EECS 559 Optimization Methods for SIPML

Lecture 9 – Alternating Direction Method of Multipliers

Instructor: Prof. Qing Qu (qingqu@umich.edu)

Lecture Agenda

- Augmented Lagrangian Method
- Alternating Direction Method of Multipliers

Two-block Problem

$$\min_{\boldsymbol{x},\boldsymbol{z}} \; g(\boldsymbol{x}) + h(\boldsymbol{z}), \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} \; = \; \boldsymbol{y}.$$

- $g : \mathbb{R}^n \mapsto \mathbb{R}$ and $h : \mathbb{R}^n \mapsto \mathbb{R}$ are convex functions;
- $\boldsymbol{A}$ and $\boldsymbol{B}$ are matrices and $\boldsymbol{y} \in \mathrm{range}([\boldsymbol{A} \mid \boldsymbol{B}])$, so that the problem is feasible.

Example I: Lasso

$$\min_{\boldsymbol{x}} \; \frac{1}{2} \left\| \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \right\|_2^2 \; + \; \lambda \left\| \boldsymbol{x} \right\|_1$$

Splitting the variable with $\boldsymbol{z} = \boldsymbol{x}$, and solve

$$\min_{\boldsymbol{x},\boldsymbol{z}} \; \underbrace{\frac{1}{2} \left\| \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \right\|_2^2}_{g(\boldsymbol{x})} + \underbrace{\lambda \left\| \boldsymbol{z} \right\|_1}_{h(\boldsymbol{z})}, \quad \text{s.t.} \quad \boldsymbol{x} - \boldsymbol{z} \; = \; \boldsymbol{0}.$$

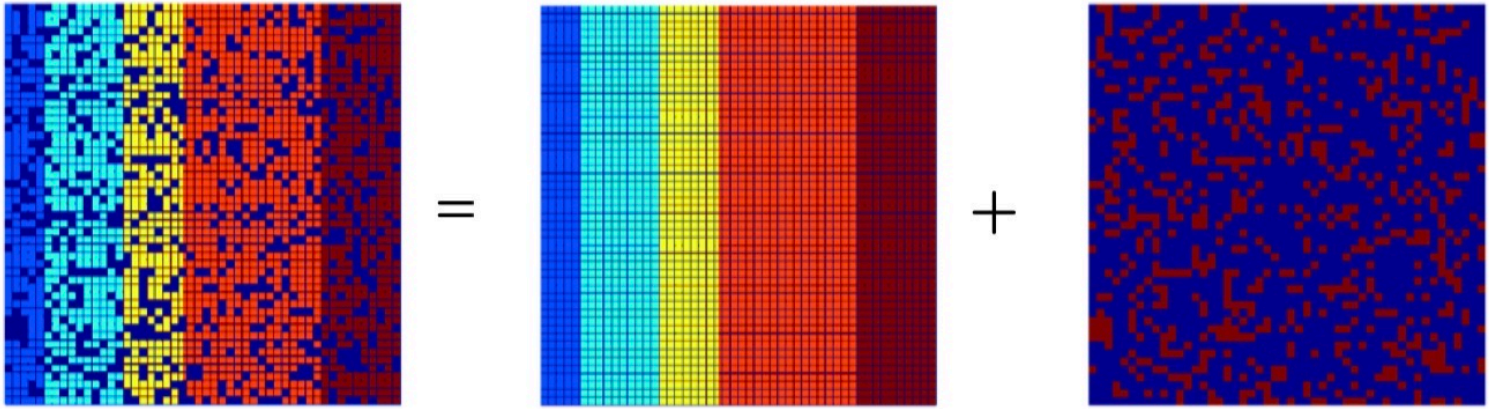
Example II: Stable Low Rank Matrix Recovery

$$\min_{\boldsymbol{X}} \ \frac{1}{2} \left\| \mathcal{A}(\boldsymbol{X}) - \boldsymbol{y} \right\|_2^2 \ + \ \lambda \left\| \boldsymbol{X} \right\|_*$$

Splitting the variable with $\boldsymbol{Z} = \boldsymbol{X}$, and solve

$$\min_{\boldsymbol{X}, \boldsymbol{Z}} \ \underbrace{\frac{1}{2} \left\| \mathcal{A}(\boldsymbol{X}) - \boldsymbol{y} \right\|_2^2}_{g(\boldsymbol{X})} + \underbrace{\lambda \left\| \boldsymbol{Z} \right\|_*}_{h(\boldsymbol{Z})}, \quad \text{s.t.} \quad \boldsymbol{X} - \boldsymbol{Z} = \boldsymbol{0}.$$

Example III: Robust PCA



observation $\boldsymbol{Y}$ low-rank $\boldsymbol{L}$ sparse $\boldsymbol{S}$

$$\min_{\boldsymbol{L},\,\boldsymbol{S}} \underbrace{\|\boldsymbol{L}\|_*}_{g(\boldsymbol{L})} + \underbrace{\lambda\,\|\boldsymbol{S}\|_1}_{h(\boldsymbol{S})}, \quad \text{s.t.} \quad \boldsymbol{L} + \boldsymbol{S} = \boldsymbol{Y}.$$

Lecture Agenda

- **Augmented Lagrange Method**

- Alternating Direction Method of Multipliers

Linear Equality Constrained Problems

Start with a simpler problem than the two-block problem

$$(\mathrm{P}) \qquad \min_{\boldsymbol{x}} \ g(\boldsymbol{x}), \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}.$$

- $g : \mathbb{R}^n \mapsto \mathbb{R}$ is a convex function;
- $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{y} \in \mathrm{range}(\boldsymbol{A})$, so that the problem is feasible.

