Mathematics of Data: From Theory to Computation

Prof. Volkan Cevher
volkan.cevher@epfl.ch

Lecture 8: Composite convex minimization I

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

EE-556 (Fall 2015)

License Information for Mathematics of Data Slides

Outline

- Today
  1. Composite convex minimization
  2. Proximal operator and computational complexity
  3. Proximal gradient methods
- Next week
  1. Proximal Newton-type methods
  2. Composite self-concordant minimization

Recommended reading material

- A. Beck and M. Tebulle, A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems, SIAM J. Imaging Sciences, 2(1), 183–202, 2009.
- Y. Nesterov, Smooth minimization of non-smooth functions, Math. Program, 103(1), 127–152, 2005.
- Q. Tran-Dinh, A. Kyrillidis and V. Cevher, Composite Self-Concordant Minimization, LIONS-EPFL Tech. Report. `http://arxiv.org/abs/1308.2867`, 2013.
- N. Parikh and S. Boyd, Proximal Algorithms, Foundations and Trends in Optimization, 1(3):123-231, 2014.

Motivation

Motivation

Data analytics problems in various disciplines can often be simplified to nonsmooth **composite convex minimization** problems. To this end, this lecture provides **efficient numerical solution methods** for such problems.

Intriguingly, composite minimization problems are far from generic nonsmooth problems and we can exploit individual function structures to obtain numerical solutions nearly as efficiently as if they are smooth problems.

Composite convex minimization

Problem (Unconstrained composite convex minimization)

$$\boxed{F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\}} \tag{1}$$

Composite convex minimization

Problem (Unconstrained composite convex minimization)

$$\boxed{F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\}} \tag{1}$$

- *f and g are both proper, closed, and convex.*

Composite convex minimization

Problem (Unconstrained composite convex minimization)

$$\boxed{F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\}} \tag{1}$$

- f *and* g *are both proper, closed, and convex.*
- *dom*(F) := *dom*$(f) \cap$ *dom*$(g) \neq \emptyset$ *and* $-\infty < F^\star < +\infty$.

Composite convex minimization

Problem (Unconstrained composite convex minimization)

$$F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\} \tag{1}$$

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

Composite convex minimization

Problem (Unconstrained composite convex minimization)

$$F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\} \tag{1}$$

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

Two remarks

- Nonsmoothness: At least one of the two functions f and g is **nonsmooth**

Composite convex minimization

Problem (Unconstrained composite convex minimization)

$$\boxed{F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\}} \qquad (1)$$

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

Two remarks

- Nonsmoothness: At least one of the two functions f and g is **nonsmooth**
  - General nonsmooth convex optimization methods (e.g., classical subgradient methods, level, or bundle methods) lack efficiency and numerical robustness.
    - Require $\mathcal{O}(\epsilon^{-2})$ iterations to reach a point $\mathbf{x}^\star_\epsilon$ such that $F(\mathbf{x}^\star_\epsilon) - F^\star \leq \epsilon$. Hence, to reach $\mathbf{x}^\star_{0.01}$ such that $F(\mathbf{x}^\star_{0.01}) - F^\star \leq 0.01$, we need $\mathcal{O}(10^4)$ iterations.

Composite convex minimization

Problem (Unconstrained composite convex minimization)

$$F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\} \tag{1}$$

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

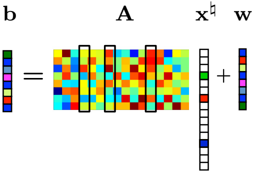
Two remarks

- Nonsmoothness: At least one of the two functions f and g is **nonsmooth**
  - General nonsmooth convex optimization methods (e.g., classical subgradient methods, level, or bundle methods) lack efficiency and numerical robustness.
    - Require $\mathcal{O}(\epsilon^{-2})$ iterations to reach a point $\mathbf{x}^\star_\epsilon$ such that $F(\mathbf{x}^\star_\epsilon) - F^\star \leq \epsilon$. Hence, to reach $\mathbf{x}^\star_{0.01}$ such that $F(\mathbf{x}^\star_{0.01}) - F^\star \leq 0.01$, we need $\mathcal{O}(10^4)$ iterations.
- Generality: it covers a wider range of problems than smooth unconstrained problems. E.g. when handling regularized M-estimation,
  - f is a loss function, a data fidelity, or negative log-likelihood function.
  - g is a regularizer, encouraging structure and/or constraints in the solution.

Example 1: Sparse regression in generalized linear models (GLMs)

Problem (Sparse regression in GLM)

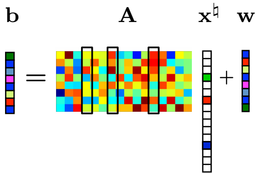
Our goal is to estimate $\mathbf{x}^\natural \in \mathbb{R}^p$ *given* $\{b_i\}_{i=1}^n$ *and* $\{\mathbf{a}_i\}_{i=1}^n$, *knowing that the likelihood function at* y_i *given* $\mathbf{a}_i$ *and* $\mathbf{x}^\natural$ *is given by* $\mathcal{L}(b_i; \langle \mathbf{a}_i, \mathbf{x}^\natural \rangle)$,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



Example 1: 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* $\mathcal{L}(b_i; \langle \mathbf{a}_i, \mathbf{x}^\natural \rangle)$*, and that* $\mathbf{x}^\natural$ *is sparse.*



Optimization formulation

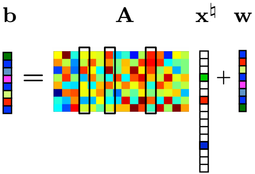
$$\min_{\mathbf{x}\in\mathbb{R}^p} \Bigg\{ \underbrace{-\sum_{i=1}^{n} \log \mathcal{L}(b_i; \langle \mathbf{a}_i, \mathbf{x} \rangle)}_{f(\mathbf{x})} + \underbrace{\rho_n \|\mathbf{x}\|_1}_{g(\mathbf{x})} \Bigg\}$$

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

Example 1: 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* $\mathcal{L}(b_i; \langle \mathbf{a}_i, \mathbf{x}^\natural \rangle)$*, and that* $\mathbf{x}^\natural$ *is sparse.*



Optimization formulation

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

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

Theorem (cf. [4, 5, 6] for details)

