Optimal first-order minimization methods

with applications to image reconstruction and ML

Donghwan Kim & Jeffrey A. Fessler

EECS Dept., BME Dept., Dept. of Radiology
University of Michigan

http://web.eecs.umich.edu/~fessler

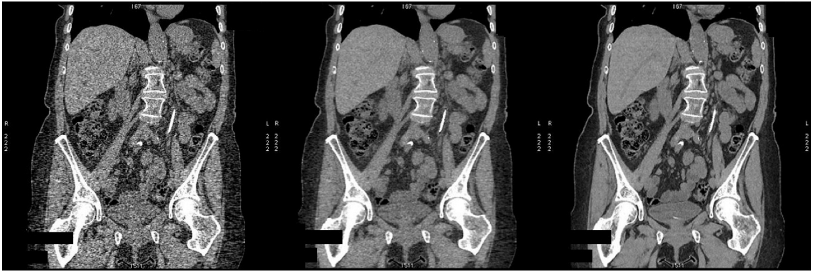
Zhejiang University Seminar

2016-09-22

Disclosure

- Research support from GE Healthcare
- Supported in part by NIH grant U01 EB018753
- Equipment support from Intel Corporation

Lower-dose X-ray CT image reconstruction



Thin-slice FBP	ASIR	Statistical
Seconds	A bit longer	Much longer

Image reconstruction as an optimization problem:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x} \succeq \boldsymbol{0}} \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_{\boldsymbol{W}}^2 + \mathsf{R}(\boldsymbol{x}),$$

$\boldsymbol{y}$ data, $\boldsymbol{A}$ system model, $\boldsymbol{W}$ statistics, $\mathsf{R}(\boldsymbol{x})$ regularizer.

(Same sinogram, so all at same dose.)

Outline

Outline

Optimization problem setting

$$\hat{\boldsymbol{x}} \in \arg\min_{\boldsymbol{x}} f(\boldsymbol{x})$$

- Unconstrained
- Large-scale (Hessian $\nabla^2 f$ too big to store and/or undefined)
  - image reconstruction / inverse problems
  - big-data / machine learning
  - ...
- Cost function assumptions (throughout)
  - $f : \mathbb{R}^M \mapsto \mathbb{R}$
  - convex (need not be strictly convex)
  - non-empty set of global minimizers:

$$\hat{\boldsymbol{x}} \in \mathcal{X}^* = \left\{\boldsymbol{x}_\star \in \mathbb{R}^M : \ f(\boldsymbol{x}_\star) \leq f(\boldsymbol{x}), \ \forall \boldsymbol{x} \in \mathbb{R}^M\right\}$$

  - smooth (differentiable with L-Lipschitz gradient)

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{z})\|_2 \leq L \|\boldsymbol{x} - \boldsymbol{z}\|_2, \quad \forall \boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^M$$

Example: Fair potential function

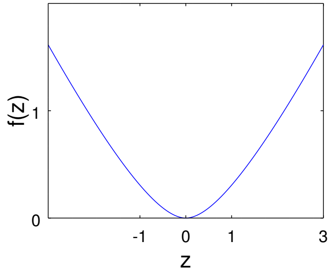
Fair's potential function [1] (similar to Huber function and hyperbola):

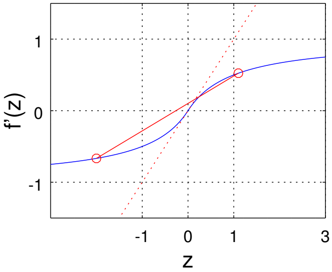
$$\psi(z) = \delta^2 \left[|z/\delta| - \log(1 + |z/\delta|) \right]$$

$$\dot{\psi}(z) = \frac{z}{1 + |z/\delta|}$$

$$\ddot{\psi}(z) = \frac{1}{(1 + |z/\delta|)^2} \le 1.$$

Thus $L = 1$.

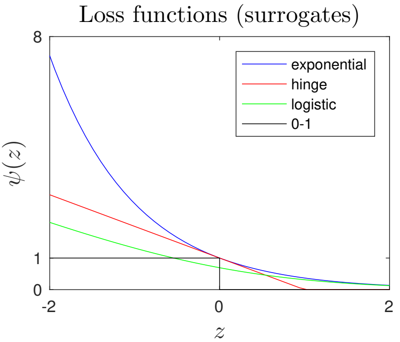




Example: Machine learning

To learn weights $\boldsymbol{x}$ of binary classifier given feature vectors $\{\boldsymbol{v}_i\}$ and labels $\{y_i = \pm 1\}$:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}), \qquad f(\boldsymbol{x}) = \sum_i \psi(y_i \langle \boldsymbol{x}, \boldsymbol{v}_i \rangle).$$



loss functions $\psi(z)$

- 0-1: $\mathbb{I}_{\{z \leq 0\}}$
- exponential: $\exp(-z)$
- logistic: $\log(1 + \exp(-z))$
- hinge: $\max\{0, 1 - z\}$

Which of these ψ fit our conditions?

Outline

Gradient descent

- Problem:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) .$$

- Initial guess $\boldsymbol{x}_0$.
- Simple recursive iteration:

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \frac{1}{L} \nabla f(\boldsymbol{x}_n) .$$

- Step size $1/L$ ensures monotonic descent of f.
- Telescoping (for intuition, not implementation):

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_0 - \frac{1}{L} \sum_{k=0}^{n} \nabla f(\boldsymbol{x}_k) .$$

Gradient descent convergence rate

- Classic $O(1/n)$ convergence rate of cost function descent:

$$\underbrace{f(\boldsymbol{x}_n) - f(\boldsymbol{x}_\star)}_{\text{inaccuracy}} \leq \frac{L \left\| \boldsymbol{x}_0 - \boldsymbol{x}_\star \right\|_2^2}{2n}.$$

- Drori & Teboulle (2014) derive tight inaccuracy bound:

$$f(\boldsymbol{x}_n) - f(\boldsymbol{x}_\star) \leq \frac{L \left\| \boldsymbol{x}_0 - \boldsymbol{x}_\star \right\|_2^2}{4n + 2}.$$

- They construct a Huber-like function f for which GD achieves that bound $\implies$ case closed for GD with step size $1/L$.
- $O(1/n)$ rate is undesirably slow.