Remove Constraints via Penalty

Idea: solve the following *unconstraint* problem instead

$$\min_{\boldsymbol{x}} \; g(\boldsymbol{x}) \; + \; \frac{\mu}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2$$

for a very large value of μ.

- **Pros:** As μ increases to $+\infty$, the solution approaches the solution of the equality constrained problem.
- **Cons:** The rate of convergence is measured by $L_{\mu \nabla f} = \mu \|\boldsymbol{A}\|_2^2$ The *larger* μ is, the *harder* the unconstrained problem is.

Lagrangian Duality

Definition. The Lagrange function of the constrained problem (P) is

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) \;:=\; g(\boldsymbol{x}) \;+\; \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \rangle \,.$$

where $\boldsymbol{\lambda} \in \mathbb{R}^m$ is called the Lagrangian multiplier.

We define the dual function of $g(\boldsymbol{x})$ as

$$d(\boldsymbol{\lambda}) \;:=\; \inf_{\boldsymbol{x}} \; g(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \rangle \,.$$

Lagrangian Duality

Definition. The Lagrange function of the constrained problem (P) is

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) \;:=\; g(\boldsymbol{x}) \;+\; \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \rangle \,.$$

where $\boldsymbol{\lambda} \in \mathbb{R}^m$ is called the Lagrangian multiplier.

Finding the optimal solution of (P) is equivalent to find the *saddle point* of the Lagrangian function:

$$\sup_{\boldsymbol{\lambda}} \inf_{\boldsymbol{x}} \; \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) \;=\; \sup_{\boldsymbol{\lambda}} \inf_{\boldsymbol{x}} \; g(\boldsymbol{x}) \;+\; \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \rangle \;=\; \sup_{\boldsymbol{\lambda}} \; d(\boldsymbol{\lambda}).$$

Dual Ascent for Lagrangian

A natural computational approach for optimizing the Lagrangian function is via the *dual ascent*

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}_k) \\ \boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k + t_{k+1} \left(\boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{y} \right). \end{aligned}$$

- For certain problem classes, dual ascent yields efficient, convergent algorithms to an optimal primal-dual solution$(\boldsymbol{x}_\star, \boldsymbol{\lambda}_\star)$
- However, it may *fail* for problems in structured signal recovery.

Example: Basis Pursuit

$$\min_{\boldsymbol{x}} \ \|\boldsymbol{x}\|_1 , \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}.$$

We can show that the dual function

$$d(\boldsymbol{\lambda}) = \inf_{\boldsymbol{x}} \ \|\boldsymbol{x}\|_1 + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \rangle = \begin{cases} -\infty & \left\|\boldsymbol{A}^\top \boldsymbol{\lambda}\right\|_\infty > 1 \\ -\langle \boldsymbol{\lambda}, \boldsymbol{y} \rangle & \left\|\boldsymbol{A}^\top \boldsymbol{\lambda}\right\|_\infty \leq 1 \end{cases}$$

Whenever $\left\|\boldsymbol{A}^\top \boldsymbol{\lambda}\right\|_\infty > 1$, the dual ascent algorithm breaks down.

Dual Ascent for Lagrangian

- This type of bad scenarios occurs more *generally*.
- It happens because the Lagrangian does *not* penalize the constraint $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$ strongly enough to lead to a useful algorithm.
- It is *only* sufficient for characterizing optimality conditions.

Remedy: Augmented Lagrangian

Definition. The augmented Lagrangian function of the problem (P) is defined as

$$\mathcal{L}_\mu(\boldsymbol{x}, \boldsymbol{\lambda}) \;:=\; g(\boldsymbol{x}) \;+\; \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \rangle \;+\; \frac{\mu}{2} \left\| \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \right\|_2^2 ,$$

where $\mu > 0$ is a penalty parameter.

The augmented Lagrangian can be regarded as the Lagrangian function for

$$\min_{\boldsymbol{x}} \; g(\boldsymbol{x}) + \frac{\mu}{2} \left\| \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \right\|_2^2 , \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} \;=\; \boldsymbol{y}.$$

Remedy: Augmented Lagrangian

Definition. The augmented Lagrangian function of the problem (P) is defined as

$$\mathcal{L}_\mu(\boldsymbol{x}, \boldsymbol{\lambda}) \;:=\; g(\boldsymbol{x}) \;+\; \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \rangle \;+\; \frac{\mu}{2} \left\| \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \right\|_2^2,$$

where $\mu > 0$ is a penalty parameter.

