An Introduction to Non-smooth Optimization

Lecture 02 - Intro of the Intro

Jingwei LIANG

Institute of Natural Sciences, Shanghai Jiao Tong University

Email: optimization.sjtu@gmail.com

Office: Room 355, No. 6 Science Building

Outline

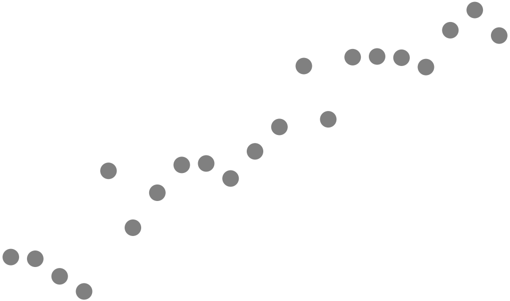
1. Motivating example
2. Regularization
3. Applications
4. First-order optimization methods

Least square regression

Let $m \in \mathbb{N}_{++}$. For $i = 1, \ldots, m$, given each $x_i \in \mathbb{R}$,

$$y_i = ax_i + b + \epsilon_i$$

with ϵ_i being random noise.



Least square regression

Let $m \in \mathbb{N}_{++}$. For $i = 1, \ldots, m$, given each $x_i \in \mathbb{R}$,

$$y_i = ax_i + b + \epsilon_i$$

with ϵ_i being random noise.

Matrix-vector representation,

$$\boldsymbol{A} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \\ x_m & 1 \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} a \\ b \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \text{ and } \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_m \end{bmatrix}.$$

The system of equations reads

$$\begin{cases} ax_1 + b + \epsilon_1 = y_1, \\ \qquad\qquad \vdots \\ ax_m + b + \epsilon_m = y_m. \end{cases} \quad \Longleftrightarrow \quad \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\epsilon}.$$

Least square regression: estimating $\boldsymbol{x}$ from $\boldsymbol{y}$

$$\min_{\boldsymbol{x}\in\mathbb{R}^2} \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|^2.$$

Assume that $\boldsymbol{A}$ has full column rank

$$\begin{aligned}\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|^2 \quad &\Longleftrightarrow \quad \mathbf{0} = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}) \\ &\Longleftrightarrow \quad \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{y} \\ &\Longleftrightarrow \quad \boldsymbol{x} = \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{y}\end{aligned}$$

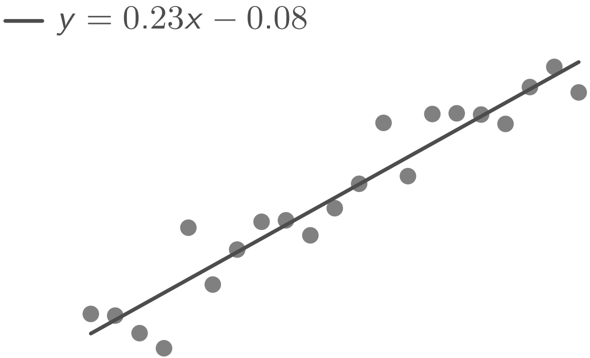
If $\boldsymbol{A}^T\boldsymbol{A}$ is not invertible,

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \gamma_k \boldsymbol{A}^T\left(\boldsymbol{A}\boldsymbol{x}^{(k)} - \boldsymbol{y}\right) \ \rightarrow \ \boldsymbol{x}^\star.$$

Least square regression

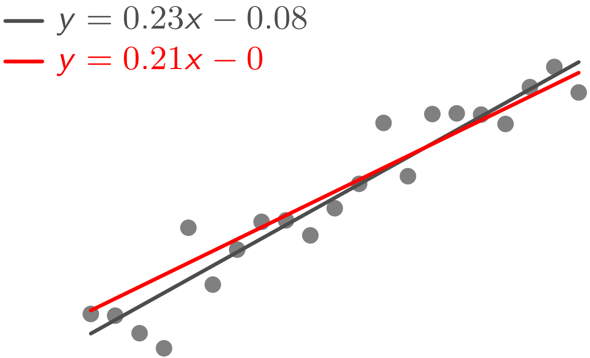
Least square regression: estimating $\boldsymbol{x}$ from $\boldsymbol{y}$

$$\min_{\boldsymbol{x}\in\mathbb{R}^2} \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|^2.$$



