Lecture 10: Issues of Large Scale Optimization

- Motivating examples: compressive sensing, matrix completion, etc
- Sparsity-promoting L_1 regularization, nonsmoothness
- Quick introduction to subgradient and subdifferential calculus
- Linear convergence without strong convexity?

Large Scale Convex Optimization

$$\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{x} \in X \end{array} \qquad (1)$$

where $f(\cdot)$ is convex differentiable, $X \subseteq \mathbb{R}^n$ is a *nonempty* convex set.

- If X is well-represented, and $f(\boldsymbol{x})$, $\nabla f(\boldsymbol{x})$, $\nabla^2 f(\boldsymbol{x})$ are easily computed, then (1) is efficiently solvable (e.g., via interior point methods in polynomial time).
- Second order methods (e.g., interior point methods) require solving a linear system of size $n \times n$, which can be slow when n is large (e.g., $> 10^5$).
- For large scale optimization problems, first order methods are preferred, since each iteration is cheaper, requiring only matrix-vector multiplications.
- First order methods typically preserve problem sparsity in large scale optimization problems.

Example: Routing in a Data Network

- Consider a directed graph with nodes $\{1, ..., N\}$ and directed links $\mathcal{A} \subseteq \{1, ..., N\} \times \{1, ..., N\}$ connecting the nodes.
- M pairs of nodes (called origin-destination (OD) pairs) labeled from 1 to M
- For each OD pair $w \in \{1, ..., M\}$, a set of directed loop-free paths from the origin to the destination, with traffic arrival rate $r_w > 0$.
- Label the given paths from 1 to P and let $\mathcal{P}_w \subseteq \{1, ..., P\}$ be the set of paths from the origin to the destination of OD pair w.
- For each link $(i, j) \in \mathcal{A}$, we are given $\bar{D}_{ij}(\boldsymbol{f}_{ij})$, the delay on this link when the flow it carries is equal to $\boldsymbol{f}_{ij}$, so that the total delay is

$$\bar{D}(\cdots, \boldsymbol{f}_{ij}, \cdots)_{(i,j)\in\mathcal{A}} = \sum_{(i,j)\in\mathcal{A}} \bar{D}_{ij}(\boldsymbol{f}_{ij}).$$

Routing in a Data Network, continued

- Let

$$X = \{ \ (\boldsymbol{x}_1, ..., \boldsymbol{x}_P) \mid \boldsymbol{x}_1 \geq 0, ..., \boldsymbol{x}_P \geq 0, \ \ \sum_{p \in \mathcal{P}_w} \boldsymbol{x}_p = r_w, w = 1, ...M \ \}.$$

- Let E denote the $|\mathcal{A}| \times P$ link-path incidence matrix associated with the given paths (i.e., an entry of E is 1 if the link corresponding to its row is on the path corresponding to its column, and is 0 otherwise).
- Then, the optimal routing problem may be formulated as the following nonlinear program:

$$\text{minimize} \ \ D(\boldsymbol{x}) = \bar{D}(E\boldsymbol{x}) \ \ \text{subject to} \ \ \boldsymbol{x} \in X.$$

- Assume $\bar{D}_{ij}$ is continuous on an interval $[0, C_{ij})$, tends to ∞ at C_{ij} (the "transmission capacity" of link (i, j)), and is strictly convex and smooth.

Routing in a Data Network, continued

- A well-known example is the M/M/1 queue delay function

$$\bar{D}_{ij}(\boldsymbol{f}_{ij}) = \boldsymbol{f}_{ij}/(C_{ij} - \boldsymbol{f}_{ij}).$$

- There may be more than one optimal routing since, although $\bar{D}$ is strictly convex, D need not be.
- Given the large network size, many OD pairs and the large number of routes, the routing problem is a large scale convex optimization problem, which may admit multiple optimal path flows.

Example: MRI, Compressive Sensing

- In a MRI system, we have measurement in the form of

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}$$

where $\boldsymbol{x} \in \mathbb{C}^n$ is the image of interest, $\boldsymbol{n} \in \mathbb{C}^m$ is measurement noise, and $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ is the measurement matrix (partial Fourier Transform matrix)

- Large under-determined linear system $m \ll n$, to reduce data volume.