Method of Multipliers with $t_{k+1} \equiv \mu$:

$$\boldsymbol{x}_{k+1} \;\in\; \arg\min_{\boldsymbol{x}} \; \mathcal{L}_\mu(\boldsymbol{x}, \boldsymbol{\lambda}_k),$$

$$\boldsymbol{\lambda}_{k+1} \;=\; \boldsymbol{\lambda}_k \;+\; \mu \left(\boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{y} \right).$$

Remedy: Augmented Lagrangian

Choice of $t_k \equiv \mu$ avoids algorithm breakdown:

$$\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}} \mathcal{L}_u(\boldsymbol{x}, \boldsymbol{\lambda}_k)$$

$$\begin{aligned} \Longrightarrow \quad \mathbf{0} &\in \partial \mathcal{L}_\mu(\boldsymbol{x}_{k+1}, \boldsymbol{\lambda}_k) \\ &= \partial g(\boldsymbol{x}_{k+1}) + \boldsymbol{A}^\top \boldsymbol{\lambda}_k + \mu \boldsymbol{A}^\top (\boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{y}) \\ &= \partial g(\boldsymbol{x}_{k+1}) + \boldsymbol{A}^\top \boldsymbol{\lambda}_{k+1} = \partial \mathcal{L}(\boldsymbol{x}_{k+1}, \boldsymbol{\lambda}_{k+1}) \end{aligned}$$

Thus, $\boldsymbol{x}_{k+1}$ minimizes the original Lagrangian $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}_{k+1})$.

Augmented Lagrange Multiplier (ALM)

Augmented Lagrange Multipler (ALM)

Problem Class:

$$\begin{array}{ll} \min_{\boldsymbol{x}} & g(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \end{array}$$

$g : \mathbb{R}^n \to \mathbb{R}$ convex, $\boldsymbol{y} \in \operatorname{range}(\boldsymbol{A})$.

Basic Iteration: set

$$\mathcal{L}_\mu(\boldsymbol{x}, \boldsymbol{\lambda}) = g(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \rangle + \tfrac{\mu}{2} \left\| \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \right\|_2^2 .$$

Repeat:

$$\boldsymbol{x}_{k+1} \in \arg\min_{\boldsymbol{x}} \ \mathcal{L}_\mu(\boldsymbol{x}, \boldsymbol{\lambda}_k),$$

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \mu \left(\boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{y} \right).$$

Convergence Guarantee:

If g is coercive, every limit point of $\{\boldsymbol{x}_k\}$ is optimal.

Convergence of ALM

Theorem. (Convergence of ALM)
Let $g : \mathbb{R}^n \mapsto \mathbb{R}$ be a convex, coercive function, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ an arbitrary matrix, and $\boldsymbol{y} \in \mathrm{range}(\boldsymbol{A})$. Then the problem

$$\min_{\boldsymbol{x}} \; g(\boldsymbol{x}), \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}.$$

has at least one optimal solution. Moreover, the ALM

$$\boldsymbol{x}_{k+1} \in \arg\min_{\boldsymbol{x}} \mathcal{L}_{\mu_k}(\boldsymbol{x}, \boldsymbol{\lambda}_k), \; \boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \mu_k(\boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{y}),$$

with sequence $\{\mu_k\}$ bounded away from 0 produces a sequence that $\{\boldsymbol{\lambda}_k\}$ converges to a dual optimal solution with the rate $O(1/k)$. Moreover, every limit point of $\{\boldsymbol{x}_k\}$ is optimal.

ALM for More Generic Problems

$$\min_{\boldsymbol{x}} \; g_0(\boldsymbol{x}), \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}, \quad \boldsymbol{x} \in \mathcal{C},$$

where $\mathcal{C}$ is a nonempty closed, convex constraint set.

We can apply ALM to the following *equivalent* problem

$$\min_{\boldsymbol{x}} g(\boldsymbol{x}) \; := \; g_0(\boldsymbol{x}) + \mathbb{1}_{\boldsymbol{x}\in\mathcal{C}}, \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}.$$

$$\text{with} \quad \mathbb{1}_{\boldsymbol{x}\in\mathcal{C}} \; = \begin{cases} 0 & \boldsymbol{x} \in \mathcal{C}, \\ +\infty & \boldsymbol{x} \notin \mathcal{C}. \end{cases}$$

ALM for Basis Pursuit

Algorithm 8.5 Augmented Lagrange Multipler (ALM) for BP

1: **Problem:** $\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1$ subject to $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, given $\boldsymbol{y} \in \mathbb{R}^m$ and $\boldsymbol{A} \in \mathbb{R}^{m \times n}$.
2: **Input:** $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\boldsymbol{\lambda}_0 \in \mathbb{R}^m$, and $\beta > 1$.
3: **for** $(k = 0, 1, 2, \ldots, K-1)$ **do**
4: $\quad \boldsymbol{x}_{k+1} \leftarrow \arg\min \mathcal{L}_{\mu_k}(\boldsymbol{x}, \boldsymbol{\lambda}_k)$ using APG.
5: $\quad \boldsymbol{\lambda}_{k+1} \leftarrow \boldsymbol{\lambda}_k + \mu_k(\boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{y})$.
6: $\quad \mu_{k+1} \leftarrow \min\{\beta\mu_k, \mu_{\max}\}$.
7: **end for**
8: **Output:** $\boldsymbol{x}_\star \leftarrow \boldsymbol{x}_K$.

