ECE 8201: Low-dimensional Signal Models for High-dimensional Data Analysis

Lecture 5: FISTA

Yuejie Chi
The Ohio State University

THE OHIO STATE UNIVERSITY

Reference

- Beck, A., & Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM journal on imaging sciences, 2(1), 183-202.

See also:

- Nesterov, Y. (2007). Gradient methods for minimizing composite objective function.

- Lecture notes by L. Vandenberghe. `http://www.seas.ucla.edu/~vandenbe/236C/lectures/fgrad.pdf`.

How to solve composite optimization problems?

General composite optimization problem:

$$\text{(COP)}: \qquad \hat{\boldsymbol{x}} = \operatorname*{argmin}_{\boldsymbol{x}} \{F(\boldsymbol{x}) = f(\boldsymbol{x}) + g(\boldsymbol{x})\}$$

- $f(\boldsymbol{x})$ is convex and differentiable,
- $g(\boldsymbol{x})$ is convex, possibly non-differentiable

Examples:

- LASSO: $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2$, and $g(\boldsymbol{x}) = \lambda\|\boldsymbol{x}\|_1$. (focus of this lecture)
- Nuclear norm minimization (later for matrix completion):

$$f(\boldsymbol{X}) = \|\mathcal{P}_\Omega(\boldsymbol{Y} - \boldsymbol{X})\|_{\mathsf{F}}^2, \qquad g(\boldsymbol{X}) = \lambda\|\boldsymbol{X}\|_*$$

where $\|\boldsymbol{X}\|_* = \sum_{i=1}^{\min(m,n)} \sigma_i(\boldsymbol{X})$, the sum of the singular values of $\boldsymbol{X} \in \mathbb{R}^{m\times n}$.

Motivation

Standard methods (e.g. subgradient methods) for solving COP has very slow convergence rate (need $O(1/\epsilon^2)$ iterations to reach ϵ accuracy).

We would discuss an algorithm called FISTA that

- is iterative, and has low computational cost (first-order algorithm, which requires computation of a single gradient per iteration);
- has quadratic convergence rate;
- performs well in practice and works for a large class of problems.

FISTA stands for Fast Iterative Shrinkage-Thresholding Algorithm.

Gradient descent

Consider the unconstrained minimization of a continuously differentiable function $f(\boldsymbol{x})$ as

$$\hat{\boldsymbol{x}} = \operatorname*{argmin}_{\boldsymbol{x}} f(\boldsymbol{x})$$

using gradient descent: start with an initialization $\boldsymbol{x}_0 \in \mathbb{R}^n$, and iterate

$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} - t_k \nabla f(\boldsymbol{x}_{k-1})$$

where t_k is a suitable step-size at step k.

Key observation: we can view the gradient descent step as solving a *proximal regularization* of the linearized function f at $\boldsymbol{x}_{k-1}$,

$$\boldsymbol{x}_k = \operatorname*{argmin}_{\boldsymbol{x}} \left\{ f(\boldsymbol{x}_{k-1}) + \langle \boldsymbol{x} - \boldsymbol{x}_{k-1}, \nabla f(\boldsymbol{x}_{k-1}) \rangle + \frac{1}{2t_k} \|\boldsymbol{x} - \boldsymbol{x}_{k-1}\|_2^2 \right\}.$$

Generalized gradient descent

In the COP,

$$\hat{\boldsymbol{x}} = \operatorname*{argmin}_{\boldsymbol{x}} f(\boldsymbol{x}) + g(\boldsymbol{x})$$

we would like to generalize the proximal regularization idea, by extending the update rule as

$$\boldsymbol{x}_k = \operatorname*{argmin}_{\boldsymbol{x}} \left\{ f(\boldsymbol{x}_{k-1}) + \langle \boldsymbol{x} - \boldsymbol{x}_{k-1}, \nabla f(\boldsymbol{x}_{k-1}) \rangle + \frac{1}{2t_k} \|\boldsymbol{x} - \boldsymbol{x}_{k-1}\|_2^2 + g(\boldsymbol{x}) \right\}.$$

This can be simplified (by ignoring constant terms) as

$$\boldsymbol{x}_k = \operatorname*{argmin}_{\boldsymbol{x}} \left\{ \frac{1}{2t_k} \|\boldsymbol{x} - (\boldsymbol{x}_{k-1} - t_k \nabla f(\boldsymbol{x}_{k-1}))\|_2^2 + g(\boldsymbol{x}) \right\} \quad (*)$$

Proximal mapping

Definition 1. *The proximal mapping (operator) of a convex function $g(\boldsymbol{x})$ is written as*

$$\operatorname{prox}_g(\boldsymbol{x}) = \underset{\boldsymbol{u}}{\operatorname{argmin}} \left\{ \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{x}\|_2^2 + g(\boldsymbol{u}) \right\}.$$

- $g(\boldsymbol{x}) = 0$: $\operatorname{prox}_g(\boldsymbol{x}) = \boldsymbol{x}$.