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

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

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

Example 2: Image processing

Problem (Imaging denoising/deblurring)

Our goal is to obtain a clean image $\mathbf{x}$ *given "dirty" observations* $\mathbf{b} \in \mathbb{R}^{n\times 1}$ *via* $\mathbf{b} = \mathcal{A}(\mathbf{x}) + \mathbf{w}$*, where* $\mathcal{A}$ *is a linear operator, which, e.g., captures camera blur as well as image subsampling, and* $\mathbf{w}$ *models perturbations, such as Gaussian or Poisson noise.*

Optimization formulation

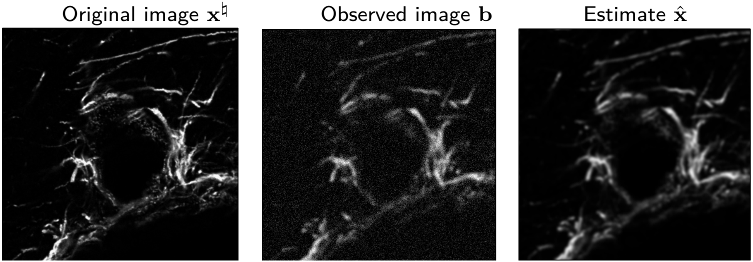
$$\text{Gaussian}: \quad \min_{\mathbf{x}\in\mathbb{R}^{n\times p}} \Big\{ \underbrace{(1/2)\|\mathcal{A}(\mathbf{x}) - \mathbf{b}\|_2^2}_{f(\mathbf{x})} + \underbrace{\rho\|\mathbf{x}\|_{\mathrm{TV}}}_{g(\mathbf{x})} \Big\}$$

$$\text{Poisson}: \quad \min_{\mathbf{x}\in\mathbb{R}^{n\times p}} \Big\{ \underbrace{\frac{1}{n}\sum_{i=1}^{n} \left[\langle \mathbf{a}_i, \mathbf{x}\rangle - b_i \ln\left(\langle \mathbf{a}_i, \mathbf{x}\rangle\right)\right]}_{f(\mathbf{x})} + \underbrace{\rho\|\mathbf{x}\|_{\mathrm{TV}}}_{g(\mathbf{x})} \Big\}$$

where $\rho > 0$ is a regularization parameter and $\|\cdot\|_{\mathrm{TV}}$ is the total variation (TV) norm:

$$\|\mathbf{x}\|_{\mathrm{TV}} := \begin{cases} \sum_{i,j} |\mathbf{x}_{i,j+1} - \mathbf{x}_{i,j}| + |\mathbf{x}_{i+1,j} - \mathbf{x}_{i,j}| & \text{anisotropic case,} \\ \sum_{i,j} \sqrt{|\mathbf{x}_{i,j+1} - \mathbf{x}_{i,j}|^2 + |\mathbf{x}_{i+1,j} - \mathbf{x}_{i,j}|^2} & \text{isotropic case} \end{cases}$$

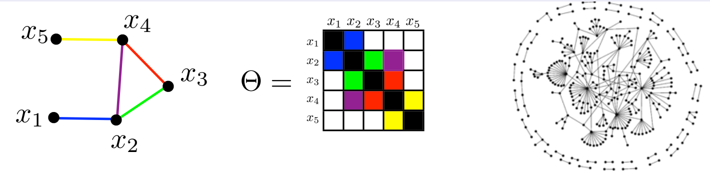
Example 3: Confocal microscopy with camera blur and Poisson observations



Example 4: Sparse inverse covariance estimation

Problem (Graphical model selection)

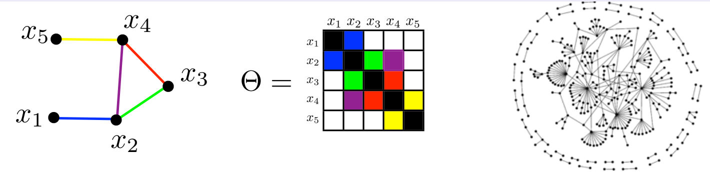
Given a data set $\mathcal{D} := \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$*, where* $\mathbf{x}_i$ *is a Gaussian random variable. Let* Σ *be the covariance matrix corresponding to the graphical model of the Gaussian Markov random field. Our goal is to learn a sparse precision matrix* Θ *(i.e., the inverse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



Example 4: 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 Θ (i.e., the inverse covariance matrix Σ^{-1}) that captures the Markov random field structure..



Optimization formulation

$$\min_{\Theta \succ 0} \Big\{ \underbrace{\mathrm{tr}(\Sigma\Theta) - \log\det(\Theta)}_{f(\mathbf{x})} + \underbrace{\lambda \|\mathrm{vec}(\Theta)\|_1}_{g(\mathbf{x})} \Big\} \quad (2)$$

where $\Theta \succ 0$ means that Θ is symmetric and positive definite and $\lambda > 0$ is a regularization parameter and vec is the vectorization operator.

Outline

- Today
  1. Composite convex minimization
  2. Proximal operator and computational complexity
  3. Proximal gradient methods
- Next week
  1. Proximal Newton-type methods
  2. Composite self-concordant minimization

Question: How do we design algorithms for finding a solution $\mathrm{x}^\star$?

Philosophy

- We cannot immediately design algorithms just based on the original formulation

$$F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\}. \quad (1)$$

- We need intermediate tools to characterize the solutions $\mathbf{x}^\star$ of this problem
- One key tool is called the **optimality condition**

Optimality condition

Theorem (Moreau-Rockafellar's theorem [8])

Let ∂f and ∂g be the subdiffierential of f and g, respectively. If $f, g \in \mathcal{F}(\mathbb{R}^p)$ and dom$(f) \cap$ dom$(g) \neq \emptyset$, then:

$$\boxed{\partial F \equiv \partial(f+g) = \partial f + \partial g.}$$

Note: dom(F) = dom$(f) \cap$ dom(g) and $\partial f(\mathbf{x})$ is defined as (cf., Lecture 2):

$$\partial f := \{\mathbf{w} \in \mathbb{R}^n : f(\mathbf{y}) - f(\mathbf{x}) \geq \mathbf{w}^T(\mathbf{y}-\mathbf{x}),\ \forall \mathbf{y} \in \mathbb{R}^n\},$$

