = \rho^{i_{bt}} \tau_k$;
      Compute

$$x^{k+1} = T_{\tau_{k+1}} y^k = \text{prox}_{\tau_{k+1}g}(y^k - \tau_{k+1}\nabla f(y^k)) \tag{1}$$

    $i_{bt} = i_{bt} + 1$;
    end while
  else if far enough from LCE **then**
   increase step-size: $\tau_{k+1} = \frac{\tau_k}{\rho}$;
   Compute x_{k+1} using (1);
  end if
   Update $q_{k+1}, \tau'_{k+1}, t_{k+1}$.
Set

$$\beta_{k+1} = \frac{1 - q_{k+1}t_{k+1}}{1 - q_{k+1}} \frac{t_k - 1}{t_{k+1}}.$$

end for

Too close/too far: how tight is (*)? Reduce costs due to (1).

Analogies/differences with FISTA-type algorithms: update rule

$$t_{k+1} = \frac{1 - \frac{q_{k+1}}{1-q_{k+1}} \frac{\tau'_k}{\tau'_{k+1}} t_k^2 + \sqrt{\left(\frac{q_{k+1}}{1-q_{k+1}} \frac{\tau'_k}{\tau'_{k+1}} t_k^2 - 1\right)^2 + 4 \frac{\tau'_k}{\tau'_{k+1}} \frac{t_k^2}{1-q_{k+1}}}}{2}$$

Analogies/differences with FISTA-type algorithms: update rule

$$t_{k+1} = \frac{1 - \frac{q_{k+1}}{1-q_{k+1}} \frac{\tau'_k}{\tau'_{k+1}} t_k^2 + \sqrt{\left(\frac{q_{k+1}}{1-q_{k+1}} \frac{\tau'_k}{\tau'_{k+1}} t_k^2 - 1\right)^2 + 4 \frac{\tau'_k}{\tau'_{k+1}} \frac{t_k^2}{1-q_{k+1}}}}{2}$$

No backtracking, convex case

If $\mu = q_k = 0$ and $\tau_k = \tau_{k+1}$ for any $k \geq 0$, this is the FISTA update rule.

Analogies/differences with FISTA-type algorithms: update rule

$$t_{k+1} = \frac{1 - \frac{q_{k+1}}{1-q_{k+1}} \frac{\tau'_k}{\tau'_{k+1}} t_k^2 + \sqrt{\left(\frac{q_{k+1}}{1-q_{k+1}} \frac{\tau'_k}{\tau'_{k+1}} t_k^2 - 1\right)^2 + 4 \frac{\tau'_k}{\tau'_{k+1}} \frac{t_k^2}{1-q_{k+1}}}}{2}$$

No backtracking, convex case

If $\mu = q_k = 0$ and $\tau_k = \tau_{k+1}$ for any $k \geq 0$, this is the FISTA update rule.

FISTA with adaptive backtracking

If $\mu = q_k = 0$ for any $k \geq 0$, the rule reduces to:

$$t_{k+1} = \frac{1 + \sqrt{1 + 4 \frac{\tau_k}{\tau_{k+1}} t_k^2}}{2},$$

which is the same as the one proposed by Scheinberg et al. '14 for fast adaptive backtracking.

Accelerated convergence rates

Recurrence

Idea: start from descent rule

$$F(\hat{x}) + (1 + \tau\mu_g)\frac{\|x - \hat{x}\|^2}{2\tau} \leq F(x) + (1 - \tau\mu_f)\frac{\|x - \bar{x}\|^2}{2\tau}, \quad \text{for any } x \in \mathcal{X},$$

and set, for $k \geq 0$

$$x = \frac{(t-1)x^k + x^*}{t}, \quad \bar{x} = y^k, \quad \hat{x} = x_{k+1} = T_\tau y^k.$$

Recurrence

Idea: start from descent rule

$$F(\hat{x}) + (1+\tau\mu_g)\frac{\|x-\hat{x}\|^2}{2\tau} \leq F(x) + (1-\tau\mu_f)\frac{\|x-\bar{x}\|^2}{2\tau}, \quad \text{for any } x \in \mathcal{X},$$

and set, for $k \geq 0$

$$x = \frac{(t-1)x^k + x^*}{t}, \quad \bar{x} = y^k, \quad \hat{x} = x_{k+1} = T_\tau y^k.$$

Multiply by suitable factors and use convexity and extrapolated sequences:

$$F(x^k) - F(x^*) \leq \theta_k \left(\tau_0' t_0^2 \left(F(x^0) - F(x^*)\right) + \frac{1}{2}\|x^* - x^0\|^2\right).$$

To study convergence speed, analyse:

$$\boxed{\theta_k := \frac{\prod_{i=1}^{k}\omega_i}{\tau_k' t_k^2}, \qquad \text{with } \omega_i = \frac{1-q_{i+1}t_{i+1}}{1-q_{i+1}} \in (0,1]}$$

Tools: technical lemmas & induction argument following Nesterov '04.