Non-negative least square regression:

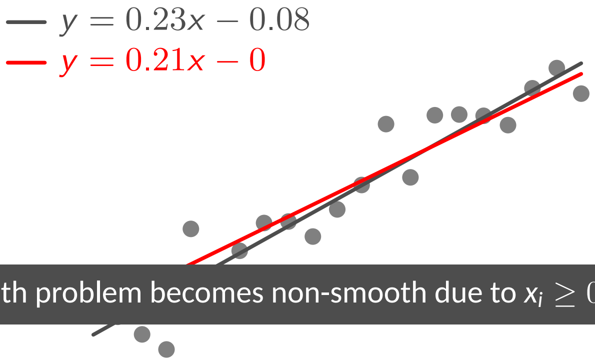
$$\min_{\boldsymbol{x} \in \mathbb{R}^2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2 \quad \text{such that} \quad x_i \geq 0, \boldsymbol{i} = 1, 2.$$



Non-negative least square

Non-negative least square regression:

$$\min_{\boldsymbol{x}\in\mathbb{R}^2} \|\boldsymbol{Ax}-\boldsymbol{y}\|^2 \quad \text{such that} \quad x_i \geq 0, i = 1, 2.$$



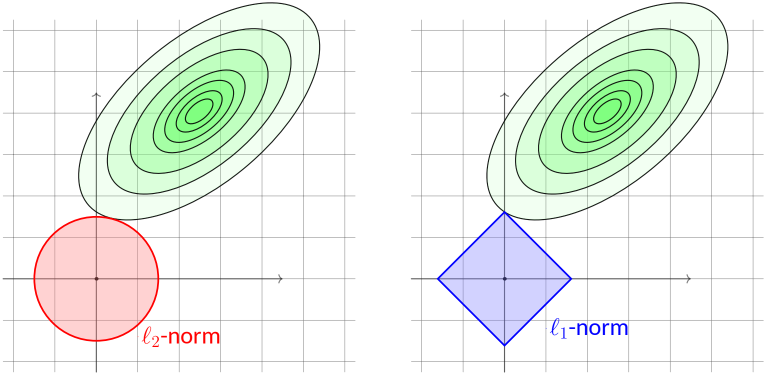
Smooth problem becomes non-smooth due to $x_i \geq 0, i = 1, 2$.

Other constraints on $\boldsymbol{x}$

Besides non-negativity, we can consider the following requirement on the solution: let $p \in \{1, 2\}$ and $\delta > 0$,

$$\min_{\boldsymbol{x}} \ \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2$$

$$\text{such that } \ \|\boldsymbol{x}\|_p \leq \delta.$$



Regularization

Sparsity, low-rank, non-negativity

Regularized least square

Let $R(\boldsymbol{x})$ be a function promoting prior information, *e.g.* non-negativity or norm constraint...

Regularized least square

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} R(\boldsymbol{x}) + \frac{1}{2\mu}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2.$$

- $\mu > 0$ provides a balance between diffusion and fidelity.
- The choices of $R(\boldsymbol{x})$ depends on the prior information.

Let $R(\boldsymbol{x})$ be a function promoting prior information, *e.g.* non-negativity or norm constraint...