Optimality condition

Theorem (Moreau-Rockafellar's theorem [8])

Let ∂f and ∂g be the subdiffierential of f and g, respectively. If $f, g \in \mathcal{F}(\mathbb{R}^p)$ and dom$(f) \cap$ dom$(g) \neq \emptyset$, then:

$$\boxed{\partial F \equiv \partial(f+g) = \partial f + \partial g.}$$

Note: dom(F) = dom$(f) \cap$ dom(g) and $\partial f(\mathbf{x})$ is defined as (cf., Lecture 2):

$$\partial f := \{\mathbf{w} \in \mathbb{R}^n : f(\mathbf{y}) - f(\mathbf{x}) \geq \mathbf{w}^T(\mathbf{y}-\mathbf{x}),\ \forall \mathbf{y} \in \mathbb{R}^n\},$$

Optimality condition

Generally, the optimality condition for (1) can be written as

$$\boxed{0 \in \partial F(\mathbf{x}^\star) \equiv \partial f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star).} \tag{3}$$

If $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^p)$, then (3) features the gradient of f as opposed to the subdifferential

$$\boxed{0 \in \partial F(\mathbf{x}^\star) \equiv \nabla f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star).} \tag{4}$$

Necessary and sufficient condition

Lemma (Necessary and sufficient condition)

A point $\mathbf{x}^\star \in$ *dom*(F) *is called a* ***globally optimal*** *solution to* (1) *(i.e.,* $F^\star := \min_{\mathbf{x}\in\mathbb{R}^p}\{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\}$

iff

$\mathbf{x}^\star$ *satisfies* (3)*:* $0 \in \partial f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star)$ (*or* (4)*:* $0 \in \nabla f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star)$ *when* $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^p)$)*.*

Necessary and sufficient condition

Lemma (Necessary and sufficient condition)

A point $\mathbf{x}^\star \in dom(F)$ *is called a* ***globally optimal*** *solution to* (1) *(i.e.,* $F^\star := \min_{\mathbf{x}\in\mathbb{R}^p}\{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\}$

iff

$\mathbf{x}^\star$ *satisfies* (3)*:* $0 \in \partial f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star)$ *(or* (4)*:* $0 \in \nabla f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star)$ *when* $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^p)$ *).*

Sketch of the proof.

- $\Rightarrow$: By definition of ∂F:

$$F(\mathbf{x}) - F(\mathbf{x}^\star) \geq \xi^T(\mathbf{x} - \mathbf{x}^\star), \text{ for any } \xi \in \partial F(\mathbf{x}^\star), \ \mathbf{x} \in \mathbb{R}^p.$$

If (3) (or (4)) is satisfied, then $F(\mathbf{x}) - F(\mathbf{x}^\star) \geq 0 \Rightarrow \mathbf{x}^\star$ is a global solution to (1).

- $\Leftarrow$: If $\mathbf{x}^\star$ is a global of (1) then

$$F(\mathbf{x}) \geq F(\mathbf{x}^\star), \forall\ \mathbf{x} \in \text{dom}(F) \quad \Leftrightarrow \quad F(\mathbf{x}) - F(\mathbf{x}^\star) \geq 0^T(\mathbf{x} - \mathbf{x}^\star), \forall \mathbf{x} \in \mathbb{R}^p.$$

This leads to $0 \in \partial F(\mathbf{x}^\star)$ or (3) (or (4)). □

A short detour: Proximal-point operators

Definition (Proximal operator [9])

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

$$\boxed{\mathrm{prox}_g(\mathbf{x}) \equiv \arg\min_{\mathbf{y} \in \mathbb{R}^p} \left\{ g(\mathbf{y}) + \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 \right\}.} \quad (5)$$

A short detour: Proximal-point operators

Definition (Proximal operator [9])

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

$$\boxed{\mathrm{prox}_g(\mathbf{x}) \equiv \arg\min_{\mathbf{y}\in\mathbb{R}^p} \left\{ g(\mathbf{y}) + \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 \right\}.} \tag{5}$$

Numerical efficiency: Why do we need proximal operator?

For problem (1):

- Many well-known convex functions g, we can compute $\mathrm{prox}_g(\mathbf{x})$ analytically or very efficiently.
- If $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^p)$, and $\mathrm{prox}_g(\mathbf{x})$ is **cheap** to compute, then solving (1) is as **efficient** as solving $\boxed{\min_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x})}$ in terms of complexity.
- If $\mathrm{prox}_f(\mathbf{x})$ and $\mathrm{prox}_g(\mathbf{x})$ are both **cheap** to compute, then *convex splitting* (1) is also efficient (cf., Lecture 8).

A short detour: Basic properties of prox-operator

Property (Basic properties of prox-operator)

1. $\mathrm{prox}_g(\mathbf{x})$ *is well-defined and single-valued (i.e., the prox-operator* (5) *has a unique solution since* $g(\cdot) + (1/2)\|\cdot - \mathbf{x}\|_2^2$ *is strongly convex).*
2. *Optimality condition:*

$$\mathbf{x} \in \mathrm{prox}_g(\mathbf{x}) + \partial g(\mathrm{prox}_g(\mathbf{x})),\ \mathbf{x} \in \mathbb{R}^p.$$

3. $\mathbf{x}^\star$ *is a fixed point of* $\mathrm{prox}_g(\cdot)$*:*

$$0 \in \partial g(\mathbf{x}^\star) \quad \Leftrightarrow \quad \mathbf{x}^\star = \mathrm{prox}_g(\mathbf{x}^\star).$$

4. *Nonexpansiveness:*

$$\|\mathrm{prox}_g(\mathbf{x}) - \mathrm{prox}_g(\tilde{\mathbf{x}})\|_2 \le \|\mathbf{x} - \tilde{\mathbf{x}}\|_2,\ \ \forall \mathbf{x}, \tilde{\mathbf{x}} \in \mathbb{R}^p.$$

Fixed-point characterization

Optimality condition as fixed-point formulation

The optimality condition (3): $0 \in \partial f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star)$ is equivalent to

$$\mathbf{x}^\star \in \mathrm{prox}_{\lambda g}\left(\mathbf{x}^\star - \lambda \partial f(\mathbf{x}^\star)\right) := \mathcal{T}_\lambda(\mathbf{x}^\star), \quad \text{for any } \lambda > 0. \tag{6}$$

The optimality condition (4): $0 \in \nabla f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star)$ is equivalent to

$$\mathbf{x}^\star = \mathrm{prox}_{\lambda g}\left(\mathbf{x}^\star - \lambda \nabla f(\mathbf{x}^\star)\right) := \mathcal{U}_\lambda(\mathbf{x}^\star), \quad \text{for any } \lambda > 0. \tag{7}$$

$\mathcal{T}_\lambda$ is a set-valued operator and $\mathcal{U}_\lambda$ is a single-valued operator.

