Mathematics of Data: From Theory to Computation

Prof. Volkan Cevher
volkan.cevher@epfl.ch

Lecture 7: Introduction to proximal-operators. Conditional gradient methods.

Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)

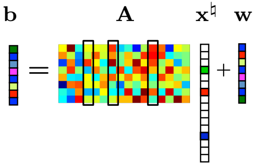
EE-556 (Fall 2023)

License Information for Mathematics of Data Slides

Outline

- Composite minimization
- Proximal gradient methods
- Introduction to Frank-Wolfe method

Recall sparse regression in generalized linear models (GLMs)



Problem (Sparse regression in GLM)

Our goal is to estimate $\mathbf{x}^\natural \in \mathbb{R}^p$ *given* $\{b_i\}_{i=1}^n$ *and* $\{\mathbf{a}_i\}_{i=1}^n$*, knowing that the likelihood function at* y_i *given* $\mathbf{a}_i$ *and* $\mathbf{x}^\natural$ *is given by* $L(\langle \mathbf{a}_i, \mathbf{x}^\natural \rangle, b_i)$*, and that* $\mathbf{x}^\natural$ *is sparse.*

Optimization formulation

$$\min_{\mathbf{x}\in\mathbb{R}^p} \Big\{ \underbrace{-\sum_{i=1}^n \log L(\langle \mathbf{a}_i, \mathbf{x}^\natural \rangle, b_i)}_{f(\mathbf{x})} + \underbrace{\rho_n \|\mathbf{x}\|_1}_{g(\mathbf{x})} \Big\}$$

where $\rho_n > 0$ is a parameter which controls the strength of sparsity regularization.

Theorem (cf. [13] for details)

Under some technical conditions, there exists $\{\rho_i\}_{i=1}^\infty$ *such that with high probability, the following holds*

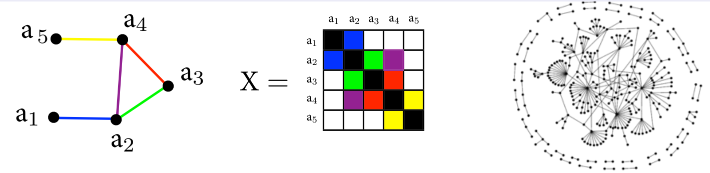
$$\| \mathbf{x}^\star - \mathbf{x}^\natural \|_2^2 = \mathcal{O}\left(\frac{s \log p}{n}\right), \quad \operatorname{supp} \mathbf{x}^\star = \operatorname{supp} \mathbf{x}^\natural.$$

Recall ML: $\| \mathbf{x}_{ML} - \mathbf{x}^\natural \|_2^2 = \mathcal{O}\left(\frac{p}{n}\right)$.

Sparse inverse covariance estimation

Problem (Graphical model selection)

Given a data set $\mathcal{D} := \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$, where $\mathbf{x}_i$ is a Gaussian random variable. Let Σ be the covariance matrix corresponding to the graphical model of the Gaussian Markov random field. Our goal is to learn a sparse precision matrix X (i.e., the inverse covariance matrix Σ^{-1}) that captures the Markov random field structure.



Optimization formulation [16]

$$\min_{X \succ 0} \Big\{ \underbrace{\operatorname{tr}(\Sigma X) - \log\det(X)}_{f(\mathbf{x})} + \underbrace{\rho_n \|\operatorname{vec}(X)\|_1}_{g(\mathbf{x})} \Big\}, \tag{1}$$

where $X \succ 0$ means that X is symmetric and positive definite and $\rho_n > 0$ is a regularization parameter and vec is the vectorization operator. Let $X^\star$ be the minimizer of (1), under some technical conditions, there exists a ρ_n such that $\| X^\star - \Sigma^{-1} \|_2^2 = \mathcal{O}(\min \frac{1}{n}\{d^2 \log p, (s+p)\log p\})$ where d is the maximum node degree.

Composite convex minimization

Problem (Composite convex minimization)

$$\boxed{F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\}} \tag{2}$$

- *f and g are both proper, closed, and convex.*
- $\mathrm{dom}(F) := \mathrm{dom}(f) \cap \mathrm{dom}(g) \neq \emptyset$ *and* $-\infty < F^\star < +\infty$.
- *The solution set* $\mathcal{S}^\star := \{\mathbf{x}^\star \in \mathrm{dom}(F) : F(\mathbf{x}^\star) = F^\star\}$ *is nonempty.*

Remarks:

◦ Without loss of generality, f is smooth and g is non-smooth in the sequel.

◦ By Moreau-Rockafellar Theorem, we have $\partial F = \partial(f+g) = \partial f + \partial g = \nabla f + \partial g$.

◦ Subgradient method attains a $\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$ rate.

◦ Without g, accelerated gradient method attains a $\mathcal{O}\left(\frac{1}{T^2}\right)$ rate.

Composite convex minimization

Problem (Composite convex minimization)

$$F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\} \tag{2}$$

- *f and g are both proper, closed, and convex.*
- $\mathrm{dom}(F) := \mathrm{dom}(f) \cap \mathrm{dom}(g) \neq \emptyset$ *and* $-\infty < F^\star < +\infty$.
- *The solution set* $\mathcal{S}^\star := \{\mathbf{x}^\star \in \mathrm{dom}(F) : F(\mathbf{x}^\star) = F^\star\}$ *is nonempty.*

Remarks:

- ◦ Without loss of generality, f is smooth and g is non-smooth in the sequel.
- ◦ By Moreau-Rockafellar Theorem, we have $\partial F = \partial(f+g) = \partial f + \partial g = \nabla f + \partial g$.
- ◦ Subgradient method attains a $\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$ rate.
- ◦ Without g, accelerated gradient method attains a $\mathcal{O}\left(\frac{1}{T^2}\right)$ rate.

Can we design algorithms that achieve a faster convergence rate for composite convex minimization?

Designing algorithms for finding a solution $\mathbf{x}^\star$

Quadratic *majorizer* for f

When f has L-Lipschitz continuous gradient, we have, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^p$

$$f(\mathbf{x}) \leq f(\mathbf{y}) + \nabla f(\mathbf{y})^T(\mathbf{x}-\mathbf{y}) + \frac{L}{2}\|\mathbf{x}-\mathbf{y}\|_2^2$$

Designing algorithms for finding a solution $\mathbf{x}^\star$

Quadratic *majorizer* for f

When f has L-Lipschitz continuous gradient, we have, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^p$

$$f(\mathbf{x}) \leq f(\mathbf{y}) + \nabla f(\mathbf{y})^T(\mathbf{x} - \mathbf{y}) + \frac{L}{2}\|\mathbf{x} - \mathbf{y}\|_2^2$$

Quadratic *majorizer* for $f + g$

When f has L-Lipschitz continuous gradient, we have, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^p$

$$f(\mathbf{x}) + g(\mathbf{x}) \leq f(\mathbf{y}) + \nabla f(\mathbf{y})^T(\mathbf{x} - \mathbf{y}) + \frac{L}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 + g(\mathbf{x}) := P_L(\mathbf{x}, \mathbf{y})$$

Designing algorithms for finding a solution $\mathbf{x}^\star$

Quadratic *majorizer* for f

When f has L-Lipschitz continuous gradient, we have, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^p$

$$f(\mathbf{x}) \leq f(\mathbf{y}) + \nabla f(\mathbf{y})^T(\mathbf{x} - \mathbf{y}) + \frac{L}{2}\|\mathbf{x} - \mathbf{y}\|_2^2$$

Quadratic *majorizer* for $f + g$

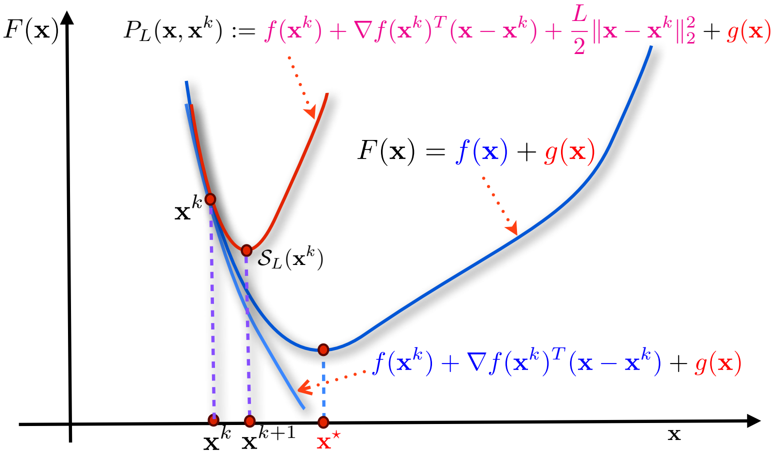
When f has L-Lipschitz continuous gradient, we have, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^p$

$$f(\mathbf{x}) + g(\mathbf{x}) \leq f(\mathbf{y}) + \nabla f(\mathbf{y})^T(\mathbf{x} - \mathbf{y}) + \frac{L}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 + g(\mathbf{x}) := P_L(\mathbf{x}, \mathbf{y})$$

Majorization-minimization for $f + g$

$$\begin{aligned} \mathbf{x}^{k+1} &= \arg\min_{\mathbf{x} \in \mathbb{R}^p} P_L(\mathbf{x}, \mathbf{x}^k) \\ &= \arg\min_{\mathbf{x} \in \mathbb{R}^p} \left\{ g(\mathbf{x}) + \frac{L}{2}\| \mathbf{x} - \left(\mathbf{x}^k - \frac{1}{L}\nabla f(\mathbf{x}^k)\right) \|^2 \right\} \end{aligned}$$

Geometric illustration



A short detour: Proximal-point operators

Definition (Proximal operator [18])

Let $g \in \mathcal{F}(\mathbb{R}^p)$, $\mathbf{x} \in \mathbb{R}^p$ and $\lambda > 0$. The proximal operator (or prox-operator) of g is defined as:

$$\boxed{\mathrm{prox}_{\lambda g}(\mathbf{y}) \equiv \arg\min_{\mathbf{x} \in \mathbb{R}^p} \left\{ g(\mathbf{x}) + \frac{1}{2\lambda}\|\mathbf{y} - \mathbf{x}\|_2^2 \right\}.} \quad (3)$$

A short detour: Proximal-point operators

Definition (Proximal operator [18])

Let $g \in \mathcal{F}(\mathbb{R}^p)$, $\mathbf{x} \in \mathbb{R}^p$ and $\lambda > 0$. The proximal operator (or prox-operator) of g is defined as:

$$\boxed{\operatorname{prox}_{\lambda g}(\mathbf{y}) \equiv \arg\min_{\mathbf{x} \in \mathbb{R}^p} \left\{ g(\mathbf{x}) + \frac{1}{2\lambda} \|\mathbf{y} - \mathbf{x}\|_2^2 \right\}.} \quad (3)$$

Remarks:

- The *proximal operator* of $\frac{1}{L} g$ evaluated at $\left(\mathbf{x}^k - \frac{1}{L}\nabla f(\mathbf{x}^k)\right)$ is given by

$$\operatorname{prox}_{\frac{1}{L} g}\left(\mathbf{x}^k - \frac{1}{L}\nabla f(\mathbf{x}^k)\right) = \arg\min_{\mathbf{x} \in \mathbb{R}^p} \left\{ g(\mathbf{x}) + \frac{L}{2} \| \mathbf{x} - \left(\mathbf{x}^k - \frac{1}{L}\nabla f(\mathbf{x}^k)\right) \|^2 \right\}.$$

- This prox-operator minimizes the majorizing bound:

$$f(\mathbf{x}) + g(\mathbf{x}) \leq f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T(\mathbf{x} - \mathbf{x}^k) + \frac{L}{2}\|\mathbf{x} - \mathbf{x}^k\|_2^2 + g(\mathbf{x})$$

- Rule of thumb: Replace gradient steps with proximal gradient steps!

Tractable prox-operators

Processing non-smooth terms in (2)

- We handle the nonsmooth term g in (2) using its proximal operator.
- However, computing proximal operator prox_g of a general convex function g

$$\mathrm{prox}_g(\mathbf{y}) \equiv \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ g(\mathbf{x}) + \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 \right\}.$$

  can be **computationally demanding**.

Definition (Tractable proximity)

- Given $g \in \mathcal{F}(\mathbb{R}^p)$. We say that g is proximally tractable if prox_g defined by (3) can be computed **efficiently**.
- "**efficiently**" $=$ {closed form solution, low-cost computation, polynomial time}.