Regularized least square

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} R(\boldsymbol{x}) + \frac{1}{2\mu}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|^2.$$

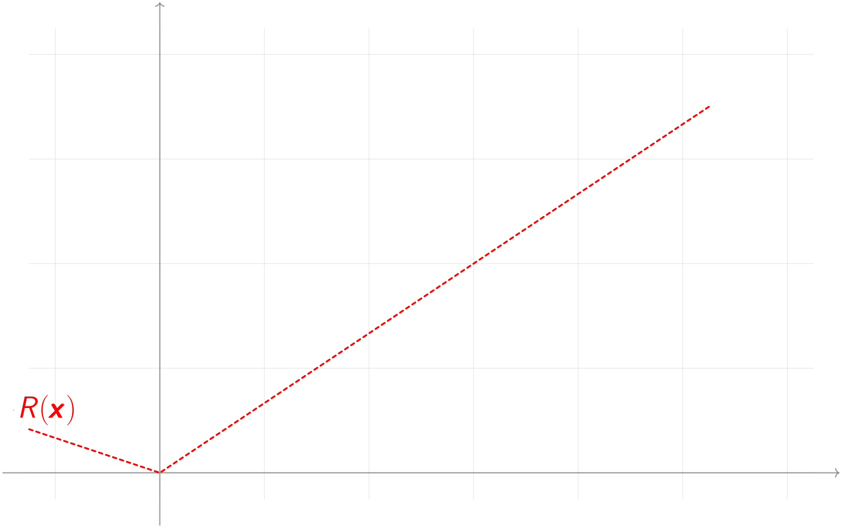
- $\mu > 0$ provides a balance between diffusion and fidelity.
- The choices of $R(\boldsymbol{x})$ depends on the prior information.

Regularization

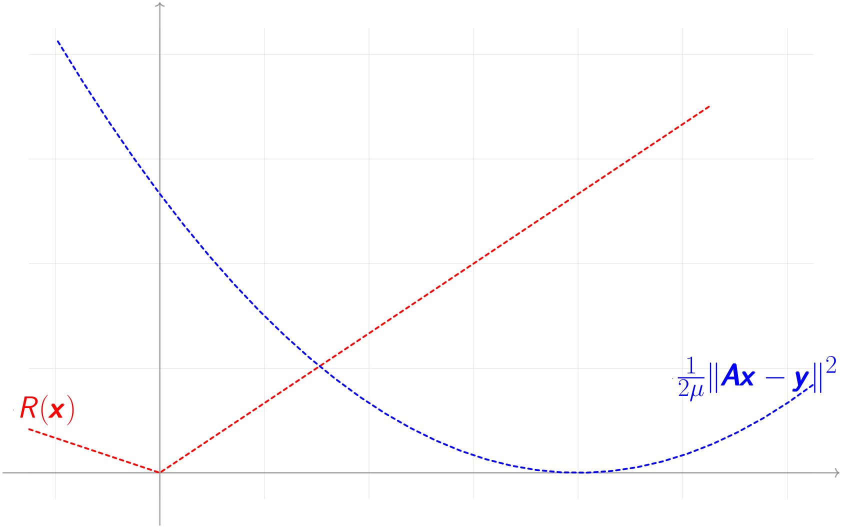
Function $R(\boldsymbol{x})$ is called regularization, it is a process that forces the solution to be “simpler”,

- obtain results for ill-posed problems (*e.g.* image processing).
- prevent overfitting (*e.g.* machine learning).

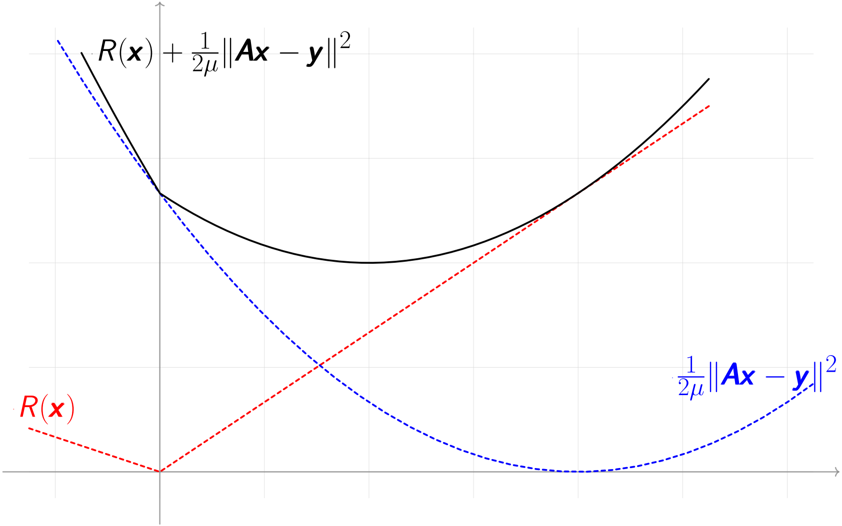
Regularized least square



Regularized least square



Regularized least square



A more general example

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} R(\boldsymbol{x}) + F(\boldsymbol{x}).$$

