# Mathematics of Data: From Theory to Computation

Prof. Volkan Cevher
*volkan.cevher@epfl.ch*

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

**EE-556** (Fall 2014)

# License Information for Mathematics of Data Slides

# Outline

- Today
    1. Composite convex minimization
    2. Proximal operator and computational complexity
    3. Proximal gradient methods
    4. Composite self-concordant minimization
    5. Smoothing for nonsmooth composite convex minimization
- Next week
    1. Nonsmooth constrained optimization

## 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 (**Mathematical formulation**)

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

*where $f$ and $g$ are both proper, closed and convex. Note that* (1) *is unconstrained.*

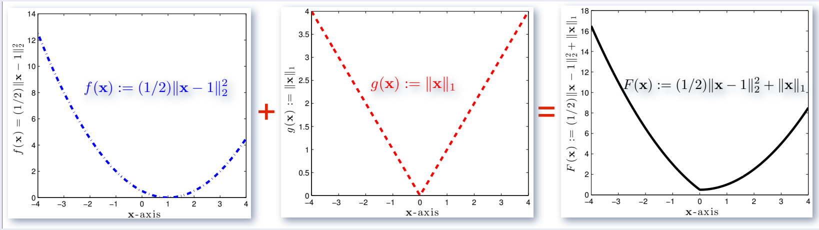
# Composite convex minimization

## Problem (**Mathematical formulation**)

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

*where $f$ and $g$ are both proper, closed and convex. Note that (1) is unconstrained.*

## A composite function illustration



## Composite convex minimization

### Problem (Mathematical formulation)

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

*where $f$ and $g$ are both proper, closed and convex. Note that* (1) *is unconstrained.*

### Two remarks

- Nonsmoothness: At least one of two functions $f$ and $g$ is **nonsmooth**
  - General nonsmooth convex optimization methods (e.g., subgradient methods or bundle methods) are not efficient and numerically robust.
    - Indeed, subgradient/bundle methods 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: (1) clearly covers a much wider variety of problems than smooth unconstrained problems. For instance, we can immediately handle regularized $M$-estimators with the following setup:
  - $f$ is a loss function, a data fidelity term or a negative log-likelihood function.
  - $g$ is a regularizer or a gauge function encouraging structure in the solution.

# Optimal solution and structure assumption

## Definition (Optimal solutions and solution set)

- (1) has solution if $F^\star$ is finite.
- $\mathbf{x}^\star \in \mathbb{R}^p$ is a solution to (1) if $\boxed{F(\mathbf{x}^\star) = F^\star}$.
- $\boxed{\mathcal{S}^\star := \{\mathbf{x}^\star \in \mathbb{R}^p \ : \ F(\mathbf{x}^\star) = F^\star\}}$ is the solution set of (1).

# Optimal solution and structure assumption

## Definition (Optimal solutions and solution set)

- (1) has solution if $F^\star$ is finite.
- $\mathbf{x}^\star \in \mathbb{R}^p$ is a solution to (1) if $\boxed{F(\mathbf{x}^\star) = F^\star}$.
- $\boxed{\mathcal{S}^\star := \{\mathbf{x}^\star \in \mathbb{R}^p \ : \ F(\mathbf{x}^\star) = F^\star\}}$ is the solution set of (1).

## Assumption (Two distinct settings for (1))

*Throughout, we assume $f$ and $g$ to feature the one of the following structures*

(a) *Both $f$ and $g$ are nonsmooth, i.e., $f, g \in \mathcal{F}(\mathbb{R}^p)$.*

(b) *Only $f$ is smooth, i.e., $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^p)$.*

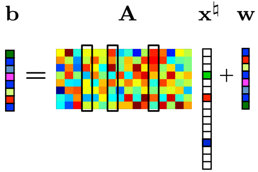
*Recall that $\mathcal{F}$ is the class of convex functions, $\mathcal{F}_L^{1,1}$ is the class of smooth convex functions with Lipschitz gradient (cf., formal definitions in Lecture 2).*

# Example 1: $\ell_1$-regularized least-squares

**Problem ($\ell_1$-regularized least-squares)**

*Compressive sensing setup:*

- $\mathbf{A}$ *is a sensing matrix (measurement matrix).*
- $\mathbf{b}$ *is an observations/measurements vector.*
- $\mathbf{x}^\natural$ *is an unknown sparse signal.*
- $\mathbf{w}$ *is unknown perturbations / noise.*

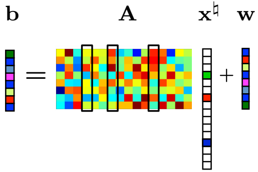


# Example 1: $\ell_1$-regularized least-squares

## Problem ($\ell_1$-regularized least-squares)

*Compressive sensing setup:*

- $\mathbf{A}$ *is a sensing matrix (measurement matrix).*
- $\mathbf{b}$ *is an observations/measurements vector.*
- $\mathbf{x}^\natural$ *is an unknown sparse signal.*
- $\mathbf{w}$ *is unknown perturbations / noise.*




## Optimization formulation

$$\min_{\mathbf{x}\in\mathbb{R}^p}\left\{\underbrace{\frac{1}{2}\|\mathbf{A}(\mathbf{x})-\mathbf{b}\|_2^2}_{f(\mathbf{x})}+\underbrace{\lambda\|\mathbf{x}\|_1}_{g(\mathbf{x})}\right\} \quad (2)$$

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

## Example 2: 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:*

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

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

# Example 2: 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:*

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

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

## Optimization formulation

$$\min_{\mathbf{x}\in\mathbb{R}^p} \Big\{ \underbrace{\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(b_i(\mathbf{a}_i^T\mathbf{x} + \mu))}_{f(\mathbf{x})} + \underbrace{\lambda\|\mathbf{x}\|_1}_{g(\mathbf{x})} \Big\}, \tag{3}$$

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

## Example 3: 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.*

# Example 3: 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

$$\text{Gaussian}: \quad \min_{\mathbf{x}\in\mathbb{R}^{n\times p}} \Big\{ \underbrace{(1/2)\|\mathcal{A}(\mathbf{x}) - \mathbf{b}\|_2^2}_{f(\mathbf{x})} + \underbrace{\lambda\|\mathbf{x}\|_{\mathrm{TV}}}_{g(\mathbf{x})} \Big\} \tag{4}$$

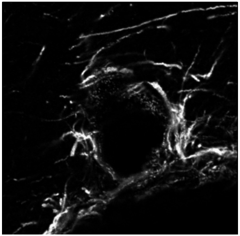
$$\text{Poisson}: \quad \min_{\mathbf{x}\in\mathbb{R}^{n\times p}} \Big\{ \underbrace{\frac{1}{n}\sum_{i=1}^{n} [\langle \mathbf{a}_i, \mathbf{x}\rangle - b_i \ln(\langle \mathbf{a}_i, \mathbf{x}\rangle)]}_{f(\mathbf{x})} + \underbrace{\lambda\|\mathbf{x}\|_{\mathrm{TV}}}_{g(\mathbf{x})} \Big\} \tag{5}$$

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

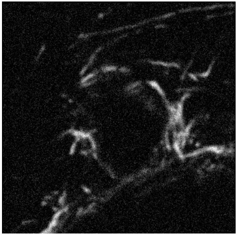
$$\|\mathbf{x}\|_{\mathrm{TV}} := \begin{cases} \sum_{i,j} |\mathbf{x}_{i,j+1} - \mathbf{x}_{i,j}| + |\mathbf{x}_{i+1,j} - \mathbf{x}_{i,j}| & \text{anisotropic case,} \\ \sum_{i,j} \sqrt{|\mathbf{x}_{i,j+1} - \mathbf{x}_{i,j}|^2 + |\mathbf{x}_{i+1,j} - \mathbf{x}_{i,j}|^2} & \text{isotropic case} \end{cases}$$

# Example 3: Confocal microscopy with camera blur and Poisson observations

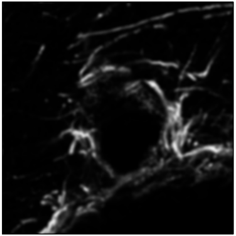
Original image $\mathbf{x}^\natural$



Observed image $\mathbf{b}$



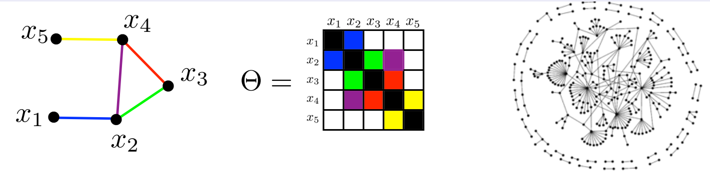
Estimate $\hat{\mathbf{x}}$ via (5)



# Example 4: Sparse inverse covariance estimation

## Problem (Graphical model selection)

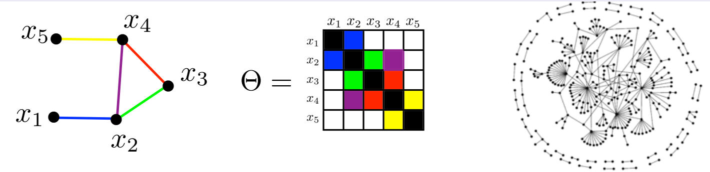
*Given a data set* $\mathcal{D} := \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$, *where* $\mathbf{x}_i$ *is a Gaussian random variable. Let* $\Sigma$ *be the covariance matrix corresponding to the graphical model of the Gaussian Markov random field. Our goal is to learn a sparse precision matrix* $\Theta$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




### Optimization formulation

$$\min_{\Theta \succ 0} \Big\{ \underbrace{\mathrm{tr}(\Sigma\Theta) - \log\det(\Theta)}_{f(\mathbf{x})} + \underbrace{\lambda \|\mathrm{vec}(\Theta)\|_1}_{g(\mathbf{x})} \Big\} \tag{6}$$

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

# Outline

- Today
  1. Composite convex minimization
  2. Proximal operator and computational complexity
  3. Proximal gradient methods
  4. Composite self-concordant minimization
  5. Smoothing for nonsmooth composite convex minimization
- Next week
  1. Nonsmooth constrained optimization

## 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

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

- 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 [9])

*Let $\partial f$ and $\partial g$ 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 [9])

*Let $\partial f$ and $\partial g$ 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:*

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

*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

$$0 \in \partial F(\mathbf{x}^\star) \equiv \partial f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star). \tag{7}$$

If $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^p)$, then (7) features the gradient of $f$ as opposed to the subdifferential

$$0 \in \partial F(\mathbf{x}^\star) \equiv \nabla f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star). \tag{8}$$

## 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* (7)*:* $0 \in \partial f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star)$ *(or* (8)*:* $0 \in \nabla f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star)$ *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* (7)*:* $0 \in \partial f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star)$ *(or* (8)*:* $0 \in \nabla f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star)$ *when* $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^p)$ *).*

### Sketch of the proof.

- $\Rightarrow$: By definition of $\partial F$:

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

If (7) (or (8)) 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 (7) (or (8)). □

## A short detour: Proximal-point operators

### Definition (Proximal operator [10])

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:

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

# A short detour: Proximal-point operators

## Definition (Proximal operator [10])

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:

$$\mathrm{prox}_g(\mathbf{x}) \equiv \arg\min_{\mathbf{y}\in\mathbb{R}^p} \left\{ g(\mathbf{y}) + \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 \right\}. \tag{9}$$

## 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* (9) *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:*
$$\|\mathrm{prox}_g(\mathbf{x}) - \mathrm{prox}_g(\tilde{\mathbf{x}})\|_2 \leq \|\mathbf{x} - \tilde{\mathbf{x}}\|_2, \quad \forall \mathbf{x}, \tilde{\mathbf{x}} \in \mathbb{R}^p.$$

## Fixed-point characterization

### Optimality condition as fixed-point formulation

The optimality condition (7): $0 \in \partial f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star)$ is equivalent to

$$\mathbf{x}^\star \in \mathrm{prox}_{\lambda g}\left(\mathbf{x}^\star - \lambda \partial f(\mathbf{x}^\star)\right) := \mathcal{T}_\lambda(\mathbf{x}^\star), \quad \text{for any } \lambda > 0. \tag{10}$$

The optimality condition (8): $0 \in \nabla f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star)$ is equivalent to

$$\mathbf{x}^\star = \mathrm{prox}_{\lambda g}\left(\mathbf{x}^\star - \lambda \nabla f(\mathbf{x}^\star)\right) := \mathcal{U}_\lambda(\mathbf{x}^\star), \quad \text{for any } \lambda > 0. \tag{11}$$

$\mathcal{T}_\lambda$ 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 (7): $0 \in \partial f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star)$ is equivalent to

$$\mathbf{x}^\star \in \mathrm{prox}_{\lambda g}\left(\mathbf{x}^\star - \lambda \partial f(\mathbf{x}^\star)\right) := \mathcal{T}_\lambda(\mathbf{x}^\star), \quad \text{for any } \lambda > 0. \tag{10}$$

The optimality condition (8): $0 \in \nabla f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star)$ is equivalent to

$$\mathbf{x}^\star = \mathrm{prox}_{\lambda g}\left(\mathbf{x}^\star - \lambda \nabla f(\mathbf{x}^\star)\right) := \mathcal{U}_\lambda(\mathbf{x}^\star), \quad \text{for any } \lambda > 0. \tag{11}$$

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

### Proof.

We prove (11) ((10) is done similarly). (8) 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 $\nabla f$ are single-valued, we obtain (11). □

## Choices of solution methods

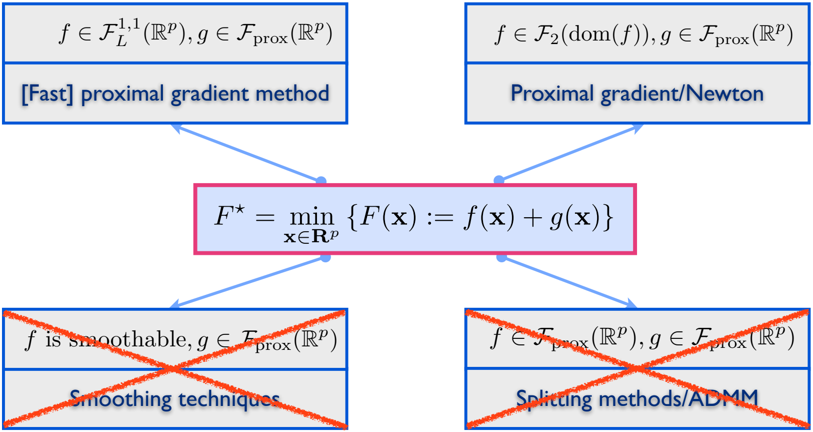
$$F^\star = \min_{\mathbf{x} \in \mathbf{R}^p} \{ F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x}) \}$$

- $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^p), g \in \mathcal{F}_{\text{prox}}(\mathbb{R}^p)$ → **[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 slides.