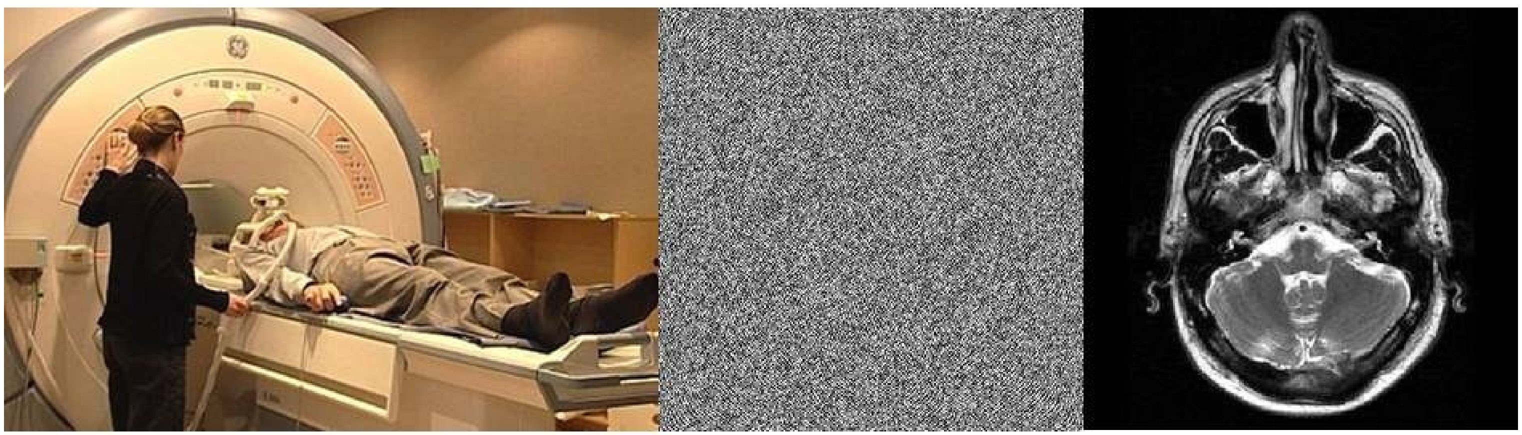


Figure 1: MRI, Sampling points, Image

MRI and Compressive Sensing

- $\boldsymbol{x}$ is typically sparse $\Rightarrow$ finding the sparsest solution

- L_0-norm minimization

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \;\; \lambda\|\boldsymbol{x}\|_0 + \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$$

  where $\lambda > 0$ is a penalty parameter

- Nonconvex, difficult to solve

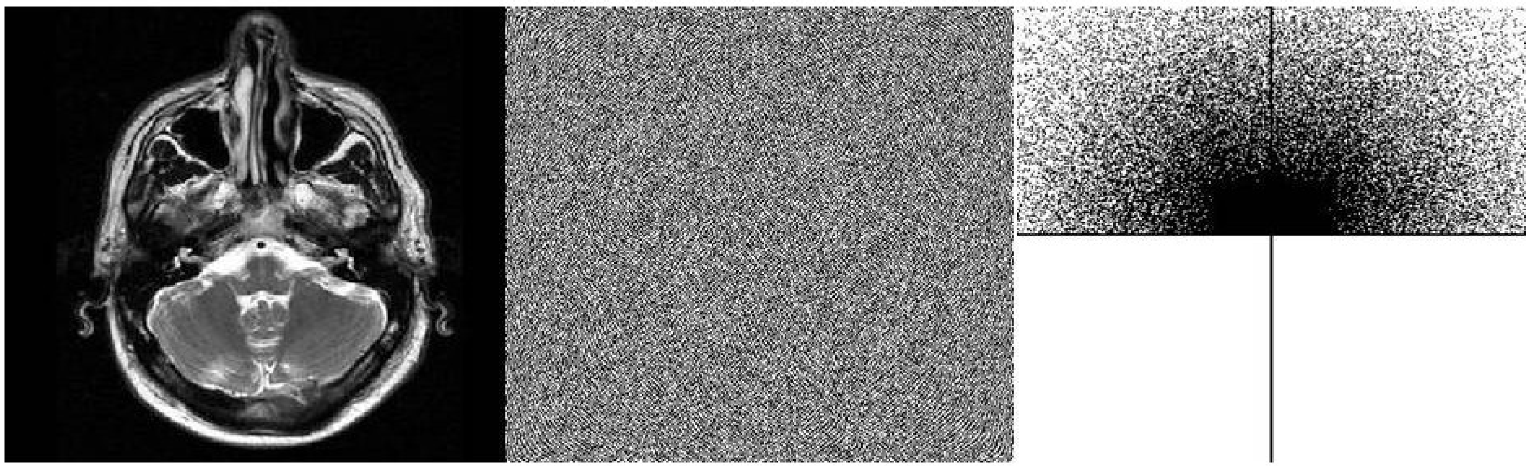
- Compressive sensing: L_1-norm minimization