Fixed-point characterization

Optimality condition as fixed-point formulation

The optimality condition (3): $0 \in \partial f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star)$ is equivalent to

$$\mathbf{x}^\star \in \mathrm{prox}_{\lambda g}\left(\mathbf{x}^\star - \lambda \partial f(\mathbf{x}^\star)\right) := \mathcal{T}_\lambda(\mathbf{x}^\star), \quad \text{for any } \lambda > 0. \tag{6}$$

The optimality condition (4): $0 \in \nabla f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star)$ is equivalent to

$$\mathbf{x}^\star = \mathrm{prox}_{\lambda g}\left(\mathbf{x}^\star - \lambda \nabla f(\mathbf{x}^\star)\right) := \mathcal{U}_\lambda(\mathbf{x}^\star), \quad \text{for any } \lambda > 0. \tag{7}$$

$\mathcal{T}_\lambda$ is a set-valued operator and $\mathcal{U}_\lambda$ is a single-valued operator.

Proof.

We prove (7) ((6) is done similarly). (4) implies

$$0 \in \nabla f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star) \Leftrightarrow \mathbf{x}^\star - \lambda \nabla f(\mathbf{x}^\star) \in \mathbf{x}^\star + \lambda \partial g(\mathbf{x}^\star) \equiv (\mathbb{I} + \lambda \partial g)(\mathbf{x}^\star).$$

Using the basic property 2 of $\mathrm{prox}_{\lambda g}$, we have

$$\mathbf{x}^\star \in \mathrm{prox}_{\lambda g}(\mathbf{x}^\star - \lambda \nabla f(\mathbf{x}^\star)).$$

Since $\mathrm{prox}_{\lambda g}$ and ∇f are single-valued, we obtain (7). □

Choices of solution methods

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

$f \in \mathcal{F}_L^{1,1}(\mathbb{R}^p), g \in \mathcal{F}_{\text{prox}}(\mathbb{R}^p)$

[Fast] proximal gradient method

$f \in \mathcal{F}_2(\text{dom}(f)), g \in \mathcal{F}_{\text{prox}}(\mathbb{R}^p)$

Proximal gradient/Newton

f is smoothable, $g \in \mathcal{F}_{\text{prox}}(\mathbb{R}^p)$

Smoothing techniques

$f \in \mathcal{F}_{\text{prox}}(\mathbb{R}^p), g \in \mathcal{F}_{\text{prox}}(\mathbb{R}^p)$

Splitting methods/ADMM

- $\mathcal{F}_L^{1,1}$ and $\mathcal{F}_2$ are the class of convex functions with Lipschitz gradient and self-concordant, respectively.
- $\mathcal{F}_{\text{prox}}$ is the class of convex functions with proximity operator (defined in the next slides).
- "smoothable" is defined in the next lectures.

Solution methods

Composite convex minimization

$$F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x}) \right\}. \tag{1}$$

Solution methods

Composite convex minimization

$$F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x}) \right\}. \quad (1)$$

Choice of numerical solution methods

- **Solve** (1) = 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 $\mathbf{x} \in \mathbb{R}^p$.

Solution methods

Composite convex minimization

$$F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \Big\{ F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x}) \Big\}. \qquad (1)$$

Choice of numerical solution methods

- **Solve** (1) = Find $\mathbf{x}^k \in \mathbb{R}^p$ such that

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

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 $\mathbf{x} \in \mathbb{R}^p$.

Using different oracle leads to different types of algorithms

Tractable prox-operators

Processing non-smooth terms in (1)

- We handle the nonsmooth term g in (1) using the proximal mapping principle.
- Computing proximal operator prox_g of a general convex function g

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

  can be computationally demanding.
- If we can efficiently compute prox_F, we can use the **proximal-point algorithm** (PPA) [3, 9] to solve (1). Unfortunately, PPA is hardly used in practice to solve (8) since computing $\mathrm{prox}_{\lambda F}(\cdot)$ can be as almost hard as solving (1).

Tractable prox-operators

Processing non-smooth terms in (1)

- We handle the nonsmooth term g in (1) using the proximal mapping principle.
- Computing proximal operator prox_g of a general convex function g

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

  can be computationally demanding.
- If we can efficiently compute prox_F, we can use the **proximal-point algorithm** (PPA) [3, 9] to solve (1). Unfortunately, PPA is hardly used in practice to solve (8) since computing $\mathrm{prox}_{\lambda F}(\cdot)$ can be as almost hard as solving (1).

Definition (Tractable proximity)

Given $g \in \mathcal{F}(\mathbb{R}^p)$. We say that g is proximally tractable if prox_g defined by (5) can be computed **efficiently**.

- "**efficiently**" $=$ {closed form solution, low-cost computation, polynomial time}.
- We denote $\mathcal{F}_{\mathrm{prox}}(\mathbb{R}^p)$ the class of proximally tractable convex functions.