## Choices of solution methods




- $\mathcal{F}_L^{1,1}$ 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 slides.

## Choices of solution methods

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

[Fast] proximal gradient method

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

Proximal gradient/Newton

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

~~$f$ is s … rox$(\mathbb{R}^p)$~~

Lecture 6

~~Smoothing techniques~~

Lectures 7 and 8

~~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 slides.

# 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

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

## 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

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

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

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 $\mathrm{prox}_g$ 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 $\mathrm{prox}_F$, we can use the **proximal-point algorithm** (PPA) [4, 10] to solve (1). Unfortunately, PPA is hardly used in practice to solve (12) 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 $\mathrm{prox}_g$ 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 $\mathrm{prox}_F$, we can use the **proximal-point algorithm** (PPA) [4, 10] to solve (1). Unfortunately, PPA is hardly used in practice to solve (12) 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 $\mathrm{prox}_g$ defined by (9) 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.

# $^{\star}$The proximal-point method

Problem (Unconstrained convex minimization)

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

$$\boxed{F^{\star} := \min_{\mathbf{x} \in \mathbb{R}^p} F(\mathbf{x}).} \tag{12}$$

## $^\star$The proximal-point method

### Problem (Unconstrained convex minimization)

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

$$F^\star := \min_{\mathbf{x} \in \mathbb{R}^p} F(\mathbf{x}). \tag{12}$$

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

# $^\star$The proximal-point method

## Problem (Unconstrained convex minimization)

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

$$F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} F(\mathbf{x}). \tag{12}$$

**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:

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

## Theorem (Convergence [4])

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

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

*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

$$\boxed{\mathrm{prox}_{\lambda g}(\mathbf{x}) := \arg\min_{\mathbf{y}\in\mathbb{R}^p} \left\{\frac{1}{2}\|\mathbf{A}\mathbf{y} - \mathbf{b}\|_2^2 + \frac{1}{2\lambda}\|\mathbf{y} - \mathbf{x}\|_2^2\right\}.} \tag{13}$$

## 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

$$\boxed{\mathrm{prox}_{\lambda g}(\mathbf{x}) := \arg\min_{\mathbf{y}\in\mathbb{R}^p} \left\{ \frac{1}{2}\|\mathbf{A}\mathbf{y} - \mathbf{b}\|_2^2 + \frac{1}{2\lambda}\|\mathbf{y} - \mathbf{x}\|_2^2 \right\}.} \tag{13}$$

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

The optimality condition implies that the solution of (13) 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

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

- 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 $\Lambda$ is a diagonal matrix) then $\mathrm{prox}_{\lambda g}(\mathbf{x})$ can be cheap
  $\mathrm{prox}_{\lambda g}(\mathbf{x}) = \mathbf{U}\left(\mathbb{I} + \lambda\Lambda\right)^{-1}\mathbf{U}^T\left(\mathbf{x} + \lambda\mathbf{A}\mathbf{b}\right)$.
    - 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 |
|---|---|---|---|
| $\ell_1$-norm | $f(\mathbf{x}) := \\|\mathbf{x}\\|_1$ | $\mathrm{prox}_{\lambda f}(\mathbf{x}) = \mathrm{sign}(\mathbf{x}) \otimes [\lvert\mathbf{x}\rvert - \lambda]_+$ | $\mathcal{O}(p)$ |
| $\ell_2$-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 $\ell_2$-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 [3, 8].

# Outline

- Today
  1. Composite convex minimization
  2. Proximal operator and computational complexity
  3. Proximal gradient methods
  4. Composite self-concordant minimization
  5. Smoothing for nonsmooth composite convex minimization
- Next week
  1. Nonsmooth constrained optimization

# Overview of algorithms/complexity

| Assumption | Algorithm | Convergence rate ($\varepsilon$) | Complexity per iteration |
|---|---|---|---|
| $f, g \in \mathcal{F}(\mathbb{R}^p)$ | Subgradient | $\mathcal{O}(1/\sqrt{k})$ | 1 sub-gradient of $f$, $g$ |
| | Bundle method | $\mathcal{O}(1/\sqrt{k})$ | 1 sub-gradient of $f$, $g$ |
| | Mirror-descent | $\mathcal{O}(1/\sqrt{k})$ | 1 sub-gradient of $f$, $g$ |
| $f \in \mathcal{F}^{1,1}_{L,\mu}(\mathbb{R}^p)$, $g \in \mathcal{F}_{\text{prox}}(\mathbb{R}^n)$ | Proximal-gradient | $\mathcal{O}(1/k)$ ($\mu = 0$), linear ($\mu > 0$) | 1 gradient, 1 prox operator |
| | Accelerated proximal-gradient | $\mathcal{O}(1/k^2)$ ($\mu = 0$), linear ($\mu > 0$) | 1 gradient, 1 or 2 prox operator(s) |
| | Proximal quasi-Newton | locally superlinear, globally sublinear | One gradient, rank-2 update |
| | Proximal Newton | locally quadratic, locally sublinear $\mathcal{O}(1/k^s)$, $1 \leq s \leq 3$ | One gradient, one Hessian inverse |
| $f, g \in \mathcal{F}_{\text{prox}}(\mathbb{R}^n)$ | Peaceman-Douglas | $\mathcal{O}(1/k)$-ergodic | $\geq 1$ prox operator(s) $f$, $g$ |
| | Douglas-Rachford | $\mathcal{O}(1/k)$-ergodic | $\geq 1$ prox operator(s) $f$, $g$ |
| | ALM | $\mathcal{O}(1/k^2)$ | $\geq 1$ prox operator(s) $f$, $g$ |
| | ADMM | $\mathcal{O}(1/k)$ | $\geq 1$ prox operator(s) $f$, $g$ |

- ALM = augmented Lagrangian method, ADMM = alternating direction method of multiplier.
- $\mathcal{F}$ = class of proper, closed convex functions.
- $\mathcal{F}^{1,1}_{L,\mu}$ = class of strongly convex functions with Lipschitz gradient.
- $\mathcal{F}_{\text{prox}}$ = class of convex functions with tractable prox-operator.

# Overview of algorithms/complexity

| Assumption | Algorithm | Convergence rate | Complexity per iteration |
|---|---|---|---|
| $f, g \in \mathcal{F}(\mathbb{R}^p)$ | Subgradient | $\mathcal{O}(1/\sqrt{k})$ | 1 sub-gradient of $f$, $g$ |
| | Bundle method | $\mathcal{O}(1/\sqrt{k})$ | 1 sub-gradient of $f$, $g$ |
| | Mirror-descent | $\mathcal{O}(1/\sqrt{k})$ | 1 sub-gradient of $f$, $g$ |
| $f \in \mathcal{F}^{1,1}_{L,\mu}(\mathbb{R}^p)$, $g \in \mathcal{F}_{\text{prox}}(\mathbb{R}^n)$ | Proximal-gradient | $\mathcal{O}(1/k)$ $(\mu = 0)$, linear $(\mu > 0)$ | 1 gradient, 1 prox operator |
| | Accelerated proximal-gradient | $\mathcal{O}(1/k^2)$ $(\mu = 0)$, linear $(\mu > 0)$ | 1 gradient, 1 or 2 prox operator(s) |
| | Proximal quasi-Newton | locally superlinear, globally sublinear | One gradient, rank-2 update |
| | Proximal Newton | locally quadratic, locally sublinear $\mathcal{O}(1/k^s)$, $1 \leq s \leq 3$ | One gradient, one Hessian inverse |
| $f, g \in \mathcal{F}_{\text{prox}}(\mathbb{R}^n)$ | Peaceman-Douglas | $\mathcal{O}(1/k)$-ergodic | $\geq 1$ prox operator(s) $f$, $g$ |
| | Douglas-Rachford | $\mathcal{O}(1/k)$-ergodic | $\geq 1$ prox operator(s) $f$, $g$ |
| | ALM | $\mathcal{O}(1/k^2)$ | $\geq 1$ prox operator(s) $f$, $g$ |
| | ADMM | $\mathcal{O}(1/k)$ | $\geq 1$ prox operator(s) $f$, $g$ |

- ALM = augmented Lagrangian method, ADMM = alternating direction method of multiplier.
- $\mathcal{F}$ = class of proper, closed convex functions.
- $\mathcal{F}^{1,1}_{L,\mu}$ = class of strongly convex functions with Lipschitz gradient.
- $\mathcal{F}_{\text{prox}}$ = class of convex functions with tractable prox-operator.

## Proximal-gradient method

### Assumption and oracle of proximal-gradient

- **Assumption A.2.**: $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^p)$ and $g \in \mathcal{F}_{\mathrm{prox}}(\mathbb{R}^p)$.
- **Oracle**: $\nabla f$ and $\mathrm{prox}_{\lambda g}$.

# Proximal-gradient method

## Assumption and oracle of proximal-gradient

- **Assumption A.2.**: $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^p)$ and $g \in \mathcal{F}_{\text{prox}}(\mathbb{R}^p)$.
- **Oracle**: $\nabla f$ and $\text{prox}_{\lambda g}$.

## Motivation ...

- [Fast] gradient methods offer a low computational cost per iteration.
- If $f \in \mathcal{F}_L^{1,1}$, then we can achieve $\mathcal{O}(1/k)$ convergence rate.
- Under **Assumption A.2.**, [fast] proximal-gradient methods have almost the same computational cost per iteration as [fast] gradient methods at the cost of one additional proximal operation per iteration.
- They maintain the same convergence rate as in [fast] gradient methods.

## A quadratic majorization perspective

### Definition (Quadratic model for $f$)

Given $\mathbf{x} \in \mathbb{R}^p$, we define:

$$Q_L(\mathbf{y}, \mathbf{x}) := f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) + \frac{L}{2}\|\mathbf{y} - \mathbf{x}\|_2^2, \ \forall \mathbf{y} \in \mathbb{R}^p. \quad (14)$$

# A quadratic majorization perspective

## Definition (Quadratic model for $f$)

Given $\mathbf{x} \in \mathbb{R}^p$, we define:

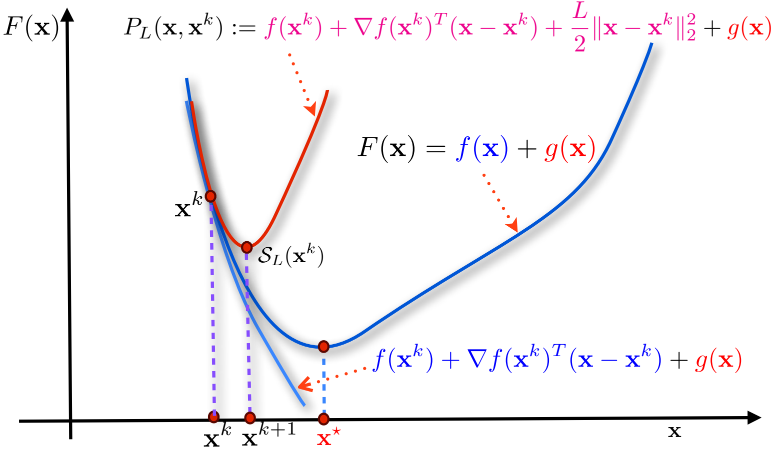
$$Q_L(\mathbf{y}, \mathbf{x}) := f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) + \frac{L}{2}\|\mathbf{y} - \mathbf{x}\|_2^2, \ \forall \mathbf{y} \in \mathbb{R}^p. \tag{14}$$

## Property (Upper and lower bounds)

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

$$\begin{cases} f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) \leq f(\mathbf{y}) \\ f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) + \frac{L_f}{2}\|\mathbf{y} - \mathbf{x}\|_2^2, \end{cases} \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^p \tag{15}$$

## Geometric illustration



## Proximal-gradient mapping

### Definition (Quadratic-convex model of $F$)

Given a point $\mathbf{x}^k \in \mathbb{R}^p$ and $L > 0$. The quadratic-convex model of $F$ at $\mathbf{x}^k$ is defined as:

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

## Proximal-gradient mapping

### Definition (Quadratic-convex model of $F$)

Given a point $\mathbf{x}^k \in \mathbb{R}^p$ and $L > 0$. The quadratic-convex model of $F$ at $\mathbf{x}^k$ is defined as:

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

### Definition (Proximal-gradient mapping [6])

$$\mathcal{S}_L(\mathbf{x}^k) := \underset{\mathbf{x} \in \mathrm{dom}(F)}{\operatorname{argmin}} P_L(\mathbf{x}, \mathbf{x}^k) \equiv \mathrm{prox}_{(1/L)g}\left(\mathbf{x}^k - (1/L)\nabla f(\mathbf{x}^k)\right). \tag{16}$$

The proximal-gradient mapping of $F$ is defined as:

$$\mathcal{PG}_{\mathcal{L}}(\mathbf{x}^k) := L(\mathbf{x}^k - \mathcal{S}_L(\mathbf{x}^k)). \tag{17}$$

**Note:** When $g \equiv 0$, we have $\mathcal{PG}_{\mathcal{L}}(\mathbf{x}^k) \equiv \nabla f(\mathbf{x}^k)$.

### Property (Optimality condition (Exercise))

*If* $\mathcal{PG}_{\mathcal{L}}(\mathbf{x}^\star) = 0$ *then* $\mathbf{x}^\star$ *is an optimal solution of* (1).