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \;\; \lambda\|\boldsymbol{x}\|_1 + \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$$

  which is a large scale (non-smooth) convex optimization problem.

- Under suitable conditions (e.g., A is random), L_0 minimization $\Leftrightarrow$ L_1 minimization

MRI and Compressive Sensing



- Pick 25% coefficients at random (with bias)
- Reconstruct image from the 25% coefficients

MRI and Compressive Sensing

Use 1/4 Fourier coefficients

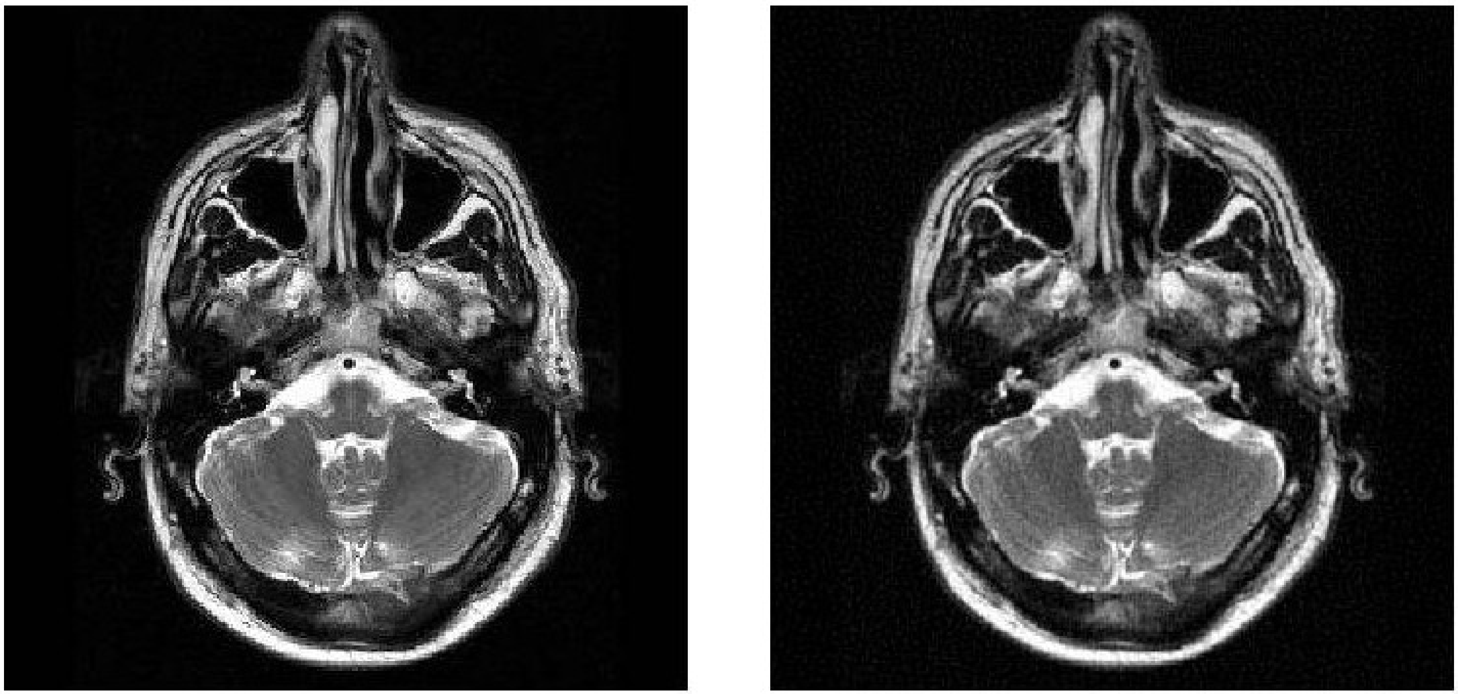


Figure 2: Original vs. Reconstructed Image (courtesy of Y. Zhang, Rice University)

L_1-minimization: Sparsity-Inducing

The optimality condition for

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}}\;\; \lambda\|\boldsymbol{x}\|_1 + \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2$$

is $-\boldsymbol{A}'(\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b}) \in \lambda\partial\|\boldsymbol{x}^*\|_1$, or

- $-\lambda \le (\boldsymbol{A}'(\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b}))_i \le \lambda$
- if $\boldsymbol{x}_i^* > 0$, then $(\boldsymbol{A}'(\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b}))_i = -\lambda$
- if $\boldsymbol{x}_i^* < 0$, then $(\boldsymbol{A}'(\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b}))_i = \lambda$

Thus, L_1 minimization is sparsity-inducing

$$-\lambda < (\boldsymbol{A}'(\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b}))_i < \lambda \quad \text{implies} \quad \boldsymbol{x}_i^* = 0.$$