Generalizing GD slightly

- GD with general step size h:

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \frac{h}{L} \nabla f(\boldsymbol{x}_n) \,.$$

- Classical monotone descent result:
  $h \in (0, 2) \implies f(\boldsymbol{x}_{n+1}) < f(\boldsymbol{x}_n)$ when $\boldsymbol{x}_n$ is not a minimizer.
- What is best h?
- If f is quadratic, then *asymptotic* best choice is:

$$h_* = \frac{2L}{\lambda_{\max}(\nabla^2 f) + \lambda_{\min}(\nabla^2 f)}.$$

Generalizing GD slightly

- GD with general step size h:

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \frac{h}{L} \nabla f(\boldsymbol{x}_n) .$$

- More generally, Taylor et al. [3] recently conjectured:

$$f(\boldsymbol{x}_N) - f(\boldsymbol{x}_\star) \leq \frac{L \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2}{2} \max\left\{ \frac{1}{2Nh+1}, \; (1-h)^{2N} \right\} .$$

  - Proof for $0 < h \leq 1$ by Drori and Teboulle [2]
  - Upper bounds achieved by Huber-like function and quadratic function $f(x) = (L/2)x^2$ respectively.
  - Best h depends on N !
    (For $N = 1$, $h_* = 1.5$; for $N = 100$, $h_* = 1.9705$.)
  - Must select N in advance?
  - Still $O(1/N)$...

Heavy ball method and momentum

- Quest for accelerated convergence.
- Heavy ball iteration (Polyak, 1987):

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \frac{\alpha}{L} \nabla f(\boldsymbol{x}_n) + \underbrace{\beta\ (\boldsymbol{x}_n - \boldsymbol{x}_{n-1})}_{\text{momentum!}}$$

(recursive form to implement)

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \frac{1}{L} \sum_{k=0}^{n} \underbrace{\alpha\beta^{n-k}}_{\text{coefficients}} \nabla f(\boldsymbol{x}_k)$$

(summation form to analyze)

- How to choose α and β?
- How to optimize coefficients more generally?

General first-order method classes

- General "first-order" (GFO) method:

$$\boldsymbol{x}_{n+1} = \text{function}(\boldsymbol{x}_0, f(\boldsymbol{x}_0), \nabla f(\boldsymbol{x}_0), \ldots, f(\boldsymbol{x}_n), \nabla f(\boldsymbol{x}_n)) .$$

- First-order (FO) methods with fixed step-size coefficients:

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \frac{1}{L} \sum_{k=0}^{n} h_{n+1,k} \nabla f(\boldsymbol{x}_k) .$$

Primary goals:

- Analyze convergence rate of FO for any given $\{h_{n,k}\}$
- Optimize step-size coefficients $\{h_{n,k}\}$
    - fast convergence
    - efficient recursive implementation
    - universal (design *prior* to iterating, independent of L)

Example: Barzilai-Borwein gradient method

Barzilai & Borwein, 1988:

$$\boldsymbol{g}^{(n)} \triangleq \nabla f(\boldsymbol{x}_n)$$

$$\alpha_n = \frac{\|\boldsymbol{x}_n - \boldsymbol{x}_{n-1}\|_2^2}{\langle \boldsymbol{x}_n - \boldsymbol{x}_{n-1},\, \boldsymbol{g}^{(n)} - \boldsymbol{g}^{(n-1)} \rangle}$$

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \alpha_n \nabla f(\boldsymbol{x}_n)\,.$$

- In "general" first-order (GFO) class, but
- not in class FO with fixed step-size coefficients.
- Likewise for methods like
  - steepest descent (with line search),
  - conjugate gradient,
  - quasi-Newton ...

Nesterov's fast gradient method (FGM1)

Nesterov (1983) iteration: Initialize: $t_0 = 1$, $\boldsymbol{z}_0 = \boldsymbol{x}_0$

$$\boldsymbol{z}_{n+1} = \boldsymbol{x}_n - \frac{1}{L} \nabla f(\boldsymbol{x}_n) \qquad \text{(usual GD update)}$$

$$t_{n+1} = \frac{1}{2}\left(1 + \sqrt{1 + 4t_n^2}\right) \qquad \text{(magic momentum factors)}$$

$$\boldsymbol{x}_{n+1} = \boldsymbol{z}_{n+1} + \frac{t_n - 1}{t_{n+1}} (\boldsymbol{z}_{n+1} - \boldsymbol{z}_n) \qquad \text{(update with momentum)} .$$

Reverts to GD if $t_n = 1, \forall n$.

FGM1 is in class FO: $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \frac{1}{L} \sum_{k=0}^{n} h_{n+1,k} \nabla f(\boldsymbol{x}_k)$

$$h_{n+1,k} = \begin{cases} \frac{t_n - 1}{t_{n+1}} h_{n,k}, & k = 0, \ldots, n-2 \\ \frac{t_n - 1}{t_{n+1}} (h_{n,n-1} - 1), & k = n-1 \\ 1 + \frac{t_n - 1}{t_{n+1}}, & k = n. \end{cases}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.25 & 0 & 0 & 0 & 0 \\ 0 & 0.10 & 1.40 & 0 & 0 & 0 \\ 0 & 0.05 & 0.20 & 1.50 & 0 & 0 \\ 0 & 0.03 & 0.11 & 0.29 & 1.57 & 0 \\ 0 & 0.02 & 0.07 & 0.18 & 0.36 & 1.62 \end{bmatrix}$$

Nesterov's FGM1 optimal convergence rate

Shown by Nesterov to be $O(1/n^2)$ for "auxiliary" sequence:

$$f(\boldsymbol{z}_n) - f(\boldsymbol{x}_\star) \leq \frac{2L \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2}{(n+1)^2}.$$

For any FO method, Nesterov constructed a function f such that

$$\frac{\frac{3}{32} L \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2}{(n+1)^2} \leq f(\boldsymbol{x}_n) - f(\boldsymbol{x}_\star).$$

Thus $O(1/n^2)$ rate of FGM1 is optimal.