Tractable prox-operators

Example

- For separable functions, the prox-operator can be efficient. When $g(\mathbf{x}) := \|\mathbf{x}\|_1 = \sum_{i=1}^{p} |\mathbf{x}_i|$, we have

$$\boxed{\mathrm{prox}_{\lambda g}(\mathbf{x}) = \mathrm{sign}(\mathbf{x}) \otimes \max\{|\mathbf{x}| - \lambda, 0\}.}$$

- Sometimes, we can compute the prox-operator via basic algebra. When $g(\mathbf{x}) := \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$, we have

$$\boxed{\mathrm{prox}_{\lambda g}(\mathbf{x}) = \left(\mathbb{I} + \lambda \mathbf{A}^T \mathbf{A}\right)^{-1} \left(\mathbf{x} + \lambda \mathbf{A}^T \mathbf{b}\right).}$$

- For the indicator functions of simple sets, e.g., $g(\mathbf{x}) := \delta_{\mathcal{X}}(\mathbf{x})$, the prox-operator is the projection operator

$$\boxed{\mathrm{prox}_{\lambda g}(\mathbf{x}) := \pi_{\mathcal{X}}(\mathbf{x}),}$$

  where $\pi_{\mathcal{X}}(\mathbf{x})$ denotes the projection of $\mathbf{x}$ onto $\mathcal{X}$. For instance, when $\mathcal{X} = \{\mathbf{x} : \|\mathbf{x}\|_1 \leq \lambda\}$, the projection can be obtained efficiently.

Computational efficiency - Example

Proximal operator of quadratic function

The **proximal operator** of a quadratic function $g(\mathbf{x}) := \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$ is defined as

$$\boxed{\mathrm{prox}_{\lambda g}(\mathbf{x}) := \arg\min_{\mathbf{y}\in\mathbb{R}^p} \left\{ \frac{1}{2}\|\mathbf{A}\mathbf{y} - \mathbf{b}\|_2^2 + \frac{1}{2\lambda}\|\mathbf{y} - \mathbf{x}\|_2^2 \right\}.} \tag{4}$$

How do we compute $\mathrm{prox}_{\lambda g}(\mathbf{x})$?

The derivation: ◦ The optimality condition implies that the solution of (4) should satisfy the following:

$$\mathbf{A}^T(\mathbf{A}\mathbf{y} - \mathbf{b}) + \lambda^{-1}(\mathbf{y} - \mathbf{x}) = 0.$$

◦ Setting $\mathbf{y} = \mathrm{prox}_{\lambda g}(\mathbf{x})$, we obtain

$$\mathrm{prox}_{\lambda g}(\mathbf{x}) = \left(\mathbb{I} + \lambda\mathbf{A}^T\mathbf{A}\right)^{-1}\left(\mathbf{x} + \lambda\mathbf{A}^T\mathbf{b}\right)$$

Remarks:

◦ The Woodbury matrix identity can be useful: $(\mathbb{I} + \lambda\mathbf{A}^T\mathbf{A})^{-1} = \mathbb{I} - \mathbf{A}^T(\lambda^{-1}\mathbb{I} + \mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A}$.

◦ When $\mathbf{A}^T\mathbf{A}$ is efficiently diagonalizable, i.e., $\mathbf{A}^T\mathbf{A} := \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$, such that

- $\mathbf{U}$ is a unitary matrix, i.e., $\mathbf{U}\mathbf{U}^T = \mathbf{U}^T\mathbf{U} = \mathbb{I}$, and $\boldsymbol{\Lambda}$ is a diagonal matrix.
- $\mathrm{prox}_{\lambda g}(\mathbf{x}) = \mathbf{U}\left(\mathbb{I} + \lambda\boldsymbol{\Lambda}\right)^{-1}\mathbf{U}^T\left(\mathbf{x} + \lambda\mathbf{A}\mathbf{b}\right)$.

A non-exhaustive list of proximal tractability functions

Name	Function	Proximal operator	Complexity
ℓ_1-norm	$f(\mathbf{x}) := \|\mathbf{x}\|_1$	$\mathrm{prox}_{\lambda f}(\mathbf{x}) = \mathrm{sign}(\mathbf{x}) \otimes [\lvert\mathbf{x}\rvert - \lambda]_+$	$\mathcal{O}(p)$
ℓ_2-norm	$f(\mathbf{x}) := \|\mathbf{x}\|_2$	$\mathrm{prox}_{\lambda f}(\mathbf{x}) = [1 - \frac{\lambda}{\|\mathbf{x}\|_2}]_+ \mathbf{x}$	$\mathcal{O}(p)$
Support function	$f(\mathbf{x}) := \max_{\mathbf{y}\in\mathcal{C}} \mathbf{x}^T\mathbf{y}$	$\mathrm{prox}_{\lambda f}(\mathbf{x}) = \mathbf{x} - \lambda\pi_{\mathcal{C}}(\mathbf{x})$	
Box indicator	$f(\mathbf{x}) := \delta_{[\mathbf{a},\mathbf{b}]}(\mathbf{x})$	$\mathrm{prox}_{\lambda f}(\mathbf{x}) = \pi_{[\mathbf{a},\mathbf{b}]}(\mathbf{x})$	$\mathcal{O}(p)$
Positive semidefinite cone indicator	$f(\mathbf{X}) := \delta_{\mathbb{S}^p_+}(\mathbf{X})$	$\mathrm{prox}_{\lambda f}(\mathbf{X}) = \mathbf{U}[\Sigma]_+\mathbf{U}^T$, where $\mathbf{X} = \mathbf{U}\Sigma\mathbf{U}^T$	$\mathcal{O}(p^3)$
Hyperplane indicator	$f(\mathbf{x}) := \delta_{\mathcal{X}}(\mathbf{x})$, $\mathcal{X} := \{\mathbf{x} : \mathbf{a}^T\mathbf{x} = b\}$	$\mathrm{prox}_{\lambda f}(\mathbf{x}) = \pi_{\mathcal{X}}(\mathbf{x}) = \mathbf{x} + \left(\frac{b - \mathbf{a}^T\mathbf{x}}{\|\mathbf{a}\|_2}\right)\mathbf{a}$	$\mathcal{O}(p)$
Simplex indicator	$f(\mathbf{x}) = \delta_{\mathcal{X}}(\mathbf{x})$, $\mathcal{X} := \{\mathbf{x} : \mathbf{x} \geq 0,\ \mathbf{1}^T\mathbf{x} = 1\}$	$\mathrm{prox}_{\lambda f}(\mathbf{x}) = (\mathbf{x} - \nu\mathbf{1})$ for some $\nu \in \mathbb{R}$, which can be efficiently calculated	$\tilde{\mathcal{O}}(p)$
Convex quadratic	$f(\mathbf{x}) := \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{q}^T\mathbf{x}$	$\mathrm{prox}_{\lambda f}(\mathbf{x}) = (\lambda\mathbb{I} + \mathbf{Q})^{-1}\mathbf{x}$	$\mathcal{O}(p\log p) \to \mathcal{O}(p^3)$
Square ℓ_2-norm	$f(\mathbf{x}) := \frac{1}{2}\|\mathbf{x}\|_2^2$	$\mathrm{prox}_{\lambda f}(\mathbf{x}) = \frac{1}{1+\lambda}\mathbf{x}$	$\mathcal{O}(p)$
log-function	$f(\mathbf{x}) := -\log(x)$	$\mathrm{prox}_{\lambda f}(x) = \frac{1}{2}(\sqrt{x^2 + 4\lambda} + x)$	$\mathcal{O}(1)$
$\log\det$-function	$f(\mathbf{x}) := -\log\det(\mathbf{X})$	$\mathrm{prox}_{\lambda f}(\mathbf{X})$ is the log-function prox applied to the individual eigenvalues of $\mathbf{X}$	$\mathcal{O}(p^3)$

Remarks:

- Here: $[\mathbf{x}]_+ := \max\{0, \mathbf{x}\}$ and $\delta_{\mathcal{X}}$ is the indicator function of the convex set $\mathcal{X}$.
- sign is the sign function, $\mathbb{S}^p_+$ is the cone of symmetric positive semidefinite matrices.
- For more functions, see [5, 15].

Solution methods

Composite convex minimization

$$F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \Big\{ F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x}) \Big\}. \tag{5}$$

Choice of numerical solution methods

- **Solve** (5) $=$ Find $\mathbf{x}^k \in \mathbb{R}^p$ such that

$$\boxed{F(\mathbf{x}^k) - F^\star \leq \varepsilon}$$

for a given tolerance $\varepsilon > 0$.

- **Oracles**: We can use one of the following configurations (oracles):
  1. $\partial f(\cdot)$ and $\partial g(\cdot)$ at any point $\mathbf{x} \in \mathbb{R}^p$.
  2. $\nabla f(\cdot)$ and $\mathrm{prox}_{\lambda g}(\cdot)$ at any point $\mathbf{x} \in \mathbb{R}^p$.
  3. $\mathrm{prox}_{\lambda f}$ and $\mathrm{prox}_{\lambda g}(\cdot)$ at any point $\mathbf{x} \in \mathbb{R}^p$.
  4. $\nabla f(\cdot)$, inverse of $\nabla^2 f(\cdot)$ and $\mathrm{prox}_{\lambda g}(\cdot)$ at any point $\mathbf{x} \in \mathbb{R}^p$.

Remark: Using different oracle leads to different types of algorithms.

Proximal-gradient algorithm

Basic proximal-gradient scheme (ISTA)
1. Choose $\mathbf{x}^0 \in \mathrm{dom}\,(F)$ arbitrarily as a starting point. 2. For $k = 0, 1, \cdots$, generate a sequence $\{\mathbf{x}^k\}_{k \geq 0}$ as: $$\mathbf{x}^{k+1} := \mathrm{prox}_{\alpha g}\left(\mathbf{x}^k - \alpha \nabla f(\mathbf{x}^k)\right),$$ where $\alpha := \frac{1}{L}$.

Proximal-gradient algorithm

Basic proximal-gradient scheme (ISTA)

1. Choose $\mathbf{x}^0 \in \mathrm{dom}\,(F)$ arbitrarily as a starting point.
2. For $k = 0, 1, \cdots$, generate a sequence $\{\mathbf{x}^k\}_{k\geq 0}$ as:

$$\mathbf{x}^{k+1} := \mathrm{prox}_{\alpha g}\left(\mathbf{x}^k - \alpha \nabla f(\mathbf{x}^k)\right),$$

where $\alpha := \frac{1}{L}$.

Theorem (Convergence of ISTA [2])

Let $\{\mathbf{x}^k\}$ be generated by ISTA. Then:

$$F(\mathbf{x}^k) - F^\star \leq \frac{L_f \|\mathbf{x}^0 - \mathbf{x}^\star\|_2^2}{2(k+1)}$$

The worst-case complexity to reach $F(\mathbf{x}^k) - F^\star \leq \varepsilon$ of (ISTA) *is $\mathcal{O}\left(\frac{L_f R_0^2}{\varepsilon}\right)$, where $R_0 := \max_{\mathbf{x}^\star \in \mathcal{S}^\star} \|\mathbf{x}^0 - \mathbf{x}^\star\|_2$.*

- **Oracles:** $\mathrm{prox}_{\alpha g}(\cdot)$ and $\nabla f(\cdot)$.
- Compared to the subgradient gradient method, the rate improves at the cost of prox-computation.

Fast proximal-gradient algorithm

Fast proximal-gradient scheme (FISTA)

1. Choose $\mathbf{x}^0 \in \mathrm{dom}\,(F)$ arbitrarily as a starting point.
2. Set $\mathbf{y}^0 := \mathbf{x}^0$ and $t_0 := 1$, $\alpha := L^{-1}$.
3. For $k = 0, 1, \ldots$, generate two sequences $\{\mathbf{x}^k\}_{k\geq 0}$ and $\{\mathbf{y}^k\}_{k\geq 0}$ as:

$$\begin{cases} \mathbf{x}^{k+1} & := \mathrm{prox}_{\alpha g}\left(\mathbf{y}^k - \alpha\nabla f(\mathbf{y}^k)\right), \\ t_{k+1} & := \frac{1}{2}(1 + \sqrt{4t_k^2 + 1}), \\ \mathbf{y}^{k+1} & := \mathbf{x}^{k+1} + \frac{t_k - 1}{t_{k+1}}(\mathbf{x}^{k+1} - \mathbf{x}^k). \end{cases}$$

Fast proximal-gradient algorithm

Fast proximal-gradient scheme (FISTA)
1. Choose $\mathbf{x}^0 \in \mathrm{dom}\,(F)$ arbitrarily as a starting point. **2.** Set $\mathbf{y}^0 := \mathbf{x}^0$ and $t_0 := 1$, $\alpha := L^{-1}$. **3.** For $k = 0, 1, \ldots$, generate two sequences $\{\mathbf{x}^k\}_{k\geq 0}$ and $\{\mathbf{y}^k\}_{k\geq 0}$ as:

$$\begin{cases} \mathbf{x}^{k+1} & := \mathrm{prox}_{\alpha g}\left(\mathbf{y}^k - \alpha\nabla f(\mathbf{y}^k)\right), \\ t_{k+1} & := \frac{1}{2}(1 + \sqrt{4t_k^2 + 1}), \\ \mathbf{y}^{k+1} & := \mathbf{x}^{k+1} + \frac{t_k - 1}{t_{k+1}}(\mathbf{x}^{k+1} - \mathbf{x}^k). \end{cases}$$

Theorem (Convergence of FISTA [2])

Let $\{\mathbf{x}^k\}$ be generated by FISTA. Then:

$$F(\mathbf{x}^k) - F^\star \leq \frac{2L_f\|\mathbf{x}^0 - \mathbf{x}^\star\|_2^2}{(k+1)^2}$$

The worst-case complexity to reach $F(\mathbf{x}^k) - F^\star \leq \varepsilon$ of (FISTA) is $\mathcal{O}\left(R_0\sqrt{\frac{L_f}{\varepsilon}}\right)$, $R_0 := \max_{\mathbf{x}^\star \in \mathcal{S}^\star} \|\mathbf{x}^0 - \mathbf{x}^\star\|_2$.