ALM for Robust PCA

Algorithm 8.6 Augmented Lagrange Multipler (ALM) for PCP

Problem: $\min_{\boldsymbol{L},\boldsymbol{S}} \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1$ subj to $\boldsymbol{L}+\boldsymbol{S}=\boldsymbol{Y}$, given $\boldsymbol{Y}$ and $\lambda > 0$.

Input: $\boldsymbol{L}_0, \boldsymbol{S}_0, \boldsymbol{\Lambda}_0 \in \mathbb{R}^{m\times n}$ and $\beta > 1$.

for $(k = 0, 1, 2, \ldots, K-1)$ **do**

$\quad \{\boldsymbol{L}_{k+1}, \boldsymbol{S}_{k+1}\} \leftarrow \arg\min \mathcal{L}_{\mu_k}(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}_k)$ using APG.

$\quad \boldsymbol{\Lambda}_{k+1} \leftarrow \boldsymbol{\Lambda}_k + \mu_k(\boldsymbol{L}_{k+1} + \boldsymbol{S}_{k+1} - \boldsymbol{Y})$.

$\quad \mu_{k+1} \leftarrow \min\{\beta\mu_k, \mu_{\max}\}$.

end for

Output: $\boldsymbol{L}_\star \leftarrow \boldsymbol{L}_K, \boldsymbol{S}_\star \leftarrow \boldsymbol{S}_K$.

Summary of ALM

- The ALM method finds a saddle point of the Augmented Lagrangian function, with $O(1/k)$ convergence rate.

- However, the ALM method needs to solve a subproblem, which could be expensive.

Further Readings

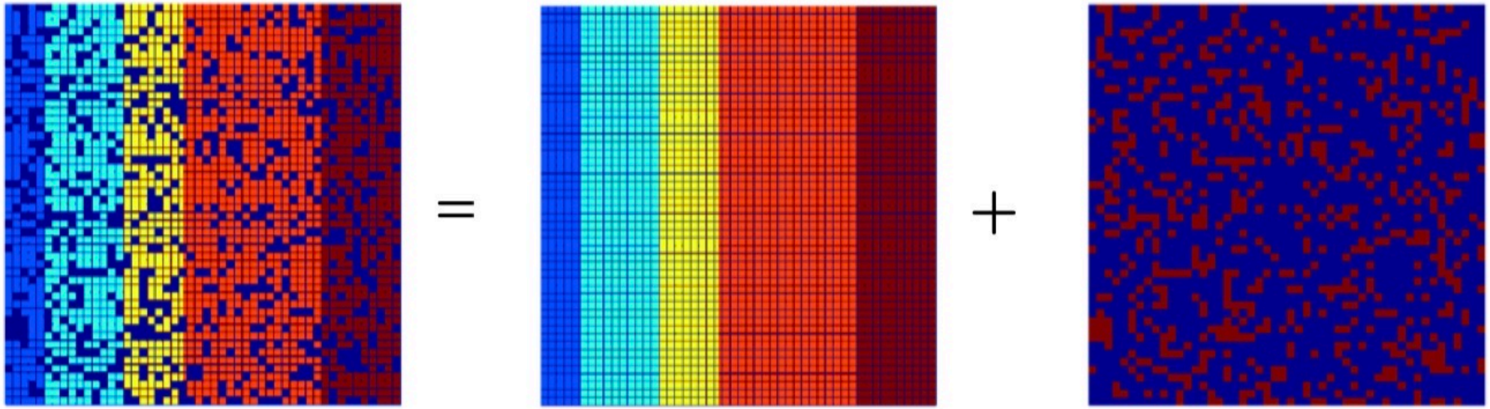
- *High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications*. John Wright, Yi Ma. (**Chapter 8.4**)

Lecture Agenda

- Augmented Lagrange Method

- **Alternating Direction Method of Multipliers**

Example: Robust PCA



observation $\boldsymbol{Y}$ = low-rank $\boldsymbol{L}$ + sparse $\boldsymbol{S}$

$$\min_{\boldsymbol{L},\,\boldsymbol{S}} \underbrace{\|\boldsymbol{L}\|_*}_{g(\boldsymbol{L})} + \underbrace{\lambda\,\|\boldsymbol{S}\|_1}_{h(\boldsymbol{S})}, \quad \text{s.t.} \quad \boldsymbol{L} + \boldsymbol{S} = \boldsymbol{Y}.$$

Two-block Problem

$$\min_{\boldsymbol{x},\boldsymbol{z}} \; g(\boldsymbol{x}) + h(\boldsymbol{z}), \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} \; = \; \boldsymbol{y}.$$

- $g : \mathbb{R}^n \mapsto \mathbb{R}$ and $h : \mathbb{R}^n \mapsto \mathbb{R}$ are convex functions;
- $\boldsymbol{A}$ and $\boldsymbol{B}$ are matrices and $\boldsymbol{y} \in \mathrm{range}([\boldsymbol{A} \mid \boldsymbol{B}])$, so that the problem is feasible.