## Proximal-gradient algorithm algorithm

| **Basic proximal-gradient scheme (ISTA)** |
|---|
| **1.** Choose $\mathbf{x}^0 \in \text{dom}(F)$ arbitrarily as a starting point. |
| **2.** For $k = 0, 1, \cdots$, generate a sequence $\{\mathbf{x}^k\}_{k \geq 0}$ as: |
| $\mathbf{x}^{k+1} := \text{prox}_{\lambda g}\left(\mathbf{x}^k - \lambda \nabla f(\mathbf{x}^k)\right),$ |
| where $\mathcal{S}_L$ is defined as (16) and $\lambda := \frac{1}{L_f}$. |

## Proximal-gradient algorithm algorithm

| **Basic proximal-gradient scheme** $(\mathrm{ISTA})$ |
| --- |
| **1.** Choose $\mathbf{x}^0 \in \mathrm{dom}(F)$ arbitrarily as a starting point.<br>**2.** For $k = 0, 1, \cdots$, generate a sequence $\{\mathbf{x}^k\}_{k\geq 0}$ as:<br>$$\mathbf{x}^{k+1} := \mathrm{prox}_{\lambda g}\left(\mathbf{x}^k - \lambda\nabla f(\mathbf{x}^k)\right),$$<br>where $\mathcal{S}_L$ is defined as (16) and $\lambda := \frac{1}{L_f}$. |

### Theorem (Convergence of ISTA [1])

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

$$F(\mathbf{x}^k) - F^\star \leq \frac{L_f\|\mathbf{x}^0 - \mathbf{x}^\star\|_2^2}{2(k+1)} \tag{18}$$

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

## Example: $\ell_1$-regularized least squares

### Problem ($\ell_1$-regularized least squares)

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

$$F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ F(\mathbf{x}) := \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_1 \right\}, \tag{19}$$

*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: $\ell_1$-regularized least squares

## Problem ($\ell_1$-regularized least squares)

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

$$F^\star := \min_{\mathbf{x} \in \mathbb{R}^p} \left\{ F(\mathbf{x}) := \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_1 \right\}, \quad (19)$$

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

## Complexity per iterations

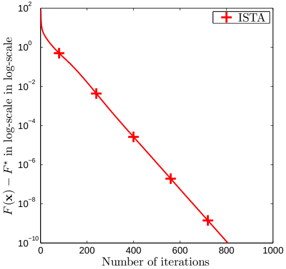
- Evaluating $\nabla f(\mathbf{x}^k) = \mathbf{A}^T(\mathbf{A}\mathbf{x}^k - \mathbf{b})$ requires one $\mathbf{A}\mathbf{x}$ and one $\mathbf{A}^T\mathbf{y}$.
- One soft-thresholding operator $\mathrm{prox}_{\lambda_g}(\mathbf{x}) = \mathrm{sign}(\mathbf{x}) \otimes \max\{|\mathbf{x}| - \lambda, 0\}$.
- **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

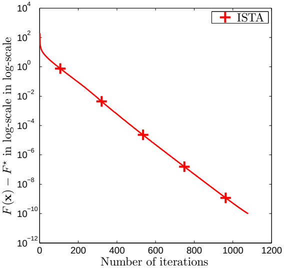
- $\mathbf{A} := \mathrm{randn}(n, p)$ - standard Gaussian $\mathcal{N}(0, \mathbb{I})$.
- $\mathbf{x}^\star$ is a $k$-sparse vector generated randomly.
- $\mathbf{b} := \mathbf{A}\mathbf{x}^\star + \mathcal{N}(0, 10^{-3})$.

## Example: Numerical test with ISTA

**Case 1:**
$n = 750, p = 2000, s = 200, \lambda = 0.1$



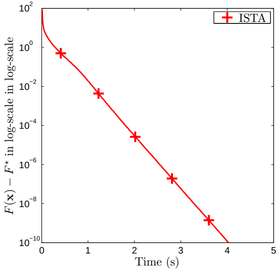

**Case 2:**
$n = 1750, p = 5000, s = 500, \lambda = 0.1$



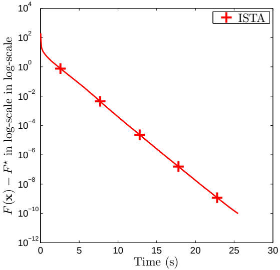

| | Case 1 | Case 2 |
|---|---|---|
| Number of iterations | 806 | 1079 |
| CPU time (s) | 4.030 | 25.509 |
| Solution error $(\times 10^{-11})$ | 9.971 | 9.810 |

# Example: Numerical test with ISTA - Performance w.r.t. time

**Case 1:**
$n = 750, p = 2000, s = 200, \lambda = 0.1$



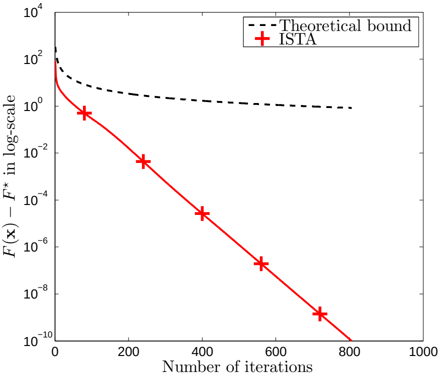

**Case 2:**
$n = 1750, p = 5000, s = 500, \lambda = 0.1$

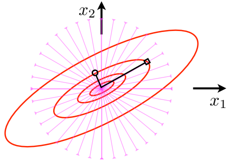



| | Case 1 | Case 2 |
|---|---|---|
| Number of iterations | 806 | 1079 |
| CPU time (s) | 4.030 | 25.509 |
| Solution error $(\times 10^{-11})$ | 9.971 | 9.810 |

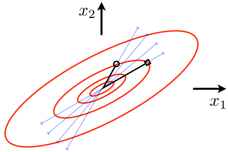
# Example: Theoretical bounds vs practical performance

- **Theoretical bound: ISTA** $:= \frac{L_f R_0^2}{2(k+1)}$.




descent directions




restricted descent directions

- $\ell_1$-regularized least squares formulation has **restricted strong convexity**. The proximal-gradient method can automatically exploit this structure.

# Fast proximal-gradient algorithm

## The need for a faster algorithm

The convergence rate of ISTA is NOT **optimal**:

$$\boxed{F(\mathbf{x}^k) - F^\star \leq \frac{L_f R_0^2}{2(k+1)}}, \tag{20}$$

where $R_0 := \min_{\mathbf{x}^\star \in \mathcal{S}^\star} \|\mathbf{x}^0 - \mathbf{x}^\star\|_2$. This is because the iterates of methods based on gradients and objective evaluations must obey

$$F(\mathbf{x}^k) - F^\star \geq \boxed{\frac{3L_f R_0^2}{32(k+2)^2}}, \quad 1 \leq k \leq (p-1)/2. \tag{21}$$

An algorithm with optimal convergence would achieve this lower bound up to a constant factor.

# Fast proximal-gradient algorithm

## The need for a faster algorithm

The convergence rate of ISTA is NOT **optimal**:

$$\boxed{F(\mathbf{x}^k) - F^\star \leq \frac{L_f R_0^2}{2(k+1)}}, \tag{20}$$

where $R_0 := \min_{\mathbf{x}^\star \in \mathcal{S}^\star} \|\mathbf{x}^0 - \mathbf{x}^\star\|_2$. This is because the iterates of methods based on gradients and objective evaluations must obey

$$F(\mathbf{x}^k) - F^\star \geq \boxed{\frac{3L_f R_0^2}{32(k+2)^2}}, \quad 1 \leq k \leq (p-1)/2. \tag{21}$$

An algorithm with optimal convergence would achieve this lower bound up to a constant factor.

## Can we design an algorithm with optimal convergence?

Answer: **YES**

# 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:

$$\begin{cases} \mathbf{x}^{k+1} & := \mathrm{prox}_{\lambda g}\left(\mathbf{y}^k - \lambda\nabla f(\mathbf{y}^k)\right), \\ t_{k+1} & := 0.5(1 + \sqrt{4t_k^2 + 1}), \\ \gamma_{k+1} & := (t_k - 1)/t_{k+1}, \\ \mathbf{y}^{k+1} & := \mathbf{x}^{k+1} + \gamma_{k+1}(\mathbf{x}^{k+1} - \mathbf{x}^k). \end{cases} \tag{22}$$

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

# 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:

$$\begin{cases} \mathbf{x}^{k+1} & := \mathrm{prox}_{\lambda g}\left(\mathbf{y}^k - \lambda\nabla f(\mathbf{y}^k)\right), \\ t_{k+1} & := 0.5(1 + \sqrt{4t_k^2 + 1}), \\ \gamma_{k+1} & := (t_k - 1)/t_{k+1}, \\ \mathbf{y}^{k+1} & := \mathbf{x}^{k+1} + \gamma_{k+1}(\mathbf{x}^{k+1} - \mathbf{x}^k). \end{cases} \tag{22}$$

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

## Complexity per iteration

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

The **cost per iteration** is almost the same as in **gradient scheme**.

## Global convergence of FISTA

### Theorem (**Global complexity** [1])

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

$$F(\mathbf{x}^k) - F^\star \leq \boxed{\frac{2L_f R_0^2}{(k+2)^2}}, \ \forall k \geq 0. \tag{23}$$

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

### Remark

The convergence rate of FISTA is **optimal** up to a constant factor based on the lowerbound we described earlier:

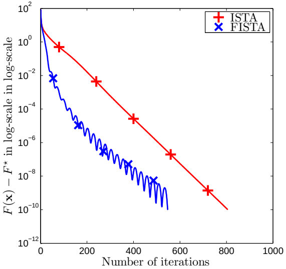
$$F(\mathbf{x}^k) - F^\star \geq \boxed{\frac{3L_f R_0^2}{32(k+2)^2}}, \quad 1 \leq k \leq (p-1)/2, \tag{24}$$

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

# Example: Numerical test with FISTA

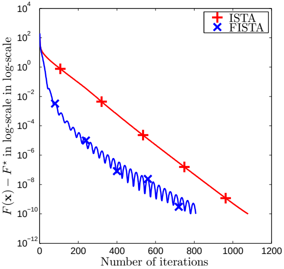
**Case 1:**

$n = 750, p = 2000, s = 200, \lambda = 0.1$




**Case 2:**

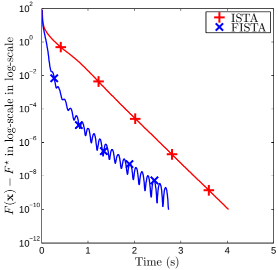
$n = 1750, p = 5000, s = 500, \lambda = 0.1$



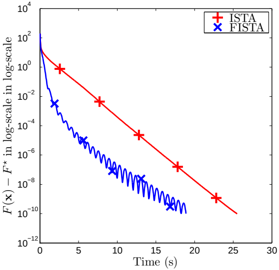

| | Case 1 | | Case 2 | |
|---|---|---|---|---|
| | ISTA | FISTA | ISTA | FISTA |
| Number of iterations | 806 | 548 | 1079 | 808 |
| CPU time (s) | 4.030 | 2.738 | 25.509 | 18.889 |
| Solution error $(\times 10^{-11})$ | 9.971 | 9.783 | 9.810 | 9.975 |

# Example: Numerical test with FISTA - Performance w.r.t. time

**Case 1:**
$n = 750, p = 2000, s = 200, \lambda = 0.1$



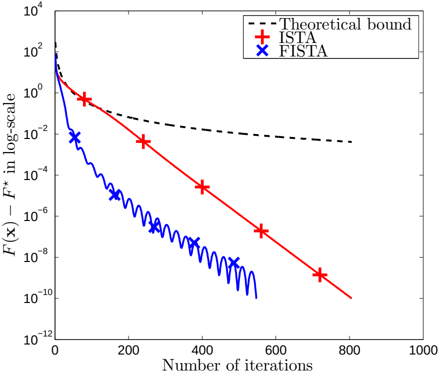

**Case 2:**
$n = 1750, p = 5000, s = 500, \lambda = 0.1$

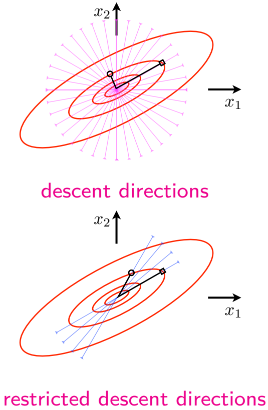



| | Case 1 | | Case 2 | |
|---|---|---|---|---|
| | ISTA | FISTA | ISTA | FISTA |
| Number of iterations | 806 | 548 | 1079 | 808 |
| CPU time (s) | 4.030 | 2.738 | 25.509 | 18.889 |
| Solution error $(\times 10^{-11})$ | 9.971 | 9.783 | 9.810 | 9.975 |

# Example: Theoretical bounds vs practical performance

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




- $\ell_1$-regularized least squares formulation has **restricted strong convexity**. The proximal-gradient method can automatically exploit this structure.

## Enhancements

### Two practical enhancements

1. Line-search for evaluating $L_f$ as LS-ISTA.
2. Adaptive restart strategies

## Enhancements

### Two practical enhancements

1. Line-search for evaluating $L_f$ as LS-ISTA.
2. Adaptive restart strategies

### When do we need a line-search procedure?

We can use a line-search procedure in one of the following cases:

- $L_f$ is **unknown**, a **line-search** procedure can approximate $L_f$.
- $L_f$ is **known** but expensive to evaluate.
- The global constant $L_f$ usually does not capture the local behavior of $f$, we want to improve this behavior.

# Enhancements

## Two practical enhancements

1. Line-search for evaluating $L_f$ as LS-ISTA.
2. Adaptive restart strategies

## When do we need a line-search procedure?

We can use a line-search procedure in one of the following cases:

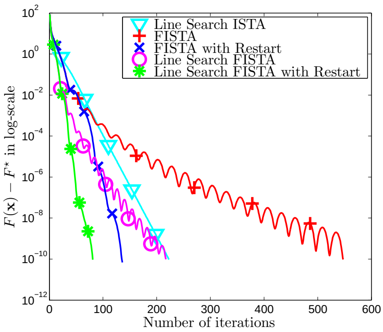
- $L_f$ is **unknown**, a **line-search** procedure can approximate $L_f$.
- $L_f$ is **known** but expensive to evaluate.
- The global constant $L_f$ usually does not capture the local behavior of $f$, we want to improve this behavior.

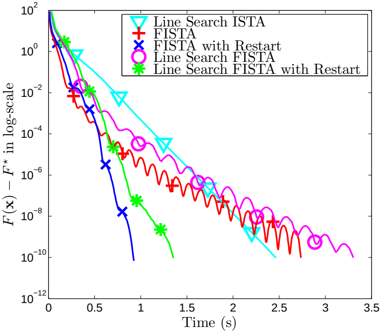
## Why do we need a restart strategy?

- FISTA is non-monotonic (i.e., $F(\mathbf{x}^{k+1}) \not\leq F(\mathbf{x}^k)$ is not necessary satisfied).
- FISTA has a periodic behavior, where the momentum depends on the local condition number $c_f := L_f/\mu_f$ ($\mu_f$ is the local strong convexity parameter). Since the momentum term is increasing, the algorithm can overshoot the optimal solution and has to backtrack.
- A **restart strategy** resets the momentum whenever we observe oscillations.

# Example: Periodic behavior of FISTA and its enhancements

**Case 1:** $n = 750, p = 2000, s = 200, \lambda = 0.1$



## Line-search proximal-gradient algorithm

**Line-search proximal-gradient scheme ($\mathrm{LS}-\mathrm{ISTA}$)**

**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:

**2.a.** Find the smallest $j \geq 0$ such that $L_k^j := 2^j L_0$ satisfying

$$F(\mathcal{S}_{L_k^j}(\mathbf{x}^k)) \leq P_{L_k^j}(\mathcal{S}_{L_k^j}(\mathbf{x}^k), \mathbf{x}^k),$$

where $L_0 > 0$ is a given.