Fast proximal-gradient algorithm

Fast proximal-gradient scheme (FISTA)

1. Choose $\mathbf{x}^0 \in \mathrm{dom}\,(F)$ arbitrarily as a starting point.
2. Set $\mathbf{y}^0 := \mathbf{x}^0$ and $t_0 := 1$, $\alpha := L^{-1}$.
3. For $k = 0, 1, \ldots$, generate two sequences $\{\mathbf{x}^k\}_{k\geq 0}$ and $\{\mathbf{y}^k\}_{k\geq 0}$ as:

$$\begin{cases} \mathbf{x}^{k+1} & := \mathrm{prox}_{\alpha g}\left(\mathbf{y}^k - \alpha\nabla f(\mathbf{y}^k)\right), \\ t_{k+1} & := \frac{1}{2}(1 + \sqrt{4t_k^2 + 1}), \\ \mathbf{y}^{k+1} & := \mathbf{x}^{k+1} + \frac{t_k - 1}{t_{k+1}}(\mathbf{x}^{k+1} - \mathbf{x}^k). \end{cases}$$

Remark: From $\mathcal{O}\left(\frac{L_f R_0^2}{\epsilon}\right)$ to $\mathcal{O}\left(R_0\sqrt{\frac{L_f}{\epsilon}}\right)$ iterations at almost no additional cost!.

Complexity per iteration

- **One** gradient $\nabla f(\mathbf{y}^k)$ and **one** prox-operator of g;
- 8 arithmetic operations for t_{k+1} and γ_{k+1};
- 2 more vector additions, and **one** scalar-vector multiplication.

The **cost per iteration** is almost the same as in **gradient scheme** if proximal operator of g is efficient.

Example 1: ℓ_1-regularized least squares

Problem (ℓ_1-regularized least squares)

Given $\mathbf{A} \in \mathbb{R}^{n \times p}$ *and* $\mathbf{b} \in \mathbb{R}^n$, *solve:*

$$F^\star := \min_{\mathbf{x} \in \mathbb{R}^p} \left\{ F(\mathbf{x}) := \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_1 \right\}, \tag{6}$$

where $\lambda > 0$ *is a regularization parameter.*

Complexity per iterations

- Evaluating $\nabla f(\mathbf{x}^k) = \mathbf{A}^T(\mathbf{A}\mathbf{x}^k - \mathbf{b})$ requires one $\mathbf{A}\mathbf{x}$ and one $\mathbf{A}^T\mathbf{y}$.
- One soft-thresholding operator $\mathrm{prox}_{\lambda g}(\mathbf{x}) = \mathrm{sign}(\mathbf{x}) \otimes \max\{|\mathbf{x}| - \lambda, 0\}$.
- **Optional**: Evaluating $L = \|\mathbf{A}^T\mathbf{A}\|$ (spectral norm) - via **power iterations**

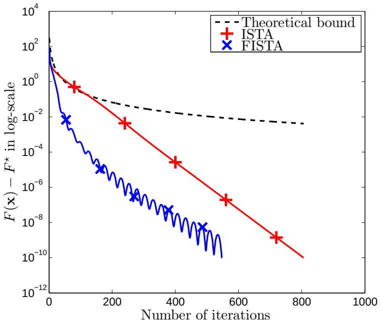
Synthetic data generation

- $\mathbf{A} := \mathrm{randn}(n, p)$ - standard Gaussian $\mathcal{N}(0, \mathbb{I})$.
- $\mathbf{x}^\star$ is a k-sparse vector generated randomly.
- $\mathbf{b} := \mathbf{A}\mathbf{x}^\star + \mathcal{N}(0, 10^{-3})$.

Example 1: Theoretical bounds vs practical performance

Theoretical bounds

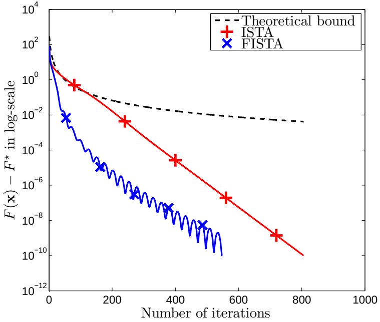
We have the following guarantees for **FISTA** $:= \frac{2L_f R_0^2}{(k+2)^2}$ and for **ISTA** $:= \frac{L_f R_0^2}{2(k+2)}$. In the figure below, ISTA's practical behavior outperforms the theoretical bound for FISTA.

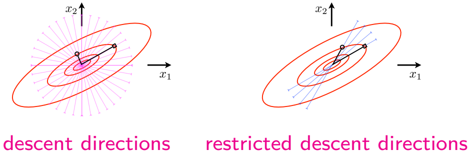


Example 1: Theoretical bounds vs practical performance

Theoretical bounds

We have the following guarantees for **FISTA** $:= \frac{2L_f R_0^2}{(k+2)^2}$ and for **ISTA** $:= \frac{L_f R_0^2}{2(k+2)}$. In the figure below, ISTA's practical behavior outperforms the theoretical bound for FISTA.

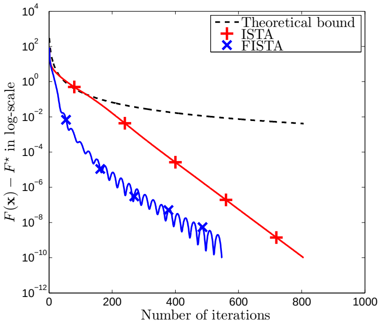


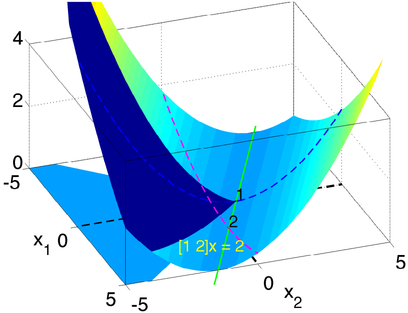


Example 1: Theoretical bounds vs practical performance

Theoretical bounds

We have the following guarantees for **FISTA** $:= \frac{2L_f R_0^2}{(k+2)^2}$ and for **ISTA** $:= \frac{L_f R_0^2}{2(k+2)}$. In the figure below, ISTA's practical behavior outperforms the theoretical bound for FISTA.





Remarks:
- ℓ_1-regularized least squares formulation has **restricted strong convexity**.
- The proximal-gradient method can automatically exploit this structure.

Example 2: Sparse logistic regression

Problem (Sparse logistic regression)

Given $\mathbf{A} \in \mathbb{R}^{n \times p}$ *and* $\mathbf{b} \in \{-1, +1\}^n$, *solve:*

$$F^\star := \min_{\mathbf{x},\beta} \left\{ F(\mathbf{x}) := \frac{1}{n} \sum_{j=1}^{n} \log \left(1 + \exp\left(-\mathbf{b}_j(\mathbf{a}_j^T \mathbf{x} + \beta)\right)\right) + \rho \|\mathbf{x}\|_1 \right\}.$$

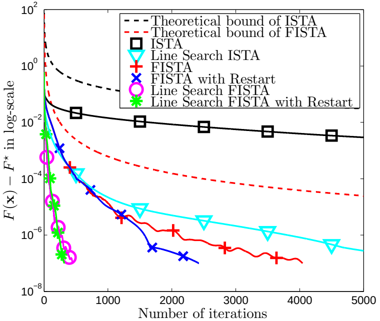
Real data

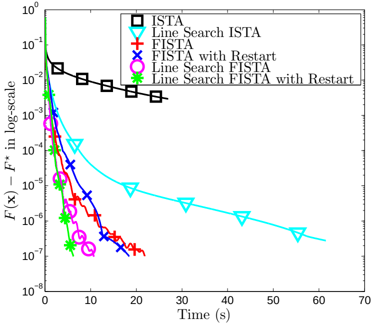
- Real data: `w8a` with $n = 49'749$ data points, $p = 300$ features
- Available at `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html`.

Parameters

- $\rho = 10^{-4}$.
- Number of iterations 5000, tolerance 10^{-7}.
- Ground truth: Solve problem up to 10^{-9} accuracy by TFOCS to get a high accuracy approximation of $\mathbf{x}^\star$ and $F^\star$.

Example 2: Sparse logistic regression - numerical results





	ISTA	LS-ISTA	FISTA	FISTA-R	LS-FISTA	LS-FISTA-R
Number of iterations	5000	5000	4046	2423	447	317
CPU time (s)	26.975	61.506	21.859	18.444	10.683	6.228
Solution error $(\times 10^{-7})$	29370	2.774	1.000	0.998	0.961	0.985

When f is strongly convex: Algorithms

Proximal-gradient scheme (ISTA$_\mu$)

1. Given $\mathbf{x}^0 \in \mathbb{R}^p$ as a starting point.
2. For $k = 0, 1, \cdots$, generate a sequence $\{\mathbf{x}^k\}_{k\geq 0}$ as:

$$\mathbf{x}^{k+1} := \mathrm{prox}_{\alpha_k g}\left(\mathbf{x}^k - \alpha_k \nabla f(\mathbf{x}^k)\right),$$

where $\alpha_k := \frac{2}{L_f + \mu}$ is the optimal step-size.

Fast proximal-gradient scheme (FISTA$_\mu$)

1. Given $\mathbf{x}^0 \in \mathbb{R}^p$ as a starting point. Set $\mathbf{y}^0 := \mathbf{x}^0$.
2. For $k = 0, 1, \cdots$, generate sequences $\{\mathbf{x}^k\}_{k\geq 0}$ and $\{\mathbf{y}^k\}_{k\geq 0}$ as:

$$\begin{cases} \mathbf{x}^{k+1} := \mathrm{prox}_{\alpha_k g}\left(\mathbf{y}^k - \alpha_k \nabla f(\mathbf{y}^k)\right), \\ \mathbf{y}^{k+1} := \mathbf{x}^{k+1} + \left(\frac{\sqrt{c_f} - 1}{\sqrt{c_f} + 1}\right)(\mathbf{x}^{k+1} - \mathbf{x}^k), \end{cases}$$

where $c_f := \frac{L_f}{\mu}$ and $\alpha_k := L_f^{-1}$ is the optimal step-size.

When f is strongly convex: Convergence

Assumption

f is strongly convex with parameter $\mu > 0$, i.e., $f \in \mathcal{F}^{1,1}_{L,\mu}(\mathbb{R}^p)$.

Condition number: $c_f := \frac{L_f}{\mu} \geq 0$.

Theorem (ISTA$_\mu$ [14])

$$F(\mathbf{x}^k) - F^\star \leq \frac{L_f}{2}\left(\frac{c_f - 1}{c_f + 1}\right)^{2k} \|\mathbf{x}^0 - \mathbf{x}^\star\|_2^2.$$

Convergence rate: **Linear** *with contraction factor:* $\omega := \left(\frac{c_f - 1}{c_f + 1}\right)^2 = \left(\frac{L_f - \mu}{L_f + \mu}\right)^2$.

Theorem (FISTA$_\mu$ [14])

$$F(\mathbf{x}^k) - F^\star \leq \frac{L_f + \mu}{2}\left(1 - \sqrt{\frac{\mu}{L_f}}\right)^{k} \|\mathbf{x}^0 - \mathbf{x}^\star\|_2^2.$$

Convergence rate: **Linear** *with contraction factor:* $\omega_f = \frac{\sqrt{L_f} - \sqrt{\mu}}{\sqrt{L_f}} < \omega$.

Summary of the worst-case complexities

Comparison

Complexity	Proximal-gradient scheme	Fast proximal-gradient scheme
Complexity $[\mu = 0]$	$\mathcal{O}\left(R_0^2 \frac{L_f}{\varepsilon}\right)$	$\mathcal{O}\left(R_0 \sqrt{\frac{L_f}{\varepsilon}}\right)$
Per iteration	1-gradient, 1-prox, 1-sv, 1-$v+$	1-gradient, 1-prox, 2-sv, 3-$v+$
Complexity $[\mu > 0]$	$\mathcal{O}\left(\kappa \log(\varepsilon^{-1})\right)$	$\mathcal{O}\left(\sqrt{\kappa} \log(\varepsilon^{-1})\right)$
Per iteration	1-gradient, 1-prox, 1-sv, 1-$v+$	1-gradient, 1-prox, 1-sv, 2-$v+$

Here: sv = scalar-vector multiplication, $v+$=vector addition.

$R_0 := \max_{\mathbf{x}^\star \in \mathcal{S}^\star} \|\mathbf{x}^0 - \mathbf{x}^\star\|$ and $\kappa = \frac{L_f}{\mu_f}$ is the condition number.