$\Rightarrow$ The larger the value of λ, the sparser is the solution $\boldsymbol{x}^*$.

Subgradient

A vector $\boldsymbol{g}$ is a **subgradient** of f (not necessarily convex) at $\boldsymbol{x}$ if

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{g}'(\boldsymbol{y} - \boldsymbol{x}) \quad \text{for all } \boldsymbol{y}$$

- subgradient gives affine global underestimator of f
- if f is convex, it has at least one subgradient at every point in $\mathbf{relint}\,\mathbf{dom}\,f$
- if f is convex and differentiable, $\nabla f(\boldsymbol{x})$ is a subgradient of f at $\boldsymbol{x}$

Example: $f = \max\{f_1, f_2\}$, with f_1, f_2 convex and differentiable

Subdifferential

The set of all subgradients of f at $\boldsymbol{x}$ is called the **subdifferential** of f at $\boldsymbol{x}$, written $\partial f(\boldsymbol{x})$

- $\partial f(\boldsymbol{x})$ is a closed convex set
- $\partial f(\boldsymbol{x})$ nonempty (if f convex, and finite near $\boldsymbol{x}$)
- $\partial f(\boldsymbol{x}) = \{\nabla f(\boldsymbol{x})\}$ if f is differentiable at $\boldsymbol{x}$
- if $\partial f(\boldsymbol{x}) = \{\boldsymbol{g}\}$, then f is differentiable at $\boldsymbol{x}$ and $\boldsymbol{g} = \nabla f(\boldsymbol{x})$

Example: $f(\boldsymbol{x}) = |x_1| + |x_2|$

Optimality condition: $\boldsymbol{x}^*$ is a minimizer of $\min f(\boldsymbol{x})$ iff $\mathbf{0} \in \partial f(\boldsymbol{x}^*)$.