Drawbacks of ALM

$$\min_{\boldsymbol{x},\boldsymbol{z}} \; g(\boldsymbol{x}) + h(\boldsymbol{z}), \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} \; = \; \boldsymbol{y}.$$

- Form the augmented Lagrangian

$$\begin{aligned} \mathcal{L}_\mu(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda}) \; &= g(\boldsymbol{x}) + h(\boldsymbol{z}) + \langle \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{y}, \boldsymbol{\lambda} \rangle \\ &\quad + \frac{\mu}{2} \left\| \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{y} \right\|_2^2 \end{aligned}$$

- Solve the problem via

$$(\boldsymbol{x}_{k+1}, \boldsymbol{z}_{k+1}) \; \in \; \arg\min_{\boldsymbol{x},\boldsymbol{z}} \; \mathcal{L}_\mu(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda}_k), \quad \text{(could be expensive)}$$

$$\boldsymbol{\lambda}_{k+1} \; = \; \boldsymbol{\lambda}_k \; + \; \mu_k \left(\boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{z}_{k+1} - \boldsymbol{y} \right).$$

Alternating Directions Method of Multipliers

$$\min_{\boldsymbol{x},\boldsymbol{z}} \; g(\boldsymbol{x}) + h(\boldsymbol{z}), \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} \; = \; \boldsymbol{y}.$$

- fix $\boldsymbol{z}$ and $\boldsymbol{\lambda}$, minimize $\boldsymbol{x}$:

$$\boldsymbol{x}_{k+1} \; \in \; \arg\min_{\boldsymbol{x}} \; \mathcal{L}_{\mu}(\boldsymbol{x}, \boldsymbol{z}_k, \boldsymbol{\lambda}_k),$$

- fix $\boldsymbol{x}$ and $\boldsymbol{\lambda}$, minimize $\boldsymbol{z}$:

$$\boldsymbol{z}_{k+1} \; \in \; \arg\min_{\boldsymbol{z}} \; \mathcal{L}_{\mu}(\boldsymbol{x}_{k+1}, \boldsymbol{z}, \boldsymbol{\lambda}_k),$$

- fix $\boldsymbol{x}$ and $\boldsymbol{z}$, take a dual ascent step on $\boldsymbol{\lambda}$:

$$\boldsymbol{\lambda}_{k+1} \; = \; \boldsymbol{\lambda}_k \; + \; \mu_k \nabla_{\boldsymbol{\lambda}} \mathcal{L}_{\mu}(\boldsymbol{x}_{k+1}, \boldsymbol{z}_{k+1}, \lambda).$$

Alternating Directions Method of Multipliers

- fix $\boldsymbol{z}$ and $\boldsymbol{\lambda}$, minimize $\boldsymbol{x}$:

$$\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}} \left\{ g(\boldsymbol{x}) + \frac{\mu}{2} \left\| \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z}_k - \boldsymbol{y} + \frac{1}{\mu}\boldsymbol{\lambda}_k \right\|_2^2 \right\},$$

- fix $\boldsymbol{x}$ and $\boldsymbol{\lambda}$, minimize $\boldsymbol{z}$:

$$\boldsymbol{z}_{k+1} = \arg\min_{\boldsymbol{z}} \left\{ h(\boldsymbol{z}) + \frac{\mu}{2} \left\| \boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{y} + \frac{1}{\mu}\boldsymbol{\lambda}_k \right\|_2^2 \right\},$$

- fix $\boldsymbol{x}$ and $\boldsymbol{z}$, take a dual ascent step on $\boldsymbol{\lambda}$:

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \mu_k \left(\boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{z}_{k+1} - \boldsymbol{y} \right).$$

Example I: Robust PCA

$$\min_{\boldsymbol{L},\boldsymbol{S}} \ \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 , \quad \text{s.t.} \quad \boldsymbol{L} + \boldsymbol{S} = \boldsymbol{Y} .$$

- Form the augmented Lagrangian:

$$\begin{aligned} \mathcal{L}_\mu(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) \ &= \ \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{L} + \boldsymbol{S} - \boldsymbol{Y} \rangle \\ &\quad + \frac{\mu}{2} \|\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{Y}\|_F^2 . \end{aligned}$$

Example I: Robust PCA

$$\min_{\boldsymbol{L},\boldsymbol{S}} \ \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 , \quad \text{s.t.} \quad \boldsymbol{L} + \boldsymbol{S} = \boldsymbol{Y} .$$

- Fix $\boldsymbol{S}_k$ and $\boldsymbol{\Lambda}_k$, find $\boldsymbol{L}_{k+1}$ via

$$\begin{aligned} \boldsymbol{L}_{k+1} &= \arg\min_{\boldsymbol{L}} \ \mathcal{L}_\mu(\boldsymbol{L}, \boldsymbol{S}_k, \boldsymbol{\Lambda}_k) \\ &= \arg\min_{\boldsymbol{L}} \left\{ \|\boldsymbol{L}\|_* + \frac{\mu}{2} \left\| \boldsymbol{L} + \boldsymbol{S}_k - \boldsymbol{Y} + \frac{1}{\mu}\boldsymbol{\Lambda}_k \right\|_F^2 \right\} \\ &= \operatorname{prox}_{\mu^{-1}\|\cdot\|_*} \left(\boldsymbol{Y} - \boldsymbol{S}_k - \mu^{-1}\boldsymbol{\Lambda}_k \right) . \end{aligned}$$

Example I: Robust PCA

$$\min_{\boldsymbol{L},\boldsymbol{S}} \ \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 , \quad \text{s.t.} \quad \boldsymbol{L} + \boldsymbol{S} = \boldsymbol{Y}.$$