Summary of the worst-case complexities

Comparison

Complexity	Proximal-gradient scheme	Fast proximal-gradient scheme
Complexity $[\mu = 0]$	$\mathcal{O}\left(R_0^2 \frac{L_f}{\varepsilon}\right)$	$\mathcal{O}\left(R_0 \sqrt{\frac{L_f}{\varepsilon}}\right)$
Per iteration	1-gradient, 1-prox, 1-sv, 1-$v+$	1-gradient, 1-prox, 2-sv, 3-$v+$
Complexity $[\mu > 0]$	$\mathcal{O}\left(\kappa \log(\varepsilon^{-1})\right)$	$\mathcal{O}\left(\sqrt{\kappa} \log(\varepsilon^{-1})\right)$
Per iteration	1-gradient, 1-prox, 1-sv, 1-$v+$	1-gradient, 1-prox, 1-sv, 2-$v+$

Here: sv = scalar-vector multiplication, $v+$=vector addition.

$R_0 := \max_{\mathbf{x}^\star \in \mathcal{S}^\star} \|\mathbf{x}^0 - \mathbf{x}^\star\|$ and $\kappa = \frac{L_f}{\mu_f}$ is the condition number.

Need alternatives when

- computing $\nabla f(\mathbf{x})$ is much costlier than computing prox_g

Summary of the worst-case complexities

Comparison

Complexity	Proximal-gradient scheme	Fast proximal-gradient scheme
Complexity $[\mu = 0]$	$\mathcal{O}\left(R_0^2 \frac{L_f}{\varepsilon}\right)$	$\mathcal{O}\left(R_0 \sqrt{\frac{L_f}{\varepsilon}}\right)$
Per iteration	1-gradient, 1-prox, 1-sv, 1-$v+$	1-gradient, 1-prox, 2-sv, 3-$v+$
Complexity $[\mu > 0]$	$\mathcal{O}\left(\kappa \log(\varepsilon^{-1})\right)$	$\mathcal{O}\left(\sqrt{\kappa} \log(\varepsilon^{-1})\right)$
Per iteration	1-gradient, 1-prox, 1-sv, 1-$v+$	1-gradient, 1-prox, 1-sv, 2-$v+$

Here: sv = scalar-vector multiplication, $v+$=vector addition.

$R_0 := \max_{\mathbf{x}^\star \in \mathcal{S}^\star} \|\mathbf{x}^0 - \mathbf{x}^\star\|$ and $\kappa = \frac{L_f}{\mu_f}$ is the condition number.

Need alternatives when

- computing $\nabla f(\mathbf{x})$ is much costlier than computing prox_g

Software

TFOCS is a good software package to learn about first order methods.

`http://cvxr.com/tfocs/`

Composite minimization: Non-convex case

Problem (Unconstrained composite minimization)

$$\boxed{F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\}} \qquad \text{(CM)}$$

- $g\colon \mathbb{R}^p \to \mathbb{R}\cup\{\infty\}$ *is proper, closed, convex, and (possibly) nonsmooth.*
- $f\colon \mathbb{R}^p \to \mathbb{R}$ *is proper and closed, dom(f) is convex, and f is L_f−smooth.*
- $\mathrm{dom}(F) := \mathrm{dom}(f) \cap \mathrm{dom}(g) \neq \emptyset$ *and* $-\infty < F^\star < +\infty$.
- *The solution set* $\mathcal{S}^\star := \{\mathbf{x}^\star \in \mathrm{dom}(F) : F(\mathbf{x}^\star) = F^\star\}$ *is nonempty.*

A different quantification of convergence: Gradient mapping

Definition (Gradient mapping)

Let prox_g denote the proximal operator of g and $\lambda > 0$ some real constant. Then, the gradient mapping operator is defined as

$$\mathcal{G}_\lambda(\mathbf{x}) := \frac{1}{\lambda}\left(\mathbf{x} - \mathrm{prox}_{\lambda g}(\mathbf{x} - \lambda \nabla f(\mathbf{x}))\right).$$

Properties [1]

- $\|\mathcal{G}_\lambda(\mathbf{x})\| = 0 \iff \mathbf{x}$ is a stationary point.
- Lipschitz continuity: $\| \mathcal{G}_{\frac{1}{L}}(\mathbf{x}) - \mathcal{G}_{\frac{1}{L}}(\mathbf{y}) \| \leq (2L + L_f)\| \mathbf{x} - \mathbf{y} \|$

Why do we care about gradient mapping?

- It is the generalization of the gradient of f, $\nabla f(\mathbf{x})$
- Recall prox-gradient update: $\mathbf{x}^{t+1} = \mathrm{prox}_{\lambda g}(\mathbf{x}^t - \lambda \nabla f(\mathbf{x}^t))$, which is equivalent to $\mathbf{x}^{t+1} = \mathbf{x}^t - \lambda \mathcal{G}_\lambda(\mathbf{x}^t)$.
- In fact, when $\mathrm{prox}_g = \mathbb{I}$, then, $\mathcal{G}_\lambda(\mathbf{x}) = \frac{1}{\lambda}(\mathbf{x} - (\mathbf{x} - \lambda \nabla f(\mathbf{x}))) = \nabla f(\mathbf{x})$.

Sufficient Decrease property for proximal-gradient

Assumption

- f *is* L_f*-smooth.*
- g *is proper, closed, convex, and (possibly) nonsmooth.* g *is proximally tractable.*

$$\boxed{\mathbf{x}^{k+1} := \mathrm{prox}_{\frac{1}{L}g}\left(\mathbf{x}^k - \frac{1}{L}\nabla f(\mathbf{x}^k)\right)}$$

Lemma (Sufficient decrease [1])

For any $\mathbf{x} \in \mathrm{int}(\mathrm{dom}(f))$ *and* $L \in (\frac{L_f}{2}, \infty)$*, it holds that*

$$F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \frac{L - \frac{L_f}{2}}{L^2}\left\|\mathcal{G}_{\frac{1}{L}}(\mathbf{x}^k)\right\|_2^2, \tag{7}$$

Corollary

$$F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \frac{1}{2L_f}\left\|\mathcal{G}_{\frac{1}{L_f}}(\mathbf{x}^k)\right\|_2^2, \qquad \textit{for } L = L_f$$

Non-convex case: Convergence

Basic proximal-gradient scheme
1. Choose $\mathbf{x}^0 \in \mathrm{dom}(F)$ arbitrarily as a starting point.
2. For $k = 0, 1, \cdots$, generate a sequence $\{\mathbf{x}^k\}_{k \geq 0}$ as: $$\mathbf{x}^{k+1} := \mathrm{prox}_{\alpha g}\left(\mathbf{x}^k - \alpha \nabla f(\mathbf{x}^k)\right),$$
where $\alpha := \left(0, \frac{2}{L_f}\right)$.

Theorem (Convergence of proximal-gradient method: Non-convex [1])

Let $\{\mathbf{x}^k\}$ be generated by proximal-gradient scheme above. Then, we have

$$\min_{i=0,1,\ldots,k} \|\mathcal{G}_\alpha(\mathbf{x}^i)\|_2^2 \leq \frac{F(\mathbf{x}^0) - F(\mathbf{x}^\star)}{M(k+1)}, \qquad \text{where } M := \alpha^2\left(\frac{1}{\alpha} - \frac{L_f}{2}\right)$$

- *When* $\alpha = \frac{1}{L_f}$, $M = \frac{1}{2L_f}$.
- *The worst-case complexity to reach* $\min_{i=0,1,\cdots,k} \|\mathcal{G}_\alpha(\mathbf{x}^i)\|_2^2 \leq \varepsilon$ *is* $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$.

Stochastic convex composite minimization

Problem (Mathematical formulation)

Consider the following composite convex minimization problem:

$$F^\star = \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ F(\mathbf{x}) := \mathbb{E}_\theta[F(\mathbf{x},\theta)] := \mathbb{E}_\theta[f(\mathbf{x},\theta) + g(\mathbf{x},\theta)] \right\}$$

- *θ is a random vector whose probability distribution is supported on set Θ.*
- *The solution set $\mathcal{S}^\star := \{\mathbf{x}^\star \in \mathrm{dom}\,(F) : F(\mathbf{x}^\star) = F^\star\}$ is nonempty.*
- **Oracles:** *(sub)gradient of $f(\cdot,\theta)$, $\nabla f(\mathbf{x},\theta)$, and stochastic prox operator of $g(\cdot,\theta)$, $\mathrm{prox}_{g(\cdot,\theta)}(\mathbf{x})$.*

Remark

◦ In this setting, we replace $\nabla f(\cdot)$ with its stochastic estimates.

◦ It is possible to replace $\mathrm{prox}_g(\cdot)$ with its stochastic estimate (advanced material).

Stochastic proximal gradient method

Stochastic proximal gradient method (SPG)

1. Choose $\mathbf{x}^0 \in \mathbb{R}^p$ and $(\gamma_k)_{k\in\mathbb{N}} \in \,]0, +\infty[^{\mathbb{N}}$.
2. For $k = 0, 1, \ldots$ perform:

$$\mathbf{x}^{k+1} = \mathrm{prox}_{\gamma_k g(\cdot,\theta)}(\mathbf{x}^k - \gamma_k G(\mathbf{x}^k, \theta_k)).$$

Definitions:

- $\mathrm{prox}_{\lambda g(\cdot,\theta)} := \arg\min_{\mathbf{y}\in\mathbb{R}^p} \left\{ g(\mathbf{y}, \theta) + \frac{1}{2\lambda} \| \mathbf{y} - \mathbf{x} \|^2 \right\}$
- $\{\theta_k\}_{k=0,1,\ldots}$: sequence of independent random variables.
- $G(\mathbf{x}^k, \theta_k) \in \partial f(\mathbf{x}^k, \theta_k)$: an unbiased estimate of the deterministic (sub)gradient:

$$\mathbb{E}[G(\mathbf{x}^k, \theta_k)] \in \partial f(\mathbf{x}^k).$$

Stochastic proximal gradient method

Stochastic proximal gradient method (SPG)

1. Choose $\mathbf{x}^0 \in \mathbb{R}^p$ and $(\gamma_k)_{k\in\mathbb{N}} \in]0, +\infty[^{\mathbb{N}}$.
2. For $k = 0, 1, \ldots$ perform:

$$\mathbf{x}^{k+1} = \mathrm{prox}_{\gamma_k g(\cdot,\theta)}(\mathbf{x}^k - \gamma_k G(\mathbf{x}^k, \theta_k)).$$

Definitions:

- $\mathrm{prox}_{\lambda g(\cdot,\theta)} := \arg\min_{\mathbf{y}\in\mathbb{R}^p} \left\{ g(\mathbf{y}, \theta) + \frac{1}{2\lambda}\|\mathbf{y} - \mathbf{x}\|^2 \right\}$
- $\{\theta_k\}_{k=0,1,\ldots}$: sequence of independent random variables.
- $G(\mathbf{x}^k, \theta_k) \in \partial f(\mathbf{x}^k, \theta_k)$: an unbiased estimate of the deterministic (sub)gradient:

$$\mathbb{E}[G(\mathbf{x}^k, \theta_k)] \in \partial f(\mathbf{x}^k).$$

Remark

Cost of computing $G(\mathbf{x}^k, \theta_k)$ is usually much cheaper than $\nabla f(\mathbf{x}^k)$.

Convergence analysis

Assumptions for the problem setting

- $f(\cdot, \theta)$ and $g(\cdot, \theta)$ are convex functions in the first argument, g is proximally-tractable.
- (Sub)gradients of F satisfy stochastic bounded gradient condition: $\exists C \geq 0, B \geq 0$ such that

$$\mathbb{E}_\theta[\| \partial F(\mathbf{x}, \theta) \|^2] \leq B^2 + C(F(\mathbf{x}) - F(\mathbf{x}^\star)).$$

- $\mathbb{E}[\| \mathbf{x}^t - \mathbf{x}^\star \|^2] \leq R^2$ for all $t \geq 0$.

Implications of the assumptions

- None of the above assumptions enforce that f is smooth.
- Stochastic bounded gradient condition holds with $C = 0$ when both $f(\cdot, \theta)$ and $g(\cdot, \theta)$ are Lipschitz continuous.
- The same condition holds when $f(\cdot, \theta)$ is L_f-smooth and $g(\cdot, \theta)$ is Lipschitz continuous.
- **However, for the upcoming theorem, we will take $C > 0$, which rules out the case when both functions are only Lipschitz continuous.**

Convergence analysis

Assumptions for the problem setting

- $f(\cdot, \theta)$ and $g(\cdot, \theta)$ are convex functions in the first argument, g is proximally-tractable.
- (Sub)gradients of F satisfy stochastic bounded gradient condition: $\exists C \geq 0, B \geq 0$ such that

$$\mathbb{E}_\theta[\| \partial F(\mathbf{x}, \theta) \|^2] \leq B^2 + C(F(\mathbf{x}) - F(\mathbf{x}^\star)).$$

- $\mathbb{E}[\| \mathbf{x}^t - \mathbf{x}^\star \|^2] \leq R^2$ for all $t \geq 0$.