New results (Donghwan Kim & JF, 2016):

- Bound on convergence rate of primary sequence $\{\boldsymbol{x}_n\}$:

$$f(\boldsymbol{x}_n) - f(\boldsymbol{x}_\star) \leq \frac{2L \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2}{(n+2)^2}.$$

- Verifies (numerically inspired) conjecture of Drori & Teboulle (2014).

Overview

First-order (FO) method with fixed step-size coefficients:

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \frac{1}{L} \sum_{k=0}^{n} h_{n+1,k} \nabla f(\boldsymbol{x}_k)$$

- Analyze (*i.e.*, bound) convergence rate as a function of
  - number of iterations N
  - Lipschitz constant L
  - step-size coefficients $H = \{h_{n+1,k}\}$
  - initial distance to a solution: $R = \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|$.
- Optimize H by minimizing the bound.
- Seek an equivalent recursive form for efficient implementation.

Ideal "universal" bound for first-order methods

For given

- number of iterations N
- Lipschitz constant L
- step-size coefficients $H = \{h_{n+1,k}\}$
- initial distance to a solution: $R = \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|$,

try to bound the worst-case convergence rate of a FO method:

$$B_1(H, R, L, N, M) \triangleq \max_{f \in \mathcal{F}_L} \max_{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^M} \max_{\substack{\boldsymbol{x}_\star \in \mathcal{X}^*(f) \\ \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\| \le R}} f(\boldsymbol{x}_N) - f(\boldsymbol{x}_\star)$$

$$\text{such that} \quad \boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \frac{1}{L} \sum_{k=0}^{n} h_{n+1,k} \nabla f(\boldsymbol{x}_k), \quad n = 0, \ldots, N-1.$$

Clearly for any FO method, this cost-function bound would hold:

$$f(\boldsymbol{x}_N) - f(\boldsymbol{x}_\star) \le B_1(H, R, L, N, M).$$

Towards practical bounds for first-order methods

For convex functions with L-Lipschitz gradients:

$$\frac{1}{2L} \|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{z})\|^2 \leq f(\boldsymbol{x}) - f(\boldsymbol{z}) - \langle \nabla f(\boldsymbol{z}), \boldsymbol{x} - \boldsymbol{z} \rangle, \quad \forall \boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^M.$$

Drori & Teboulle (2014) use this inequality to propose a "more tractable" (finite-dimensional) relaxed bound:

$$B_2(H, R, L, N, M) \triangleq \max_{\boldsymbol{g}_0, \ldots, \boldsymbol{g}_N \in \mathbb{R}^M} \max_{\delta_0, \ldots, \delta_N \in \mathbb{R}} \max_{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^M} \max_{\boldsymbol{x}_\star : \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\| \leq R} LR\delta_N^2$$

$$\text{such that} \quad \boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \frac{1}{L} \sum_{k=0}^{n} h_{n+1,k} R \, \boldsymbol{g}_k, \quad n = 0, \ldots, N-1,$$

$$\frac{1}{2} \left\| \boldsymbol{g}_i - \boldsymbol{g}_j \right\|^2 \leq \delta_i - \delta_j - \frac{1}{R} \langle \boldsymbol{g}_j, \boldsymbol{x}_i - \boldsymbol{x}_j \rangle, \quad i, j = 0, \ldots, N, *,$$

where $\boldsymbol{g}_n = \frac{1}{LR} \nabla f(\boldsymbol{x}_n)$ and $\delta_n = \frac{1}{LR} \left(f(\boldsymbol{x}_n) - f(\boldsymbol{x}_\star) \right)$.

For any FO method:

$$f(\boldsymbol{x}_N) - f(\boldsymbol{x}_\star) \leq B_1(H, R, L, N, M) \leq B_2(H, R, L, N, M)$$

However, even B_2 is as of yet unsolved.

- Drori & Teboulle (2014) further relax the bound:

$$f(\boldsymbol{x}_N) - f(\boldsymbol{x}_\star) \leq B_1(H, \ldots) \leq B_2(H, \ldots) \leq B_3(H, R, L, N).$$

- For given step-size coefficients H, and given number of iterations N, they use a semi-definite program (SDP) to compute B_3 numerically.
- They find numerically that for the FGM1 choice of H, the convergence bound B_3 is slightly below $\frac{2L \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2}{(N+1)^2}$.
- This suggested that improvements on FGM1 could exist.

Optimizing step-size coefficients numerically

Drori & Teboulle (2014) also computed numerically the minimizer over H of their relaxed bound for given N using a SDP:

$$H^* = \arg\min_H B_3(H, R, L, N).$$

Numerical solution for H^* for $N = 5$ iterations: [2, Ex. 3]

0. Input: $f \in C_L^{1,1}(\mathbb{R}^d)$, $x_0 \in \mathbb{R}^d$,
1. $x_1 = x_0 - \frac{1.6180}{L} f'(x_0)$,
2. $x_2 = x_1 - \frac{0.1741}{L} f'(x_0) - \frac{2.0194}{L} f'(x_1)$,
3. $x_3 = x_2 - \frac{0.0756}{L} f'(x_0) - \frac{0.4425}{L} f'(x_1) - \frac{2.2317}{L} f'(x_2)$,
4. $x_4 = x_3 - \frac{0.0401}{L} f'(x_0) - \frac{0.2350}{L} f'(x_1) - \frac{0.6541}{L} f'(x_2) - \frac{2.3656}{L} f'(x_3)$,
5. $x_5 = x_4 - \frac{0.0178}{L} f'(x_0) - \frac{0.1040}{L} f'(x_1) - \frac{0.2894}{L} f'(x_2) - \frac{0.6043}{L} f'(x_3) - \frac{2.0778}{L} f'(x_4)$.

Drawbacks:

- Must choose N in advance
- Requires $O(N)$ memory for all gradient vectors $\{\nabla f(\boldsymbol{x}_n)\}_{n=1}^N$
- $O(N^2)$ computation for N iterations

Benefit: convergence bound (for specific N) $\approx 2\times$ lower than for Nesterov's FGM1.