*The proximal-point method

Problem (Unconstrained convex minimization)

*Given $F \in \mathcal{F}(\mathbb{R}^p)$, our **goal** is to solve*

$$F^{\star} := \min_{\mathbf{x} \in \mathbb{R}^p} F(\mathbf{x}). \tag{8}$$

⋆The proximal-point method

Problem (Unconstrained convex minimization)

*Given $F \in \mathcal{F}(\mathbb{R}^p)$, our **goal** is to solve*

$$F^\star := \min_{\mathbf{x} \in \mathbb{R}^p} F(\mathbf{x}). \quad (8)$$

Proximal-point algorithm (PPA):

1. Choose $\mathbf{x}^0 \in \mathbb{R}^p$ and a positive sequence $\{\lambda_k\}_{k \geq 0} \subset \mathbb{R}_{++}$.
2. For $k = 0, 1, \cdots$, update:

$$\mathbf{x}^{k+1} := \mathrm{prox}_{\lambda_k F}(\mathbf{x}^k)$$

*The proximal-point method

Problem (Unconstrained convex minimization)

Given $F \in \mathcal{F}(\mathbb{R}^p)$, our ***goal*** *is to solve*

$$\boxed{F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} F(\mathbf{x}).} \tag{8}$$

Proximal-point algorithm (PPA):

1. Choose $\mathbf{x}^0 \in \mathbb{R}^p$ and a positive sequence $\{\lambda_k\}_{k\geq 0} \subset \mathbb{R}_{++}$.
2. For $k = 0, 1, \cdots,$ update:

$$\boxed{\mathbf{x}^{k+1} := \mathrm{prox}_{\lambda_k F}(\mathbf{x}^k)}$$

Theorem (Convergence [3])

Let $\{\mathbf{x}^k\}_{k\geq 0}$ be a sequence generated by PPA. If $0 < \lambda_k < +\infty$ then

$$\boxed{F(\mathbf{x}^k) - F^\star \leq \frac{\|\mathbf{x}^0 - \mathbf{x}^\star\|_2^2}{2\sum_{j=0}^k \lambda_j}}, \quad \forall \mathbf{x}^\star \in \mathcal{S}^\star,\ k \geq 0.$$

If $\lambda_k \geq \lambda > 0$, then the convergence rate of PPA is $\mathcal{O}(1/k)$.

Tractable prox-operators

Example

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

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

- For smooth functions, we can computer the prox-operator via basic algebra. For instance, $g(\mathbf{x}) := (1/2)\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$, one has

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

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

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

  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

$$\boxed{\mathrm{prox}_{\lambda g}(\mathbf{x}) := \arg\min_{\mathbf{y} \in \mathbb{R}^p} \left\{ \frac{1}{2}\|\mathbf{A}\mathbf{y} - \mathbf{b}\|_2^2 + \frac{1}{2\lambda}\|\mathbf{y} - \mathbf{x}\|_2^2 \right\}.} \tag{9}$$

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

$$\boxed{\operatorname{prox}_{\lambda g}(\mathbf{x}) := \arg\min_{\mathbf{y}\in\mathbb{R}^p} \left\{ \frac{1}{2}\|\mathbf{A}\mathbf{y} - \mathbf{b}\|_2^2 + \frac{1}{2\lambda}\|\mathbf{y} - \mathbf{x}\|_2^2 \right\}.} \tag{9}$$

How to compute $\operatorname{prox}_{\lambda g}(\mathbf{x})$?

The optimality condition implies that the solution of (9) should satisfy the following linear system: $\mathbf{A}^T(\mathbf{A}\mathbf{y} - \mathbf{b}) + \lambda^{-1}(\mathbf{y} - \mathbf{x}) = 0$. As a result, we obtain

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

- When $\mathbf{A}^T\mathbf{A}$ is efficiently diagonalizable (e.g., $\mathbf{U}^T\mathbf{A}^T\mathbf{A}\mathbf{U} := \Lambda$, where $\mathbf{U}$ is a unitary matrix and Λ is a diagonal matrix) then $\operatorname{prox}_{\lambda g}(\mathbf{x})$ can be cheap $\operatorname{prox}_{\lambda g}(\mathbf{x}) = \mathbf{U}\left(\mathbb{I} + \lambda\Lambda\right)^{-1}\mathbf{U}^T(\mathbf{x} + \lambda\mathbf{A}\mathbf{b})$.
  - Matrices $\mathbf{A}$ such as TV operator with periodic boundary conditions, index subsampling operators (e.g., as in matrix completion), and circulant matrices (e.g., typical image blur operators) are efficiently diagonalizable with the Fast Fourier transform $\mathbf{U}$.
- If $\mathbf{A}\mathbf{A}^T$ is diagonalizable (e.g., a tight frame $\mathbf{A}$), then we can use the identity
$$(\mathbb{I} + \lambda\mathbf{A}^T\mathbf{A})^{-1} = \mathbb{I} - \mathbf{A}^T(\lambda^{-1}\mathbb{I} + \mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A}.$$

A non-exhaustive list of proximal tractability functions

Name	Function	Proximal operator	Complexity
ℓ_1-norm	$f(\mathbf{x}) := \\|\mathbf{x}\\|_1$	$\mathrm{prox}_{\lambda f}(\mathbf{x}) = \mathrm{sign}(\mathbf{x}) \otimes [\|\mathbf{x}\| - \lambda]_+$	$\mathcal{O}(p)$
ℓ_2-norm	$f(\mathbf{x}) := \\|\mathbf{x}\\|_2$	$\mathrm{prox}_{\lambda f}(\mathbf{x}) = [1 - \lambda/\\|\mathbf{x}\\|_2]_+ \mathbf{x}$	$\mathcal{O}(p)$
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	$f(\mathbf{x}) := (1/2)\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{q}^T\mathbf{x}$	$\mathrm{prox}_{\lambda f}(\mathbf{x}) = (\lambda\mathbb{I} + \mathbf{Q})^{-1}\mathbf{x}$	$\mathcal{O}(p\log p) \to \mathcal{O}(p^3)$
Square ℓ_2-norm	$f(\mathbf{x}) := (1/2)\\|\mathbf{x}\\|_2^2$	$\mathrm{prox}_{\lambda f}(\mathbf{x}) = (1/(1+\lambda))\mathbf{x}$	$\mathcal{O}(p)$
log-function	$f(\mathbf{x}) := -\log(x)$	$\mathrm{prox}_{\lambda f}(x) = ((x^2 + 4\lambda)^{1/2} + x)/2$	$\mathcal{O}(1)$
$\log\det$-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 $\mathbf{X}$	$\mathcal{O}(p^3)$

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 [2, 7].