- Fix $\boldsymbol{L}_{k+1}$ and $\boldsymbol{\Lambda}_k$, find $\boldsymbol{S}_{k+1}$ via

$$\begin{aligned}
\boldsymbol{S}_{k+1} &= \arg\min_{\boldsymbol{S}} \mathcal{L}_\mu(\boldsymbol{L}_{k+1}, \boldsymbol{S}, \boldsymbol{\Lambda}_k) \\
&= \arg\min_{\boldsymbol{S}} \left\{ \lambda \|\boldsymbol{S}\|_1 + \frac{\mu}{2} \left\| \boldsymbol{L}_{k+1} + \boldsymbol{S} - \boldsymbol{Y} + \frac{1}{\mu} \boldsymbol{\Lambda}_k \right\|_F^2 \right\} \\
&= \operatorname{prox}_{\lambda\mu^{-1}\|\cdot\|_1} \left(\boldsymbol{Y} - \boldsymbol{L}_{k+1} - \mu^{-1} \boldsymbol{\Lambda}_k \right).
\end{aligned}$$

Example I: Robust PCA

- Fix $\boldsymbol{S}_k$ and $\boldsymbol{\Lambda}_k$, find $\boldsymbol{L}_{k+1}$ via

$$\boldsymbol{L}_{k+1} \;=\; \operatorname{prox}_{\mu^{-1}\|\cdot\|_*}\left(\boldsymbol{Y} - \boldsymbol{S}_k - \mu^{-1}\boldsymbol{\Lambda}_k\right).$$

- Fix $\boldsymbol{L}_{k+1}$ and $\boldsymbol{\Lambda}_k$, find $\boldsymbol{S}_{k+1}$ via

$$\boldsymbol{S}_{k+1} \;=\; \operatorname{prox}_{\lambda\mu^{-1}\|\cdot\|_1}\left(\boldsymbol{Y} - \boldsymbol{L}_{k+1} - \mu^{-1}\boldsymbol{\Lambda}_k\right).$$

- Fix $\boldsymbol{L}_{k+1}$ and $\boldsymbol{S}_{k+1}$, take a dual ascent step on $\boldsymbol{\Lambda}$:

$$\boldsymbol{\Lambda}_{k+1} \;=\; \boldsymbol{\Lambda}_k \;+\; \mu_k\left(\boldsymbol{L}_{k+1} + \boldsymbol{S}_{k+1} - \boldsymbol{Y}\right).$$

Example I: Robust PCA

Algorithm 8.7 ADMM for Principal Component Pursuit

1: **Problem:** $\min_{\boldsymbol{L},\boldsymbol{S}} \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{L}+\boldsymbol{S}-\boldsymbol{Y}\rangle + \frac{\mu}{2}\|\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{Y}\|_F^2$, given $\boldsymbol{Y}$, $\lambda, \mu > 0$.

2: **Input:** $\boldsymbol{L}_0, \boldsymbol{S}_0, \boldsymbol{\Lambda}_0 \in \mathbb{R}^{m\times n}$.

3: **for** $(k = 0, 1, 2, \ldots, K-1)$ **do**

4: $\quad \boldsymbol{L}_{k+1} \leftarrow \operatorname{prox}_{\mu^{-1}\|\cdot\|_*}\left[\boldsymbol{Y} - \boldsymbol{S}_k - \mu^{-1}\boldsymbol{\Lambda}_k\right]$.

5: $\quad \boldsymbol{S}_{k+1} \leftarrow \operatorname{prox}_{\lambda\mu^{-1}\|\cdot\|_1}\left[\boldsymbol{Y} - \boldsymbol{L}_{k+1} - \mu^{-1}\boldsymbol{\Lambda}_k\right]$.

6: $\quad \boldsymbol{\Lambda}_{k+1} \leftarrow \boldsymbol{\Lambda}_k + \mu(\boldsymbol{L}_{k+1} + \boldsymbol{S}_{k+1} - \boldsymbol{Y})$.

7: **end for**

8: **Output:** $\boldsymbol{L}_\star \leftarrow \boldsymbol{L}_K; \boldsymbol{S}_\star \leftarrow \boldsymbol{S}_K$.

Example II: Sparse Graphical Lasso

- When learning a sparse Gaussian graphical model, one resorts to

$$\min_{\boldsymbol{\Theta}} \; -\log\det\boldsymbol{\Theta} + \langle \boldsymbol{\Theta}, \boldsymbol{S} \rangle + \lambda \cdot \left\| \boldsymbol{\Theta} \right\|_1, \quad \text{s.t.} \quad \boldsymbol{\Theta} \succeq \mathbf{0}.$$

- This is equivalent to

$$\min_{\boldsymbol{\Theta},\,\boldsymbol{\Phi}} \; -\log\det\boldsymbol{\Theta} + \langle \boldsymbol{\Theta}, \boldsymbol{S} \rangle + \mathbb{1}_{\boldsymbol{\Theta} \in \mathcal{S}_+} + \lambda \cdot \left\| \boldsymbol{\Phi} \right\|_1, \quad \text{s.t.} \quad \boldsymbol{\Theta} = \boldsymbol{\Phi}.$$