Theorem (Ergodic convergence [12])

- *Assume the above assumptions hold with $C > 0$.*
- *Let the sequence $\{\mathbf{x}^k\}_{k \geq 0}$ be generated by SPG.*
- *Set $\gamma_k = \frac{1}{C\sqrt{k}}$.*

Conclusion:

- *Define $\bar{\mathbf{x}}^k = \frac{1}{k}\sum_{i=0}^{k-1} \mathbf{x}^i$, then*

$$\mathbb{E}[F(\bar{\mathbf{x}}^k) - F(\mathbf{x}^\star)] \leq \frac{1}{\sqrt{k}} \left(R^2 C + \frac{B^2}{C} \right), \quad \forall k \geq 1.$$

Revisiting a special composite structure

A basic constrained problem setting

$$f^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) + \delta_{\mathcal{X}}(\mathbf{x}) \right\} := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) : \mathbf{x} \in \mathcal{X} \right\}, \quad (8)$$

Assumptions

- $\mathcal{X}$ is nonempty, convex and compact (closed and bounded) where $\delta_{\mathcal{X}}$ is its indicator function.
- $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^p)$ (i.e., convex with Lipschitz gradient).

Recall proximal gradient algorithm

Basic proximal-gradient scheme (ISTA)
1. Choose $\mathbf{x}^0 \in \mathrm{dom}\,(F)$ arbitrarily as a starting point. **2.** For $k = 0, 1, \cdots$, generate a sequence $\{\mathbf{x}^k\}_{k\geq 0}$ as: $$\mathbf{x}^{k+1} := \mathrm{prox}_{\alpha g}\left(\mathbf{x}^k - \alpha\nabla f(\mathbf{x}^k)\right)$$ where $\alpha := \frac{1}{L}$.

- Prox-operator of indicator of $\mathcal{X}$ is projection onto $\mathcal{X}$ $\Longrightarrow$ ensures feasibility

How else can we ensure feasibility?

Frank-Wolfe's approach - I

$$f^\star := \min_{\mathbf{x}\in\mathbb{R}^p}\left\{f(\mathbf{x}) : \mathbf{x}\in\mathcal{X}\right\},$$

Conditional gradient method (CGM, see [10] for review)

A plausible strategy which dates back to 1956 [6]. At iteration k:

1. Consider the linear approximation of f at $\mathbf{x}^k$

$$\phi_k(\mathbf{x}) := f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T(\mathbf{x}-\mathbf{x}^k)$$

2. Minimize this approximation within constraint set

$$\hat{\mathbf{x}}^k \in \min_{x\in\mathcal{X}} \phi_k(\mathbf{x}) = \min_{x\in\mathcal{X}} \nabla f(\mathbf{x}^k)^T\mathbf{x}$$

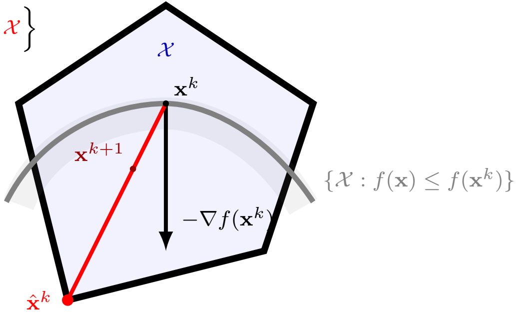
3. Take a step towards $\hat{\mathbf{x}}^k$ with step-size $\gamma_k \in [0,1]$

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \gamma_k(\hat{\mathbf{x}}^k - \mathbf{x}^k)$$

- $\mathbf{x}^{k+1}$ is feasible since it is convex combination of two other feasible points.

Frank-Wolfe's approach - II

$$f^\star := \min_{\mathbf{x}\in\mathbb{R}^p}\left\{f(\mathbf{x}) : \mathbf{x}\in\mathcal{X}\right\}$$



Conditional gradient method (CGM)

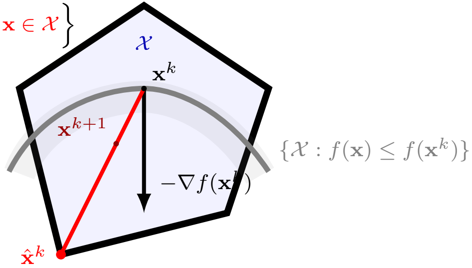
1. Choose $\mathbf{x}^0 \in \mathcal{X}$.
2. For $k = 0, 1, \ldots$ perform:

$$\begin{cases} \hat{\mathbf{x}}^k & := \arg\min_{\mathbf{x}\in\mathcal{X}} \nabla f(\mathbf{x}^k)^T\mathbf{x} \\ \mathbf{x}^{k+1} & := (1-\gamma_k)\mathbf{x}^k + \gamma_k\hat{\mathbf{x}}^k, \end{cases}$$

where $\gamma_k := \frac{2}{k+2}$.

On the linear minimization oracle

$$f^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) : \mathbf{x} \in \mathcal{X} \right\}$$



Definition (Linear minimization oracle)

Let $\mathcal{X}$ be a convex, closed and bounded set. Then, the linear minimization oracle of $\mathcal{X}$ ($\mathrm{lmo}_{\mathcal{X}}$) returns a vector $\hat{\mathbf{x}}$ such that

$$\boxed{\mathrm{lmo}_{\mathcal{X}}(\mathbf{x}) := \hat{\mathbf{x}} \in \arg\min_{\mathbf{y}\in\mathcal{X}} \ \mathbf{x}^T\mathbf{y}} \tag{9}$$

- $\mathrm{lmo}_{\mathcal{X}}$ returns an extreme point of $\mathcal{X}$.
- $\mathrm{lmo}_{\mathcal{X}}$ is arguably cheaper than projection.
- $\mathrm{lmo}_{\mathcal{X}}$ is not single valued, note $\in$ in the definition.

Convergence guarantees of CGM

Problem setting

$$f^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) : \mathbf{x} \in \mathcal{X} \right\},$$

Assumptions

- $\mathcal{X}$ is nonempty, convex, closed and bounded.
- $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^p)$ (i.e., convex with Lipschitz gradient).

Theorem

*Under **assumptions** listed above, CGM with step size $\gamma_k = \frac{2}{k+2}$ satisfies*

$$\boxed{f(\mathbf{x}^k) - f(\mathbf{x}^\star) \leq \frac{4LD_{\mathcal{X}}^2}{k+1}} \tag{10}$$

where $D_{\mathcal{X}} := \max_{\mathbf{x},\mathbf{y}\in\mathcal{X}} \|\mathbf{x}-\mathbf{y}\|_2$ is diameter of constraint set.

*Convergence guarantees of CGM: A faster rate

Problem setting

$$f^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \Big\{ f(\mathbf{x}) : \mathbf{x} \in \mathcal{X} \Big\},$$

Assumptions

- $\mathcal{X}$ is nonempty, α-strongly convex, closed and bounded.
- $f \in \mathcal{F}^{1,1}_{L,\mu}(\mathbb{R}^p)$ (i.e., strongly convex with Lipschitz gradient).

Definition (α-strongly convex set) [7]

A convex set $\mathcal{X} \in \mathbb{R}^{p\times p}$ is α-strongly convex with respect to $\|\cdot\|$ if for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, any $\gamma \in [0,1]$ and any vector $\mathbf{z} \in \mathbb{R}^{p\times p}$ such that $\|\mathbf{z}\| = 1$, it holds that

$$\gamma\mathbf{x} + (1-\gamma)\mathbf{y} + \gamma(1-\gamma)\frac{\alpha}{2}\|\mathbf{x}-\mathbf{y}\|^2\mathbf{z} \in \mathcal{X}$$

That is, for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, the ball centered at $\gamma\mathbf{x} + (1-\gamma)\mathbf{y}$ with radius $\gamma(1-\gamma)\frac{\alpha}{2}\|\mathbf{x}-\mathbf{y}\|^2$ is contained in $\mathcal{X}$.

*CGM for strongly convex objective + strongly convex set

Conditional gradient method - CGM2
1. Choose $\mathbf{x}^0 \in \mathcal{X}$.
2. For $k = 0, 1, \ldots$ perform:
$\begin{cases} \hat{\mathbf{x}}^k & := \arg\min_{\mathbf{x}\in\mathcal{X}} \nabla f(\mathbf{x}^k)^T \mathbf{x} \\ \gamma_k & := \arg\min_{\gamma\in[0,1]} \gamma \left\langle \hat{\mathbf{x}}^k - \mathbf{x}^k, \nabla f(\mathbf{x}^k) \right\rangle + \gamma^2 \frac{L}{2} \| \hat{\mathbf{x}}^k - \mathbf{x}^k \|^2 \\ \mathbf{x}^{k+1} & := (1-\gamma_k)\mathbf{x}^k + \gamma_k \hat{\mathbf{x}}^k, \end{cases}$

Theorem ([7])

*Under **assumptions** listed previously, CGM2 satisfies*

$$\boxed{f(\mathbf{x}^k) - f(\mathbf{x}^\star) = \mathcal{O}\left(\frac{1}{k^2}\right)}. \tag{11}$$

Example: lmo of nuclear-norm bal

Consider $\delta_{\mathcal{X}}$, the indicator of nuclear-norm ball $\mathcal{X} := \left\{\mathbf{X} : \mathbf{X} \in \mathbb{R}^{p\times p},\ \|\mathbf{X}\|_* \leq \alpha\right\}$

lmo of nuclear-norm ball

$$\text{lmo}_{\mathcal{X}}(\mathbf{X}) := \hat{\mathbf{X}} \in \arg\min_{\mathbf{Y}\in\mathcal{X}} \ \langle \mathbf{Y}, \mathbf{X}\rangle$$

This can be computed as follows:

- Compute top singular vectors of $\mathbf{X}$ $\Longrightarrow$ $(\mathbf{u}_1, \sigma_1, \mathbf{v}_1) = \mathbf{svds}(\mathbf{X}, 1)$.
- Form the rank-1 output $\Longrightarrow$ $\mathbf{X} = -\mathbf{u}_1 \alpha \mathbf{v}_1^T$

We can efficiently approximate top singular vectors by power method!

Proximal gradient vs. Frank-Wolfe

Definitions:

- Here: sv = scalar-vector multiplication, $v+$=vector addition.
- $R_0 := \max_{\mathbf{x}^\star \in \mathcal{S}^\star} \|\mathbf{x}^0 - \mathbf{x}^\star\|$ is the maximum initial distance.
- $D_{\mathcal{X}} := \max_{\mathbf{x},\mathbf{y} \in \mathcal{X}} \|\mathbf{x} - \mathbf{y}\|_2$ is diameter of constraint set $\mathcal{X}$.

Algorithm	Proximal-gradient scheme	Frank-Wolfe method
Rate	$\mathcal{O}\left(\frac{L_f R_0^2}{k}\right)$	$\mathcal{O}\left(\frac{L_f D_{\mathcal{X}}^2}{k}\right)$
Complexity	$\mathcal{O}\left(R_0^2 \frac{L_f}{\varepsilon}\right)$	$\mathcal{O}\left(D_{\mathcal{X}}^2 \frac{L_f}{\varepsilon}\right)$
Per iteration	1-gradient, 1-prox, 1-sv, 1-$v+$	1-gradient, 1-lmo, 2-sv, 1-$v+$

How do prox operator and lmo compare in practice?

An example with matrices

Problem Definition

$$\min_{\mathbf{X} \in \mathbb{R}^{p \times p}} f(\mathbf{X}) + g(\mathbf{X})$$

- Define $g(\mathbf{X}) = \delta_{\mathcal{X}}(\mathbf{X})$, where $\mathcal{X} := \left\{\mathbf{X} : \mathbf{X} \in \mathbb{R}^{p\times p},\ \|\mathbf{X}\|_* \leq \alpha\right\}$ is nuclear norm ball.
- This problem is equivalent to:

$$\min_{\mathbf{X} \in \mathcal{X}} f(\mathbf{X})$$

Observations

- $\mathrm{prox}_g = \pi_{\mathcal{X}}$. Projection requires full SVD, $\mathcal{O}(p^3)$.
- lmo computes (approximately) top singular vectors, roughly in $\approx \mathcal{O}(p^2)$ with Lanczos algorithm.

Example: Phase retrieval

Phase retrieval

Aim: Recover signal $\mathbf{x}^\natural \in \mathbb{C}^p$ from the measurements $\mathbf{b} \in \mathbb{R}^n$:

$$b_i = \left|\langle \mathbf{a}_i, \mathbf{x}^\natural \rangle\right|^2 + \omega_i.$$

($\mathbf{a}_i \in \mathbb{C}^p$ are known measurement vectors, ω_i models noise).

• Non-linear measurements $\rightarrow$ **non-convex** maximum likelihood estimators.

PhaseLift [4]

Phase retrieval can be solved as a convex matrix completion problem, following a combination of

- semidefinite relaxation $(\mathbf{x}^\natural {\mathbf{x}^\natural}^H = \mathbf{X}^\natural)$
- convex relaxation $(\texttt{rank} \rightarrow \|\cdot\|_*)$

albeit in terms of the lifted variable $\mathbf{X} \in \mathbb{C}^{p\times p}$.