Outline

- Today
  1. Composite convex minimization
  2. Proximal operator and computational complexity
  3. Proximal gradient methods
- Next week
  1. Proximal Newton-type methods
  2. Composite self-concordant minimization

Proximal-gradient method: A quadratic majorization perspective

Definition (Moreau proximal operator [?, 9])

Let $g \in \mathcal{F}(\mathbb{R}^p)$. The proximal operator (or prox-operator) of g is defined as:

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

Proximal-gradient method: A quadratic majorization perspective

Definition (Moreau proximal operator [?, 9])

Let $g \in \mathcal{F}(\mathbb{R}^p)$. The proximal operator (or prox-operator) of g is defined as:

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

Quadratic upper bound for f

For $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^p)$, we have, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^p$

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

Proximal-gradient method: A quadratic majorization perspective

Definition (Moreau proximal operator [?, 9])

Let $g \in \mathcal{F}(\mathbb{R}^p)$. The proximal operator (or prox-operator) of g is defined as:

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

Quadratic upper bound for f

For $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^p)$, we have, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^p$

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

Quadratic *majorizer* for $f + g$ [?]

Of course, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^p$,

$$f(\mathbf{x}) \leq Q_L(\mathbf{x}, \mathbf{y}) \quad \Rightarrow \quad f(\mathbf{x}) + g(\mathbf{x}) \leq Q_L(\mathbf{x}, \mathbf{y}) + g(\mathbf{x}) := P_L(\mathbf{x}, \mathbf{y})$$

Proximal-gradient method: A quadratic majorization perspective

Definition (Moreau proximal operator [?, 9])

Let $g \in \mathcal{F}(\mathbb{R}^p)$. The proximal operator (or prox-operator) of g is defined as:

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

Quadratic upper bound for f

For $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^p)$, we have, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^p$

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

Quadratic *majorizer* for $f + g$ [?]

Of course, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^p$,

$$f(\mathbf{x}) \leq Q_L(\mathbf{x},\mathbf{y}) \quad \Rightarrow \quad f(\mathbf{x}) + g(\mathbf{x}) \leq Q_L(\mathbf{x},\mathbf{y}) + g(\mathbf{x}) := P_L(\mathbf{x},\mathbf{y})$$

Proximal-gradient from the majorize-minimize perspective [?]

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} P_L(\mathbf{x},\mathbf{x}^k)$$

Proximal-gradient method: A quadratic majorization perspective

Definition (Moreau proximal operator [?, 9])

Let $g \in \mathcal{F}(\mathbb{R}^p)$. The proximal operator (or prox-operator) of g is defined as:

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

Quadratic upper bound for f

For $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^p)$, we have, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^p$

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

Quadratic *majorizer* for $f + g$ [?]

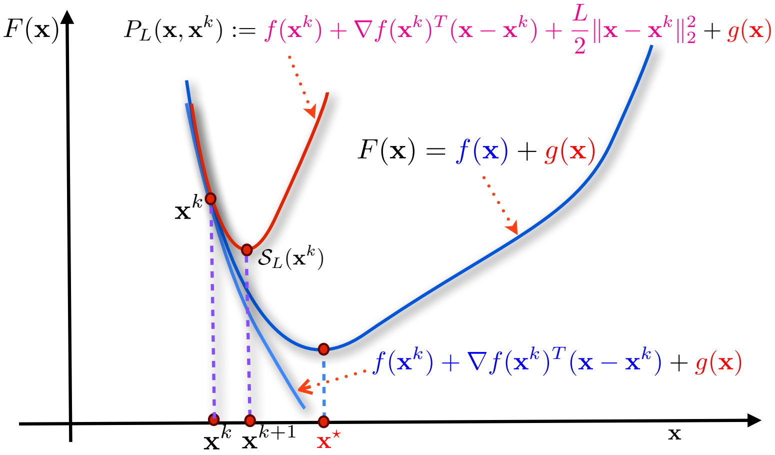
Of course, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^p$,

$$f(\mathbf{x}) \le Q_L(\mathbf{x}, \mathbf{y}) \quad \Rightarrow \quad f(\mathbf{x}) + g(\mathbf{x}) \le Q_L(\mathbf{x}, \mathbf{y}) + g(\mathbf{x}) := P_L(\mathbf{x}, \mathbf{y})$$

Proximal-gradient from the majorize-minimize perspective [?]

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} P_L(\mathbf{x}, \mathbf{x}^k) = \mathrm{prox}_{g/L}(\mathbf{x} - \nabla f(\mathbf{x}^k)/L)$$

Geometric illustration



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 [1])

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 $R_0 := \max_{\mathbf{x}^\star \in \mathcal{S}^\star} \|\mathbf{x}^0 - \mathbf{x}^\star\|_2$.

- A line-search procedure can be used to estimate L_k for L based on $(0 < c \leq 1)$:

$$f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k) - \frac{c}{2L_k}\|\nabla f(\mathbf{x}^k)\|^2.$$

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 $t_0 := 1$. **3.** For $k = 0, 1, \ldots$, generate two sequences $\{\mathbf{x}^k\}_{k\geq 0}$ and $\{\mathbf{y}^k\}_{k\geq 0}$ as: $$\begin{cases} \mathbf{x}^{k+1} & := \mathrm{prox}_{\alpha g}\left(\mathbf{y}^k - \alpha \nabla f(\mathbf{y}^k)\right), \\ t_{k+1} & := (1 + \sqrt{4t_k^2 + 1})/2, \\ \mathbf{y}^{k+1} & := \mathbf{x}^{k+1} + \frac{t_k - 1}{t_{k+1}}(\mathbf{x}^{k+1} - \mathbf{x}^k). \end{cases}$$ where $\alpha := L^{-1}$.

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!.

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 $t_0 := 1$.
3. For $k = 0, 1, \ldots,$ generate two sequences $\{\mathbf{x}^k\}_{k\geq 0}$ and $\{\mathbf{y}^k\}_{k\geq 0}$ as:

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

where $\alpha := L^{-1}$.

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:*

$$F^\star := \min_{\mathbf{x} \in \mathbb{R}^p} \left\{ F(\mathbf{x}) := \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_1 \right\}, \tag{10}$$

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** (e.g., 20 iterations, each iteration requires one $\mathbf{A}\mathbf{x}$ and one $\mathbf{A}^T\mathbf{y}$).