New analytical solution

- Analytical solution for optimized step-size coefficients [8], [9]:

$$H^* \; : \quad h_{n+1,k} = \begin{cases} \frac{\theta_n - 1}{\theta_{n+1}} h_{n,k}, & k = 0, \ldots, n-2 \\ \frac{\theta_n - 1}{\theta_{n+1}} \left(h_{n,n-1} - 1 \right), & k = n-1 \\ 1 + \frac{2\theta_n - 1}{\theta_{n+1}}, & k = n. \end{cases}$$

$$\theta_n = \begin{cases} 1, & n = 0 \\ \frac{1}{2}\left(1 + \sqrt{1 + 4\theta_{n-1}^2}\right), & n = 1, \ldots, N-1 \\ \frac{1}{2}\left(1 + \sqrt{1 + 8\theta_{n-1}^2}\right), & n = N. \end{cases}$$

- Analytical convergence bound for this optimized H^*:

$$f(\boldsymbol{x}_N) - f(\boldsymbol{x}_\star) \le B_3(H^*, R, L, N) = \frac{1 L \left\|\boldsymbol{x}_0 - \boldsymbol{x}_\star\right\|_2^2}{(N+1)(N+1+\sqrt{2})}.$$

  - Of course bound is $O(1/N^2)$, but constant is twice better.
  - No numerical SDP needed $\Longrightarrow$ feasible for large N.
  - (History: sought banded / structured lower-triangular form)

Outline

- Motivation
- Problem setting
- Existing algorithms
  - Gradient descent
  - Nesterov's "optimal" first-order method
- Optimizing first-order minimization methods
- Numerical examples
  - Logistic regression for machine learning
  - CT image reconstruction
  - Further acceleration using OS
- Generalizing OGM
- Summary / future work

Optimized gradient method (OGM1)

Donghwan Kim & JF (2016) also found efficient recursive iteration:
Initialize: $\theta_0 = 1$, $\boldsymbol{z}_0 = \boldsymbol{x}_0$

$$\boldsymbol{z}_{n+1} = \boldsymbol{x}_n - \frac{1}{L} \nabla f(\boldsymbol{x}_n)$$

$$\theta_n = \begin{cases} \frac{1}{2}\left(1 + \sqrt{1 + 4\theta_{n-1}^2}\right), & n = 1, \ldots, N-1 \\ \frac{1}{2}\left(1 + \sqrt{1 + 8\theta_{n-1}^2}\right), & n = N \end{cases}$$

$$\boldsymbol{x}_{n+1} = \boldsymbol{z}_{n+1} + \frac{\theta_n - 1}{\theta_{n+1}} (\boldsymbol{z}_{n+1} - \boldsymbol{z}_n) + \underbrace{\frac{\theta_n}{\theta_{n+1}} (\boldsymbol{z}_{n+1} - \boldsymbol{x}_n)}_{\text{new momentum}}.$$

Reverts to Nesterov's FGM1 if the new term is removed.

- Very simple modification of existing Nesterov code.
- No need to solve SDP.
- Factor of 2 better bound than Nesterov's "optimal" FGM1.

(Proofs omitted.)

Recent refinement of OGM1

New version OGM1' [10], [11]

$$\boldsymbol{z}_{n+1} = \boldsymbol{x}_n - \frac{1}{L} \nabla f(\boldsymbol{x}_n) \qquad \text{(usual GD update)}$$

$$t_{n+1} = \frac{1}{2}\left(1 + \sqrt{1 + 4t_n^2}\right) \qquad \text{(momentum factors)}$$

$$\boldsymbol{x}_{n+1} = \boldsymbol{z}_{n+1} + \frac{t_n - 1}{t_{n+1}} (\boldsymbol{z}_{n+1} - \boldsymbol{z}_n) + \underbrace{\frac{t_n}{t_{n+1}} (\boldsymbol{z}_{n+1} - \boldsymbol{x}_n)}_{\text{OGM1 momentum}}$$

- New convergence bound for *every iteration*:

$$f(\boldsymbol{z}_n) - f(\boldsymbol{x}_\star) \leq \frac{1 L \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2}{(n+1)^2}.$$

- Simpler and more practical implementation.
- Need not pick N in advance.

Optimized gradient method (OGM) is optimal!

For the class of first-order (FO) methods with fixed step sizes:

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \frac{1}{L}\sum_{k=0}^{n} h_{n+1,k} \nabla f(\boldsymbol{x}_k),$$

we optimized OGM and proved the convergence rate upper bound:

$$f(\boldsymbol{x}_N) - f(\boldsymbol{x}_\star) \leq \frac{L \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2}{N^2}.$$

Recently Y. Drori [12] considered the class of general FO methods:

$$\boldsymbol{x}_{n+1} = F(\boldsymbol{x}_0, f(\boldsymbol{x}_0), \nabla f(\boldsymbol{x}_0), \ldots, f(\boldsymbol{x}_n), \nabla f(\boldsymbol{x}_n)),$$

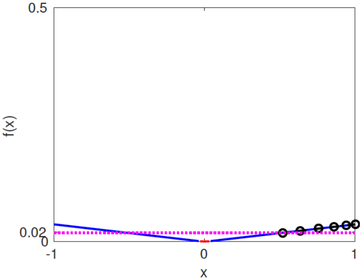
and showed any algorithm in this case has a function f such that

$$\frac{L \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2}{N^2} \leq f(\boldsymbol{x}_N) - f(\boldsymbol{x}_\star),$$

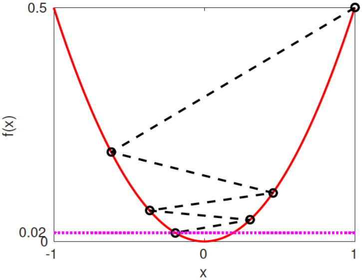
for $d > N$ (large-scale). Thus OGM has optimal complexity among all FO methods!