Choices of $F(\boldsymbol{x})$: let $\boldsymbol{a} \in \mathbb{R}^n$ and $b \in \mathbb{R}$

- **Quadratic loss**

$$F(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{a}^T\boldsymbol{x} - b)^2.$$

- **Logistic loss**

$$F(\boldsymbol{x}) = \log\left(1 + \mathrm{e}^{-b\boldsymbol{a}^T\boldsymbol{x}}\right).$$

- **Squared hinge loss**

$$F(\boldsymbol{x}) = \max\left\{1 - t(\boldsymbol{a}^T\boldsymbol{x} + b),\ 0\right\}, \quad t \in \{-1, 1\}.$$

Choices of $R(\boldsymbol{x})$ $\longrightarrow$

Examples: norms

Let $\boldsymbol{x} \in \mathbb{R}^n$

- **ℓ_1-norm**

$$\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i|.$$

- **ℓ_2-norm**

$$\|\boldsymbol{x}\|_2 = \left(\sum_{i=1}^{n} (x_i)^2\right)^{1/2}.$$

- **group ℓ_1-norm** let $\mathcal{G} = \{g_1, g_2, ..., g_\ell\}$ be a partition of $\{1, 2, ..., n\}$,

$$\|\boldsymbol{x}\|_{1,2} = \sum_{g_j \in \mathcal{G}} \left(\sum_{i \in g_i} (x_i)^2\right)^{1/2}.$$

Example - Total variation (TV) [Rudin, Osher & Fatemi '92]

Let ∇ be the discrete gradient operator

$$\|\nabla \boldsymbol{x}\|_1$$

In 1D case:

$$\nabla = \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \\ & & & & 0 \end{bmatrix}.$$

Examples: total variation

Example - Total variation (TV) [Rudin, Osher & Fatemi '92]

Let ∇ be the discrete gradient operator

$$\|\nabla \boldsymbol{x}\|_1$$

Original image

Horizontal gradient

Vertical gradient

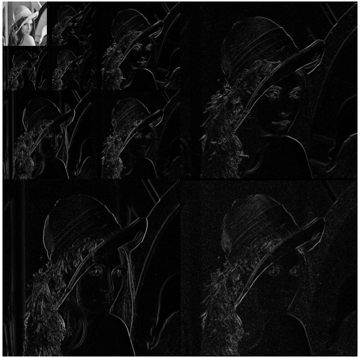
Examples: wavelet frames

Example - Wavelet [Morlet, Meyer, Mallat, Daubechies, et al]

Family of filters $\{\psi_{j,k} : j, k \in \mathbb{Z}\}$

$$\psi_{j,k}(\cdot) = 2^{j/2}\psi(2^j \cdot -k).$$

Original image



Wavelet coefficients

Examples: low rank

Example - Nuclear norm [Recht, Fazel and Parrilo '10]

Let $A \in \mathbb{R}^{m\times n}$, and $A = USV^T$ be its singular value decomposition

$$\|A\|_* = \sum_{i=1}^{\min\{m,n\}} S_{i,i}.$$

Rank 20

Rank 80

Rank 140

Examples: low rank

Example - Nuclear norm [Recht, Fazel and Parrilo '10]

Let $A \in \mathbb{R}^{m \times n}$, and $A = USV^T$ be its singular value decomposition

$$\|A\|_* = \sum_{i=1}^{\min\{m,n\}} S_{i,i}.$$

Rank 20

Rank 80

How to use regularization?