Convergence rates: worst-case analysis

Define:

$$L_w := \max\left\{\frac{L_f}{\rho}, \rho L_0\right\}, \qquad q_w := \frac{\mu}{L_w + \mu_g},$$

with q_w being the *worst-case* inverse condition number.

Theorem

Let $x_0 \in \mathcal{X}$, $\tau_0 > 0$ and let (x_k) the sequence produced by the GFISTA algorithm with backtracking. If $t_0 \geq 0$ and $\sqrt{q_0}\, t_0 \leq 1$, we have:

$$F(x^k) - F(x^*) \leq r_k \, \boldsymbol{(L_w - \mu_f)} \left(\frac{\tau_0 t_0^2}{1 - \mu_f \tau_0} \Big(F(x^0) - F(x^*) \Big) + \frac{1}{2}\|x^0 - x^*\|^2 \right)$$

where the decay rate is defined as:

$$r_k := \min\left\{ \frac{4}{(k+1)^2}, (1 - \sqrt{q_w})^{k-1}, \frac{(1-\sqrt{q_w})^k}{t_0^2} \right\}.$$

Convergence rates: worst-case analysis

Define:

$$L_w := \max\left\{\frac{L_f}{\rho}, \rho L_0\right\}, \qquad q_w := \frac{\mu}{L_w + \mu_g},$$

with q_w being the *worst-case* inverse condition number.

Theorem

Let $x_0 \in \mathcal{X}$, $\tau_0 > 0$ and let (x_k) the sequence produced by the GFISTA algorithm with backtracking. If $t_0 \geq 0$ and $\sqrt{q_0}\, t_0 \leq 1$, we have:

$$F(x^k) - F(x^*) \leq r_k \,(\boldsymbol{L_w - \mu_f}) \left(\frac{\tau_0 t_0^2}{1 - \mu_f \tau_0}\Big(F(x^0) - F(x^*)\Big) + \frac{1}{2}\|x^0 - x^*\|^2\right)$$

where the decay rate is defined as:

$$r_k := \min\left\{\frac{4}{(k+1)^2}, (1 - \sqrt{q_w})^{k-1}, \frac{(1 - \sqrt{q_w})^k}{t_0^2}\right\}.$$

Disclaimer

Compare recent work by Florea, Vorobyov (preprint, '17) where the same result is obtained via a *generalised estimate sequence* argument.

Monotone variants

In order to make the convergence non-increasing[3], we can simply set:

$$y^k = x^k + \beta_k \left(\left(x^k - x^{k-1} \right) + \frac{t_k}{t_k - 1} \left(T_{\tau_k}(y^{k-1}) - x^k \right) \right).$$

This suggests an easy rule to select x^{k+1} at any iteration:

$$x^{k+1} = \begin{cases} T_{\tau_{k+1}}(y^k) & \text{if } F(T_{\tau_{k+1}}(y^k)) \leq F(x^k), \\ x^k & \text{otherwise.} \end{cases}$$

Same computations and the same convergence rates!

[3] Beck, Teboulle '09, Tseng '08, Tao, Boley, Zhang '16

Imaging applications

Huber-TV Gaussian denoising

Given noisy $u^0 \in \mathbb{R}^{m \times n}$ corrupted by noise $\mathcal{N}(0, \sigma^2)$, use TV ROF[4] model:

$$\min_u \ \lambda \|Du\|_{2,1} + \frac{1}{2}\|u - u^0\|_2^2,$$

$$\|Du\|_{2,1} = \sum_{i,j=1}^{m,n} \sqrt{(Du)_{i,j,1}^2 + (Du)_{i,j,2}^2},$$

where Du is the finite-difference-discretised gradient and $\lambda > 0$ is a regularisation parameter.