Subdifferential Calculus

Assumption: all functions are finite near $\boldsymbol{x}$

- $\partial f(\boldsymbol{x}) = \{\nabla f(\boldsymbol{x})\}$ if f is differentiable at $\boldsymbol{x}$
- **scaling:** $\partial(\alpha f) = \alpha \partial f$ (if $\alpha > 0$)
- **addition:** $\partial(f_1 + f_2) = \partial f_1 + \partial f_2$ (RHS is addition of sets)
- **affine transformation of variables:** if $g(\boldsymbol{x}) = f(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b})$, then $\partial g(\boldsymbol{x}) = \boldsymbol{A}^T \partial f(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b})$
- **pointwise maximum:** if $f = \max_{i=1,\ldots,m} f_i$, then

$$\partial f(\boldsymbol{x}) = \textbf{Co} \bigcup \{\partial f_i(\boldsymbol{x}) \mid f_i(\boldsymbol{x}) = f(\boldsymbol{x})\},$$

i.e.,, convex hull of union of subdifferentials of 'active' functions at $\boldsymbol{x}$

Compressive Sensing: Other Convex Formulations

Variations of formulations

- $\min_{\boldsymbol{x}\in\mathbb{C}^n} \|\boldsymbol{x}\|_1,$ subject to: $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$

- $\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{x}\|_1,$ subject to: $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}; \boldsymbol{x} \geq 0$

- $\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2,$ subject to: $\|\boldsymbol{x}\|_1 \leq \nu$

- The interior point methods are ineffective. The classical first order methods are preferred, such as the coordinate descent, the alternating direction method of multipliers, LASSO - Least Absolute Shrinkage and Selection Operator

- Note $\boldsymbol{A}$ is a fat matrix, thus not full column rank $\Rightarrow$ existing (rate of) convergence theory does not apply.

- Non-smoothness is involved in the objective function, "causing problem" to the coordinate descent method??

Coordinate Descent for Compressive Sensing

Consider the nonsmooth formulation:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \lambda\|\boldsymbol{x}\|_1 + \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2$$

- Coordinate descent algorithm (CD): iteratively and cyclically minimize wrt each variable
- Soft thresholding: let $x^+ = \arg\min_{y\in\mathbb{R}} \lambda|y| + \frac{1}{2}(y-x)^2$, then

$$x^+ = \begin{cases} x+\lambda, & x \le -\lambda \\ 0, & -\lambda \le x \le \lambda \\ x-\lambda, & x \ge \lambda \end{cases}$$

- Simple, fast, sparsity-inducing ... how about convergence??