**2.b.** Update

$$\mathbf{x}^{k+1} := \mathcal{S}_{L_k^j}(\mathbf{x}^k) \equiv \mathrm{prox}_{1/L_k^j g}\left(\mathbf{x}^k - (1/L_k^j)\nabla f(\mathbf{x}^k)\right)$$

We can use $L_0 := \frac{\|\nabla f(\mathbf{x}^1) - \nabla f(\mathbf{x}^0)\|_2}{\|\mathbf{x}^1 - \mathbf{x}^0\|_2}$.

# Line-search proximal-gradient algorithm

**Line-search proximal-gradient scheme ($\mathrm{LS} - \mathrm{ISTA}$)**

**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:

**2.a.** Find the smallest $j \geq 0$ such that $L_k^j := 2^j L_0$ satisfying

$$F(\mathcal{S}_{L_k^j}(\mathbf{x}^k)) \leq P_{L_k^j}(\mathcal{S}_{L_k^j}(\mathbf{x}^k), \mathbf{x}^k),$$

where $L_0 > 0$ is a given.

**2.b.** Update

$$\mathbf{x}^{k+1} := \mathcal{S}_{L_k^j}(\mathbf{x}^k) \equiv \mathrm{prox}_{1/L_k^j g}\left(\mathbf{x}^k - (1/L_k^j)\nabla f(\mathbf{x}^k)\right)$$

We can use $L_0 := \frac{\|\nabla f(\mathbf{x}^1) - \nabla f(\mathbf{x}^0)\|_2}{\|\mathbf{x}^1 - \mathbf{x}^0\|_2}$.

## Complexity per iteration of LS-ISTA

- One gradient of $f$ and one prox-operator of $g$
- Requires roughly $2$ function evaluations of $F$ in average for each iteration.

## Example: $\ell_1$-regularized least squares

### Problem ($\ell_1$-regularized least squares)

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

$$F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ F(\mathbf{x}) := \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_1 \right\}, \tag{25}$$

*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: $\ell_1$-regularized least squares

## Problem ($\ell_1$-regularized least squares)

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

$$F^\star := \min_{\mathbf{x} \in \mathbb{R}^p} \left\{ F(\mathbf{x}) := \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_1 \right\}, \tag{25}$$

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

## Complexity per iterations

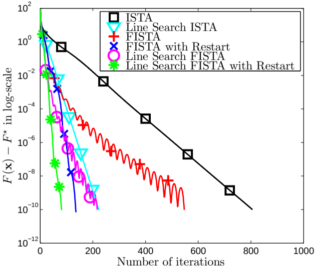
- Evaluating $\nabla f(\mathbf{x}^k) = \mathbf{A}^T(\mathbf{A}\mathbf{x}^k - \mathbf{b})$ requires one $\mathbf{A}\mathbf{x}$ and one $\mathbf{A}^T\mathbf{y}$.
- One soft-thresholding operator $\mathrm{prox}_{\lambda g}(\mathbf{x}) = \mathrm{sign}(\mathbf{x}) \otimes \max\{|\mathbf{x}| - \lambda, 0\}$.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

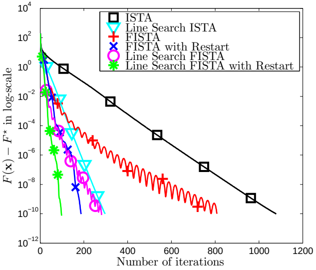
- $\mathbf{A} := \mathrm{randn}(n, p)$ - standard Gaussian $\mathcal{N}(0, \mathbb{I})$.
- $\mathbf{x}^\star$ is a $k$-sparse vector generated randomly.
- $\mathbf{b} := \mathbf{A}\mathbf{x}^\star + \mathcal{N}(0, 10^{-3})$.

# Example: $\ell_1$-regularized least squares

**Case 1:**
$n = 750, p = 2000, s = 200, \lambda = 0.1$




**Case 2:**
$n = 1750, p = 5000, s = 500, \lambda = 0.1$

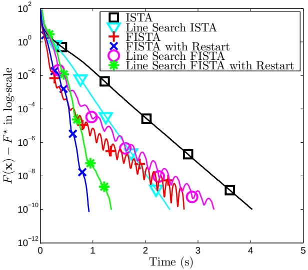



| | Case 1 | | | | | | Case 2 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ISTA | LS-ISTA | FISTA | FISTA-R | LS-FISTA | LS-FISTA-R | ISTA | LS-ISTA | FISTA | FISTA-R | LS-FISTA | LS-FISTA-R |
| Number of iterations | 806 | 223 | 548 | 136 | 217 | 81 | 1079 | 297 | 808 | 187 | 281 | 98 |
| CPU time (s) | 4.030 | 2.466 | 2.738 | 0.926 | 3.303 | 1.354 | 25.509 | 15.743 | 18.889 | 5.788 | 20.473 | 7.861 |
| Solution error $(\times 10^{-11})$ | 9.971 | 9.487 | 9.783 | 6.875 | 9.664 | 9.251 | 9.810 | 9.775 | 9.975 | 9.432 | 8.153 | 7.427 |

# Example: $\ell_1$-regularized least squares: Performance w.r.t. time

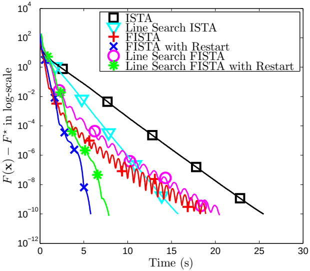
**Case 1:**

$n = 750, p = 2000, s = 200, \lambda = 0.1$




**Case 2:**

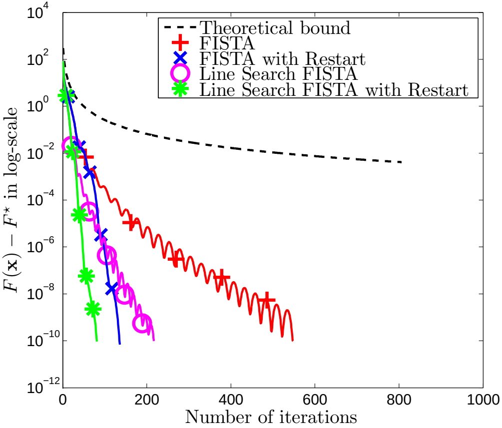
$n = 1750, p = 5000, s = 500, \lambda = 0.1$




| | Case 1 | | | | | | Case 2 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ISTA | LS-ISTA | FISTA | FISTA-R | LS-FISTA | LS-FISTA-R | ISTA | LS-ISTA | FISTA | FISTA-R | LS-FISTA | LS-FISTA-R |
| Number of iterations | 806 | 223 | 548 | 136 | 217 | 81 | 1079 | 297 | 808 | 187 | 281 | 98 |
| CPU time (s) | 4.030 | 2.466 | 2.738 | 0.926 | 3.303 | 1.354 | 25.509 | 15.743 | 18.889 | 5.788 | 20.473 | 7.861 |
| Solution error ($\times 10^{-11}$) | 9.971 | 9.487 | 9.783 | 6.875 | 9.664 | 9.251 | 9.810 | 9.775 | 9.975 | 9.432 | 8.153 | 7.427 |

# Example: BP - Theoretical bounds vs actual performance

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



## 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:*

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

# 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:*

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

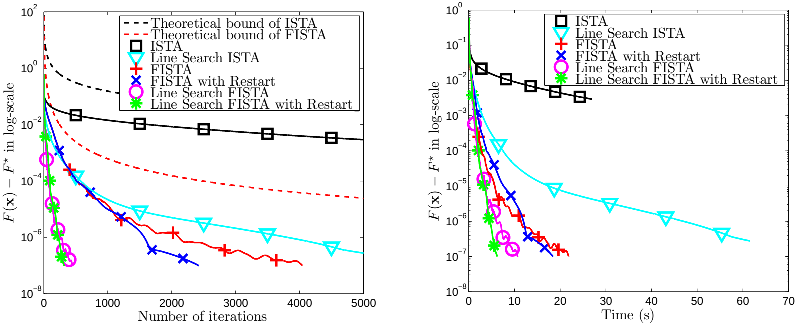
## Real data

- Real data: `w8a` with $n = 49749$ data points, $p = 300$ features
- Available at `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html`.

## Parameters

- $\lambda = 10^{-4}$.
- 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.<br>**2.** For $k = 0, 1, \cdots$, generate a sequence $\{\mathbf{x}^k\}_{k\geq 0}$ as:<br>$$\mathbf{x}^{k+1} := \mathrm{prox}_{\alpha_k g}\left(\mathbf{x}^k - \alpha_k \nabla f(\mathbf{x}^k)\right),$$<br>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 $\mathbf{y}^0 := \mathbf{x}^0$.<br>**2.** For $k = 0, 1, \cdots$, generate two sequences $\{\mathbf{x}^k\}_{k\geq 0}$ and $\{\mathbf{y}^k\}_{k\geq 0}$ as:<br>$$\begin{cases} \mathbf{x}^{k+1} := \mathrm{prox}_{\alpha_k g}\left(\mathbf{y}^k - \alpha_k \nabla f(\mathbf{y}^k)\right), \\ \mathbf{y}^{k+1} := \mathbf{x}^{k+1} + \left(\frac{\sqrt{c_f}-1}{\sqrt{c_f}+1}\right)(\mathbf{x}^{k+1} - \mathbf{x}^k), \end{cases}$$<br>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**$_\mu$ [6])

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

***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$_\mu$ [6])

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

***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$_\mu$ [6])

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

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

# Summary of the complexity

## Comparison with gradient scheme

| 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 $[\mu > 0]$ | $\mathcal{O}\left(c_f \log(\varepsilon^{-1})\right)$ | $\mathcal{O}\left(\sqrt{c_f}\log(\varepsilon^{-1})\right)$ |
| 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.

# Summary of the complexity

## Comparison with gradient scheme

| 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 $[\mu > 0]$ | $\mathcal{O}\left(c_f\log(\varepsilon^{-1})\right)$ | $\mathcal{O}\left(\sqrt{c_f}\log(\varepsilon^{-1})\right)$ |
| 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.

## 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 $\varepsilon$ is a given tolerance.

# Outline

- Today
  1. Composite convex minimization
  2. Proximal operator and computational complexity
  3. Proximal gradient methods
  4. Composite self-concordant minimization
  5. Smoothing for nonsmooth composite convex minimization
- Next week
  1. Nonsmooth constrained optimization

# The idea of the proximal-Newton method

**Assumptions A.2**

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

# The idea of the proximal-Newton method

## Assumptions A.2

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

## The idea of proximal-Newton method

- Under Assumptions A.2, we can linearize the smooth term of the optimality condition of (1): $0 \in \nabla f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star)$ as

$$0 \in \nabla f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star) \approx \nabla f(\mathbf{x}^k) + \nabla^2 f(\mathbf{x}^k)^T(\mathbf{x}^\star - \mathbf{x}^k) + \partial g(\mathbf{x}^\star).$$

- Similar to the classical Newton method in Lecture 3, we can generate an iterative sequence $\{\mathbf{x}^k\}_{k\geq 0}$ by solving the **inclusion**:

$$0 \in \nabla f(\mathbf{x}^k) + \nabla^2 f(\mathbf{x}^k)^T(\mathbf{x} - \mathbf{x}^k) + \partial g(\mathbf{x}) \tag{26}$$

  to obtain $\mathbf{x}^{k+1}$.

- The last condition is equivalent to

$$\mathbf{x}^{k+1} := \arg\min_{\mathbf{x}\in\mathbb{R}^p}\left\{\frac{1}{2}(\mathbf{x}-\mathbf{x}^k)^T\nabla^2 f(\mathbf{x}^k)(\mathbf{x}-\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T(\mathbf{x}-\mathbf{x}^k) + g(\mathbf{x})\right\}. \tag{27}$$

## Proximal-Newton-type scheme

- The sequence $\{\mathbf{x}^k\}$ generated by (27) is not necessarily convergent. Hence, a sufficient descent condition is required.
- We can replace $\nabla^2 f(\mathbf{x}^k)$ by a given approximate matrix $\mathbf{H}_k$.

## Proximal-Newton-type scheme

- The sequence $\{\mathbf{x}^k\}$ generated by (27) is not necessarily convergent. Hence, a sufficient descent condition is required.
- We can replace $\nabla^2 f(\mathbf{x}^k)$ by a given approximate matrix $\mathbf{H}_k$.

**Proximal-Newton-type scheme:**

- Let $\mathbf{H}_k \approx \nabla^2 f(\mathbf{x}^k)$ be a symmetric positive definite (SDP) matrix. From (26), we have

$$\mathbf{x}^k - \mathbf{H}_k^{-1}\nabla f(\mathbf{x}^k) \in (\mathbb{I} + \mathbf{H}_k^{-1}\partial g)(\mathbf{x}),$$

  which leads to

$$\mathbf{x}^{k+1} := \mathrm{prox}_{\mathbf{H}_k^{-1} g}\Big(\mathbf{x}^k - \mathbf{H}_k^{-1}\nabla f(\mathbf{x}^k)\Big). \quad (28)$$

- By letting $\mathbf{d}^k := \mathbf{x}^{k+1} - \mathbf{x}^k$, (28) is equivalent to

$$\mathbf{d}^k := \arg\min_{\mathbf{d}\in\mathbb{R}^p}\Big\{\frac{1}{2}\mathbf{d}^T\mathbf{H}_k\mathbf{d} + \nabla f(\mathbf{x}^k)^T\mathbf{d} + g(\mathbf{x}^k + \mathbf{d})\Big\}. \quad (29)$$

  Then $\mathbf{d}^k$ is called a proximal-Newton-type direction.
- Proximal-Newton-type algorithm generates a sequence $\{\mathbf{x}^k\}_{k\geq 0}$ starting from $\mathbf{x}^0 \in \mathbb{R}^p$ and update:

$$\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k\mathbf{d}^k, \quad (30)$$

  where $\mathbf{d}^k$ is given by (29) and $\alpha_k \in (0, 1]$ is a damed step-size.