Worst-case functions for OGM

From [10], [11], worst-case behavior is:



(c) $N = 5$: $f_{1,\mathrm{OGM}}(\boldsymbol{x};5)$



(d) $N = 5$: $f_2(\boldsymbol{x})$

OGM has two worst-case functions (like GM):
a Huber-like function and a quadratic function.
Worst-case means:

$$f(\boldsymbol{x}_N) - f(\boldsymbol{x}_\star) = \frac{LR^2}{\theta_N^2} \leq \frac{LR^2}{(N+1)(N+1+\sqrt{2})} \leq \frac{LR^2}{(N+1)^2}.$$

Outline

Machine learning (logistic regression)

To learn weights $\boldsymbol{x}$ of binary classifier given feature vectors $\{\boldsymbol{v}_i\}$ and labels $\{y_i = \pm 1\}$:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}), \qquad f(\boldsymbol{x}) = \sum_i \psi(y_i \langle \boldsymbol{x}, \boldsymbol{v}_i \rangle) + \beta \frac{1}{2} \|\boldsymbol{x}\|_2^2 .$$

logistic:
$$\psi(z) = \log(1 + \mathrm{e}^{-z}), \quad \dot{\psi}(z) = \frac{-1}{\mathrm{e}^z + 1}, \quad \ddot{\psi}(z) = \frac{\mathrm{e}^z}{(\mathrm{e}^z + 1)^2} \in \left(0, \frac{1}{4}\right].$$

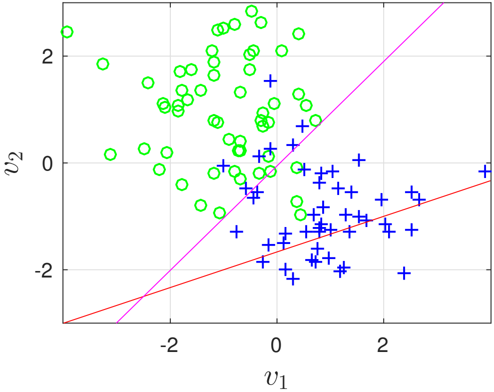
Gradient $\nabla f(\boldsymbol{x}) = \sum_i y_i \boldsymbol{v}_i \dot{\psi}(y_i \langle \boldsymbol{x}, \boldsymbol{v}_i \rangle) + \beta \boldsymbol{x}$

Hessian is positive definite so strictly convex:

$$\nabla^2 f(\boldsymbol{x}) = \sum_i \boldsymbol{v}_i \ddot{\psi}(y_i \langle \boldsymbol{x}, \boldsymbol{v}_i \rangle) \boldsymbol{v}_i' + \beta \boldsymbol{I} \preceq \frac{1}{4} \sum_i \boldsymbol{v}_i \boldsymbol{v}_i' + \beta \boldsymbol{I}$$

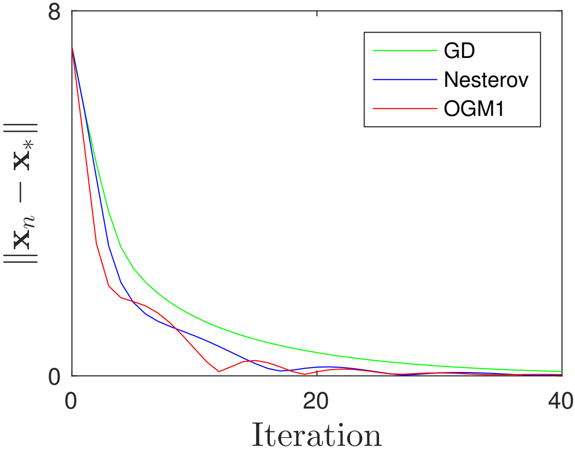
$$\Longrightarrow L \triangleq \frac{1}{4} \rho\left(\sum_i \boldsymbol{v}_i \boldsymbol{v}_i'\right) + \beta \geq \max_{\boldsymbol{x}} \rho\left(\nabla^2 f(\boldsymbol{x})\right)$$

Numerical Results: logistic regression

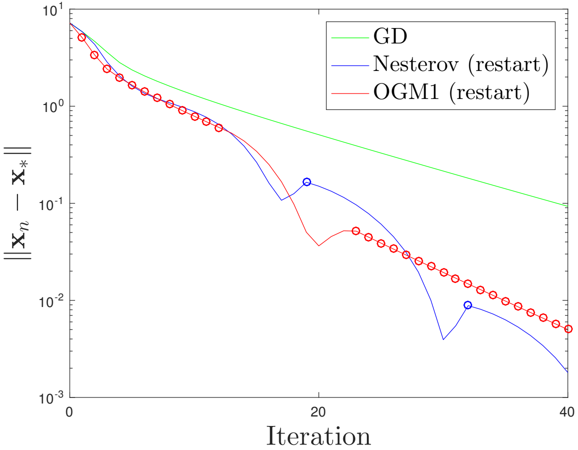


Training data (points); initial decision boundary (red); final decision boundary (magenta).

Numerical Results: convergence rates



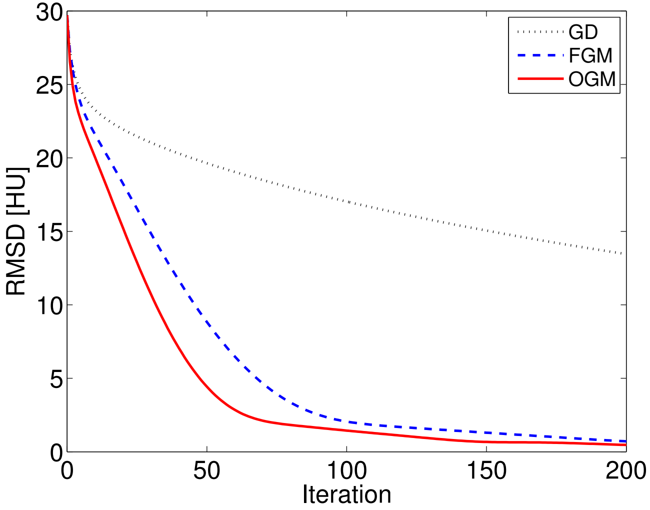
Numerical Results: adaptive restart



FGM restart, O'Donoghue & Candès, 2015.