[4] Rudin, Osher, Fatemi, '92

Huber-TV Gaussian denoising

Given noisy $u^0 \in \mathbb{R}^{m\times n}$ corrupted by noise $\mathcal{N}(0,\sigma^2)$, use TV ROF[4] model:

$$\min_u \ \lambda\|Du\|_{2,1} + \frac{1}{2}\|u-u^0\|_2^2,$$

$$\|Du\|_{2,1} = \sum_{i,j=1}^{m,n} \sqrt{(Du)_{i,j,1}^2 + (Du)_{i,j,2}^2},$$

where Du is the finite-difference-discretised gradient and $\lambda > 0$ is a regularisation parameter.

Strongly convex variant: for $\varepsilon \ll 1$, C^1-Huber regularisation

$$h_\varepsilon(t) := \begin{cases} \frac{t^2}{2\varepsilon} & \text{for } |t| \le \varepsilon, \\ |t| - \frac{\varepsilon}{2} & \text{for } |t| > \varepsilon. \end{cases}$$

[4]Rudin, Osher, Fatemi, '92

Huber-TV Gaussian denoising

Given noisy $u^0 \in \mathbb{R}^{m\times n}$ corrupted by noise $\mathcal{N}(0,\sigma^2)$, use TV ROF[4] model:

$$\min_u \ \lambda H_\varepsilon(u) + \frac{1}{2}\|u - u^0\|_2^2$$

$$H_\varepsilon(u) := \sum_{i,j=1}^{m,n} h_\varepsilon\left(\sqrt{(Du)_{i,j,1}^2 + (Du)_{i,j,2}^2}\right)$$

where Du is the finite-difference-discretised gradient and $\lambda > 0$ is a regularisation parameter.

Strongly convex variant: for $\varepsilon \ll 1$, C^1-Huber regularisation

$$h_\varepsilon(t) := \begin{cases} \frac{t^2}{2\varepsilon} & \text{for } |t| \le \varepsilon, \\ |t| - \frac{\varepsilon}{2} & \text{for } |t| > \varepsilon. \end{cases}$$

[4]Rudin, Osher, Fatemi, '92

Huber-TV Gaussian denoising: dual formulation

The Huber-TV dual problem reads:

$$\min_{\boldsymbol{p}} \frac{1}{2}\|D^*\boldsymbol{p} - u^0\|_2^2 + \frac{\varepsilon}{2\lambda}\|\boldsymbol{p}\|_2^2 + \delta_{\{\|\cdot\|_{2,\infty}\leq\lambda\}}(\boldsymbol{p}),$$

where D^* is the discretised negative finite-difference divergence and:

$$\delta_{\{\|\cdot\|_{2,\infty}\leq\lambda\}}(\boldsymbol{p}) = \begin{cases} 0 & \text{if } |\boldsymbol{p}_{i,j}|_2 \leq \lambda \text{ for any } i,j, \\ +\infty & \text{otherwise.} \end{cases}$$

Huber-TV Gaussian denoising: dual formulation

The Huber-TV dual problem reads:

$$\min_{\boldsymbol{p}} \underbrace{\frac{1}{2}\|D^*\boldsymbol{p} - u^0\|_2^2}_{\text{“}f\text{”}} + \underbrace{\frac{\varepsilon}{2\lambda}\|\boldsymbol{p}\|_2^2 + \delta_{\{\|\cdot\|_{2,\infty}\leq\lambda\}}(\boldsymbol{p})}_{\text{“}g\text{”}},$$

where D^* is the discretised negative finite-difference divergence and:

$$\delta_{\{\|\cdot\|_{2,\infty}\leq\lambda\}}(\boldsymbol{p}) = \begin{cases} 0 & \text{if } |\boldsymbol{p}_{i,j}|_2 \leq \lambda \text{ for any } i,j, \\ +\infty & \text{otherwise.} \end{cases}$$

Note:

- $\nabla f(\boldsymbol{p}) = D(D^*\boldsymbol{p} - u^0) \Longrightarrow L_f \leq 8$;
- $\text{prox}_{\tau g}$ is easy to compute and $\mu_g = \mu = \frac{\varepsilon}{\lambda}$.