## How to find step size $\alpha_k$?

**Lemma (Descent lemma [5])**

*Let* $\mathbf{x}^k(\alpha) := \mathbf{x}^k + \alpha\mathbf{d}^k$ *for sufficiently small* $\alpha \in (0, 1]$ *and* $\mathbf{H}_k \succ 0$*. Then, we have:*

$$F(\mathbf{x}^k(\alpha)) \leq F(\mathbf{x}^k) - (1/2)\alpha(\mathbf{d}^k)^T\mathbf{H}_k\mathbf{d}^k + \mathcal{O}(\alpha^2).$$

## How to find step size $\alpha_k$?

**Lemma (Descent lemma [5])**

*Let $\mathbf{x}^k(\alpha) := \mathbf{x}^k + \alpha\mathbf{d}^k$ for sufficiently small $\alpha \in (0,1]$ and $\mathbf{H}_k \succ 0$. Then, we have:*

$$F(\mathbf{x}^k(\alpha)) \leq F(\mathbf{x}^k) - (1/2)\alpha(\mathbf{d}^k)^T\mathbf{H}_k\mathbf{d}^k + \mathcal{O}(\alpha^2).$$

Since $\mathbf{H}_k \succ 0$, this lemma tells us that:

- If $\mathbf{d}^k \neq 0$, then the exist $\alpha > 0$ such that $F(\mathbf{x}^k(\alpha)) < F(\mathbf{x}^k)$.
- The value of $\alpha$ can be computed via **backtracking line search**.
- If $\mathbf{d}^k = 0$, then we can easily check that $\mathbf{x}^k$ is a solution of (1).

## How to find step size $\alpha_k$?

### Lemma (Descent lemma [5])

*Let* $\mathbf{x}^k(\alpha) := \mathbf{x}^k + \alpha\mathbf{d}^k$ *for sufficiently small* $\alpha \in (0,1]$ *and* $\mathbf{H}_k \succ 0$. *Then, we have:*

$$F(\mathbf{x}^k(\alpha)) \leq F(\mathbf{x}^k) - (1/2)\alpha(\mathbf{d}^k)^T\mathbf{H}_k\mathbf{d}^k + \mathcal{O}(\alpha^2).$$

Since $\mathbf{H}_k \succ 0$, this lemma tells us that:

- If $\mathbf{d}^k \neq 0$, then the exist $\alpha > 0$ such that $F(\mathbf{x}^k(\alpha)) < F(\mathbf{x}^k)$.
- The value of $\alpha$ can be computed via **backtracking line search**.
- If $\mathbf{d}^k = 0$, then we can easily check that $\mathbf{x}^k$ is a solution of (1).

### Backtracking line-search

- Let
$$r_k := \nabla f(\mathbf{x}^k)^T\mathbf{d}^k + g(\mathbf{x}^k + \mathbf{d}^k) - g(\mathbf{x}^k).$$
- Find the smallest integer number $j \geq 0$ such that $\alpha_k := \beta^j$ and
$$F(\mathbf{x}^k + \alpha_k\mathbf{d}^k) \leq F(\mathbf{x}^k) + c\alpha_k r_k, \tag{31}$$
where $c \in (0, 0.5]$ and $\beta \in (0,1)$ are two given constants (e.g., $c = 0.1$ and $\beta = 0.5$).

## The proximal-Newton-type algorithm

We can summary the **proximal-Newton-type method** as follows:

| **Proximal-Newton algorithm (PNA)** |
| --- |
| **1.** Given $\mathbf{x}^0 \in \mathbb{R}^p$ as a starting point. Choose $c := 0.1$ and $\beta := 0.5$ |
| **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. Find the smallest integer number $j \geq 0$ such that $$F(\mathbf{x}^k + \beta^j \mathbf{d}^k) \leq F(\mathbf{x}^k) + c\beta^j r_k$$ and set $\alpha_k := \beta^j$. |
| 2.4. Update $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k \mathbf{d}^k$. |

## The proximal-Newton-type algorithm

We can summary the **proximal-Newton-type method** as follows:

**Proximal-Newton algorithm (PNA)**

**1.** Given $\mathbf{x}^0 \in \mathbb{R}^p$ as a starting point. Choose $c := 0.1$ and $\beta := 0.5$

**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. Find the smallest integer number $j \geq 0$ such that

$$F(\mathbf{x}^k + \beta^j \mathbf{d}^k) \leq F(\mathbf{x}^k) + c\beta^j r_k$$

and set $\alpha_k := \beta^j$.

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

- If $\mathbf{H}_k \equiv \nabla^2 f(\mathbf{x}^k)$, then **PNA** becomes a pure proximal-Newton algorithm.
- If $\mathbf{H}_k \approx \nabla^2 f(\mathbf{x}^k)$, then **PNA** becomes a proximal-quasi-Newton algorithm.
- Main computation is Step 2.2, which requires 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

### Assumption A.3.

- Problem (1): $\min_{\mathbf{x}}\{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\}$ admits a solution $\mathbf{x}^\star$.
- The subproblem $\mathrm{prox}_{\mathbf{H}_k^{-1} g}\left(\mathbf{x}^k + \mathbf{H}_k^{-1}\nabla f(\mathbf{x}^k)\right)$ is solved exactly for all $k \geq 0$.

## Convergence analysis

**Assumption A.3.**

- Problem (1): $\min_{\mathbf{x}}\{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\}$ admits a solution $\mathbf{x}^\star$.
- The subproblem $\mathrm{prox}_{\mathbf{H}_k^{-1}g}\left(\mathbf{x}^k + \mathbf{H}_k^{-1}\nabla f(\mathbf{x}^k)\right)$ is solved exactly for all $k \geq 0$.

**Theorem (Global convergence [5])**

***Assumptions:***

- *The sequence $\{\mathbf{x}^k\}_{k\geq 0}$ is generated by PNA.*
- *Assumption A.3. is satisfied.*
- *Exists $\mu > 0$ such that $\mathbf{H}_k \succeq \mu\mathbb{I}$ for all $k \geq 0$.*

***Conclusion:***

- $\{\mathbf{x}^k\}_{k\geq 0}$ *globally converges to a solution $\mathbf{x}^\star$ of* (1).

- So far, we have not yet obtained a **global convergence rate** of proximal-Newton methods.

## Convergence analysis

### Assumption A.3.

- Problem (1): $\min_{\mathbf{x}}\{F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\}$ admits a solution $\mathbf{x}^\star$.
- The subproblem $\mathrm{prox}_{\mathbf{H}_k^{-1}g}\left(\mathbf{x}^k + \mathbf{H}_k^{-1}\nabla f(\mathbf{x}^k)\right)$ is solved exactly for all $k \geq 0$.

### Theorem (Local convergence [5])

***Assumptions:***

- *The sequence $\{\mathbf{x}^k\}_{k\geq 0}$ is generated by PNA.*
- *Assumption A.3. is satisfied.*
- *Exist $0 < \mu \leq L_2 < +\infty$ such that $\mu\mathbb{I} \preceq \mathbf{H}_k \preceq L_2\mathbb{I}$ for all sufficiently large $k$.*

***Conclusion:***

- *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:*

$$\lim_{k\to+\infty} \frac{\|(\mathbf{H}_k - \nabla^2 f(\mathbf{x}^\star))(\mathbf{x}^{k+1} - \mathbf{x}^k)\|}{\|\mathbf{x}^{k+1} - \mathbf{x}^k\|} = 0, \tag{32}$$

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

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

- Solving $\mathrm{prox}_{\mathbf{H}_k^{-1} g}\left(\mathbf{x}^k + \mathbf{H}_k^{-1}\nabla f(\mathbf{x}^k)\right)$ exactly for non-diagonal matrix $\mathbf{H}_k$ is impractical.
- This problem is solved iteratively by using, e.g., FISTA except for the special cases of $\mathbf{H}_k$.

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

- Solving $\mathrm{prox}_{\mathbf{H}_k^{-1} g}\left(\mathbf{x}^k + \mathbf{H}_k^{-1}\nabla f(\mathbf{x}^k)\right)$ exactly for non-diagonal matrix $\mathbf{H}_k$ is impractical.
- This problem is solved iteratively by using, e.g., FISTA except for the special cases of $\mathbf{H}_k$.

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

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

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

## Advantages and disadvantages

### Advantages

- PNA has fast local convergence rate (super-linear or quadratic)
- Numerical robustness under the inexactness/noise (inexact proximal-Newton method [5]).
- Quasi-Newton method is useful if the evaluation of $\nabla^2 f$ is expensive.
- Suitable for problems with many data points but few parameters. For example, problems of the form:

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

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

## Advantages and disadvantages

### Advantages

- PNA has fast local convergence rate (super-linear or quadratic)
- Numerical robustness under the inexactness/noise (inexact proximal-Newton method [5]).
- Quasi-Newton method is useful if the evaluation of $\nabla^2 f$ is expensive.
- Suitable for problems with many data points but few parameters. For example, problems of the form:

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

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

### Disadvantages

- Expensive iteration compared to proximal-gradient methods.
- Global convergence rate may be worse than accelerated proximal-gradient methods.
- Require a good initial point to get a fast local convergence, which is hard to find.
- Require 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:*

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

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

# 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:*

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

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

## Optimization formulation

$$\min_{\mathbf{x}\in\mathbb{R}^p} \Big\{ \underbrace{\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(b_i(\mathbf{a}_i^T\mathbf{x} + \mu))}_{f(\mathbf{x})} + \underbrace{\lambda\|\mathbf{x}\|_1}_{g(\mathbf{x})} \Big\}, \quad (33)$$

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

## Example: Sparse logistic regression

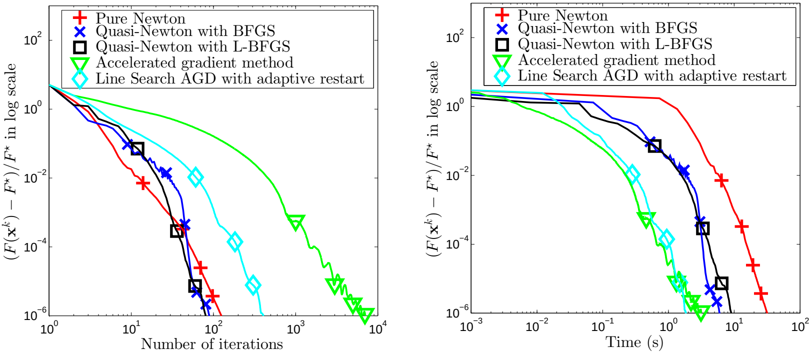
### Real data

- 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: $\ell_1$-regularized least squares

## Problem ($\ell_1$-regularized least squares)

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

$$F^\star := \min_{\mathbf{x} \in \mathbb{R}^p} \left\{ F(\mathbf{x}) := \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_1 \right\}, \tag{34}$$

*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 2: $\ell_1$-regularized least squares

## Problem ($\ell_1$-regularized least squares)

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

$$F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ F(\mathbf{x}) := \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_1 \right\}, \tag{34}$$

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

## Complexity per iterations

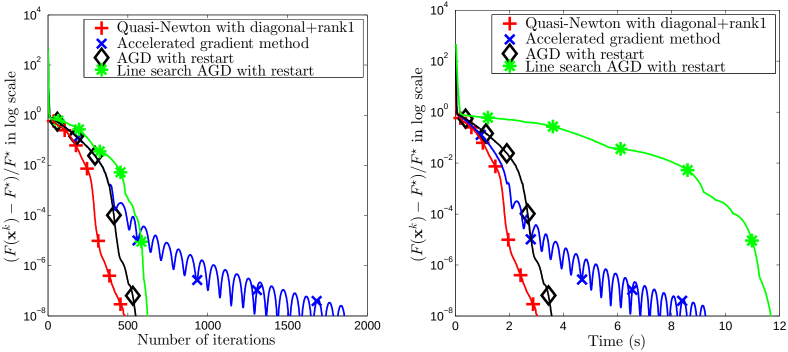
- Evaluating $\nabla f(\mathbf{x}^k) = \mathbf{A}^T(\mathbf{A}\mathbf{x}^k - \mathbf{b})$ requires one $\mathbf{A}\mathbf{x}$ and one $\mathbf{A}^T\mathbf{y}$.
- One soft-thresholding operator $\mathrm{prox}_{\lambda g}(\mathbf{x}) = \mathrm{sign}(\mathbf{x}) \otimes \max\{|\mathbf{x}| - \lambda, 0\}$.
- **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 $s$-sparse vector generated randomly.
- $\mathbf{b} := \mathbf{A}\mathbf{x}^\star + \mathcal{N}(0, 10^{-3})$.

# Example 2: $\ell_1$-regularized least squares - Numerical results

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



# Outline

- Today
  1. Composite convex minimization
  2. Proximal operator and computational complexity
  3. Proximal gradient methods
  4. Composite self-concordant minimization
  5. Smoothing for nonsmooth composite convex minimization
- Next week
  1. Nonsmooth constrained optimization

## Composite self-concordant minimization

### Composite self-concordant minimization (CSM) problem [11]

$$F^\star := \min_{\mathbf{x}\in\mathrm{dom}(F)} \Big\{ F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x}) \Big\}, \tag{35}$$

- $f \in \mathcal{F}_2(\mathrm{dom}(f))$ - self-concordant on $\mathrm{dom}(f) := \{\mathbf{x} \in \mathbb{R}^p \ : \ f(\mathbf{x}) < +\infty\}$
- $g \in \mathcal{F}_{\text{prox}}(\mathbb{R}^p)$
- $\mathrm{dom}(F) := \mathrm{dom}(f) \cap \mathrm{dom}(g)$

## Composite self-concordant minimization

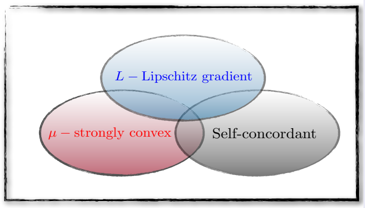
### Composite self-concordant minimization (CSM) problem [11]

$$F^\star := \min_{\mathbf{x} \in \mathrm{dom}(F)} \Big\{ F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x}) \Big\}, \tag{35}$$

- $f \in \mathcal{F}_2(\mathrm{dom}(f))$ - self-concordant on $\mathrm{dom}(f) := \{\mathbf{x} \in \mathbb{R}^p \;:\; f(\mathbf{x}) < +\infty\}$
- $g \in \mathcal{F}_{\mathrm{prox}}(\mathbb{R}^p)$
- $\mathrm{dom}(F) := \mathrm{dom}(f) \cap \mathrm{dom}(g)$

### Why is composite self-concordant minimization?

- A self-concordant function is not necessarily Lipschitz gradient.




- Cover many well-known examples.

## Self-concordant functions in higher dimensions

### Definition (Self-concordant functions [7, 6])

- A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be self-concordant with parameter $M \geq 0$ if

$$|\varphi'''(t)| \leq M\varphi''(t)^{3/2},$$

  where $\varphi(t) := f(\mathbf{x} + t\mathbf{v})$ for all $t \in \mathbb{R}$, $\mathbf{x} \in \text{dom}(f)$ and $\mathbf{v} \in \mathbb{R}^n$ and $\mathbf{x} + t\mathbf{v} \in \text{dom}(f)$.
- When $M = 2$, the function $f$ is said to be a standard self-concordant.