Rank 140

Applications

Image processing, computer vision

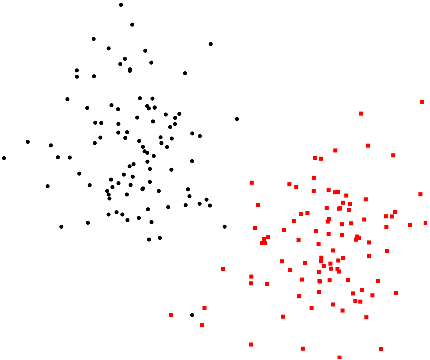
Example: machine learning

Example - Sparse logistic regression

Let $(\boldsymbol{a}_i, b_i) \in \mathbb{R}^n \times \{\pm 1\}, i = 1, ..., m,$

$$\min_{(\boldsymbol{x},y)\in\mathbb{R}^n\times\mathbb{R}} \mu\|\boldsymbol{x}\|_1 + \frac{1}{m}\sum_{i=1}^m f(\boldsymbol{x}^T\boldsymbol{a}_i + y; b_i),$$

where $f(u_i; b_i) = \log(1 + \mathrm{e}^{-u_i b_i})$.

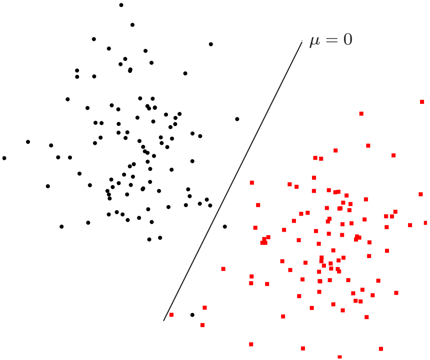


Example - Sparse logistic regression

Let $(\boldsymbol{a}_i, b_i) \in \mathbb{R}^n \times \{\pm 1\}, i = 1, \ldots, m$,

$$\min_{(\boldsymbol{x}, y) \in \mathbb{R}^n \times \mathbb{R}} \mu \|\boldsymbol{x}\|_1 + \frac{1}{m} \sum_{i=1}^{m} f(\boldsymbol{x}^T \boldsymbol{a}_i + y; b_i),$$

where $f(u_i; b_i) = \log(1 + e^{-u_i b_i})$.

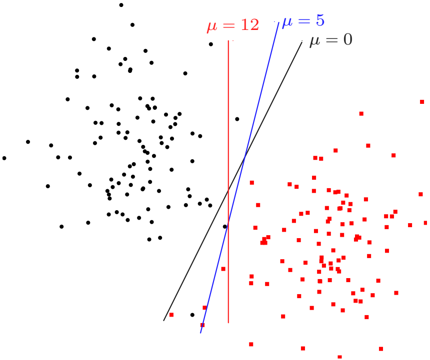


Example - Sparse logistic regression

Let $(\boldsymbol{a}_i, b_i) \in \mathbb{R}^n \times \{\pm 1\}, i = 1, ..., m,$

$$\min_{(\boldsymbol{x},y)\in\mathbb{R}^n\times\mathbb{R}} \mu\|\boldsymbol{x}\|_1 + \frac{1}{m}\sum_{i=1}^m f(\boldsymbol{x}^T\boldsymbol{a}_i + y; b_i),$$

where $f(u_i; b_i) = \log(1 + \mathrm{e}^{-u_i b_i})$.



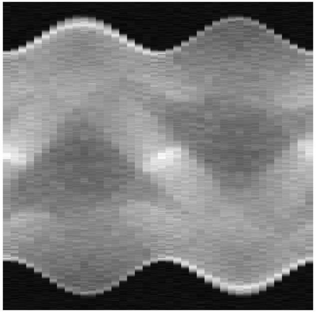
Mathematical formulation

$$\boldsymbol{f} = \mathcal{F}\bar{\boldsymbol{x}} + \varepsilon,$$

where

- $\bar{\boldsymbol{x}}$ is the true image which is **piecewise constant/smooth** — TV.
- $\mathcal{F}$ is partial Fourier transform.
- ε is additive noise.