Example: Phase retrieval - II

Problem formulation

We solve the following PhaseLift variant:

$$f^\star := \min_{\mathbf{X}\in\mathbb{C}^{p\times p}} \left\{ \frac{1}{2}\|\mathcal{A}(\mathbf{X}) - \mathbf{b}\|_2^2 : \quad \|\mathbf{X}\|_* \leq \kappa, \quad \mathbf{X} \succeq 0 \right\}. \tag{12}$$

Experimental setup [19]

Coded diffraction pattern measurements, $\mathbf{b} = [\mathbf{b}_1, \ldots, \mathbf{b}_L]$ with $L = 20$ different masks

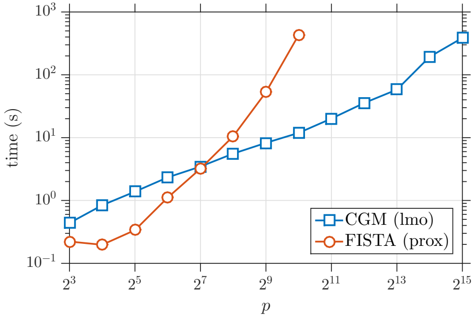
$$\mathbf{b}_\ell = |\texttt{fft}(\mathbf{d}_\ell^H \odot \mathbf{x}^\natural)|^2$$

→ $\odot$ denotes Hadamard product; $|\cdot|^2$ applies element-wise

→ $\mathbf{d}_\ell$ are randomly generated octonary masks (distributions as proposed in [4])

→ Parametric choices: $\lambda^0 = \mathbf{0}^n$; $\quad \epsilon = 10^{-2}$; $\quad \kappa = \texttt{mean}(\mathbf{b})$.

Example: Phase retrieval - III



Test with synthetic data: Prox vs sharp

→ Synthetic data: $\mathbf{x}^\natural = \texttt{randn}(p, 1) + i \cdot \texttt{randn}(p, 1)$.

→ Stopping criteria: $\frac{\|\mathbf{x}^\natural - \mathbf{x}^k\|_2}{\|\mathbf{x}^\natural\|_2} \leq 10^{-2}$.

→ Averaged over 10 Monte-Carlo iterations.

Note that the problem is $p \times p$ dimensional!

A basic constrained non-convex problem

Problem setting

$$f^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) : \mathbf{x} \in \mathcal{X} \right\},$$

Assumptions

- $\mathcal{X}$ is nonempty, convex, closed and bounded.
- f has L-Lipschitz continuous gradients, but it is **non-convex**.

Stationary point

Due to constraints, $\|\nabla f(\mathbf{x}^\star)\| = 0$ may not hold!

Frank-Wolfe gap: Following measure, known as FW-gap, generalizes the definition of stationary point for constrained problems:

$$g_{FW}(\mathbf{x}) := \max_{\mathbf{y}\in\mathcal{X}} \ (\mathbf{x}-\mathbf{y})^T \nabla f(\mathbf{x})$$

- $g_{FW}(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathcal{X}$.
- $\mathbf{x} \in \mathcal{X}$ is a stationary point if and only if $g_{FW}(\mathbf{x}) = 0$.

CGM for non-convex problems

CGM for non-convex problems

1. Choose $\mathbf{x}^0 \in \mathcal{X}$, $K > 0$ total number of iterations.
2. For $k = 0, 1, \ldots, K-1$ perform:

$$\begin{cases} \hat{\mathbf{x}}^k & := \mathrm{lmo}_{\mathcal{X}}(\nabla f(\mathbf{x}^k)) \\ \mathbf{x}^{k+1} & := (1-\gamma_k)\mathbf{x}^k + \gamma_k \hat{\mathbf{x}}^k, \end{cases}$$

where $\gamma_k := \frac{1}{\sqrt{K+1}}$.

Theorem

Denote $\bar{\mathbf{x}}$ chosen uniformly random from $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^K\}$. Then, CGM satisfies

$$\min_{k=1,2,\ldots,K} g_{FW}(\mathbf{x}^k) \leq \mathbb{E}[g_{FW}(\bar{\mathbf{x}})] \leq \frac{1}{\sqrt{K}}\left(f(\mathbf{x}^0) - f^\star + \frac{LD^2}{2}\right).$$

* There exist stochastic CGM methods for non-convex problems. See [17] for details.

A basic constrained stochastic problem

Problem setting (Stochastic)

$$f^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ \mathbb{E}[f(\mathbf{x},\theta)] : \mathbf{x} \in \mathcal{X} \right\}, \tag{13}$$

Assumptions

- θ is a random vector whose probability distribution is supported on set Θ
- $\mathcal{X}$ is nonempty, convex, closed and bounded.
- $f(\cdot,\theta) \in \mathcal{F}_L^{1,1}(\mathbb{R}^p)$ for all θ (i.e., convex with Lipschitz gradient).

Example (Finite-sum model)

$$\mathbb{E}[f(\mathbf{x},\theta)] = \frac{1}{n}\sum_{j=1}^{n} f_j(\mathbf{x})$$

- $j = \theta$ is a drawn uniformly from $\Theta = \{1, 2, \ldots, n\}$
- $f_j \in \mathcal{F}_L^{1,1}(\mathbb{R}^p)$ for all j (i.e., convex with Lipschitz gradient).

Stochastic conditional gradient method

Stochastic conditional gradient method (SFW)
1. Choose $\mathbf{x}^0 \in \mathcal{X}$.
2. For $k = 0, 1, \ldots$ perform:
$$\begin{cases} \hat{\mathbf{x}}^k & := \mathrm{lmo}_{\mathcal{X}}(\tilde{\nabla} f(\mathbf{x}^k, \theta_k)) \\ \mathbf{x}^{k+1} & := (1-\gamma_k)\mathbf{x}^k + \gamma_k \hat{\mathbf{x}}^k, \end{cases}$$
where $\gamma_k := \frac{2}{k+2}$, and $\tilde{\nabla} f$ is an unbiased estimator of ∇f.

Theorem [9]

Assume that the following variance condition holds

$$\mathbb{E}\| \nabla f(\mathbf{x}^k) - \tilde{\nabla} f(\mathbf{x}^k, \theta_k) \|^2 \leq \left(\frac{LD}{k+1}\right)^2. \qquad (\star)$$

Then, the iterates of SFW satisfies

$$\mathbb{E}[f(\mathbf{x}^k, \theta)] - f^\star \leq \frac{4LD^2}{k+1}.$$

$(\star)$ $\rightarrow$ SFW requires decreasing variance!

Stochastic conditional gradient method

Stochastic conditional gradient method (SFW)

1. Choose $\mathbf{x}^0 \in \mathcal{X}$.
2. For $k = 0, 1, \ldots$ perform:

$$\begin{cases} \hat{\mathbf{x}}^k & := \mathrm{lmo}_{\mathcal{X}}(\tilde{\nabla} f(\mathbf{x}^k, \theta_k)) \\ \mathbf{x}^{k+1} & := (1-\gamma_k)\mathbf{x}^k + \gamma_k \hat{\mathbf{x}}^k, \end{cases}$$

where $\gamma_k := \frac{2}{k+2}$, and $\tilde{\nabla} f$ is an unbiased estimator of ∇f.

Example (Finite-sum model)

$$\mathbb{E}[f(\mathbf{x}, \theta)] = \frac{1}{n}\sum_{j=1}^{n} f_j(\mathbf{x})$$

Assume f_j is G-Lipschitz continuous for all j. Suppose that $\mathcal{S}_k$ is a random sampling (with replacement) from $\Theta = \{1, 2, \ldots, n\}$. Then,

$$\tilde{\nabla} f(\mathbf{x}^k, \theta_k) := \frac{1}{|\mathcal{S}_k|}\sum_{j \in \mathcal{S}_k} f_j(\mathbf{x}^k) \quad \Longrightarrow \quad \mathbb{E}\| \nabla f(\mathbf{x}) - \tilde{\nabla} f(\mathbf{x}, \theta_k) \|^2 \leq \frac{G^2}{|\mathcal{S}_k|}.$$

Hence, by choosing $|\mathcal{S}_k| = (\frac{G(k+1)}{LD})^2$ we satisfy the variance condition for SFW.

Wrap up!

◦ Monday: Transition from variance reduction to deep learning...

*Expanding on prox operator and optimality condition

Notes

- By definition, $g(\mathbf{y}) + \frac{1}{2\lambda}\|\mathbf{y} - \mathbf{x}\|^2$ attains its minimum when $\mathbf{y} = \mathrm{prox}_{\lambda g}(\mathbf{x})$.
- One can see that $g(\mathbf{y}) + \frac{1}{2\lambda}\|\mathbf{y} - \mathbf{x}\|^2$ is convex, and prox operator computes its minimizer over $\mathbb{R}^p$.
- As a result, subdifferential of $g(\mathbf{y}) + \frac{1}{2\lambda}\|\mathbf{y} - \mathbf{x}\|^2$ at the minimizer $(\mathbf{y} = \mathrm{prox}_{\lambda g}(\mathbf{x}))$ should include 0.
- Hence, $0 \in \partial g(\mathrm{prox}_{\lambda g}(\mathbf{x})) + \frac{1}{\lambda}\left(\mathrm{prox}_{\lambda g}(\mathbf{x}) - \mathbf{x}\right)$.

- After rearranging the above inclusion we obtain: $\mathbf{x} \in \lambda\partial g(\mathrm{prox}_{\lambda g}(\mathbf{x})) + \mathrm{prox}_{\lambda g}(\mathbf{x})$
- We can rewrite the RHS as a single function: $\lambda\partial g(\mathrm{prox}_{\lambda g}(\mathbf{x})) + \mathrm{prox}_{\lambda g}(\mathbf{x}) = (\lambda\partial g + \mathbb{I})(\mathrm{prox}_{\lambda g}(\mathbf{x}))$
- The inclusion becomes: $\mathbf{x} \in (\lambda\partial g + \mathbb{I})(\mathrm{prox}_{\lambda g}(\mathbf{x}))$.

- Finally, we compute the inverse of $(\lambda\partial g + \mathbb{I})(\cdot)$ to conclude: $\mathrm{prox}_{\lambda g}(\mathbf{x}) = (\lambda\partial g + \mathbb{I})^{-1}(\mathbf{x})$.

∘ In the literature, $(\lambda\partial g + \mathbb{I})^{-1}$ is called the *resolvent of the subdifferential of g with parameter λ*.
∘ This is just a technical term that stands for *proximal operator of λg*, as we have defined in this course.

*A short detour: Basic properties of prox-operator

Property (Basic properties of prox-operator)

1. $\text{prox}_g(\mathbf{x})$ *is well-defined and single-valued (i.e., the prox-operator* (3) *has a unique solution since* $g(\cdot) + \frac{1}{2}\|\cdot - \mathbf{x}\|_2^2$ *is strongly convex).*
2. *Optimality condition:*
$$\mathbf{x} \in \text{prox}_g(\mathbf{x}) + \partial g(\text{prox}_g(\mathbf{x})), \ \mathbf{x} \in \mathbb{R}^p.$$
3. $\mathbf{x}^\star$ *is a fixed point of* $\text{prox}_g(\cdot)$*:*
$$0 \in \partial g(\mathbf{x}^\star) \quad \Leftrightarrow \quad \mathbf{x}^\star = \text{prox}_g(\mathbf{x}^\star).$$
4. *Nonexpansiveness:*
$$\|\text{prox}_g(\mathbf{x}) - \text{prox}_g(\tilde{\mathbf{x}})\|_2 \le \|\mathbf{x} - \tilde{\mathbf{x}}\|_2, \quad \forall \mathbf{x}, \tilde{\mathbf{x}} \in \mathbb{R}^p.$$

Note: An operator is called *non-expansive* if it is L-Lipschitz continuous with $L = 1$.

*Adaptive Restart

It is possible the preserve $\mathcal{O}(\frac{1}{k^2})$ convergence guarantee !

One needs to slightly modify the algorithm as below.