How to best "restart" OGM1 is an open question.

Low-dose 2D X-ray CT image reconstruction simulation



Outline

Combining ordered subsets (OS) with momentum

- Optimization problems in image reconstruction (and machine learning) involve sums of many similar terms:

$$f(\mathbf{x}) = \sum_{m=1}^{M} f_m(\mathbf{x}) .$$

- Approximate gradients using just one term at a time:

$$\nabla f(\mathbf{x}) \approx M \nabla f_m(\mathbf{x})$$

  - Ordered subsets (OS) in tomography [15]
  - Incremental gradients in optimization / machine learning
- Combining OS with momentum dramatically accelerates!

OS + OGM1 method

Initialize: $\theta_0 = 1$, $\boldsymbol{z}_0 = \boldsymbol{x}_0$ (D. Kim, S. Ramani, JF, 2015) [16]

For each iteration n

For each subset $m = 1, \ldots, M$

$$k = nM + m - 1$$

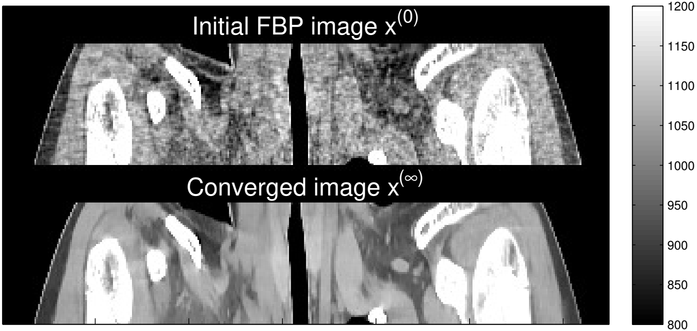
$$\boldsymbol{z}_{k+1} = \boldsymbol{x}_k - \frac{M}{L} \nabla f_m(\boldsymbol{x}_k) \quad \text{(usual OS update)}$$

$$\theta_k = \frac{1}{2}\left(1 + \sqrt{1 + 4\theta_{k-1}^2}\right) \quad \text{(momentum factors)}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{z}_{k+1} + \frac{\theta_k - 1}{\theta_{k+1}} (\boldsymbol{z}_{k+1} - \boldsymbol{z}_k) + \underbrace{\frac{\theta_k}{\theta_{k+1}} (\boldsymbol{z}_{k+1} - \boldsymbol{x}_k)}_{\text{new momentum}}.$$

- Simple modification of existing OS code
- $\approx O(1/(Mn)^2)$ decrease of cost function f in early iterations

- 3D cone-beam helical CT scan with pitch 0.5

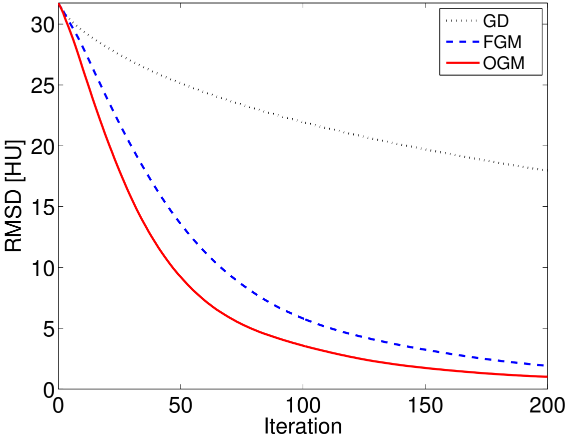


- Convergence rate in RMSD [HU], within ROI, versus iteration:

$$\mathrm{RMSD}_{\mathrm{ROI}}(\boldsymbol{x}_n) \triangleq \frac{||x_{\mathrm{ROI}}^{(n)} - \hat{x}_{\mathrm{ROI}}||_2}{\sqrt{N_{\mathrm{ROI}}}}.$$

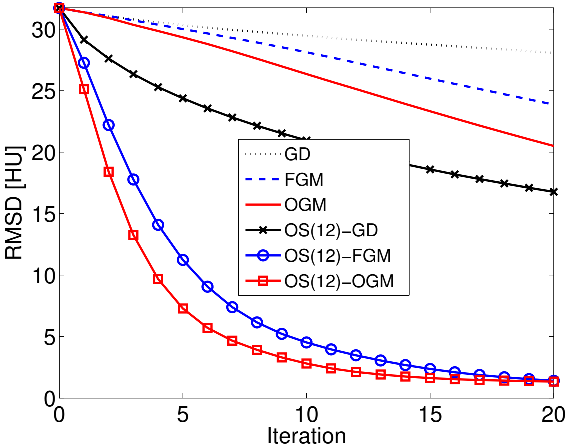
(Disclaimer: RMSD may not relate to task performance...)

Results: RMSD [HU] vs. iteration: without OS



- Computation time: OGM $<$ FGM $\ll$ GD
- OGM requires about $\frac{1}{\sqrt{2}}$-times fewer iterations than FGM to reach the same RMSD.

Results: RMSD [HU] vs. iteration: with OS



- $M = 12$ subsets in OS algorithm.
- Proposed OS-OGM converges faster than OS-FGM.
- Computation time per iteration of all algorithms are similar.

Outline

Generalizing OGM - faster gradient norm decrease

- Cost function decrease: $f(\boldsymbol{x}_n) - f(\boldsymbol{x}_\star) \sim O(1/n^2)$
- Gradient norm decrease? $\|\nabla f(\boldsymbol{x}_n)\| \to 0$ at what rate?

Important especially for problems involving duality.