$\bar{\boldsymbol{x}}$



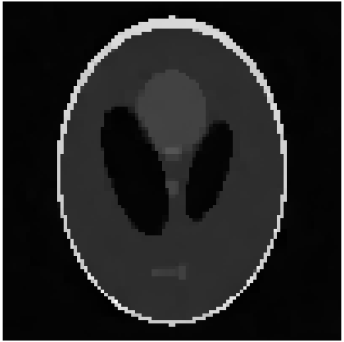
$\boldsymbol{f}$

Example - TV based MRI reconstruction

Let $p \in \{1, 2\}$,

$$\min_{\mathbf{x} \in \mathbb{R}^{m \times n}} \mu \|\nabla \mathbf{x}\|_1 + \frac{1}{2}\|\mathbf{f} - \mathcal{F}\mathbf{x}\|_p^p.$$

$\bar{\mathbf{x}}$



recovered $\mathbf{x}$

Video decomposition

Mathematical formulation

$$\boldsymbol{f} = \bar{\boldsymbol{l}} + \bar{\boldsymbol{s}} + \varepsilon,$$

where

- $\bar{\boldsymbol{l}}$ is the background which is **low rank** — nuclear norm.
- $\bar{\boldsymbol{s}}$ is the foreground which is **sparse** — ℓ_1-norm.
- ε is additive white Gaussian noise.

Video decomposition

Mathematical formulation

$$\boldsymbol{f} = \bar{\boldsymbol{l}} + \bar{\boldsymbol{s}} + \varepsilon,$$

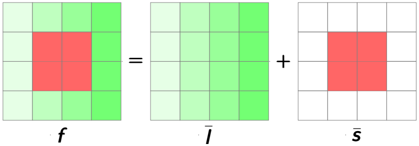
where

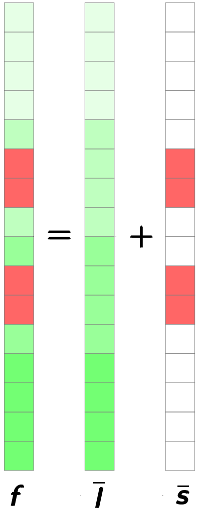
- $\bar{\boldsymbol{l}}$ is the background which is **low rank** — nuclear norm.
- $\bar{\boldsymbol{s}}$ is the foreground which is **sparse** — ℓ_1-norm.
- ε is additive white Gaussian noise.

Example - Principal component pursuit [Candès et al '11]

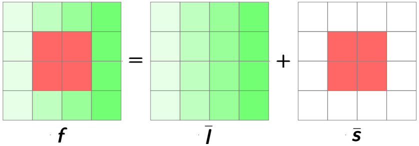
$$\min_{\boldsymbol{l},\boldsymbol{s}\in\mathbb{R}^{m\times n}} \ \mu\left(\|\boldsymbol{l}\|_* + \nu\|\boldsymbol{s}\|_1\right) + \frac{1}{2}\|\boldsymbol{l} + \boldsymbol{s} - \boldsymbol{f}\|^2.$$

Video decomposition





Video decomposition



First-order optimization methods

Non-smooth optimization, first-order methods

Non-smooth optimization: a typical example

Problem - Non-smooth optimization problem

Let $r \in \mathbb{N}_{++}$

$$\min_{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_r} \left\{ \Phi(\boldsymbol{x}) \stackrel{\text{def}}{=} F(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_r) + \sum\nolimits_{i=1}^{r} R_i(\boldsymbol{K}_i \boldsymbol{x}_i) \right\},$$

where

- F: smooth data fidelity term...
- R_i: non-smooth regularization terms...
- $\boldsymbol{K}_i$: linear/nonlinear operators...