- $g(\boldsymbol{x}) = I_C(\boldsymbol{x})$ is an indicator function of a convex set C, then

$$\operatorname{prox}_g(\boldsymbol{x}) = \underset{\boldsymbol{u}\in C}{\operatorname{argmin}} \|\boldsymbol{u} - \boldsymbol{x}\|_2^2$$

- $g(\boldsymbol{x}) = \lambda\|\boldsymbol{x}\|_1$: $\operatorname{prox}_g(\boldsymbol{x})$ is the shrinkage (soft-thresholding) operator and can be decomposed entry-wise:

$$\operatorname{prox}_g(x_i) := \mathcal{T}_\lambda(x_i) = \begin{cases} x_i - \lambda, & x_i \geq \lambda \\ 0, & |x_i| < \lambda \\ x_i + \lambda, & x_i \leq -\lambda \end{cases}$$

Generalized gradient descent and ISTA

- The generalized gradient descent (*) can be regarded as a proximal mapping:

$$\begin{aligned}\boldsymbol{x}_k &= \operatorname*{argmin}_{\boldsymbol{x}} \left\{ \frac{1}{2t_k} \|\boldsymbol{x} - (\boldsymbol{x}_{k-1} - t_k \nabla f(\boldsymbol{x}_{k-1}))\|_2^2 + g(\boldsymbol{x}) \right\} \\ &= \mathrm{prox}_{t_k g}(\boldsymbol{x}_{k-1} - t_k \nabla f(\boldsymbol{x}_{k-1}))\end{aligned}$$

- When $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2$, and $g(\boldsymbol{x}) = \lambda\|\boldsymbol{x}\|_1$, this gives the update rule for ISTA (Iterative Shrinkage-Thresholding Algorithm), or proximal gradient descent:

$$\begin{aligned}\boldsymbol{x}_k &= \mathrm{prox}_{\lambda t_k \|\boldsymbol{x}\|_1}(\boldsymbol{x}_{k-1} - t_k \nabla f(\boldsymbol{x}_{k-1})) \\ &= \mathrm{prox}_{\lambda t_k \|\boldsymbol{x}\|_1}(\boldsymbol{x}_{k-1} - t_k \nabla f(\boldsymbol{x}_{k-1})) \\ &= \mathcal{T}_{\lambda t_k}(\boldsymbol{x}_{k-1} - t_k \nabla f(\boldsymbol{x}_{k-1}))\end{aligned}$$

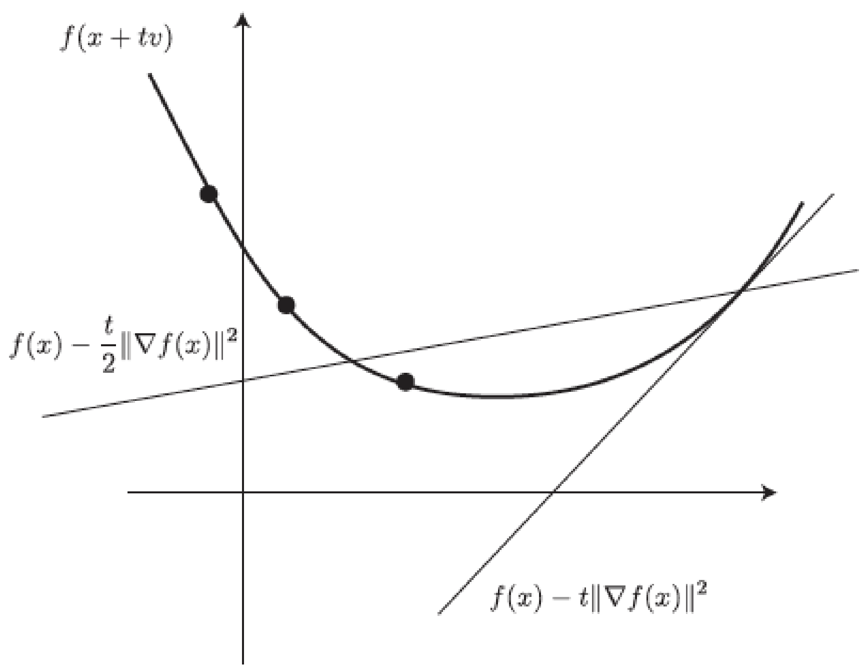
where $\nabla f(\boldsymbol{x}_{k-1}) = \boldsymbol{A}^\mathsf{T}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y})$. This can be efficiently computed.