Bounds on gradient norm decrease

- Known bounds [17] [19]:

$$\text{GM:} \quad \min_{0 \le n \le N} \|\nabla f(\boldsymbol{x}_n)\| = \|\nabla f(\boldsymbol{x}_N)\| \le \frac{\sqrt{2}}{N} LR$$

$$\text{FGM:} \quad \|\nabla f(\boldsymbol{x}_N)\| \le \frac{2}{N} LR.$$

- New very recent bounds (DK & JF, 2016) [20], [21]:

$$\text{FGM:} \quad \min_{0 \le n \le N} \|\nabla f(\boldsymbol{x}_n)\| \le \frac{2\sqrt{3}}{N^{3/2}} LR$$

$$\text{OGM:} \quad \min_{0 \le n \le N} \|\nabla f(\boldsymbol{x}_n)\| \le \|\nabla f(\boldsymbol{x}_N)\| \le \frac{\sqrt{2}}{N} LR.$$

- Can one do better than FGM?

Generalized OGM (GOGM) recursive iteration

Very recent generalization (DK & JF, 2016) [20], [21]

Input: $f \in \mathcal{F}_L$, $\boldsymbol{x}_0 \in \mathbb{R}^N$, $\boldsymbol{z}_0 = \boldsymbol{x}_0$, $t_0 \in (0,1]$.
for $n = 0, 1, \ldots$

$$\boldsymbol{z}_{n+1} = \boldsymbol{x}_n - \frac{1}{L} \nabla f(\boldsymbol{x}_n)$$

$$t_{n+1} > 0 \text{ s.t. } t_{n+1}^2 \leq T_{n+1} \triangleq \sum_{k=0}^{n+1} t_k \qquad \text{(momentum factors)}$$

$$\boldsymbol{x}_{n+1} = \boldsymbol{z}_{n+1} + \frac{(T_n - t_n) t_{n+1}}{T_{n+1} t_n} (\boldsymbol{z}_{n+1} - \boldsymbol{z}_n) + \frac{(2t_n^2 - T_n) t_{n+1}}{T_{n+1} t_n} (\boldsymbol{z}_{n+1} - \boldsymbol{x}_n) .$$

- Simple implementation
- Best choice of factors t_n (in terms of gradient norm decrease)?

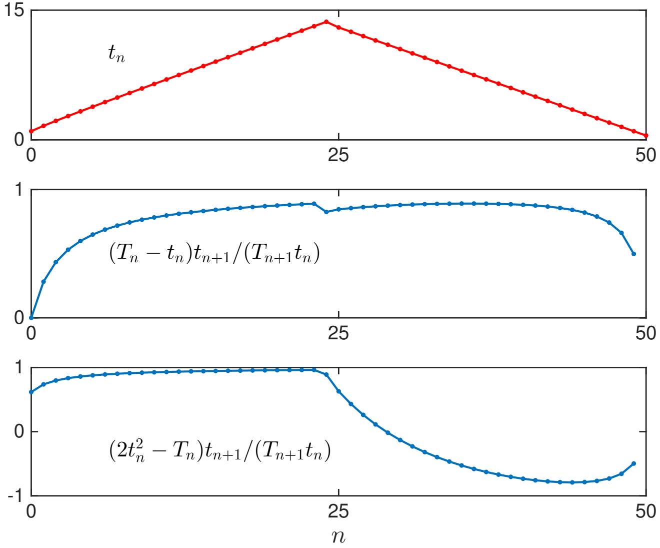
Generalized OGM (GOGM)

Optimized choice of momentum factors (for decreasing gradient norm) (DK & JF, 2016) [20], [21] :

$$t_n \triangleq \begin{cases} 1, & n = 0, \\ \frac{1}{2}\left(1 + \sqrt{1 + 4t_{n-1}^2}\right), & n = 0, \ldots, \lfloor N/2 \rfloor - 1, \\ (N - n + 1)/2, & n = \lfloor N/2 \rfloor, \ldots N. \end{cases}$$

Dubbed "OGM-OG" for OGM with optimized gradients.

Optimized parameters for OGM-OG



OGM-OG convergence rate bounds

- Convergence bound for cost function for OGM-OG:

$$f(\boldsymbol{z}_N) - f(\boldsymbol{x}_\star) \leq \frac{2L \left\| \boldsymbol{x}_0 - \boldsymbol{x}_\star \right\|_2^2}{N^2}.$$

- Same as Nesterov's FGM.
- Convergence bound for gradient norm is best known:

$$\min_{0 \leq n \leq N} \|\nabla f(\boldsymbol{z}_n)\| \leq \min_{0 \leq n \leq N} \|\nabla f(\boldsymbol{x}_n)\| \leq \frac{\sqrt{6}}{N^{3/2}} LR.$$

- $\sqrt{2}$ better than FGM's *smallest* gradient norm bound.
- Variations that do not require choosing N in advance, but that have slightly larger constants in bounds.
- Derivation uses relaxations that are not tight.
- Is $N^{3/2}$ best possible? What is best possible constant?

Summary of (fast?) gradient decreasing FO methods

From [20], [21]:

Algorithm	Asymptotic convergence rate bound		Require selecting N in advance
	Cost function	Gradient norm	
GM	$\frac{1}{4}N^{-1}$	$\sqrt{2}N^{-1}$	No
FGM	$2N^{-2}$	$2\sqrt{3}N^{-\frac{3}{2}}$	No
OGM	N^{-2}	$\sqrt{2}N^{-1}$	No
OGM-H	$4N^{-2}$	$4N^{-\frac{3}{2}}$	Yes
OGM-OG	$2N^{-2}$	$\sqrt{6}N^{-\frac{3}{2}}$	Yes
OGM-a ($a > 2$)	$\frac{a}{2}N^{-2}$	$\frac{a\sqrt{6}}{2\sqrt{a-2}}N^{-\frac{3}{2}}$	No
OGM-$a=4$	$2N^{-2}$	$2\sqrt{3}N^{-\frac{3}{2}}$	

Numerical examples are work-in-progress.