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:*

$$F^\star := \min_{\mathbf{x} \in \mathbb{R}^p} \left\{ F(\mathbf{x}) := \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_1 \right\}, \tag{10}$$

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

Complexity per iterations

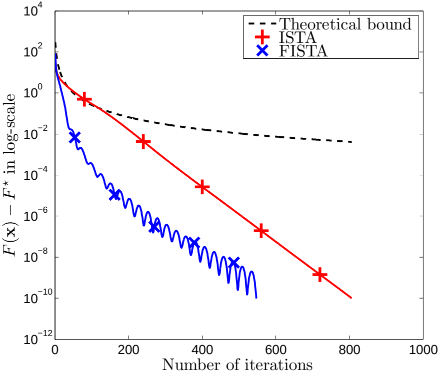
- Evaluating $\nabla f(\mathbf{x}^k) = \mathbf{A}^T(\mathbf{A}\mathbf{x}^k - \mathbf{b})$ requires one $\mathbf{A}\mathbf{x}$ and one $\mathbf{A}^T\mathbf{y}$.
- One soft-thresholding operator $\mathrm{prox}_{\lambda g}(\mathbf{x}) = \mathrm{sign}(\mathbf{x}) \otimes \max\{|\mathbf{x}| - \lambda, 0\}$.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

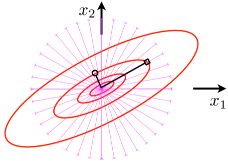
Synthetic data generation

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

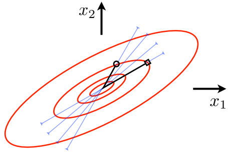
Example 1: Theoretical bounds vs practical performance

- **(Theoretical bounds)** **FISTA** $:= \frac{2L_f R_0^2}{(k+2)^2}$ and **ISTA** $:= \frac{L_f R_0^2}{2(k+2)}$.





descent directions



restricted descent directions

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

Adaptive Restart

It is possible the preserve $\mathcal{O}(1/k^2)$ 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 $\theta_0 = \theta_{-1} = 1$
3. For $k = 0, 1, \ldots$, generate two sequences $\{\mathbf{x}^k\}_{k\geq 0}$ and $\{\mathbf{y}^k\}_{k\geq 0}$ as:

$$\left\{\begin{array}{l} \mathbf{y}^k := \mathbf{x}^k + \theta_k(\theta_{k-1}^{-1} - 1)(\mathbf{x}^k - \mathbf{x}^{k-1}) \\ \mathbf{x}^{k+1} := \mathrm{prox}_{\lambda g}\left(\mathbf{y}^k - \lambda\nabla f(\mathbf{y}^k)\right), \\ \textbf{if } \text{restart test holds} \\ \qquad \mathbf{y}^k = \mathbf{x}^k \\ \qquad \mathbf{x}^{k+1} := \mathrm{prox}_{\lambda g}\left(\mathbf{y}^k - \lambda\nabla f(\mathbf{y}^k)\right) \end{array}\right. \tag{11}$$

where $\lambda := L_f^{-1}$.

θ_k is chosen so that it satisfies

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

Adaptive Restart: Guarantee

Theorem (Global complexity [10])

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

$$F(\mathbf{x}^k) - F^\star \leq \frac{2L_f}{(k+2)^2}\left(R_0^2 + \sum_{k_i \leq k}\left(\|\mathbf{x}^* - \mathbf{x}^{k_i}\|_2^2 - \|\mathbf{x}^* - \mathbf{z}^{k_i}\|_2^2\right)\right) \ \forall k \geq 0. \quad (12)$$

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* $k_i, i = 1...$ *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: $\langle (L_f(\mathbf{y}^k - \mathbf{x}^{k+1}), \mathbf{x}^{k+1} - \mathbf{x}^k\rangle > 0$

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:*

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

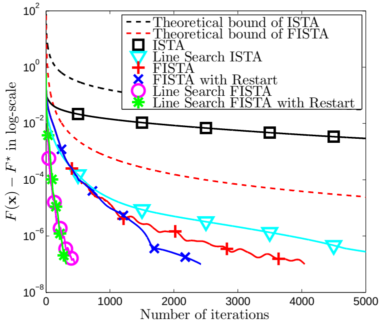
Real data

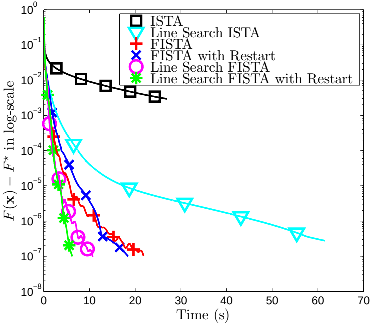
- Real data: `w8a` with $n = 49'749$ data points, $p = 300$ features
- Available at `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html`.

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

Strong convexity case: 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: $$\mathbf{x}^{k+1} := \mathrm{prox}_{\alpha_k g}\Big(\mathbf{x}^k - \alpha_k \nabla f(\mathbf{x}^k)\Big),$$ where $\alpha_k := 2/(L_f + \mu)$ 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 two sequences $\{\mathbf{x}^k\}_{k\geq 0}$ and $\{\mathbf{y}^k\}_{k\geq 0}$ as: $$\begin{cases} \mathbf{x}^{k+1} := \mathrm{prox}_{\alpha_k g}\Big(\mathbf{y}^k - \alpha_k \nabla f(\mathbf{y}^k)\Big), \\ \mathbf{y}^{k+1} := \mathbf{x}^{k+1} + \Big(\frac{\sqrt{c_f}-1}{\sqrt{c_f}+1}\Big)(\mathbf{x}^{k+1} - \mathbf{x}^k), \end{cases}$$ where $\alpha_k := L_f^{-1}$ is the optimal step-size.

Strong convexity case: 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: $c_f := \frac{L_f}{\mu} \geq 0$.

Strong convexity case: 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**$_\mu$ [7])

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

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

Strong convexity case: 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**$_\mu$ [7])

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

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**$_\mu$ [7])

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

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

A practical issue

Stopping criterion

Fact: If $\mathcal{PG}_{\mathcal{L}}(\mathbf{x}^\star) = 0$, then $\mathbf{x}^\star$ is optimal to (1), where

$$\mathcal{PG}_{\mathcal{L}}(\mathbf{x}) = L\left(\mathbf{x} - \mathrm{prox}_{(1/L)g}\left(\mathbf{x} - (1/L)\nabla f(\mathbf{x})\right)\right).$$

Stopping criterion: (relative solution change)

$$L_k\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_2 \leq \varepsilon \max\{L_0\|\mathbf{x}^1 - \mathbf{x}^0\|_2, 1\},$$

where ε is a given tolerance.