Choice of step size

- Constant step-size: $t_k = t$
- Backtracking line search: start with t_0 and do $t = \beta t$ until

$$f(\boldsymbol{x} - t\nabla f(\boldsymbol{x})) \leq f(\boldsymbol{x}) - \alpha t \|\nabla f(\boldsymbol{x})\|_2^2$$

with $0 < \alpha, \beta < 1$, e.g. $\alpha = 1/2$.

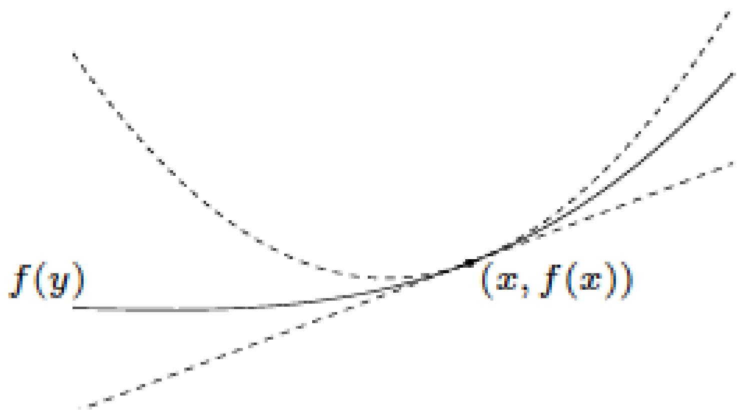


Assumptions

- $g : \mathbb{R}^n \mapsto \mathbb{R}$ is a continuous convex function, possibly nonsmooth;
- $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a smooth convex function that is continuously differentiable with *Lipschitz constant*:

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq L_f \|\boldsymbol{x} - \boldsymbol{y}\|, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n.$$

  Example: For LASSO problems, we have $L_f = \sigma_{\max}(\boldsymbol{A}^\top \boldsymbol{A})$.



- The optimal value of $F = f + g$ is $F^\star$ with optimal solution $\boldsymbol{x}^\star$.

Convergence of ISTA

Theorem 1. [Convergence for generalized gradient descent] *Fix step size* $t_k = t \leq 1/L$,

$$F(\boldsymbol{x}_k) - F^* \leq \frac{\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2}{2tk}$$

Similar results hold with backtracking for step size.

- Similar to the convergence of gradient descent

- The best possible is $O(1/k^2)$ for first-order methods – can we achieve it?

The answer is yes, with minimal additional computational cost.

Accelerated Gradient Descent

ISTA reaches an accuracy within $O(1/k)$ after k steps; this is not optimal (which is $O(1/k^2)$. The methods of Nesterov meet the optimal bound with the same computational cost (one gradient computation per iteration).

- We will first examine Nesterov's acceleration method (1983) for smooth convex functions;
- We then extend it to optimizing composite functions, using FISTA (Beck and Teboulle, 2009), which extends Nesterov's method.

Nesterov's ACG for convex smooth function

Consider minimizing a convex smooth function $f(\boldsymbol{x})$ with Lipschitz constant L:

$$\hat{\boldsymbol{x}} = \operatorname*{argmin}_{\boldsymbol{x}} f(\boldsymbol{x})$$

Nesterov's Accelerated Gradient Descent performs attains a rate of $O(1/k^2)$. It proceeds as below:

- Start with an initialization $\boldsymbol{x}_0 = \boldsymbol{x}_{-1}$, $\theta_0 = 0$;
- for $k = 1, 2, \ldots,$

$$\theta_k = \frac{1 + \sqrt{1 + 4\theta_{k-1}^2}}{2},$$

$$\boldsymbol{y}_k = \boldsymbol{x}_{k-1} + \left(\frac{\theta_{k-1} - 1}{\theta_k} \right) (\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-2})$$

$$\boldsymbol{x}_k = \boldsymbol{y}_k - t_k \nabla f(\boldsymbol{y}_k)$$

Remark: other choice of the momentum term with $\theta_k = \frac{k+1}{2}$:

$$\boldsymbol{y}_k = \boldsymbol{x}_{k-1} + \frac{k-2}{k+1}(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-2})$$

Theorem 2. [Nesterov 1983] *The Nesterov's AGD satisfies*

$$f(\boldsymbol{y}_k) - f(\boldsymbol{x}^\star) \leq \frac{2\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2}{Lk^2}$$

Achieves the optimal rate!

FISTA

The FISTA algorithm with step size t_k (e.g. $t_k = \frac{1}{L}$, where L_f is the Lipschitz constant of f):

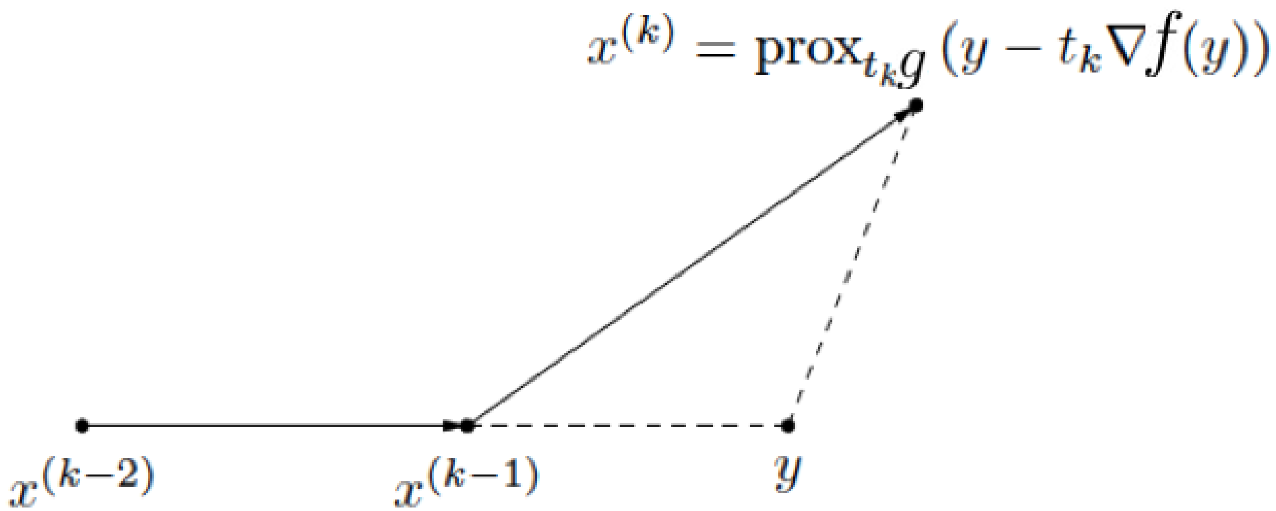
- Initialization: $\boldsymbol{x}_0 = \boldsymbol{x}_{-1} \in \mathbb{R}^n$, $\theta_0 = 1$,
- For $k = 1, 2, \ldots,$

$$
\begin{aligned}
\theta_k &= \frac{1 + \sqrt{1 + 4\theta_{k-1}^2}}{2} \\
\boldsymbol{y}_k &= \boldsymbol{x}_{k-1} + \left(\frac{\theta_{k-1} - 1}{\theta_k} \right) (\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-2}) \\
\boldsymbol{x}_k &= \operatorname{prox}_{t_k g} \left(\boldsymbol{y}_k - t_k \nabla f(\boldsymbol{y}_k) \right)
\end{aligned}
$$

FISTA is computationally efficient when the proximal operator can be computed efficiently (e.g. LASSO).

Interpretation

- first iteration is a proximal gradient step at $\boldsymbol{y}_1 = \boldsymbol{x}_0$
- next iterations are proximal gradient steps at extrapolated points $\boldsymbol{y}_k$, $k \geq 2$, with the linear combinations carefully chosen.