Here, we define $\mathcal{S}_+ := \left\{ \boldsymbol{X} \in \mathbb{R}^{n \times n} \;\middle|\; \boldsymbol{X} \succeq \mathbf{0} \right\}.$

Example II: Sparse Graphical Lasso

$$\min_{\boldsymbol{\Theta},\ \boldsymbol{\Phi}} \; -\log\det\boldsymbol{\Theta} + \langle \boldsymbol{\Theta}, \boldsymbol{S} \rangle + \mathbb{1}_{\boldsymbol{\Theta}\in\mathcal{S}_+} + \lambda\cdot\|\boldsymbol{\Phi}\|_1\,, \quad \text{s.t.} \quad \boldsymbol{\Theta} = \boldsymbol{\Phi}.$$

- Form the augmented Lagrangian:

$$\begin{aligned}\mathcal{L}_\mu(\boldsymbol{\Theta},\boldsymbol{\Phi},\boldsymbol{\Lambda}) \;=\; & -\log\det\boldsymbol{\Theta} + \langle \boldsymbol{\Theta}, \boldsymbol{S} \rangle + \mathbb{1}_{\boldsymbol{\Theta}\in\mathcal{S}_+} + \lambda\cdot\|\boldsymbol{\Phi}\|_1 \\ & + \langle \boldsymbol{\Lambda}, \boldsymbol{\Theta}-\boldsymbol{\Phi} \rangle + \frac{\mu}{2}\|\boldsymbol{\Theta}-\boldsymbol{\Phi}\|_F^2\end{aligned}$$

Example II: Sparse Graphical Lasso

- Fix $\mathbf{\Phi}_k$ and $\mathbf{\Lambda}_k$, find $\mathbf{\Theta}_{k+1}$ via

$$
\begin{aligned}
\mathbf{\Theta}_{k+1} &= \arg\min_{\mathbf{\Theta}} \mathcal{L}_\mu(\mathbf{\Theta}, \mathbf{\Phi}_k, \mathbf{\Lambda}_k) \\
&= \arg\min_{\mathbf{\Theta} \succeq \mathbf{0}} \left\{ -\log\det \mathbf{\Theta} + \frac{\mu}{2} \left\| \mathbf{\Theta} - \mathbf{\Phi}_k + \frac{1}{\mu}\mathbf{\Lambda}_k + \frac{1}{\mu}\boldsymbol{S} \right\|_F^2 \right\} \\
&= \mathcal{F}_\mu \left(\mathbf{\Phi}_k - \frac{1}{\mu}\mathbf{\Lambda}_k - \frac{1}{\mu}\boldsymbol{S} \right).
\end{aligned}
$$

Here, $\mathcal{F}_\mu(\boldsymbol{X}) := \frac{1}{2\mu}\boldsymbol{U} \operatorname{diag}\left(\left\{\lambda_i + \sqrt{\lambda_i^2 + 4/\mu}\right\}\right)\boldsymbol{U}^\top$
with $\boldsymbol{X} = \boldsymbol{U}\mathbf{\Lambda}\boldsymbol{U}^\top$.

Example II: Sparse Graphical Lasso

- Fix $\mathbf{\Theta}_{k+1}$ and $\mathbf{\Lambda}_k$, find $\mathbf{\Phi}_{k+1}$ via

$$
\begin{aligned}
\mathbf{\Phi}_{k+1} &= \arg\min_{\mathbf{\Phi}} \mathcal{L}_\mu(\mathbf{\Theta}_{k+1}, \mathbf{\Phi}, \mathbf{\Lambda}_k) \\
&= \arg\min_{\mathbf{\Phi}} \left\{ \lambda \left\| \mathbf{\Phi} \right\|_1 + \frac{\mu}{2} \left\| \mathbf{\Theta}_{k+1} + \frac{1}{\mu} \mathbf{\Lambda}_k - \mathbf{\Phi} \right\|_F^2 \right\} \\
&= \operatorname{prox}_{\lambda\mu^{-1}\|\cdot\|_1} \left(\mathbf{\Theta}_{k+1} + \frac{1}{\mu} \mathbf{\Lambda}_k \right).
\end{aligned}
$$

- Fix $\mathbf{\Theta}_{k+1}$ and $\mathbf{\Phi}_{k+1}$, take a dual ascent step on $\mathbf{\Lambda}$:

$$
\mathbf{\Lambda}_{k+1} = \mathbf{\Lambda}_k + \mu (\mathbf{\Theta} - \mathbf{\Phi}).
$$

Example II: Sparse Graphical Lasso

- Fix $\boldsymbol{\Phi}_k$ and $\boldsymbol{\Lambda}_k$, find $\boldsymbol{\Theta}_{k+1}$ via

$$\boldsymbol{\Theta}_{k+1} = \mathscr{F}_{\mu}\left(\boldsymbol{\Phi}_k - \frac{1}{\mu}\boldsymbol{\Lambda}_k - \frac{1}{\mu}\boldsymbol{S}\right).$$