Summary of the worst-case complexities

Software

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

http://cvxr.com/tfocs/

Comparison with gradient scheme $(F(\mathbf{x}^k) - F^\star \leq \varepsilon)$

Complexity	Proximal-gradient scheme	Fast proximal-gradient scheme
Complexity $[\mu = 0]$	$\mathcal{O}\left(R_0^2(L_f/\varepsilon)\right)$	$\mathcal{O}\left(R_0\sqrt{L_f/\varepsilon}\right)$
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 $\kappa = L_f/\mu_f$ is the condition number.

Summary of the worst-case complexities

Software

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

http://cvxr.com/tfocs/

Comparison with gradient scheme $\left(F(\mathbf{x}^k) - F^\star \leq \varepsilon\right)$

Complexity	Proximal-gradient scheme	Fast proximal-gradient scheme
Complexity $[\mu = 0]$	$\mathcal{O}\left(R_0^2(L_f/\varepsilon)\right)$	$\mathcal{O}\left(R_0\sqrt{L_f/\varepsilon}\right)$
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 $\kappa = L_f/\mu_f$ is the condition number.

Need alternatives when

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

Examples

Example (Sparse graphical model selection)

$$\min_{\Theta \succ 0} \Big\{ \underbrace{\mathrm{tr}(\Sigma\Theta) - \log\det(\Theta)}_{f(\mathbf{x})} + \underbrace{\rho\|\mathrm{vec}(\Theta)\|_1}_{g(\mathbf{x})} \Big\}$$

where $\Theta \succ 0$ means that Θ is symmetric and positive definite, and $\rho > 0$ is a regularization parameter and vec is the vectorization operator.

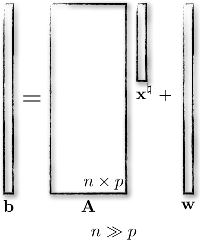
- Computing the gradient is expensive: $\nabla f(\Theta) = \Theta^{-1}$.
- $f \in \mathcal{F}_2$ is self-concordant. However, if $\alpha\mathbf{I} \preceq \Theta \preceq \beta\mathbf{I}$, then $f \in \mathcal{F}^{2,1}_{L,\mu}$ with $L = \sqrt{p}/\alpha^2$ and $\mu = (\beta^2\sqrt{p})^{-1}$.

Example (ℓ_1-regularized Lasso)

$$\min_{\mathbf{x}} \underbrace{\frac{1}{2}\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2}_{f(\mathbf{x})} + \underbrace{\rho\|\mathbf{x}\|_1}_{g(\mathbf{x})}$$

where $n \gg p$, $\mathbf{A} \in \mathbb{R}^{n \times p}$ is a full column-rank matrix, and $\rho > 0$ is a regularization parameter.

- $f \in \mathcal{F}^{2,1}_{L,\mu}$ and computing the gradient is $\mathcal{O}(n)$.



References I

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

[2] Patrick L. Combettes and Jean-Christophe Pesquet.
Proximal splitting methods in signal processing.
In Heinz Bauschke, Regina S. Burachik, Patrick Combettes, Veit Elser, D. Russell Luck, and Henry Wolkowicz, editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, chapter 10. Springer, New York, NY, 2011.

[3] Osman Güler.
On the convergence of the proximal point algorithm for convex minimization.
SIAM J. Control Optim., 29(2):403–419, March 1991.

[4] Yen-Huan Li, Ya-Ping Hsieh, and Volkan Cevher.
A geometric view on constrained M-estimators.
EPFL-REPORT-205083, École Polytechnique Fédérale de Lausanne, 2015.

[5] Yen-Huan Li, Jonathan Scarlett, Pradeep Ravikumar, and Volkan Cevher.
Sparsistency of ℓ_1-regularized M-estimators.
In *Proc. 18th Inf. Conf. Artificial Intelligence and Statistics*, pages 644–652, 2015.

[6] 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.

[7] Yurii Nesterov.
Introductory Lectures on Convex Optimization.
Kluwer, Boston, MA, 2004.

[8] R. T. Rockafellar.
On the maximal monotonicity of subdifferential mappings.
Pac. J. Math., 33(1):209–216, 1970.

[1] A. Beck and M. Teboulle.
A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems.
SIAM J. Imaging Sciences, 2(1):183–202, 2009.

References II

[2] P. Combettes and Pesquet J.-C.
Signal recovery by proximal forward-backward splitting.
In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer-Verlag, 2011.

[3] O. Güler.
On the convergence of the proximal point algorithm for convex minimization.
SIAM J. Control Optim., 29(2):403–419, 1991.

[4] Y. Nesterov.
Introductory lectures on convex optimization: a basic course, volume 87 of *Applied Optimization*.
Kluwer Academic Publishers, 2004.

[5] Y. Nesterov.
Smooth minimization of non-smooth functions.
Math. Program., 103(1):127–152, 2005.

[6] Y. Nesterov and A. Nemirovski.
Interior-point Polynomial Algorithms in Convex Programming.
Society for Industrial Mathematics, 1994.

[7] N. Parikh and S. Boyd.
Proximal algorithms.
Foundations and Trends in Optimization, 1(3):123–231, 2013.

[8] R. T. Rockafellar.
Convex Analysis, volume 28 of *Princeton Mathematics Series*.
Princeton University Press, 1970.

[9] R.T. Rockafellar.
Monotone operators and the proximal point algorithm.
SIAM Journal on Control and Optimization, 14:877–898, 1976.

[10] P. Giselsson and S. Boyd.
Monotonicity and Restart in Fast Gradient Methods.
IEEE 53rd Annual Conference on Decision and Control, 5058–5063, 2014.

References III

[11] Q. Tran-Dinh, A. Kyrillidis, and V. Cevher.
Composite self-concordant minimization.
Tech. report, Lab. for Information and Inference Systems (LIONS), EPFL, Switzerland, CH-1015 Lausanne, Switzerland, January 2013.