- Signal/imaging processing, compressed sensing, inverse problems
- Statistics, data science, machine learning
- Control theory, operation research, game theory
- $\cdots$

Non-smooth, (non-convex), composite, high dimension

First-order methods: two basic ingredients

$$\min_{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_r} F(\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_r) + \sum_{i=1}^{r} R_i(\boldsymbol{K}_i\boldsymbol{x}_i)$$

First-order methods: two basic ingredients

$$\min_{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_r} F(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_r) + \sum_{i=1}^{r} R_i(\boldsymbol{K}_i \boldsymbol{x}_i)$$

Divide and Conquer.

$$\boldsymbol{K}_1, \boldsymbol{K}_2, \cdots, \boldsymbol{K}_r$$
$$\boldsymbol{K}_1^*, \boldsymbol{K}_2^*, \cdots, \boldsymbol{K}_r^*$$

$$F(\boldsymbol{x}_1, \cdot, \ldots, \cdot)$$
$$F(\cdot, \boldsymbol{x}_2, \ldots, \cdot)$$
$$F(\cdot, \ldots, \cdot, \boldsymbol{x}_r)$$

$$R_1, R_2, \cdots, R_r$$
$$R_1^*, R_2^*, \cdots, R_r^*$$

Algorithm - Gradient descent [Cauchy '1847]

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} F(\boldsymbol{x})$$

where F is convex smooth differentiable with ∇F being L-Lipschitz.

Gradient descent (GD):

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \gamma_k \nabla F(\boldsymbol{x}^{(k)}), \quad \gamma_k \in]0, 2/L[.$$

First-order methods: two basic ingredients

Algorithm - Gradient descent [Cauchy '1847]

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} F(\boldsymbol{x})$$

where F is convex smooth differentiable with ∇F being L-Lipschitz.

Gradient descent (GD):

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \gamma_k \nabla F(\boldsymbol{x}^{(k)}), \quad \gamma_k \in]0, 2/L[.$$

Algorithm - Proximal point algorithm [Rockafellar '76]

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} R(\boldsymbol{x})$$

with R being proper closed convex. Define "proximity operator" by

$$\operatorname{prox}_{\gamma R}(\boldsymbol{v}) \stackrel{\text{def}}{=} \operatorname{argmin}_{\boldsymbol{x}} \gamma R(\boldsymbol{x}) + \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{v}\|^2.$$

Proximal point algorithm (PPA):

$$\boldsymbol{x}^{(k+1)} = \operatorname{prox}_{\gamma_k R}(\boldsymbol{x}^{(k)}), \quad \gamma_k > 0.$$

First-order methods: a rich class

Definition - First-order methods

Numerical schemes that use *at most* the first-order differentiability, *e.g.* *gradient or sub-gradient*, of the objective.

$F + R$ Forward–Backward splitting [Lions & Mercier '79]

$R_1 + R_2$ Douglas–Rachford splitting [Douglas & Rachford '56; Lions & Mercier '79]

ADMM [Glowinski & Marrocco '75; Gabay & Mercier '76]...

$F + R(\boldsymbol{K}\cdot)$ Primal–Dual splitting methods [Arrow, Hurwicz & Uzawa '58; Esser, Zhang & Chan '10; Chambolle & Pock '11]

$F + \sum_i R_i$ Generalized Forward–Backward splitting [Raguet, Fadili & Peyré '13]

– $\cdots$

`Origins from numerical PDE back to 1950s, now ubiquitous in signal/image processing, inverse problems, data science, statistics, machine learning...`

References

- Leonid I. Rudin, Stanley Osher, and Emad Fatemi. "Nonlinear total variation based noise removal algorithms." Physica D: nonlinear phenomena 60.1-4 (1992): 259-268.
- Ingrid Daubechies. "Ten lectures on wavelets." Society for industrial and applied mathematics, 1992.
- Stéphane Mallat. "A wavelet tour of signal processing." Elsevier, 1999.
- Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization." SIAM review 52.3 (2010): 471-501.
- Emmanuel J. Candès, Xiaodong Li, Yi Ma, John Wright. "Robust principal component analysis?." Journal of the ACM (JACM) 58.3 (2011): 1-37.