Coordinate Descent Method

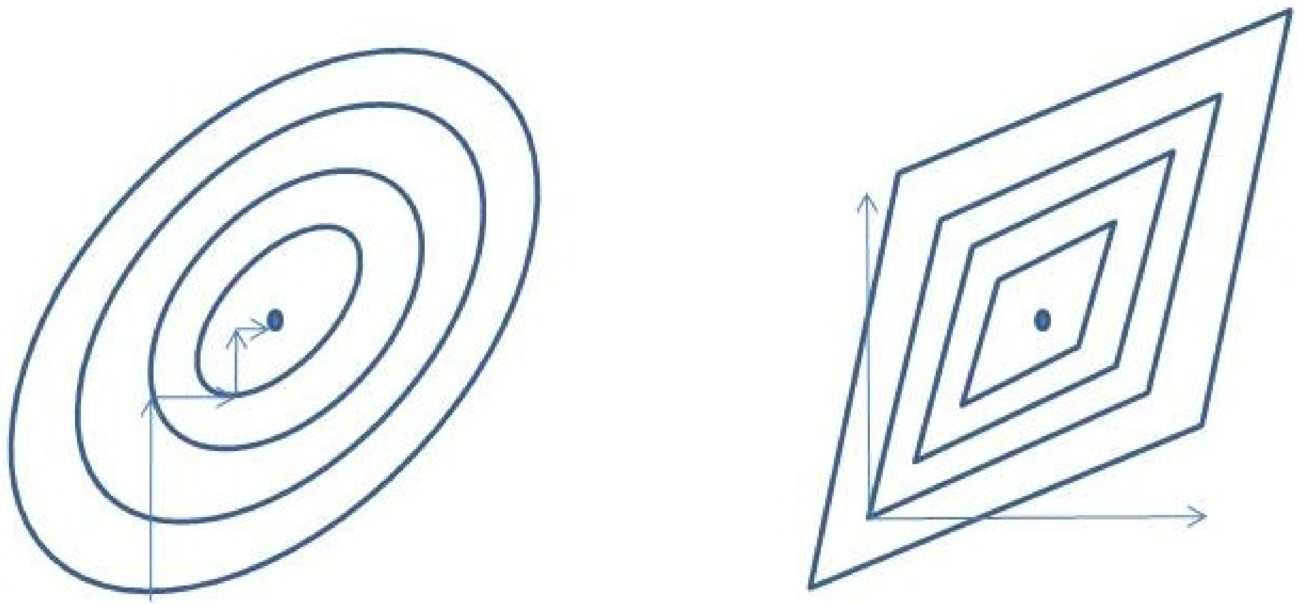


Figure 3: CD method for smooth/non-smooth minimization

Non-smoothness can cause the CD method to get stuck!

Example: Matrix Completion

- Restore an image $\boldsymbol{M} \in \mathbb{R}^{m \times n}$ using a subset E of pixel values
- $\boldsymbol{M}$ is a large matrix, but low rank (say rank $r \ll \min\{m, n\}$)
- Matrix completion: assume M is rank r, let $\boldsymbol{X} \in \mathbb{R}^{m \times r}$, $\boldsymbol{Y} \in \mathbb{R}^{n \times r}$, solve

$$\begin{array}{ll} \text{minimize}_{\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}} & \|\boldsymbol{Z} - \boldsymbol{X}\boldsymbol{Y}'\|_2^2 \\ \text{subject to} & \boldsymbol{Z}_{ij} = \boldsymbol{M}_{ij}, \quad (i,j) \in E. \end{array} \tag{2}$$

- Can do coordinate descent by cyclically minimizing $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$.
- Nuclear norm minimization: let P_E denote projection to the entries in E

$$\min_{\boldsymbol{Z}} \ \mathrm{rank}(\boldsymbol{Z}) + \lambda \|P_E(\boldsymbol{Z} - \boldsymbol{M})\|^2 \ \Rightarrow \ \min_{\boldsymbol{Z}} \|\boldsymbol{Z}\|_* + \lambda \|P_E(\boldsymbol{Z} - \boldsymbol{M})\|^2.$$

  $\Rightarrow$ relaxation to a large scale convex optimization problem
- Related to the Netflix problem: $\boldsymbol{M}$ contains the movie rankings from customers

Matrix Completion

- Under suitable conditions (e.g., randomly generated low rank matrix $\boldsymbol{M}$, $|E| \geq O^*(nr)$, ignoring $\log$ factors), the nuclear norm minimization leads to exact recovery of $\boldsymbol{M}$ w.h.p.. That is, matrix completion via convex optimization
- Nuclear norm minimization $\Leftrightarrow$ Semidefinite programming, slow for large m, n.
- Solving the nonconvex formulation (2) using CD is much more efficient; also has exact recovery property?
- Special case: $E = \{1, 2, ..., m\} \times \{1, 2, ..., n\}$. Then the nonconvex formulation (2) becomes

$$\min_{\boldsymbol{X} \in \mathbb{R}^{m \times r}, \boldsymbol{Y} \in \mathbb{R}^{n \times r}} \|\boldsymbol{M} - \boldsymbol{X}\boldsymbol{Y}'\|^2 \tag{3}$$

  whose optimal solution is given by the SVD of $\boldsymbol{M}$.
- **Surprise(?):** For almost all initial values of $\boldsymbol{X}$, $\boldsymbol{Y}$, the CD method for (3) converges to the global optimal solution (i.e., the SVD solution) linearly.