## Self-concordant functions in higher dimensions

### Definition (Self-concordant functions [7, 6])

- A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be self-concordant with parameter $M \geq 0$ if

$$|\varphi'''(t)| \leq M\varphi''(t)^{3/2},$$

where $\varphi(t) := f(\mathbf{x} + t\mathbf{v})$ for all $t \in \mathbb{R}$, $\mathbf{x} \in \mathrm{dom}(f)$ and $\mathbf{v} \in \mathbb{R}^n$ and $\mathbf{x} + t\mathbf{v} \in \mathrm{dom}(f)$.

- When $M = 2$, the function $f$ is said to be a standard self-concordant.

### Example

The function $f(x) = -\log x$ is self-concordant. To see this, observe:

$$f''(x) = 1/x^2, \quad f'''(x) = -2/x^3.$$

Thus:

$$\frac{|f'''(x)|}{2f''(x)^{3/2}} = \frac{2/x^3}{2(1/x^2)^{3/2}} = 1$$

# Self-concordant functions in higher dimensions

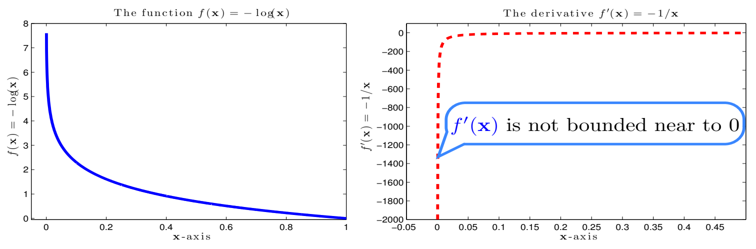
## Definition (Self-concordant functions [7, 6])

- A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be self-concordant with parameter $M \geq 0$ if

$$|\varphi'''(t)| \leq M\varphi''(t)^{3/2},$$

  where $\varphi(t) := f(\mathbf{x} + t\mathbf{v})$ for all $t \in \mathbb{R}$, $\mathbf{x} \in \mathrm{dom}(f)$ and $\mathbf{v} \in \mathbb{R}^n$ and $\mathbf{x} + t\mathbf{v} \in \mathrm{dom}(f)$.
- When $M = 2$, the function $f$ is said to be a standard self-concordant.

## $f(\mathbf{x}) = -\log(\mathbf{x})$ and its derivative $f'(\mathbf{x})$



# Self-concordant functions in higher dimensions

## Definition (Self-concordant functions [7, 6])

- A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be self-concordant with parameter $M \geq 0$ if

$$|\varphi'''(t)| \leq M\varphi''(t)^{3/2},$$

where $\varphi(t) := f(\mathbf{x} + t\mathbf{v})$ for all $t \in \mathbb{R}$, $\mathbf{x} \in \mathrm{dom}(f)$ and $\mathbf{v} \in \mathbb{R}^n$ and $\mathbf{x} + t\mathbf{v} \in \mathrm{dom}(f)$.

- When $M = 2$, the function $f$ is said to be a standard self-concordant.

## Example

Similarly, the following example functions are self-concordant

1. $f(x) = x \log x - \log x$,
2. $f(x) = \sum_{i=1}^{m} \log(b_i - \mathbf{a}_i^T\mathbf{x})$ with domain $\mathrm{dom}(f) = \left\{\mathbf{x} \ : \ \mathbf{a}_i^T\mathbf{x} < b_i, i = 1, \ldots, m\right\}$,
3. $f(\mathbf{X}) = -\log\det(\mathbf{X})$ with domain $\mathrm{dom}(f) = \mathbb{S}_n^{++}$,
4. $f(\mathbf{x}) = -\log\left(\mathbf{x}^T\mathbf{P}\mathbf{x} + \mathbf{q}^T\mathbf{x} + r\right)$ with domain $\mathrm{dom}(f) = \left\{\mathbf{x} \ : \ \mathbf{x}^T\mathbf{P}\mathbf{x} + \mathbf{q}^T\mathbf{x} + r > 0\right\}$ and $-\mathbf{P} \in \mathbb{S}_n^{++}$.

## Two well-known examples

### Graphical model selection

$$\boxed{\min_{\Theta \succ 0} \Big\{ \underbrace{\mathrm{tr}(\Sigma\Theta) - \log\det(\Theta)}_{f(\mathbf{x})} + \underbrace{\lambda \|\mathrm{vec}(\Theta)\|_1}_{g(\mathbf{x})} \Big\}} \tag{36}$$

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

## Two well-known examples

### Graphical model selection

$$\min_{\Theta \succ 0} \left\{ \underbrace{\mathrm{tr}(\Sigma\Theta) - \log\det(\Theta)}_{f(\mathbf{x})} + \underbrace{\lambda\|\mathrm{vec}(\Theta)\|_1}_{g(\mathbf{x})} \right\} \tag{36}$$

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

### Poisson imaging reconstruction (with TV-norm regularizer)

$$\min_{\mathbf{x} \in \mathbb{R}^{n \times p}} \left\{ \underbrace{\sum_{i=1}^{n} (\mathbf{K}\mathbf{x})_i - \sum_{i=1}^{n} y_i \log((\mathbf{K}\mathbf{x})_i)}_{f(\mathbf{x})} + \underbrace{\lambda\|\mathbf{x}\|_{\mathrm{TV}}}_{g(\mathbf{x})} \right\} \tag{37}$$

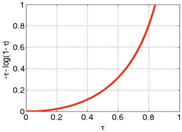
- $\mathbf{K}$ is a linear operator, $\mathbf{y} = (y_1, \ldots, y_n)^T \in \mathbb{Z}_+^n$ is the observed vector of photon counts.
- $\lambda > 0$ is a regularization parameter,
- $\|\mathbf{x}\|_{\mathrm{TV}}$ is the TV-norm of $\mathbf{x}$ (see the above example).

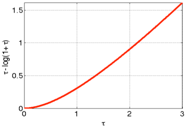
# Some geometric intuition behind self-concordant functions

## Local norm

**Local norm:** $\|\mathbf{u}\|_{\mathbf{x}} := \left[\mathbf{u}^T \nabla^2 f(\mathbf{x}) \mathbf{u}\right]^{1/2}$

**Utility functions:** $\omega_*(\tau) = -\tau - \ln(1-\tau),\ \tau \in [0,1)$ $\qquad$ $\omega(\tau) = \tau - \ln(1+\tau),\ \tau \geq 0$



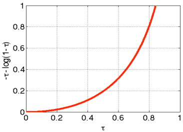



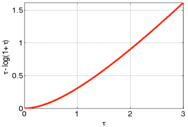
# Some geometric intuition behind self-concordant functions

## Local norm

**Local norm:** $\|\mathbf{u}\|_{\mathbf{x}} := \left[\mathbf{u}^T \nabla^2 f(\mathbf{x}) \mathbf{u}\right]^{1/2}$

**Utility functions:** $\omega_*(\tau) = -\tau - \ln(1-\tau),\ \tau \in [0, 1)$ $\quad$ $\omega(\tau) = \tau - \ln(1+\tau),\ \tau \geq 0$




## Basic properties [6]

| | | |
|---|---|---|
| Lower surrogate | $f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + \omega\left(\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}\right)$ | $\mathbf{x}, \mathbf{y} \in \mathrm{dom}(f)$ |
| Upper surrogate | $f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + \omega_*\left(\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}\right)$ | $\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}} < 1$ |
| Hessian surrogates | $(1-\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}})^2 \nabla^2 f(\mathbf{x}) \preceq \nabla^2 f(\mathbf{y}) \preceq (1-\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}})^{-2} \nabla^2 f(\mathbf{x})$ | $\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}} < 1$ |

Local

**Bound on gradient:**

$$\frac{\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}^2}{1+\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}} \leq \langle \nabla f(\mathbf{y}) - \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x} \rangle \leq \frac{\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}^2}{1-\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}}, \quad \forall \mathbf{x}, \mathbf{y} \in \mathrm{dom}(f).$$

The right-hand side inequality holds for $\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}} < 1$.

# Variable metric proximal-gradient algorithm for SCM

## Variable metric proximal operator

Given $\mathbf{H} \succ 0$ and $g \in \mathcal{F}(\mathbb{R}^p)$. The variable metric proximal operator of $g$ is defined as

$$\boxed{\mathrm{prox}_{\mathbf{H}g}(\mathbf{x}) := \arg\min_{\mathbf{y}\in\mathbb{R}^p} \left\{ g(\mathbf{y}) + (1/2)(\mathbf{y}-\mathbf{x})^T\mathbf{H}^{-1}(\mathbf{y}-\mathbf{x}) \right\}} \tag{38}$$

# Variable metric proximal-gradient algorithm for SCM

## Variable metric proximal operator

Given $\mathbf{H} \succ 0$ and $g \in \mathcal{F}(\mathbb{R}^p)$. The variable metric proximal operator of $g$ is defined as

$$\boxed{\mathrm{prox}_{\mathbf{H}g}(\mathbf{x}) := \arg\min_{\mathbf{y}\in\mathbb{R}^p}\left\{g(\mathbf{y}) + (1/2)(\mathbf{y}-\mathbf{x})^T\mathbf{H}^{-1}(\mathbf{y}-\mathbf{x})\right\}} \qquad (38)$$

## Property (Basis properties of variable metric proximal operator)

1. $\mathrm{prox}_{\mathbf{H}g}(\mathbf{x})$ *is well-defined and single-valued (i.e.,* (38) *has unique solution).*
2. *Optimality condition:*

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

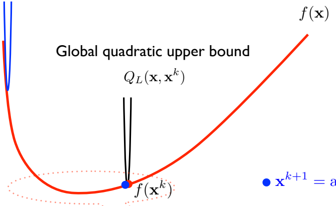
3. $\mathbf{x}^\star$ *is a fixed point of* $\mathrm{prox}_{\mathbf{H}g}(\cdot)$*:*

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

4. *Non-expansiveness:*

$$\|\mathrm{prox}_{\mathbf{H}g}(\mathbf{x}) - \mathrm{prox}_{\mathbf{H}g}(\tilde{\mathbf{x}})\|_{\mathbf{H}}^* \le \|\mathbf{x}-\tilde{\mathbf{x}}\|_{\mathbf{H}}, \quad \forall \mathbf{x}, \tilde{\mathbf{x}} \in \mathbb{R}^p.$$

# How can we better adapt to the local geometry?

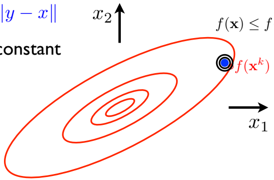



- $\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \left\{ f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k \rangle + \frac{L}{2}\|\mathbf{x} - \mathbf{x}^k\|_2^2 \right\}$

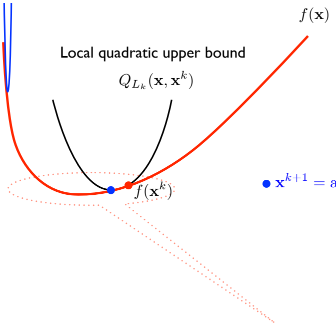
$\|\nabla f(x) - \nabla f(y)\| \leq L\|y - x\|$

L is a global worst-case constant

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



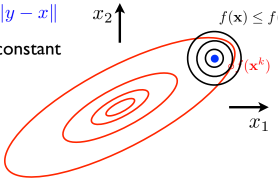
# How can we better adapt to the local geometry?




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

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

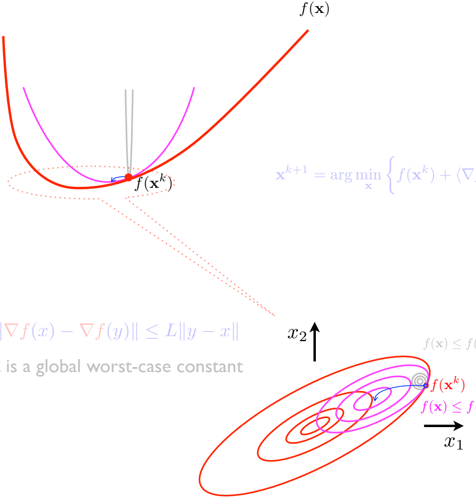
L is a global worst-case constant




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

applies only locally

# How can we better adapt to the local geometry?




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

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

L is a global worst-case constant

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

$$f(\mathbf{x}) \leq f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T(\mathbf{x} - \mathbf{x}^k) + \frac{1}{2}\|\mathbf{x} - \mathbf{x}^k\|_{H_k^{-1}}^2$$

# Variable metric proximal-gradient algorithm

**Variable metric proximal-gradient algorithm [11]**

1. Choose $\mathbf{x}^0 \in \mathbb{R}^p$ as a starting point and $\mathbf{H}_0 \succ 0$.
2. For $k = 0, 1, \cdots$, perform:

$$\begin{cases} \mathbf{d}^k & := \mathrm{prox}_{\mathbf{H}_k g}\left(\mathbf{x}^k - \mathbf{H}_k \nabla f(\mathbf{x}^k)\right) - \mathbf{x}^k, \\ \mathbf{x}^{k+1} & := \mathbf{x}^k + \alpha_k \mathbf{d}^k, \end{cases} \tag{39}$$

where $\alpha_k \in (0, 1]$ is a given step size. Update $\mathbf{H}_{k+1} \succ 0$ if necessary.

# Variable metric proximal-gradient algorithm

**Variable metric proximal-gradient algorithm [11]**

1. Choose $\mathbf{x}^0 \in \mathbb{R}^p$ as a starting point and $\mathbf{H}_0 \succ 0$.
2. For $k = 0, 1, \cdots$, perform:

$$\begin{cases} \mathbf{d}^k & := \mathrm{prox}_{\mathbf{H}_k g}\left(\mathbf{x}^k - \mathbf{H}_k \nabla f(\mathbf{x}^k)\right) - \mathbf{x}^k, \\ \mathbf{x}^{k+1} & := \mathbf{x}^k + \alpha_k \mathbf{d}^k, \end{cases} \tag{39}$$

where $\alpha_k \in (0, 1]$ is a given step size. Update $\mathbf{H}_{k+1} \succ 0$ if necessary.

## Common choices of $\mathbf{H}_k$

- $\mathbf{H}_k := \lambda_k \mathbb{I}$, we have $\mathrm{prox}_{\mathbf{H}g} \equiv \mathrm{prox}_{\lambda g}$ and obtain a proximal-gradient method.
- $\mathbf{H}_k := \mathbf{D}$ a diagonal matrix, $\mathrm{prox}_{\mathbf{H}g}$ can be transformed into $\mathrm{prox}_{\lambda g}$ (by scaling the variables) and we obtain a proximal-gradient method.
- $\mathbf{H}_k := \nabla^2 f(\mathbf{x}^k)^{-1}$, we obtain a proximal-Newton method.
- $\mathbf{H}_k \approx \nabla^2 f(\mathbf{x}^k)^{-1}$, we obtain a proximal quasi-Newton method.