Case Study: LASSO

For LASSO: set $\boldsymbol{y}_1 = \boldsymbol{x}_0 \in \mathbb{R}^n$, $\theta_1 = 0$, and $t_k = 1/\sigma_{\max}(\boldsymbol{A}^\top \boldsymbol{A})$ (constant step-size), iterate

$$\begin{aligned}
\theta_k &= \frac{1 + \sqrt{1 + 4\theta_{k-1}^2}}{2} \\
\boldsymbol{y}_k &= \boldsymbol{x}_{k-1} + \left(\frac{\theta_{k-1} - 1}{\theta_k}\right)(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-2}) \\
\boldsymbol{x}_k &= \mathcal{T}_{\lambda t_k}\left(\boldsymbol{y}_k - t_k \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{y})\right)
\end{aligned}$$

The main computation cost to apply $\boldsymbol{A}$ and $\boldsymbol{A}^\top$; no matrix inversion is needed.

Convergence of FISTA

Theorem 3.

$$F(\boldsymbol{x}_k) - F(\boldsymbol{x}^\star) \leq \frac{2L\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2}{(k+1)^2} \sim O\left(\frac{1}{k^2}\right)$$

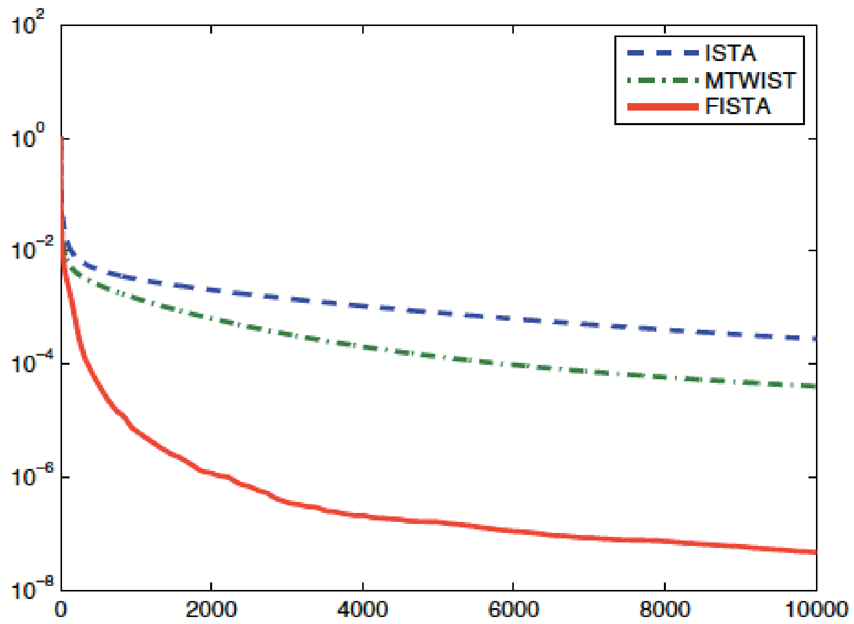


Figure 5. *Comparison of function value errors $F(\mathbf{x}_k) - F(\mathbf{x}^*)$ of ISTA, MTWIST, and FISTA.*

Proof of Theorem 3

- Introduce another sequence $\boldsymbol{v}_k$, which satisfies

$$\boldsymbol{v}_k := \boldsymbol{x}_{k-1} + \theta_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$$
$$\boldsymbol{y}_k = \frac{1}{\theta_k}\boldsymbol{v}_{k-1} + \left(1 - \frac{1}{\theta_k}\right)\boldsymbol{x}_{k-1}$$

- Two useful facts:

1. $\boldsymbol{v}_k = \boldsymbol{v}_{k-1} + \theta_k(\boldsymbol{x}_k - \boldsymbol{y}_k)$
2. $\left(1 - \frac{1}{\theta_k}\right)\theta_k^2 = \theta_{k-1}^2$

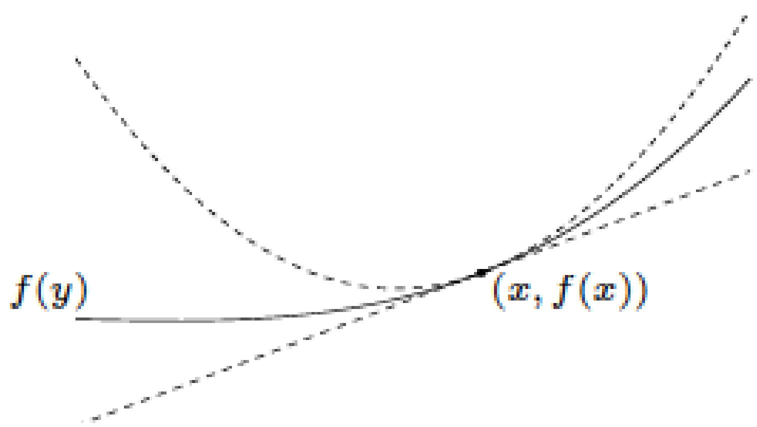
Important inequalities

Upper bound of f from Lipschitz property:

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq L_f \|\boldsymbol{x} - \boldsymbol{y}\|, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n.$$

we have

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^{\mathsf{T}}(\boldsymbol{y} - \boldsymbol{x}) + \frac{L_f}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2, \quad \forall \boldsymbol{x}, \boldsymbol{y}$$



Important inequalities

Upper bound of g from definition of proximal operator:

$$g(\boldsymbol{y}) \leq g(\boldsymbol{z}) + \frac{1}{t}(\boldsymbol{w} - \boldsymbol{y})^{\mathsf{T}}(\boldsymbol{y} - \boldsymbol{z}), \quad \forall \boldsymbol{w}, \boldsymbol{z}, \boldsymbol{y} = \mathsf{prox}_{tg}(\boldsymbol{w})$$

Proof: since $\boldsymbol{y} = \mathsf{prox}_{tg}(\boldsymbol{w})$ minimizes $tg(\boldsymbol{u}) + \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{u}\|_2^2$ by definition, we have

$$0 \in t\partial g(\boldsymbol{y}) + (\boldsymbol{y} - \boldsymbol{w})$$

i.e.

$$\frac{1}{t}(\boldsymbol{w} - \boldsymbol{y}) \in \partial g(\boldsymbol{y}),$$

By the definition of subgradient we have $\forall \boldsymbol{z}$,

$$g(\boldsymbol{z}) \geq g(\boldsymbol{y}) + \frac{1}{t}(\boldsymbol{w} - \boldsymbol{y})^{\mathsf{T}}(\boldsymbol{z} - \boldsymbol{y})$$

Progress in one iteration

Define $\boldsymbol{x}^+ = \boldsymbol{x}_k$, $\boldsymbol{x} = \boldsymbol{x}_{k-1}$, $\boldsymbol{y} = \boldsymbol{y}_k$, $\theta = \theta_k$, $\boldsymbol{v} = \boldsymbol{v}_{k-1}$, $\boldsymbol{v}^+ = \boldsymbol{v}_k$,

- upper bound from Lipschitz property: if $0 < t \le 1/L$,

$$f(\boldsymbol{x}^+) \le f(\boldsymbol{y}) + \nabla f(\boldsymbol{y})^\mathsf{T}(\boldsymbol{x}^+ - \boldsymbol{y}) + \frac{1}{2t}\|\boldsymbol{y} - \boldsymbol{x}^+\|_2^2$$

- upper bound from the definition of prox-operator $\left(\boldsymbol{x}^+ = \mathrm{prox}_{tg}(\boldsymbol{y} - t\nabla f(\boldsymbol{y}))\right)$:

$$g(\boldsymbol{x}^+) \le g(\boldsymbol{z}) + \nabla f(\boldsymbol{y})^\mathsf{T}(\boldsymbol{z} - \boldsymbol{x}^+) + \frac{1}{t}(\boldsymbol{x}^+ - \boldsymbol{y})^\mathsf{T}(\boldsymbol{z} - \boldsymbol{x}^+), \quad \forall \boldsymbol{z}$$

- add the upper bounds and use convexity of f:

$$F(\boldsymbol{x}^+) \le F(\boldsymbol{z}) + \frac{1}{t}(\boldsymbol{x}^+ - \boldsymbol{y})^\mathsf{T}(\boldsymbol{z} - \boldsymbol{x}^+) + \frac{1}{2t}\|\boldsymbol{y} - \boldsymbol{x}^+\|_2^2, \quad \forall \boldsymbol{z}$$