Non-smooth (composite) convex problems

Composite cost function:

$$\arg\min_{\boldsymbol{x}} F(\boldsymbol{x}), \quad F(\boldsymbol{x}) \triangleq f(\boldsymbol{x}) + g(\boldsymbol{x})$$

$f(\boldsymbol{x})$: convex, smooth with Lipshitz gradient
$g(\boldsymbol{x})$: convex but possibly (usually) non-smooth
Examples:

- $g(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$
- $g(\boldsymbol{x})$ characteristic function of a convex constraint

Fast iterative soft thresholding algorithm (FISTA) (Beck & Teboulle, 2009) [22]
AKA "fast proximal gradient method" (FPGM)
Simple recursive iteration with $O(1/n^2)$ cost function convergence rate

Improving on FISTA

DK & JF, 2016 [23], [24]

Algorithm	Asymptotic convergence rate bound		Require selecting N in advance
	Cost function ($\times LR^2$)	Proximal gradient ($\times LR$)	
PGM	$\frac{1}{2}N^{-1}$	$2N^{-1}$	No
FPGM [5]	$\mathbf{2N^{-2}}$	$2N^{-1}$	No
FPGM-σ ($0 < \sigma < 1$) [22]	$\frac{2}{\sigma^2}N^{-2}$	$\frac{2\sqrt{3}}{\sigma^2}\sqrt{\frac{1+\sigma}{1-\sigma}}N^{-\frac{3}{2}}$	No
FPGM-$\sigma = 0.78$	$3.3N^{-2}$	$16.2N^{-\frac{3}{2}}$	
FPGM-H	$8N^{-2}$	$5.7N^{-\frac{3}{2}}$	Yes
FPGM-OPG	$4N^{-2}$	$\mathbf{4.9N^{-\frac{3}{2}}}$	Yes
FPGM-a ($a > 2$)	aN^{-2}	$\frac{a\sqrt{6}}{\sqrt{a-2}}N^{-\frac{3}{2}}$	No
FPGM-$a = 4$	$4N^{-2}$	$6.9N^{-\frac{3}{2}}$	

FPGM with "optimized proximal gradient" (FPGM-OPG).
Best known bound on proximal gradient convergence rate.

Summary

- New optimized first-order minimization algorithm (optimal!)
- Simple implementation akin to Nesterov's FGM
- Analytical converge rate bound
- Bound on cost function decrease is $2\times$ better than Nesterov

Future work

- Constraints
- Non-smooth cost functions, *e.g.*, ℓ_1
- Tighter bounds
- Strongly convex case
- Asymptotic / local convergence rates
- Incremental gradients
- Stochastic gradient descent
- Adaptive restart
- Distributed computation
- Low-dose 3D X-ray CT image reconstruction

Bibliography I

[1] R. C. Fair, "On the robust estimation of econometric models," *Ann. Econ. Social Measurement*, vol. 2, 667–77, Oct. 1974.

[2] Y. Drori and M. Teboulle, "Performance of first-order methods for smooth convex minimization: A novel approach," *Mathematical Programming*, vol. 145, no. 1-2, 451–82, Jun. 2014.

[3] A. B. Taylor, J. M. Hendrickx, and François. Glineur, "Smooth strongly convex interpolation and exact worst-case performance of first- order methods," *Mathematical Programming*, 2016.

[4] B. T. Polyak, *Introduction to optimization*. New York: Optimization Software Inc, 1987.

[5] J. Barzilai and J. Borwein, "Two-point step size gradient methods," *IMA J. Numerical Analysis*, vol. 8, no. 1, 141–8, 1988.

[6] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$," *Dokl. Akad. Nauk. USSR*, vol. 269, no. 3, 543–7, 1983.

[7] ——,"Smooth minimization of non-smooth functions," *Mathematical Programming*, vol. 103, no. 1, 127–52, May 2005.

[8] D. Kim and J. A. Fessler, *Optimized first-order methods for smooth convex minimization*, arxiv 1406.5468, 2014.

[9] ——,"Optimized first-order methods for smooth convex minimization," *Mathematical Programming*, vol. 159, no. 1, 81–107, Sep. 2016.

[10] ——,*On the convergence analysis of the optimized gradient methods*, arxiv 1510.08573, 2015.

[11] ——,"On the convergence analysis of the optimized gradient methods," *J. Optim. Theory Appl.*, 2016, Submitted.

[12] Y. Drori, *The exact information-based complexity of smooth convex minimization*, arxiv 1606.01424, 2016.

Bibliography II

[13] D. Böhning and B. G. Lindsay, "Monotonicity of quadratic approximation algorithms," *Ann. Inst. Stat. Math.*, vol. 40, no. 4, 641–63, Dec. 1988.

[14] B. O'Donoghue and E. Candès, "Adaptive restart for accelerated gradient schemes," *Found. Comp. Math.*, vol. 15, no. 3, 715–32, Jun. 2015.

[15] H. Erdoğan and J. A. Fessler, "Ordered subsets algorithms for transmission tomography," *Phys. Med. Biol.*, vol. 44, no. 11, 2835–51, Nov. 1999.

[16] D. Kim, S. Ramani, and J. A. Fessler, "Combining ordered subsets and momentum for accelerated X-ray CT image reconstruction," *IEEE Trans. Med. Imag.*, vol. 34, no. 1, 167–78, Jan. 2015.

[17] Y. Nesterov, *How to make the gradients small*, Optima 88, 2012.

[18] A. Beck and M. Teboulle, "A fast dual proximal gradient algorithm for convex minimization and applications," *Operations Research Letters*, vol. 42, no. 1, 1–6, Jan. 2014.

[19] I. Necoara and A. Patrascu, "Iteration complexity analysis of dual first order methods for conic convex programming," *Optimization Methods and Software*, vol. 31, no. 3, 645–78, 2016.

[20] D. Kim and J. A. Fessler, *Generalizing the optimized gradient method for smooth convex minimization*, arxiv 1607.06764, 2016.

[21] ——,"Generalizing the optimized gradient method for smooth convex minimization," *Mathematical Programming*, 2016, Submitted.

[22] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. Im. Proc.*, vol. 18, no. 11, 2419–34, Nov. 2009.

[23] D. Kim and J. A. Fessler, *Another look at the "Fast iterative shrinkage/Thresholding algorithm (FISTA)*, arxiv 1608.03861, 2016.

[24] ——,"Another look at the "Fast iterative shrinkage/Thresholding algorithm (FISTA)"," *SIAM J. Optim.*, 2016, Submitted.