Generalized fast proximal-gradient scheme

1. Choose $\mathbf{x}^0 = \mathbf{x}^{-1} \in \mathrm{dom}\,(F)$ arbitrarily as a starting point.
2. Set $\theta_0 = \theta_{-1} = 1$, $\lambda := L_f^{-1}$
3. For $k = 0, 1, \ldots$, generate two sequences $\{\mathbf{x}^k\}_{k\geq 0}$ and $\{\mathbf{y}^k\}_{k\geq 0}$ as:

$$
\begin{cases}
\mathbf{y}^k := \mathbf{x}^k + \theta_k(\theta_{k-1}^{-1} - 1)(\mathbf{x}^k - \mathbf{x}^{k-1}) \\
\mathbf{x}^{k+1} := \mathrm{prox}_{\lambda g}\left(\mathbf{y}^k - \lambda \nabla f(\mathbf{y}^k)\right), \\
\text{if restart test holds} \\
\qquad \theta_{k-1} = \theta_k = 1 \\
\qquad \mathbf{y}^k = \mathbf{x}^k \\
\qquad \mathbf{x}^{k+1} := \mathrm{prox}_{\lambda g}\left(\mathbf{y}^k - \lambda \nabla f(\mathbf{y}^k)\right)
\end{cases}
\tag{14}
$$

θ_k is chosen so that it satisfies

$$
\theta_{k+1} = \frac{\sqrt{\theta_k^4 + 4\theta_k^2} - \theta_k^2}{2} < \frac{2}{k+3}
$$

⋆Adaptive Restart: Guarantee

Theorem (Global complexity [8])

The sequence $\{\mathbf{x}^k\}_{k\geq 0}$ *generated by the modified algorithm satisfies*

$$F(\mathbf{x}^k) - F^\star \leq \frac{2L_f}{(k+2)^2}\left(R_0^2 + \sum_{k_i \leq k}\left(\|\mathbf{x}^\star - \mathbf{x}^{k_i}\|_2^2 - \|\mathbf{x}^\star - \mathbf{z}^{k_i}\|_2^2\right)\right) \quad \forall k \geq 0. \tag{15}$$

where $R_0 := \min_{\mathbf{x}^\star \in \mathcal{S}^\star} \|\mathbf{x}^0 - \mathbf{x}^\star\|$, $\mathbf{z}^k = \mathbf{x}^{k-1} + \theta_{k-1}^{-1}(\mathbf{x}^k - \mathbf{x}^{k-1})$ *and* $k_i, i = 1\ldots$ *are the iterations for which the restart test holds.*

Various restarts tests that might coincide with $\|\mathbf{x}^* - \mathbf{x}^{k_i}\|_2^2 \leq \|\mathbf{x}^* - \mathbf{z}^{k_i}\|_2^2$

- Exact non-monotonicity test: $F(\mathbf{x}^{k+1}) - F(\mathbf{x}^k) > 0$
- Non-monotonicity test: $\langle (L_F(\mathbf{y}^{k-1} - \mathbf{x}^k), \mathbf{x}^{k+1} - \frac{1}{2}(\mathbf{x}^k + y^{k-1})\rangle > 0$ (implies exact non-monotonicity and it is advantageous when function evaluations are expensive)
- Gradient-mapping based test: $\langle (L_f(\mathbf{y}^k - \mathbf{x}^{k+1}), \mathbf{x}^{k+1} - \mathbf{x}^k\rangle > 0$

*Recall: Composite convex minimization

Problem (Unconstrained composite convex minimization)

$$F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\}$$

- *f and g are both proper, closed, and convex.*
- $\mathrm{dom}(F) := \mathrm{dom}(f) \cap \mathrm{dom}(g) \neq \emptyset$ *and* $-\infty < F^\star < +\infty$.
- *The solution set* $\mathcal{S}^\star := \{\mathbf{x}^\star \in \mathrm{dom}(F) : F(\mathbf{x}^\star) = F^\star\}$ *is nonempty.*

*Recall: Composite convex minimization guarantees

Proximal gradient method(ISTA) vs. fast proximal gradient method (FISTA)

Assumptions, step sizes and convergence rates

Proximal gradient method:

$$f \in \mathcal{F}_L^{1,1}, \quad \alpha = \frac{1}{L} \qquad\qquad F(\mathbf{x}^k) - F(\mathbf{x}^\star) \leq \epsilon, \quad \mathcal{O}\left(\frac{1}{\epsilon}\right).$$

Fast proximal gradient method:

$$f \in \mathcal{F}_L^{1,1}, \quad \alpha = \frac{1}{L} \qquad\qquad F(\mathbf{x}^k) - F(\mathbf{x}^\star) \leq \epsilon, \quad \mathcal{O}\left(\frac{1}{\sqrt{\epsilon}}\right).$$

⋆Recall: Composite convex minimization guarantees

Proximal gradient method(ISTA) vs. fast proximal gradient method (FISTA)

Assumptions, step sizes and convergence rates

Proximal gradient method:

$$f \in \mathcal{F}_L^{1,1}, \quad \alpha = \frac{1}{L} \qquad\qquad F(\mathbf{x}^k) - F(\mathbf{x}^\star) \leq \epsilon, \quad \mathcal{O}\left(\frac{1}{\epsilon}\right).$$

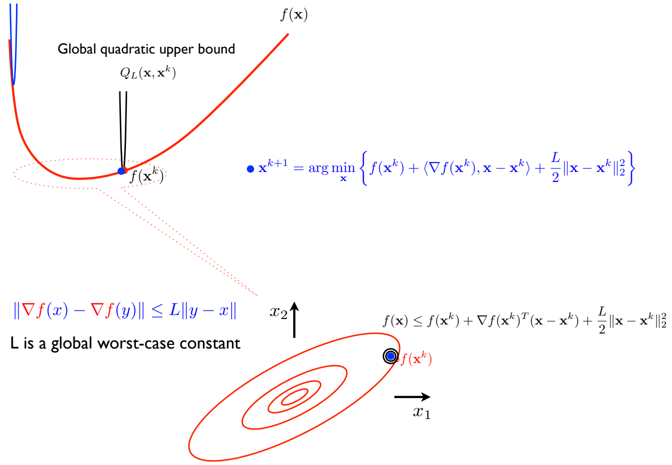
Fast proximal gradient method:

$$f \in \mathcal{F}_L^{1,1}, \quad \alpha = \frac{1}{L} \qquad\qquad F(\mathbf{x}^k) - F(\mathbf{x}^\star) \leq \epsilon, \quad \mathcal{O}\left(\frac{1}{\sqrt{\epsilon}}\right).$$

- ∘ We require α_k to be a function of L.
- ∘ It may not be possible to know exactly the Lipschitz constant. Line-search ?
- ∘ Adaptation to local geometry $\rightarrow$ may lead to larger steps.

*How can we better adapt to the local geometry?

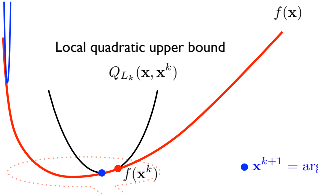
Non-adaptive:



- $\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \left\{ f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k \rangle + \frac{L}{2}\|\mathbf{x} - \mathbf{x}^k\|_2^2 \right\}$

⋆How can we better adapt to the local geometry?

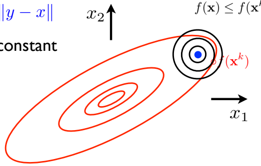
Line-search:



- $\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \left\{ f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k \rangle + \frac{L_k}{2} \|\mathbf{x} - \mathbf{x}^k\|_2^2 \right\}$

$\|\nabla f(x) - \nabla f(y)\| \leq L\|y - x\|$

L is a global worst-case constant

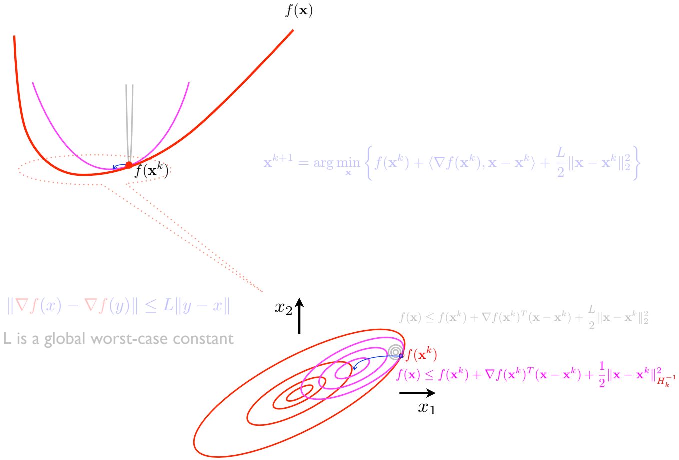


$f(\mathbf{x}) \leq f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T(\mathbf{x} - \mathbf{x}^k) + \frac{L_k}{2}\|\mathbf{x} - \mathbf{x}^k\|_2^2$

applies only locally

*How can we better adapt to the local geometry?

Variable metric:



*The idea of the proximal-Newton method

Assumptions A.2

Assume that $f \in \mathcal{F}^{2,1}_{L,\mu}(\mathbb{R}^p)$ and $g \in \mathcal{F}_{\text{prox}}(\mathbb{R}^p)$.

*Proximal-Newton update

- Similar to classical newton, proximal-newton suggests the following update scheme using second order Taylor series expansion near $\mathbf{x}_k$.

$$\mathbf{x}^{k+1} := \arg\min_{\mathbf{x}\in\mathbb{R}^p} \Big\{ \underbrace{\frac{1}{2}(\mathbf{x}-\mathbf{x}^k)^T \nabla^2 f(\mathbf{x}^k)(\mathbf{x}-\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T(\mathbf{x}-\mathbf{x}^k)}_{\text{2nd-order Taylor expansion near } \mathbf{x}^k} + g(\mathbf{x}) \Big\}. \tag{16}$$

*The proximal-Newton-type algorithm

Proximal-Newton algorithm (PNA)

1. Given $\mathbf{x}^0 \in \mathbb{R}^p$ as a starting point.

2. For $k = 0, 1, \cdots$, perform the following steps:

2.1. Evaluate an SDP matrix $\mathbf{H}_k \approx \nabla^2 f(\mathbf{x}^k)$ and $\nabla f(\mathbf{x}^k)$.

2.2. Compute $\mathbf{d}^k := \mathrm{prox}_{\mathbf{H}_k^{-1} g}\left(\mathbf{x}^k - \mathbf{H}_k^{-1}\nabla f(\mathbf{x}^k)\right) - \mathbf{x}^k$.

2.3. Update $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k \mathbf{d}^k$.

⋆The proximal-Newton-type algorithm

Proximal-Newton algorithm (PNA)

1. Given $\mathbf{x}^0 \in \mathbb{R}^p$ as a starting point.
2. For $k = 0, 1, \cdots$, perform the following steps:

2.1. Evaluate an SDP matrix $\mathbf{H}_k \approx \nabla^2 f(\mathbf{x}^k)$ and $\nabla f(\mathbf{x}^k)$.

2.2. Compute $\mathbf{d}^k := \mathrm{prox}_{\mathbf{H}_k^{-1} g}\left(\mathbf{x}^k - \mathbf{H}_k^{-1}\nabla f(\mathbf{x}^k)\right) - \mathbf{x}^k$.

2.3. Update $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k \mathbf{d}^k$.

Remark

- $\mathbf{H}_k \equiv \nabla^2 f(\mathbf{x}^k) \Longrightarrow$ proximal-Newton algorithm.
- $\mathbf{H}_k \approx \nabla^2 f(\mathbf{x}^k) \Longrightarrow$ proximal-quasi-Newton algorithm.
- A generalized prox-operator: $\mathrm{prox}_{\mathbf{H}_k^{-1} g}\left(\mathbf{x}^k + \mathbf{H}_k^{-1}\nabla f(\mathbf{x}^k)\right)$.

*Convergence analysis

Theorem (Global convergence [11])

Assume generalized-prox subproblem is solved exactly for the algorithm and there exists $\mu > 0$ such that $\mathbf{H}_k \succeq \mu\mathbb{I}$ for all $k \geq 0$. Then;

$$\boxed{\{\mathbf{x}^k\}_{k\geq 0} \text{ globally converges to a solution } \mathbf{x}^\star \text{ of } (2).}$$

*Convergence analysis

Theorem (Global convergence [11])

Assume generalized-prox subproblem is solved exactly for the algorithm and there exists $\mu > 0$ such that $\mathbf{H}_k \succeq \mu\mathbb{I}$ for all $k \geq 0$. Then;

$$\boxed{\{\mathbf{x}^k\}_{k\geq 0} \text{ globally converges to a solution } \mathbf{x}^\star \text{ of } (2).}$$

Theorem (Local convergence [11])

Assume generalized-prox subproblem is solved exactly for the algorithm there exists $0 < \mu \leq L_2 < +\infty$ such that $\mu\mathbb{I} \preceq \mathbf{H}_k \preceq L_2\mathbb{I}$ for all sufficiently large k. Then;

- *If $\mathbf{H}_k \equiv \nabla^2 f(\mathbf{x}^k)$, then $\alpha_k = 1$ for k sufficiently large (full-step).*
- *If $\mathbf{H}_k \equiv \nabla^2 f(\mathbf{x}^k)$, then $\{\mathbf{x}^k\}$ locally converges to $\mathbf{x}^\star$ at a quadratic rate.*
- *If $\mathbf{H}_k$ satisfies the Dennis-Moré condition:*

$$\lim_{k\to+\infty} \frac{\|(\mathbf{H}_k - \nabla^2 f(\mathbf{x}^\star))(\mathbf{x}^{k+1} - \mathbf{x}^k)\|}{\|\mathbf{x}^{k+1} - \mathbf{x}^k\|} = 0, \tag{17}$$

  then $\{\mathbf{x}^k\}$ locally converges to $\mathbf{x}^\star$ at a super linear rate.

⋆How to compute the approximation $\mathbf{H}_k$?

How to update $\mathbf{H}_k$?

Matrix $\mathbf{H}_k$ can be updated by using low-rank updates.

- **BFGS update**: maintain the **Dennis-Moré condition** and $\mathbf{H}_k \succ 0$.

$$\mathbf{H}_{k+1} := \mathbf{H}_k + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{s}_k^T \mathbf{y}_k} - \frac{\mathbf{H}_k \mathbf{s}_k \mathbf{s}_k^T \mathbf{H}_k}{\mathbf{s}_k^T \mathbf{H}_k \mathbf{s}_k}, \quad \mathbf{H}_0 := \gamma \mathbb{I}, \ (\gamma > 0).$$

  where $\mathbf{y}_k := \nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k)$ and $\mathbf{s}_k := \mathbf{x}^{k+1} - \mathbf{x}^k$.