- make convex combination of upper bounds for $\boldsymbol{z} = \boldsymbol{x}$ and $\boldsymbol{z} = \boldsymbol{x}^\star$:

$$
\begin{aligned}
F(\boldsymbol{x}^+) - F^\star - \left(1 - \frac{1}{\theta}\right)(F(\boldsymbol{x}) - F^\star) &= F(\boldsymbol{x}^+) - \frac{1}{\theta}F^\star - \left(1 - \frac{1}{\theta}\right)F(\boldsymbol{x}) \\
&\le \frac{1}{t}(\boldsymbol{x}^+ - \boldsymbol{y})^\mathsf{T}\left(\frac{1}{\theta}\boldsymbol{x}^\star + (1 - \frac{1}{\theta})\boldsymbol{x} - \boldsymbol{x}^+\right) + \frac{1}{2t}\|\boldsymbol{y} - \boldsymbol{x}^+\|_2^2 \\
&= \frac{1}{2t}\left(\|\boldsymbol{y} - \frac{1}{\theta}\boldsymbol{x}^\star + (1 - \frac{1}{\theta})\boldsymbol{x}\|_2^2 - \|\boldsymbol{x}^+ - \frac{1}{\theta}\boldsymbol{x}^\star + (1 - \frac{1}{\theta})\boldsymbol{x}\|_2^2\right) \\
&= \frac{1}{2\theta^2 t}\left(\|\boldsymbol{v} - \boldsymbol{x}^\star\|_2^2 - \|\boldsymbol{v}^+ - \boldsymbol{x}^\star\|_2^2\right)
\end{aligned}
$$

We now have, at the kth iteration:

$$
\begin{aligned}
\theta_k^2 t\left(F(\boldsymbol{x}_k) - F^\star\right) + \frac{1}{2}\|\boldsymbol{v}_k - \boldsymbol{x}^\star\|_2^2 &\le \left(\theta_k^2 - \theta_k\right)t(F(\boldsymbol{x}_{k-1}) - F^\star) + \frac{1}{2}\|\boldsymbol{v}_{k-1} - \boldsymbol{x}^\star\|_2^2 \\
&= \theta_{k-1}^2 t(F(\boldsymbol{x}_{k-1}) - F^\star) + \frac{1}{2}\|\boldsymbol{v}_{k-1} - \boldsymbol{x}^\star\|_2^2
\end{aligned}
$$

Applying the above relationship recursively, we obtain

$$\theta_k^2 t\left(F(\boldsymbol{x}_k) - F^\star\right) + \frac{1}{2}\|\boldsymbol{v}_k - \boldsymbol{x}^\star\|_2^2 \le \theta_0^2 t(F(\boldsymbol{x}_0) - F^\star) + \frac{1}{2}\|\boldsymbol{v}_0 - \boldsymbol{x}^\star\|_2^2$$
$$= \frac{1}{2}\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2$$

therefore, plug in $t = \frac{1}{L}$,

$$F(\boldsymbol{x}_k) - F^\star \le \frac{1}{2\theta_k^2 t}\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2 \le \frac{2L}{(k+1)^2}\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2.$$

Alternative formulation

Alternative formulation:

- Initialization: $\boldsymbol{y}_1 = \boldsymbol{x}_0 \in \mathbb{R}^n$, and L_f is the Lipschitz constant;
- Fix step size $t_k = \frac{1}{L}$.
- For $k = 1, 2, \ldots,$

$$\boldsymbol{x}_k = \mathrm{prox}_{t_k g}\left(\boldsymbol{y}_k - t_k \nabla f(\boldsymbol{y}_k)\right)$$

$$\boldsymbol{y}_{k+1} = \boldsymbol{x}_k + \left(\frac{k-2}{k+1}\right)(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$$

Convergence speed $O(1/k^2)$ in k steps.

Computational-Statistical Trade-off

If there is indeed a ground truth $\boldsymbol{x}^\star$ and we wish $\hat{\boldsymbol{x}}$ is close to $\boldsymbol{x}^\star$; we have a sequence of $\{\boldsymbol{x}_k\}$ and hope $\boldsymbol{x}_k$ converges to $\hat{\boldsymbol{x}}$. At a fixed k, we may bound

$$\|\boldsymbol{x}_k - \boldsymbol{x}^\star\|_2 \le \underbrace{\|\boldsymbol{x}_k - \hat{\boldsymbol{x}}\|_2}_{\text{computational error}} + \underbrace{\|\hat{\boldsymbol{x}} - \boldsymbol{x}^\star\|_2}_{\text{statistical error}}$$

Active research in studying the computational-statistical trade-offs in statistical estimation.