# Proximal-Newton method for CSM

**Proximal-Newton algorithm (PNA)**

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

$$\begin{cases} \mathbf{B}_k & := \nabla^2 f(\mathbf{x}^k), \\ \mathbf{d}^k & := \mathrm{prox}_{\mathbf{B}_k^{-1} g}\left(\mathbf{x}^k - \mathbf{B}_k^{-1}\nabla f(\mathbf{x}^k)\right) - \mathbf{x}^k, \quad \text{(PN direction)} \\ \lambda_k & := \|\mathbf{d}\|_{\mathbf{x}^k}, \quad \text{(PN decrement)} \\ \alpha_k & = (1+\lambda_k)^{-1}, \quad \text{(step-size)} \\ \mathbf{x}^{k+1} & := \mathbf{x}^k + \alpha_k \mathbf{d}^k. \end{cases} \tag{40}$$

# Proximal-Newton method for CSM

**Proximal-Newton algorithm (PNA)**

1. Choose $\mathbf{x}^0 \in \text{dom}(F)$ as a starting point.
2. For $k = 0, 1, \cdots$, perform:

$$\begin{cases} \mathbf{B}_k & := \nabla^2 f(\mathbf{x}^k), \\ \mathbf{d}^k & := \text{prox}_{\mathbf{B}_k^{-1} g}\left(\mathbf{x}^k - \mathbf{B}_k^{-1}\nabla f(\mathbf{x}^k)\right) - \mathbf{x}^k, \quad \text{(PN direction)} \\ \lambda_k & := \|\mathbf{d}\|_{\mathbf{x}^k}, \quad \text{(PN decrement)} \\ \alpha_k & = (1+\lambda_k)^{-1}, \quad \text{(step-size)} \\ \mathbf{x}^{k+1} & := \mathbf{x}^k + \alpha_k \mathbf{d}^k. \end{cases} \tag{40}$$

## Complexity per iteration

- Evaluation of $\nabla^2 f(\mathbf{x}^k)$ and $\nabla f(\mathbf{x}^k)$ (closed form expressions).
- Computing $\text{prox}_{\mathbf{H}_k g}$ requires to solve a strongly convex program (38).
- Computing proximal-Newton decrement $\lambda_k$ requires $(\mathbf{d}^k)^T \nabla f^2(\mathbf{x}^k)\mathbf{d}^k$.

## Global convergence

**Lemma (Descent lemma [11])**

*Let* $\{\mathbf{x}^k\}_{k\geq 0}$ *be the sequence generated by PNA. Then*

$$\boxed{F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \omega(\lambda_k)} \tag{41}$$

*where* $\omega(\tau) := \tau - \ln(1+\tau) > 0$ *for* $\tau > 0$.

## Global convergence

### Lemma (Descent lemma [11])

*Let* $\{\mathbf{x}^k\}_{k\geq 0}$ *be the sequence generated by PNA. Then*

$$\boxed{F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \omega(\lambda_k)} \quad (41)$$

*where* $\omega(\tau) := \tau - \ln(1+\tau) > 0$ *for* $\tau > 0$.

### Consequences

- $[F(\mathbf{x}^{k+1}) - F^\star] \leq [F(\mathbf{x}^k) - F^\star] - \omega(\lambda_k)$ for all $k \geq 0$.
- $[F(\mathbf{x}^k) - F(\mathbf{x}^\star] \leq [F(\mathbf{x}^0) - F^\star] - \sum_{j=0}^{k-1}\omega(\lambda_j)$.
- If $\lambda_k \geq \lambda > 0$ for $k = 0, \ldots, K$, then

  $$[F(\mathbf{x}^K) - F^\star] \leq [F(\mathbf{x}^0) - F^\star] - K\omega(\lambda).$$

  The **number of iterations** to reach $F(\mathbf{x}^K) - F^\star \leq \varepsilon$ is

  $$K := \left\lfloor \frac{[F(\mathbf{x}^0) - F^\star] - \varepsilon}{\omega(\lambda)} \right\rfloor + 1.$$
- Global convergence rate is just sublinear, i.e. $\mathcal{O}(1/k)$.

## Proof of (41)

### Sketch of proof.

- Let $\mathbf{s}^k := \mathbf{x}^k + \mathbf{d}^k$. We have $\mathbf{x}^{k+1} - \mathbf{x}^k = \alpha_k \mathbf{d}^k$ and $\mathbf{x}^{k+1} = (1-\alpha_k)\mathbf{x}^k + \alpha_k \mathbf{s}^k$.
- By convexity of $g$:

$$g(\mathbf{x}^{k+1}) \leq (1-\alpha_k) g(\mathbf{x}^k) + \alpha_k g(\mathbf{s}^k), \ \ \alpha_k \in (0,1]. \tag{42}$$

- By subgradient definition:

$$g(\mathbf{s}^k) \leq g(\mathbf{x}^k) + \mathbf{v}(\mathbf{s}^k)^T(\mathbf{s}^k - \mathbf{x}^k), \ \ \forall \ \mathbf{v}(\mathbf{s}^k) \in \partial g(\mathbf{s}^k). \tag{43}$$

- Substituting (43) into (42) we get

$$g(\mathbf{x}^{k+1}) \leq g(\mathbf{x}^k) + \alpha_k \mathbf{v}(\mathbf{s}^k)^T \mathbf{d}^k. \tag{44}$$

- By self-concordance of $f$ (upper bound inequality):

$$f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)(\mathbf{x}^{k+1} - \mathbf{x}^k) + \omega_*(\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}), \tag{45}$$

under condition $\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k} < 1$.

□

## Proof of (41) (cont)

### Sketch of proof (cont).

- Summing up (44) and (45) and using $F := f + g$, we get

$$F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) + \alpha_k[\nabla f(\mathbf{x}^k) + \mathbf{v}(\mathbf{s}^k)]^T\mathbf{d}^k + \omega_*(\alpha_k\|\mathbf{d}^k\|_{\mathbf{x}^k}). \tag{46}$$

- From the optimality property 2 of (38) we have

$$\nabla f(\mathbf{x}^k) + \mathbf{v}(\mathbf{s}^k) = -\nabla^2 f(\mathbf{x}^k)\mathbf{d}^k. \tag{47}$$

- Plug (48) into (46) and use $\lambda_k := \|\mathbf{d}^k\|_{\mathbf{x}^k}$, we get

$$F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \alpha_k\lambda_k^2 + \omega_*(\alpha_k\lambda_k). \tag{48}$$

- Let $\psi(\alpha) := \alpha\lambda_k^2 - \omega_*(\alpha\lambda_k) = \alpha\lambda_k^2 + \alpha\lambda_k + \ln(1-\alpha\lambda_k)$. This function attains the maximum at $\alpha_k = (1+\lambda_k)^{-1}$ and $\psi(\alpha_k) = \lambda_k - \ln(1+\lambda_k)$. Hence, we have

$$F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \omega(\lambda_k),$$

which is (41).

□

# Local convergence

**Theorem (Local quadratic convergence [11])**

*Let* $\{\mathbf{x}^k\}$ *be the sequence generated by* ***PNA***. *If* $\|\mathbf{x}^0 - \mathbf{x}^\star\|_{\mathbf{x}^\star} \leq \sigma_0 := 0.08763$, *then*

$$\boxed{\|\mathbf{x}^{k+1} - \mathbf{x}^\star\|_{\mathbf{x}^\star} \leq c^* \|\mathbf{x}^k - \mathbf{x}^\star\|^2_{\mathbf{x}^\star}}, \quad k \geq 0,$$

*where* $c^* := 3.57$.
*Consequently,* $\{\mathbf{x}^k\}_{k\geq 0}$ *converges to* $\mathbf{x}^\star$ *at a quadratic rate.*

# Local convergence

## Theorem (Local quadratic convergence [11])

*Let* $\{\mathbf{x}^k\}$ *be the sequence generated by* ***PNA****. If* $\|\mathbf{x}^0 - \mathbf{x}^\star\|_{\mathbf{x}^\star} \leq \sigma_0 := 0.08763$*, then*

$$\boxed{\|\mathbf{x}^{k+1} - \mathbf{x}^\star\|_{\mathbf{x}^\star} \leq c^* \|\mathbf{x}^k - \mathbf{x}^\star\|^2_{\mathbf{x}^\star}}, \quad k \geq 0,$$

*where* $c^* := 3.57$*.*
*Consequently,* $\{\mathbf{x}^k\}_{k\geq 0}$ *converges to* $\mathbf{x}^\star$ *at a quadratic rate.*

## Quadratic convergence region

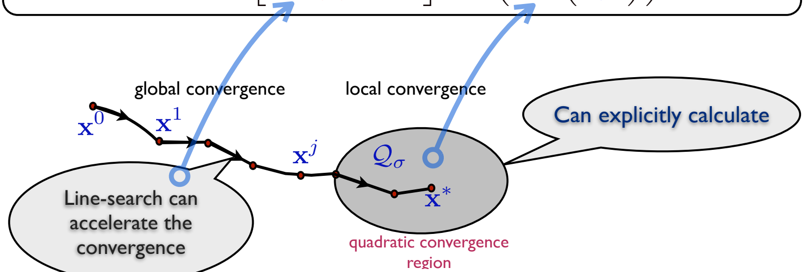
Let $\sigma := 0.08763$. Then the **quadratic convergence region** $\mathcal{Q}_\sigma$ is defined as:

$$\boxed{\mathcal{Q}_\sigma := \{\mathbf{x} \in \mathrm{dom}(F) \ : \ \|\mathbf{x} - \mathbf{x}^\star\|_{\mathbf{x}^\star} \leq \sigma\}.}$$

For any $\mathbf{x}^0 \in \mathcal{Q}_\sigma$, $\{\mathbf{x}^k\}$ converges to $\mathbf{x}^\star$ at a quadratic rate.

# Overall analytical worst-case complexity

$$\#\text{iterations} = \left\lfloor \frac{F(\mathbf{x}^0) - F^\star}{0.021} \right\rfloor + O\left(\ln\ln\left(\frac{4.56}{\varepsilon}\right)\right)$$




$$\mathcal{Q}_\sigma := \{\mathbf{x} \in \mathrm{dom}(F) \ : \ \|\mathbf{x} - \mathbf{x}^\star\|_{\mathbf{x}^\star} \leq \sigma\}$$

## Enhancements

### Two new line-search strategies

The optimal step-size $\alpha_k^* := (1+\lambda_k)^{-1}$ provides a **lower bound**. Perform line-search on $[\alpha_k^*, 1]$.

- **Forward line-search**: Start from $\alpha_k$ and increase the step-size until meet $1$.
- **Enhanced backtracking**: Start from $1$ and decrease the step size until meet $\alpha_k^*$

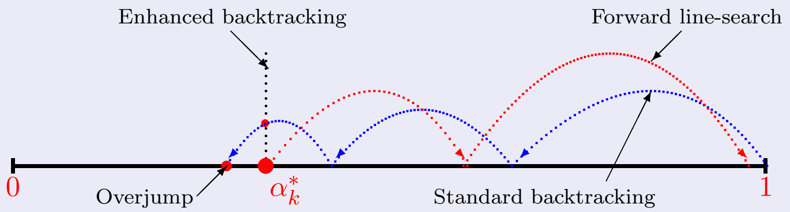
## Enhancements

### Two new line-search strategies

The optimal step-size $\alpha_k^* := (1+\lambda_k)^{-1}$ provides a **lower bound**. Perform line-search on $[\alpha_k^*, 1]$.

- **Forward line-search**: Start from $\alpha_k$ and increase the step-size until meet $1$.
- **Enhanced backtracking**: Start from $1$ and decrease the step size until meet $\alpha_k^*$

### Illustration of three line-search strategies



# Example: Graphical model selection

## Graphical model selection

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

# Example: Graphical model selection

## Graphical model selection

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

## Computational cost

- $\nabla f(\Theta) = \mathrm{vec}(\Sigma - \Theta_k^{-1})$ and $\nabla^2 f(\Theta^k) = \Theta_k^{-1} \otimes \Theta_k^{-1}$ ($\otimes$-Kronecker product).
- Compute the **search direction** $\mathbf{d}_k$.

$$\mathbf{U}_k = \operatorname*{argmin}_{\|\mathrm{vec}(\mathbf{U})\|_1 \leq 1} \Big\{ (1/2)\mathrm{trace}((\Theta_k \mathbf{U})^2) + \mathrm{trace}(\mathbf{Q}_k \mathbf{U}) \Big\},$$

where $\mathbf{Q}_k := \lambda^{-1}(\Theta_k \Sigma \Theta_k - 2\Theta_k)$. Then $\mathbf{d}^k := -((\Theta_k\Sigma - \mathbb{I})\Theta_k + \lambda\Theta_k\mathbf{U}_k\Theta_k)$.
- The proximal-Newton decrement $\lambda_k$:

$$\lambda_k := (p - 2\mathrm{trace}(\mathbf{W}_k) + \mathrm{trace}(\mathbf{W}_k^2))^{1/2}, \quad \mathbf{W}_k := \Theta_k(\Sigma + \lambda\mathbf{U}_k).$$

# Example: Graphical model selection

## Graphical model selection

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

## Computational cost

- $\nabla f(\Theta) = \mathrm{vec}(\Sigma - \Theta_k^{-1})$ and $\nabla^2 f(\Theta^k) = \Theta_k^{-1} \otimes \Theta_k^{-1}$ ($\otimes$-Kronecker product).
- Compute the **search direction** $\mathbf{d}_k$ via dualization:

$$\mathbf{U}_k = \operatorname*{argmin}_{\|\mathrm{vec}(\mathbf{U})\|_1 \leq 1} \Big\{ (1/2)\mathrm{trace}((\Theta_k\mathbf{U})^2) + \mathrm{trace}(\mathbf{Q}_k\mathbf{U}) \Big\},$$

where $\mathbf{Q}_k := \lambda^{-1}(\Theta_k\Sigma\Theta_k - 2\Theta_k)$. Then $\mathbf{d}^k := -((\Theta_k\Sigma - \mathbb{I})\Theta_k + \lambda\Theta_k\mathbf{U}_k\Theta_k)$.