Box Constrained Convex QP

Consider a convex quadratic minimization problem over $\mathbb{R}^n_+$

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\boldsymbol{x}'\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}'\boldsymbol{x} \\ \text{subject to} & \boldsymbol{x} \in X \end{array} \tag{4}$$

If $X = \mathbb{R}^n$, we can solve (4) by a simple coordinate descent (Gauss-Seidel type) method: write $\boldsymbol{A} = \boldsymbol{A}_{\text{low}} + \boldsymbol{D} + \boldsymbol{A}_{\text{upp}} \succeq 0$ (assume $\boldsymbol{D} \succ 0$) and iterate

$$\begin{aligned} (\boldsymbol{A}_{\text{low}} + \boldsymbol{D})\boldsymbol{x}^{r+1} + \boldsymbol{A}_{\text{upp}}\boldsymbol{x}^r + \boldsymbol{b} = 0 \\ \Rightarrow \boldsymbol{x}^{r+1} = -(\boldsymbol{A}_{\text{low}} + \boldsymbol{D})^{-1}(\boldsymbol{A}_{\text{upp}}\boldsymbol{x}^r + \boldsymbol{b}). \end{aligned}$$

If $X = \mathbb{R}^n_+$, the coordinate descent (CD) algorithm becomes

$$\boldsymbol{x}^{r+1} = [\boldsymbol{x}^{r+1} - (\boldsymbol{A}_{\text{low}} + \boldsymbol{D})\boldsymbol{x}^{r+1} - \boldsymbol{A}_{\text{upp}}\boldsymbol{x}^r - \boldsymbol{b}]_+$$

where $[u]+ = \max\{0, u\}$; related to SOR, block SOR.

Convex QP

Consider a convex quadratic minimization problem over $\mathbb{R}^n_+$

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\boldsymbol{x}'\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}'\boldsymbol{x} \\ \text{subject to} & \boldsymbol{x} \in X \end{array}$$

- CD has many applications in image processing, MRI, ... Convergence was known for symmetric $\boldsymbol{A} \succ 0$, with unique optimal solution

- Convergence for the case symmetric $A \succeq 0$ was unresolved for many years ..., unbounded optimal solution set

$$X^* = \{\boldsymbol{x}^* \mid \boldsymbol{x}^* = \mathrm{proj}_X[\boldsymbol{x}^* - \alpha \nabla f(\boldsymbol{x}^*)]\}, \quad \alpha > 0$$

with $\mathrm{proj}_X[\cdot] =$ projection to X.

- Studied by Hildreth, Cryer, Mangasarian, Pang, ... in various contexts since 1950's.

Gradient Projection Method

Given $\boldsymbol{x}^0 \in X$, generate

$$\boldsymbol{x}^{r+1} = \mathrm{proj}_X[x^r - \alpha^r \nabla f(\boldsymbol{x}^r)], \quad r = 0, 1, 2, \ldots$$

where $\alpha^r > 0$ is the stepsize (chosen as constant or by Armijo-like rule).

Convergence for the non-degenerate case $\boldsymbol{A} \succ 0$: $\exists$ a unique solution $\boldsymbol{x}^*$ satisfying $\boldsymbol{x}^* = \mathrm{proj}_X[\boldsymbol{x}^* - \alpha^r \nabla f(\boldsymbol{x}^*)]$, so

$$\begin{aligned}
\|\boldsymbol{x}^{r+1} - \boldsymbol{x}^*\| &= \|\,\mathrm{proj}_X[\boldsymbol{x}^r - \alpha^r \nabla f(\boldsymbol{x}^r)] - \mathrm{proj}_X[\boldsymbol{x}^* - \alpha^r \nabla f(\boldsymbol{x}^*)]\| \\
&\le \|(\boldsymbol{x}^r - \boldsymbol{x}^*) - \alpha^r(\nabla f(\boldsymbol{x}^r) - \nabla f(\boldsymbol{x}^*))\| \\
&= \|(\boldsymbol{I} - \alpha^r \boldsymbol{A})(\boldsymbol{x}^r - \boldsymbol{x}^*)\| \\
&\le \left(\frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}}\right)\|\boldsymbol{x}^r - \boldsymbol{x}^*\| = \left(1 - \frac{2}{\kappa + 1}\right)\|\boldsymbol{x}^r - \boldsymbol{x}^*\|.
\end{aligned}$$

Convergence rate depends on $\kappa = \lambda_{\max}/\lambda_{\min}$, but *independent* of dimension. Requires $O(1)\kappa \ln(1/\epsilon)$ to find an ϵ-relative optimal solution.