Huber-TV Gaussian denoising: dual formulation

The Huber-TV dual problem reads:

$$\min_{\boldsymbol{p}} \underbrace{\frac{1}{2}\|D^*\boldsymbol{p} - u^0\|_2^2}_{\text{“}f\text{”}} + \underbrace{\frac{\varepsilon}{2\lambda}\|\boldsymbol{p}\|_2^2 + \delta_{\{\|\cdot\|_{2,\infty}\leq\lambda\}}(\boldsymbol{p})}_{\text{“}g\text{”}},$$

where D^* is the discretised negative finite-difference divergence and:

$$\delta_{\{\|\cdot\|_{2,\infty}\leq\lambda\}}(\boldsymbol{p}) = \begin{cases} 0 & \text{if } |\boldsymbol{p}_{i,j}|_2 \leq \lambda \text{ for any } i,j, \\ +\infty & \text{otherwise.} \end{cases}$$

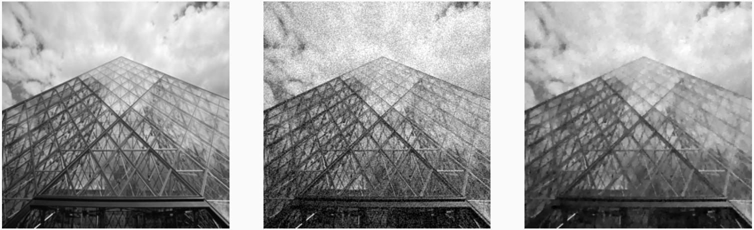
Note:

- $\nabla f(\boldsymbol{p}) = D(D^*\boldsymbol{p} - u^0) \Longrightarrow L_f \leq 8$;
- $\text{prox}_{\tau g}$ is easy to compute and $\mu_g = \mu = \frac{\varepsilon}{\lambda}$.

Use monotone GFISTA with backtracking. . .

Huber-TV Gaussian denoising: results

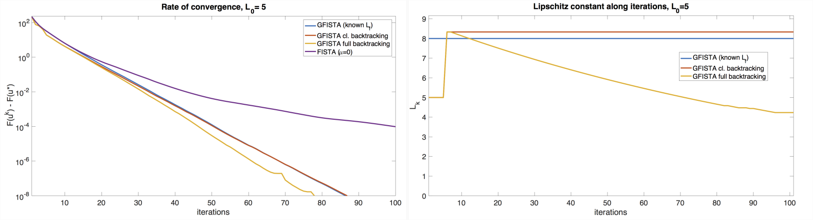
Parameters: $u^0 \in \mathbb{R}^{256\times256}$, $\sigma^2 = 0.005$, $\varepsilon = 0.01$, $\lambda = 0.1$. Have: $\mu = 0.1$



(a) Ground truth **(b)** u^0 **(c)** Reference u^*

Huber-TV Gaussian denoising: results

Parameters: $u^0 \in \mathbb{R}^{256\times256}$, $\sigma^2 = 0.005$, $\varepsilon = 0.01$, $\lambda = 0.1$. Have: $\mu = 0.1$

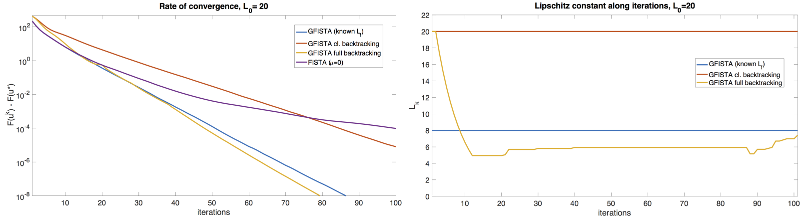


(a) Convergence rates. **(b)** Lipschitz constant estimate.

Figure 1: Underestimating $L_0 = 5$. GFISTA parameters: $\rho = 0.9$, $t_0 = 1$, $\boldsymbol{p_0} = Du^0$.

Huber-TV Gaussian denoising: results

Parameters: $u^0 \in \mathbb{R}^{256\times 256}$, $\sigma^2 = 0.005$, $\varepsilon = 0.01$, $\lambda = 0.1$. Have: $\mu = 0.1$

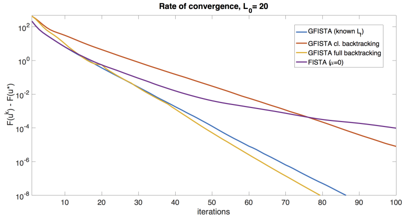


(a) Convergence rates. **(b)** Lipschitz constant estimate.