- The proximal-Newton decrement $\lambda_k$:

$$\lambda_k := (p - 2\mathrm{trace}(\mathbf{W}_k) + \mathrm{trace}(\mathbf{W}_k^2))^{1/2}, \quad \mathbf{W}_k := \Theta_k(\Sigma + \lambda\mathbf{U}_k).$$

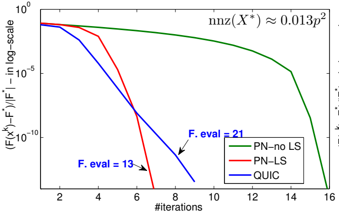
Only need **matrix-matrix multiplications**
**No** Cholesky factorizations or matrix inversions

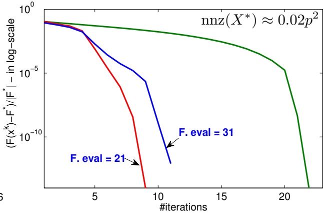
# Test on the real-data: Lymph

## Our method vs QUIC [Hseih2011]

- QUIC subproblem solver: special block-coordinate descent
- Our subproblem solver: general proximal algorithms

## Convergence behaviour [rho = 0.5]: Lymph [p = 587] (left), Leukemia [p = 1255] (right)




## Step-size selection strategies: Arabidopsis [p = 834], Leukemia [p = 1255], Hereditary [p = 1869]

| LS Scheme | Synthetic ($\rho = 0.01$) | | | Arabidopsis ($\rho = 0.5$) | | | Leukemia ($\rho = 0.1$) | | | Hereditary ($\rho = 0.1$) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | #iter | #chol | #Mm | #iter | #chol | #Mm | #iter | #chol | #Mm | #iter | #chol | #Mm |
| NoLS | 25.4 | - | 3400 | 18 | - | 1810 | 44 | - | 9842 | 72 | - | 20960 |
| BtkLS | 25.5 | 37.0 | 2436 | 11 | 25 | 718 | 15 | 50 | 1282 | 19 | 63 | 2006 |
| E-BtkLS | 25.5 | 36.2 | 2436 | 11 | 24 | 718 | 15 | 49 | 1282 | 15 | 51 | 1282 |
| FwLS | 18.1 | 26.2 | 1632 | 10 | 17 | 612 | 12 | 34 | 844 | 14 | 44 | 1126 |

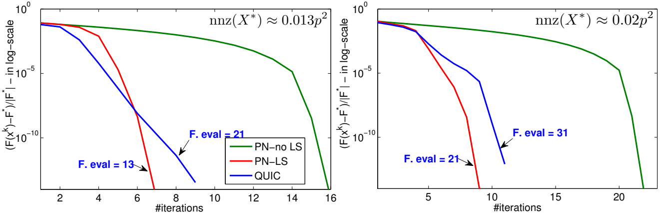
# Test on the real-data: Lymph

## Our method vs QUIC [Hseih2011]

- QUIC subproblem solver: special block-coordinate descent
- Our subproblem solver: general proximal algorithms

**On the average x5 acceleration (up to x15) over Matlab QUIC**

## Convergence behaviour [rho = 0.5]: Lymph [p = 587] (left), Leukemia [p = 1255] (right)




## Step-size selection strategies: Arabidopsis [p = 834], Leukemia [p = 1255], Hereditary [p = 1869]

| | Synthetic ($\rho = 0.01$) | | | Arabidopsis ($\rho = 0.5$) | | | Leukemia ($\rho = 0.1$) | | | Hereditary ($\rho = 0.1$) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LS Scheme | #iter | #chol | #Mm | #iter | #chol | #Mm | #iter | #chol | #Mm | #iter | #chol | #Mm |
| NoLS | 25.4 | - | 3400 | 18 | - | 1810 | 44 | - | 9842 | 72 | - | 20960 |
| BtkLS | 25.5 | 37.0 | 2436 | 11 | 25 | 718 | 15 | 50 | 1282 | 19 | 63 | 2006 |
| E-BtkLS | 25.5 | 36.2 | 2436 | 11 | 24 | 718 | 15 | 49 | 1282 | 15 | 51 | 1282 |
| FwLS | 18.1 | 26.2 | 1632 | 10 | 17 | 612 | 12 | 34 | 844 | 14 | 44 | 1126 |

## Proximal-gradient method for CSM

### Choice of variable matrix and line-search condition

$$\boxed{\mathbf{H}_k := L_k \mathbb{I}, \quad L_k > 0}$$

**Line search condition**: Find the largest $L_k$ such that:

$$\boxed{L_k \leq \eta_k := \frac{\lambda_k^2}{\|\mathbf{d}^k\|_2^2}.} \tag{49}$$

# Proximal-gradient method for CSM

## Choice of variable matrix and line-search condition

$$\boxed{\mathbf{H}_k := L_k \mathbb{I}, \quad L_k > 0}$$

**Line search condition**: Find the largest $L_k$ such that:

$$\boxed{L_k \leq \eta_k := \frac{\lambda_k^2}{\|\mathbf{d}^k\|_2^2}.} \tag{49}$$

**Proximal-gradient algorithm (PGA)**

1. Given $\varepsilon > 0$. Choose $\mathbf{x}^0 \in \text{dom}(F)$ as a starting point.
2. For $k = 0, 1, \cdots$, perform:
   - 2.1. Choose $L_k > 0$ satisfies (49).
   - 2.2. $\mathbf{d}^k := \text{prox}_{\lambda_k g}(\mathbf{x}^k - \gamma_k \nabla f(\mathbf{x}^k)) - \mathbf{x}^k$, with $\gamma_k := 1/L_k$.
   - 2.3. $\lambda_k := \|\mathbf{d}^k\|_{\mathbf{x}^k}$, $\beta_k := \sqrt{L_k}\|\mathbf{d}^k\|_2$.
   - 2.4. If $\beta_k \leq \varepsilon$, terminate.
   - 2.5. *Step size*: $\alpha_k := \beta_k^2/(\lambda_k(\lambda_k + \beta_k^2))$.
   - 2.6. Update $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k \mathbf{d}^k$.

## Global convergence and local convergence

### Theorem (Global convergence [11])

- *If* $L_k \geq \underline{L} > 0$ *for all* $k \geq 0$ *and* $\mathcal{L}_F(F(\mathbf{x}^0)) := \{\mathbf{x} \in \textit{dom}(F) : F(\mathbf{x}) \leq F(\mathbf{x}^0)\}$ *is bounded from below, then* $\{\mathbf{x}^k\}$ *generated by PGA converges to* $\mathbf{x}^\star$*.*
- *Let*

$$\bar{\mathbf{x}}^k := S_k^{-1} \sum_{j=0}^{k} \alpha_k \mathbf{x}^j, \quad \textit{where } S_k := \sum_{j=0}^{k} \alpha_j > 0.$$

*Then* $\boxed{F(\bar{\mathbf{x}}^k) - F^\star \leq \frac{\bar{L}_k}{2S_k}\|\mathbf{x}^0 - \mathbf{x}^\star\|_2^2}$*, where* $\bar{L}_k := \max_{0 \leq j \leq k} L_j$*.*

# Global convergence and local convergence

## Theorem (Global convergence [11])

- *If* $L_k \geq \underline{L} > 0$ *for all* $k \geq 0$ *and* $\mathcal{L}_F(F(\mathbf{x}^0)) := \{\mathbf{x} \in \mathrm{dom}(F) : F(\mathbf{x}) \leq F(\mathbf{x}^0)\}$ *is bounded from below, then* $\{\mathbf{x}^k\}$ *generated by PGA converges to* $\mathbf{x}^\star$.
- *Let*

$$\bar{\mathbf{x}}^k := S_k^{-1} \sum_{j=0}^{k} \alpha_k \mathbf{x}^j, \quad \text{where } S_k := \sum_{j=0}^{k} \alpha_j > 0.$$

*Then* $\boxed{F(\bar{\mathbf{x}}^k) - F^\star \leq \frac{\bar{L}_k}{2S_k}\|\mathbf{x}^0 - \mathbf{x}^\star\|_2^2}$, *where* $\bar{L}_k := \max_{0 \leq j \leq k} L_j$.

## Theorem (Local convergence [11])

***Assumptions:***

- *Let* $\mathbf{x}^\star$ *be the unique solution of* $(1)$ *such that* $\nabla^2 f(\mathbf{x}^\star) \succ 0$.
- *For* $k$ *sufficiently large, if* $\mathbf{D}_k := L_k \mathbb{I}$ *and* $\max\{|1 - \frac{L_k}{\sigma^*_{\min}}|, |1 - \frac{L_k}{\sigma^*_{\max}}|\} < \frac{1}{2}$.

***Conclusion****:* $\{\mathbf{x}^k\}_{k \geq 0}$ *generated by PGA converges to* $\mathbf{x}^\star$ *at a linear rate.*

# Example 1: Graphical model selection

## Graphical model selection

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

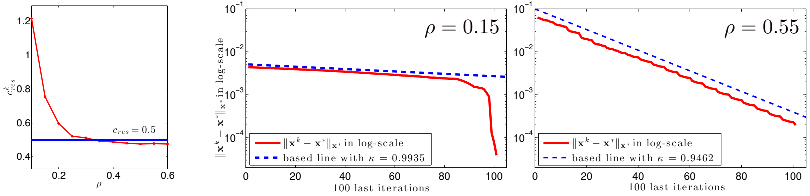
# Example 1: Graphical model selection

## Graphical model selection

$$\min_{\Theta \succ 0} \left\{ \underbrace{\mathrm{tr}(\Sigma\Theta) - \log\det(\Theta)}_{f(\Theta)} + \underbrace{\lambda \|\mathrm{vec}(\Theta)\|_1}_{g(\Theta)} \right\}.$$

## Linear convergence of PGA

### Graph learning: Lymph [p = 587]



# Improvement - greedy proximal gradient variant

## Mathematical observation

Let us define

- $\mathbf{s}_g^k := \mathbf{x}^k + \mathbf{d}^k$
- $\hat{\mathbf{x}}^k = (1-\alpha_k)\mathbf{x}^k + \alpha_k \mathbf{s}^k$ for $\alpha_k \in (0,1]$.

If $F(\mathbf{s}_g^k) \leq F(\mathbf{x}^k)$, then by convexity of $F$:

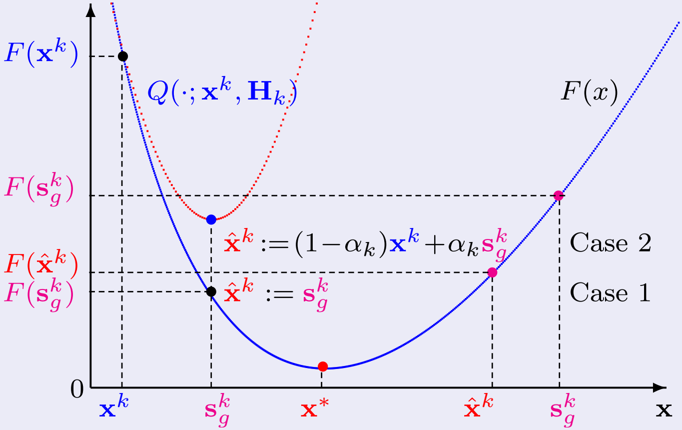
$$F(\hat{\mathbf{x}}^k) = F((1-\alpha_k)\mathbf{x}^k + \alpha_k) \leq (1-\alpha_k)F(\mathbf{x}^k) + \alpha_k F(\mathbf{s}_g^k) \overset{F(\mathbf{s}_g^k)\leq F(\mathbf{x}^k)}{\leq} F(\mathbf{x}^k)$$

By comparing $F(\mathbf{x}^k)$, $F(\mathbf{s}_g^k)$ and $F(\hat{\mathbf{x}}^k)$ we can pick $\mathbf{x}^{k+1}$ as

$$\mathbf{x}^{k+1} = \begin{cases} \mathbf{s}_g^k & \text{if } \mathbf{s}_g^k \in \mathrm{dom}(F) \text{ and } F(\mathbf{s}_g^k) \leq F(\mathbf{x}^k), \\ \hat{\mathbf{x}}^k & \text{otherwise.} \end{cases}$$

# Improvement - greedy proximal gradient variant

## Visualization of the idea



## Example 2: Poisson imaging reconstruction

### Optimization problem with TV-norm

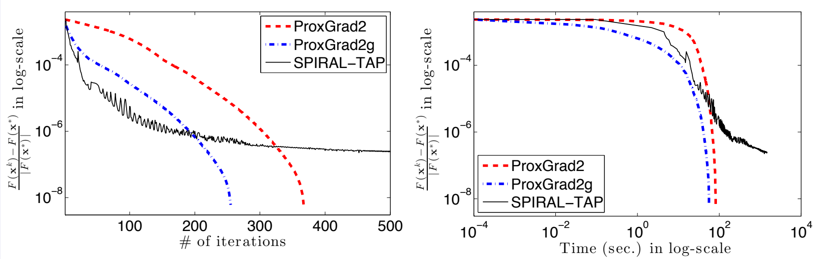
$$\min_{\mathbf{x}\in\mathbb{R}^{n\times p}} \left\{ \underbrace{\sum_{i=1}^{n}(\mathbf{K}\mathbf{x})_i - \sum_{i=1}^{n} y_i \log((\mathbf{K}\mathbf{x})_i)}_{f(\mathbf{x})} + \underbrace{\lambda\|\mathbf{x}\|_{\mathrm{TV}}}_{g(\mathbf{x})} \right\}$$

## Example 2: Poisson imaging reconstruction

### Optimization problem with TV-norm

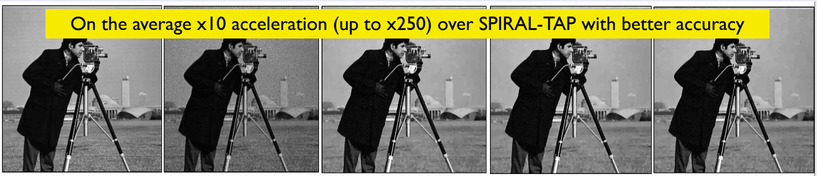
$$\min_{\mathbf{x}\in\mathbb{R}^{n\times p}} \Big\{ \underbrace{\sum_{i=1}^{n} (\mathbf{K}\mathbf{x})_i - \sum_{i=1}^{n} y_i \log((\mathbf{K}\mathbf{x})_i)}_{f(\mathbf{x})} + \underbrace{\lambda\|\mathbf{x}\|_{\mathrm{TV}}}_{g(\mathbf{x})} \Big\}$$

### Convergence of PGA, greedy PGA and SPIRAL-TAP



# Example 2: Poisson imaging reconstruction - cont.

## Visualization of the outcome - cameraman




Original image | Poisson noise image | Reconstructed image (ProxGrad) | Reconstructed image (ProxGradNewton) | Reconstructed image (SPIRAL–TAP

# References

[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.

[2] S. Becker and M.J. Fadili.
A quasi-Newton proximal splitting method.
In *Proceedings of Neutral Information Processing Systems Foundation*, 2012.

[3] 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.

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

[5] J.D. Lee, Y. Sun, and M.A. Saunders.
Proximal newton-type methods for convex optimization.
*Tech. Report.*, pages 1–25, 2012.

## References

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

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

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

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

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

[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.