- **Diagonal+Rank-1 [3]**: computing PN direction $\mathbf{d}^k$ is in polynomial time, but it does not maintain the Dennis-Moré condition:

$$\mathbf{H}_k := \mathbf{D}_k + \mathbf{u}_k \mathbf{u}_k^T, \quad \mathbf{u}_k := \frac{\mathbf{s}_k - \mathbf{H}_0 \mathbf{y}_k}{\sqrt{(\mathbf{s}_k - \mathbf{H}_0 \mathbf{y}_k)^T \mathbf{y}_k}},$$

  where $\mathbf{D}_k$ is a positive diagonal matrix.

*Pros and cons

Pros

- Fast local convergence rate (super-linear or quadratic)
- Numerical robustness under the inexactness/noise ([11]).
- Well-suited for problems with many data points but few parameters. For example,

$$F^* := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ \sum_{j=1}^{n} \ell_j(\mathbf{a}_j^T\mathbf{x} + b_j) + g(\mathbf{x}) \right\},$$

  where ℓ_j is twice continuously differentiable and convex, $g \in \mathcal{F}_{\text{prox}}$, $p \ll n$.

*Pros and cons

Pros

- Fast local convergence rate (super-linear or quadratic)
- Numerical robustness under the inexactness/noise ([11]).
- Well-suited for problems with many data points but few parameters. For example,

$$F^* := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ \sum_{j=1}^{n} \ell_j(\mathbf{a}_j^T\mathbf{x} + b_j) + g(\mathbf{x}) \right\},$$

  where ℓ_j is twice continuously differentiable and convex, $g \in \mathcal{F}_{\text{prox}}$, $p \ll n$.

Cons

- Expensive iteration compared to proximal-gradient methods.
- Global convergence rate may be worse than accelerated proximal-gradient methods.
- Requires a good initial point to get fast local convergence.
- Requires strict conditions for global/local convergence analysis.

⋆Example 1: Sparse logistic regression

Problem (Sparse logistic regression)

Given a sample vector $\mathbf{a} \in \mathbb{R}^p$ *and a binary class label vector* $\mathbf{b} \in \{-1, +1\}^n$. *The conditional probability of a label* b *given* $\mathbf{a}$ *is defined as:*

$$\mathbb{P}(b|\mathbf{a}, \mathbf{x}, \mu) = \frac{1}{1 + e^{-b(\mathbf{x}^T\mathbf{a}+\mu)}},$$

where $\mathbf{x} \in \mathbb{R}^p$ *is a weight vector,* μ *is called the intercept.*
Goal: *Find a sparse-weight vector* $\mathbf{x}$ *via the maximum likelihood principle.*

Optimization formulation

$$\min_{\mathbf{x}\in\mathbb{R}^p} \Big\{ \underbrace{\frac{1}{n}\sum_{i=1}^{n} L(b_i(\mathbf{a}_i^T\mathbf{x} + \mu))}_{f(\mathbf{x})} + \underbrace{\rho\|\mathbf{x}\|_1}_{g(\mathbf{x})} \Big\}, \tag{18}$$

where $\mathbf{a}_i$ is the i-th row of data matrix $\mathbf{A}$ in $\mathbb{R}^{n\times p}$, $\rho > 0$ is a regularization parameter, and ℓ is the logistic loss function $\ell(\tau) := \log(1 + e^{-\tau})$.

*Example: Sparse logistic regression

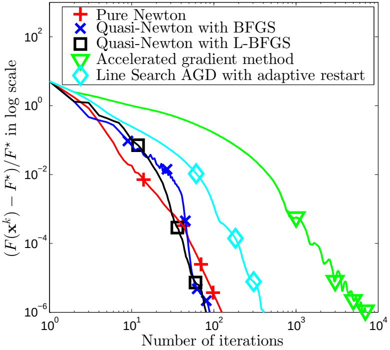
Real data

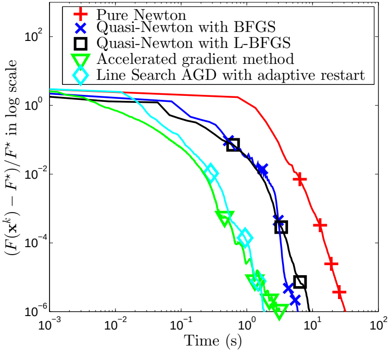
- Real data: w2a with $n = 3470$ data points, $p = 300$ features
- Available at http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html.

Parameters

- Tolerance 10^{-6}.
- L-BFGS memory $m = 50$.
- Ground truth: Get a high accuracy approximation of $\mathbf{x}^\star$ and $f^\star$ by TFOCS with tolerance 10^{-12}.

⋆Example: Sparse logistic regression-Numerical results





*Example 2: ℓ_1-regularized least squares

Problem (ℓ_1-regularized least squares)

Given $\mathbf{A} \in \mathbb{R}^{n \times p}$ *and* $\mathbf{b} \in \mathbb{R}^n$, *solve:*

$$F^\star := \min_{\mathbf{x} \in \mathbb{R}^p} \left\{ F(\mathbf{x}) := \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \rho\|\mathbf{x}\|_1 \right\},$$

where $\rho > 0$ *is a regularization parameter.*

Complexity per iterations

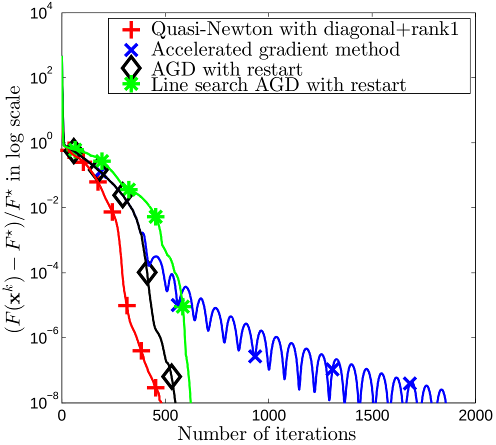
- Evaluating $\nabla f(\mathbf{x}^k) = \mathbf{A}^T(\mathbf{A}\mathbf{x}^k - \mathbf{b})$ requires one $\mathbf{A}\mathbf{x}$ and one $\mathbf{A}^T\mathbf{y}$.
- One soft-thresholding operator $\mathrm{prox}_{\lambda g}(\mathbf{x}) = \mathrm{sign}(\mathbf{x}) \otimes \max\{|\mathbf{x}| - \rho, 0\}$.
- **Optional**: Evaluating $L = \|\mathbf{A}^T\mathbf{A}\|$ (spectral norm) - via **power iterations** (e.g., 20 iterations, each iteration requires one $\mathbf{A}\mathbf{x}$ and one $\mathbf{A}^T\mathbf{y}$).

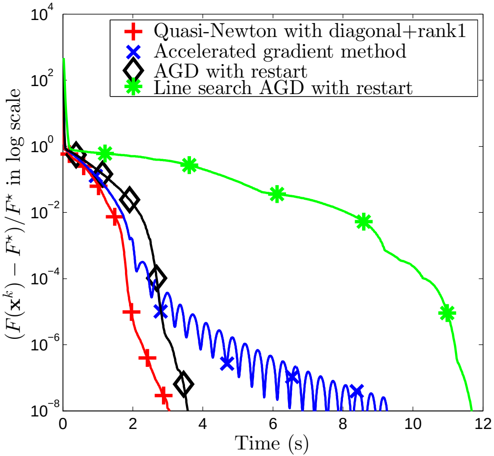
Synthetic data generation

- $\mathbf{A} := \mathrm{randn}(n, p)$ - standard Gaussian $\mathcal{N}(0, \mathbb{I})$.
- $\mathbf{x}^\star$ is a s-sparse vector generated randomly.
- $\mathbf{b} := \mathbf{A}\mathbf{x}^\star + \mathcal{N}(0, 10^{-3})$.

*Example 2: ℓ_1-regularized least squares - Numerical results - Trial 1

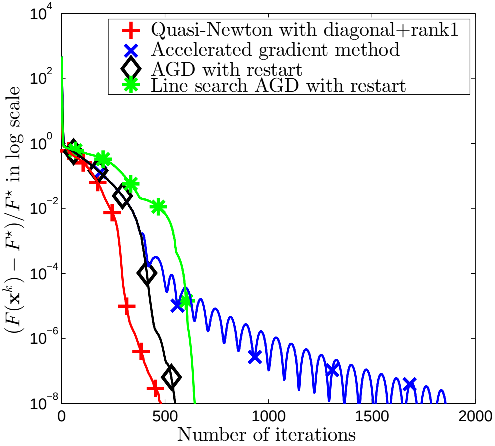
Parameters: $n = 750, p = 2000, s = 200, \rho = 1$

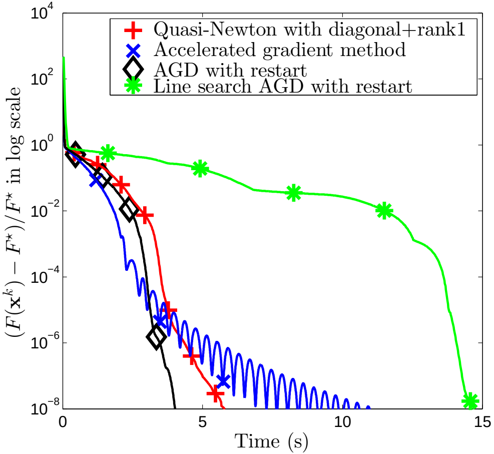




*Example 2: ℓ_1-regularized least squares - Numerical results - Trial 2

Parameters: $n = 750, p = 2000, s = 200, \rho = 1$





References I

[1] Amir Beck.
First-order methods in optimization, volume 25.
SIAM, 2017.
(Cited on pages 36, 37, and 38.)

[2] Amir Beck and Marc Teboulle.
A fast iterative shrinkage-thresholding algorithm for linear inverse problems.
SIAM J. Imaging Sci., 2(1):183–202, 2009.
(Cited on pages 19, 20, and 22.)

[3] S. Becker and M. J. Fadili.
A quasi-newton proximal splitting method.
In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'12, pages 2618–2626, Red Hook, NY, USA, 2012. Curran Associates Inc.
(Cited on page 78.)

[4] E.J. Candès, T. Strohmer, and V. Voroninski.
Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming.
IEE Trans. Signal Processing, 60(5):2422–2432, 2012.
(Cited on pages 54 and 55.)

References II

[5] P. Combettes and J.-C. Pesquet.
Fixed-Point Algorithms for Inverse Problems in Science and Engineering, chapter Proximal Splitting Methods in Signal Processing, pages 185–212.
Springer-Velarg, 2011.
(Cited on page 17.)

[6] Marguerite Frank and Philip Wolfe.
An algorithm for quadratic programming.
Naval Res. Logis. Quart., 3:95–110, 1956.
(Cited on page 45.)

[7] Dan Garber and Elad Hazan.
Faster rates for the frank-wolfe method over strongly-convex sets.
In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, '15, pages 541–549. JMLR.org, 2015.
(Cited on pages 49 and 50.)

[8] Pontus Giselsson and Stephen Boyd.
Monotonicity and restart in fast gradient methods.
In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 5058–5063. IEEE, 2014.
(Cited on page 66.)

References III

[9] E. Hazan and H. Luo.
Variance-reduced and projection-free stochastic optimization.
In *Proc. 33rd Int. Conf. Machine Learning*, 2016.
(Cited on page 60.)

[10] M. Jaggi.
Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization.
JMLR W&CP, 28(1):427–435, 2013.
(Cited on page 45.)

[11] J.D. Lee, Y. Sun, and M.A. Saunders.
Proximal newton-type methods for convex optimization.
Tech. Report., pages 1–25, 2012.
(Cited on pages 76, 77, 79, and 80.)

[12] Ion Necoara.
General convergence analysis of stochastic first order methods for composite optimization, 2020.
(Cited on page 43.)

References IV

[13] Sahand N. Negahban, Pradeep Ravikumar, Martin J. Wainwright, and Bin Yu.
A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers.
Stat. Sci., 27(4):538–557, 2012.
(Cited on page 4.)

[14] Yurii Nesterov.
Introductory lectures on convex optimization: A basic course, volume 87.
Springer Science & Business Media, 2013.
(Cited on page 31.)

[15] N. Parikh and S. Boyd.
Proximal algorithms.
Found. Trends Opt., 1(3):123–231, 2013.
(Cited on page 17.)

[16] Pradeep Ravikumar, Martin J. Wainwright, Garvesh Raskutti, and Bin Yu.
High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence.
Elec. J. Stats., 5:935–980, 2011.
(Cited on page 5.)

References V

[17] Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alex Smola.
Stochastic frank-wolfe methods for nonconvex optimization.
arXiv preprint arXiv:1607.08254, 2016.
(Cited on page 58.)

[18] R Tyrrell Rockafellar.
Monotone operators and the proximal point algorithm.
SIAM journal on control and optimization, 14(5):877–898, 1976.
(Cited on pages 12 and 13.)

[19] Alp Yurtsever, Ya-Ping Hsieh, and Volkan Cevher.
Scalable convex methods for phase retrieval.
In *6th IEEE Intl. Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, 2015.
(Cited on page 55.)