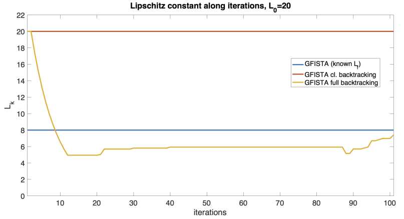
Figure 1: Overestimating $L_0 = 20$. GFISTA parameters: $\rho = 0.9$, $t_0 = 1$, $\boldsymbol{p_0} = Du^0$.

Huber-TV Gaussian denoising: results

Parameters: $u^0 \in \mathbb{R}^{256\times 256}$, $\sigma^2 = 0.005$, $\varepsilon = 0.01$, $\lambda = 0.1$. Have: $\mu = 0.1$



(a) Convergence rates.



(b) Lipschitz constant estimate.

Figure 1: Overestimating $L_0 = 20$. GFISTA parameters: $\rho = 0.9$, $t_0 = 1$, $\boldsymbol{p_0} = Du^0$.

Remark

Note that using naive FISTA ($\mu = 0$) results in $O(1/k^2)$ convergence.

Strongly convex TV-Poisson denoising: primal formulation

Poisson noise is typical in astronomy and microscopy imaging...

For $\varepsilon \ll 1$, consider the ε-strongly convex TV Poisson denoising model:

$$\min_u \; \lambda\|Du\|_{2,1} + \frac{\varepsilon}{2}\|u\|_2^2 + \tilde{KL}(u_0, u),$$

where $\tilde{KL}(u^0, u)$ is a differentiable version of the Kullback-Leibler function[5]:

$$\tilde{KL}(u^0, u) := \sum_{i,j=1}^{m,n} \begin{cases} u_{i,j} + b_{i,j} - u^0_{i,j} + u^0_{i,j} \log\left(\frac{u^0_{i,j}}{u_{i,j}+b_{i,j}}\right) & \text{if } u_{i,j} \geq 0, \\ \frac{u^0_{i,j}}{2b^2_{i,j}} u^2_{i,j} + \left(1 - \frac{u^0_{i,j}}{b_{i,j}}\right) u_{i,j} + b_{i,j} - u^0_{i,j} + u^0_{i,j} \log\left(\frac{u^0_{i,j}}{b_{i,j}}\right) & \text{else,} \end{cases}$$

and $b \in \mathbb{R}^{m\times n}$ is the background image. We can estimate:

$$L_f = \max_{i,j} \frac{u^0_{i,j}}{b^2_{i,j}}, \qquad \text{for } u^0, b > 0.$$

Moreover, $\text{prox}_{\tau g}$ can be computed solving TV ROF model.

[5] Chambolle, Ehrhardt, Richtarik, Schönlieb, '17

Strongly convex TV-Poisson denoising: primal formulation

Poisson noise is typical in astronomy and microscopy imaging...

For $\varepsilon \ll 1$, consider the ε-strongly convex TV Poisson denoising model:

$$\min_u \underbrace{\lambda \|Du\|_{2,1} + \frac{\varepsilon}{2}\|u\|_2^2}_{\text{“}g\text{”}} + \underbrace{\tilde{KL}(u_0, u)}_{\text{“}f\text{”}},$$

where $\tilde{KL}(u^0, u)$ is a differentiable version of the Kullback-Leibler function[5]:

$$\tilde{KL}(u^0, u) := \sum_{i,j=1}^{m,n} \begin{cases} u_{i,j} + b_{i,j} - u^0_{i,j} + u^0_{i,j} \log\left(\frac{u^0_{i,j}}{u_{i,j}+b_{i,j}}\right) & \text{if } u_{i,j} \geq 0, \\ \frac{u^0_{i,j}}{2b^2_{i,j}} u^2_{i,j} + \left(1 - \frac{u^0_{i,j}}{b_{i,j}}\right) u_{i,j} + b_{i,j} - u^0_{i,j} + u^0_{i,j} \log\left(\frac{u^0_{i,j}}{b_{i,j}}\right) & \text{else,} \end{cases}$$

and $b \in \mathbb{R}^{m\times n}$ is the background image. We can estimate:

$$L_f = \max_{i,j} \frac{u^0_{i,j}}{b^2_{i,j}}, \qquad \text{for } u^0, b > 0.$$

Moreover, $\text{prox}_{\tau g}$ can be computed solving TV ROF model.

[5] Chambolle, Ehrhardt, Richtarik, Schönlieb, ’17

Strongly convex TV-Poisson denoising: results

Parameters: $u^0 \in \mathbb{R}^{256\times 256}$, $\varepsilon = \mu = 0.15$, $\lambda = 0.2$. b constant, $L_f \leq 45$.

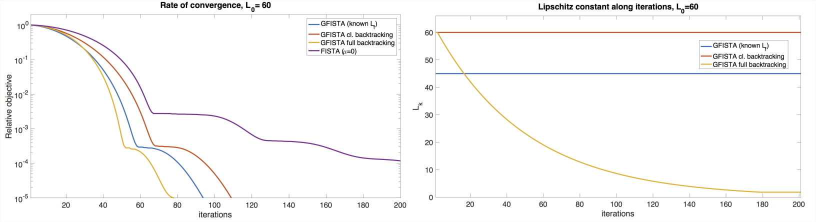
(a) Ground truth

(b) u^0

(c) Reference u^*

Strongly convex TV-Poisson denoising: results

Parameters: $u^0 \in \mathbb{R}^{256\times 256}$, $\varepsilon = \mu = 0.15$, $\lambda = 0.2$. b constant, $L_f \leq 45$.



(a) Convergence rates. **(b)** Lipschitz constant estimate.

Figure 2: Overestimating $L_0 = 60$. GFISTA parameters: $\rho = 0.8$, $t_0 = 1$, $u_0 = u^0$. Relative objective: $\frac{F(u^k)-F(u^*)}{F(u^0)-F(u^*)}$.

Strongly convex TV-Poisson denoising: results

Parameters: $u^0 \in \mathbb{R}^{256\times 256}$, $\varepsilon = \mu = 0.15$, $\lambda = 0.2$. b constant, $L_f \leq 45$.

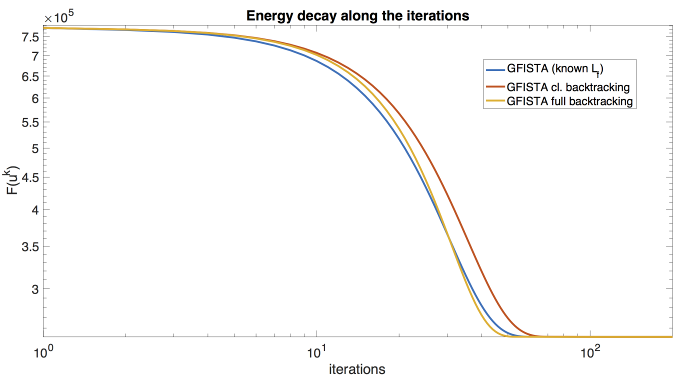


Figure 2: Monotone decay with/without backtracking.

Conclusions & outlook

Take-home messages:

- If $\mu_f, \mu_g > 0$, **linear convergence** can be shown for GFISTA;
- If L_f is not known, use **adaptive backtracking** to get a *local* estimate L_k along the iterations.
- GFISTA with backtracking can be easily implemented!

Conclusions & outlook

Take-home messages:

- If $\mu_f, \mu_g > 0$, **linear convergence** can be shown for GFISTA;
- If L_f is not known, use **adaptive backtracking** to get a *local* estimate L_k along the iterations.
- GFISTA with backtracking can be easily implemented!

Outlook:

- Estimate of μ_f and μ_g? (Nesterov '13, Ferocq, Qu, '16).
- Application to other imaging problems?

Main references

- L. Calatroni, A. Chambolle, *Backtracking strategies for accelerated descent methods with smooth composite objectives*, arXiv:1709.09004, 2017.

- K. Scheinberg, D. Goldfarb, and X. Bai, *Fast first-order methods for composite convex optimization with backtracking*, Foundations of Computational Mathematics 14, 3, 2014.
- M. I. Florea, S. Vorobyov, *A generalized accelerated composite gradient method: uniting Nesterov's fast gradient method and FISTA*, arXiv:1705.10266, 2017.
- A. Chambolle, T. Pock, *An introduction to continuous optimization for imaging*, Acta Numerica 25, 2016.

Merci pour votre attention !

Questions ?

luca.calatroni@polytechnique.edu