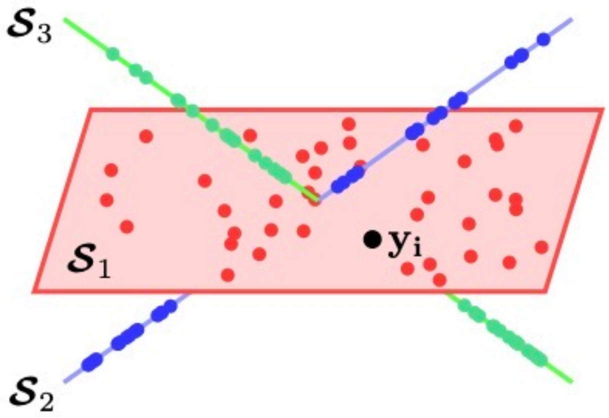
- Fix $\boldsymbol{\Theta}_{k+1}$ and $\boldsymbol{\Lambda}_k$, find $\boldsymbol{\Phi}_{k+1}$ via

$$\boldsymbol{\Phi}_{k+1} = \operatorname{prox}_{\lambda\mu^{-1}\|\cdot\|_1}\left(\boldsymbol{\Theta}_{k+1} + \frac{1}{\mu}\boldsymbol{\Lambda}_k\right).$$

- Fix $\boldsymbol{\Theta}_{k+1}$ and $\boldsymbol{\Phi}_{k+1}$, take a dual ascent step on $\boldsymbol{\Lambda}$:

$$\boldsymbol{\Lambda}_{k+1} = \boldsymbol{\Lambda}_k + \mu\left(\boldsymbol{\Theta} - \boldsymbol{\Phi}\right).$$

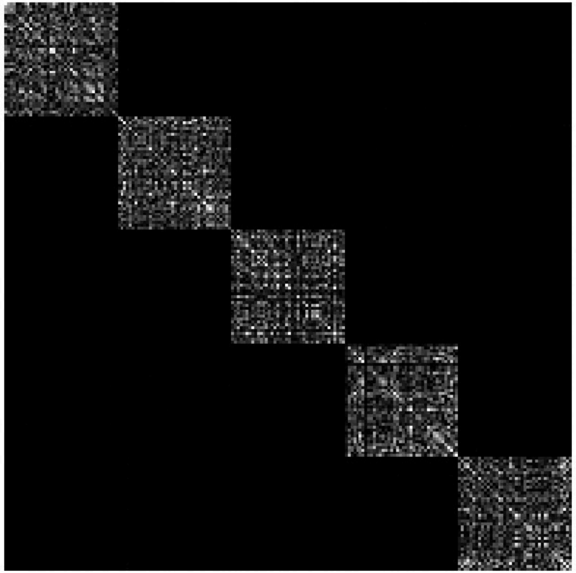
Example III: Sparse Subspace Clustering



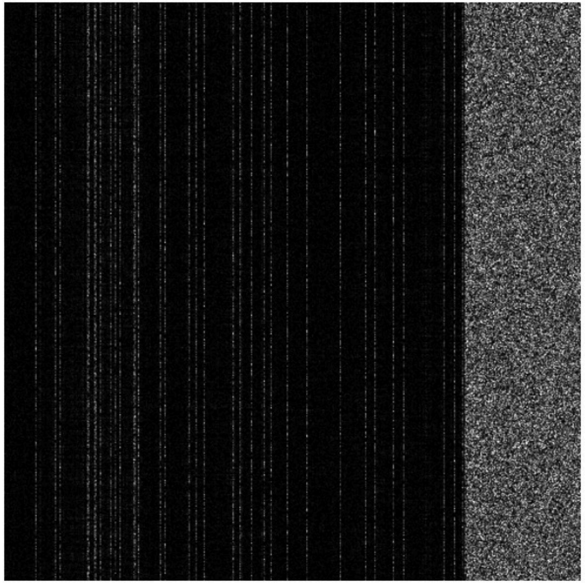
$$\boldsymbol{Y} = \boldsymbol{Y} \cdot \underbrace{\boldsymbol{C}}_{\text{self expressiveness}} + \underbrace{\boldsymbol{E}}_{\text{outliers}} + \underbrace{\boldsymbol{Z}}_{\text{noise}}$$

$$\min_{\boldsymbol{C},\boldsymbol{E}} \|\boldsymbol{C}\|_1 + \lambda_e \|\boldsymbol{E}\|_1 + \frac{\lambda_z}{2} \|\boldsymbol{Y} - \boldsymbol{Y}\boldsymbol{C} - \boldsymbol{E}\|_F^2, \quad \text{s.t.} \quad \mathrm{diag}(\boldsymbol{C}) = \boldsymbol{0}.$$

Example III: Sparse Subspace Clustering



$\boldsymbol{C}$



$\boldsymbol{E}$

Example III: Sparse Subspace Clustering

Fig. 1. Motion segmentation: given feature points on multiple rigidly moving objects tracked in multiple frames of a video (top), the goal is to separate the feature trajectories according to the moving objects (bottom).

Example III: Sparse Subspace Clustering

Fig. 1. Motion segmentation: given feature points on multiple rigidly moving objects tracked in multiple frames of a video (top), the goal is to separate the feature trajectories according to the moving objects (bottom).

Fig. 2. Face clustering: given face images of multiple subjects (top), the goal is to find images that belong to the same subject (bottom).

Example III: Sparse Subspace Clustering

- Split the variable $\boldsymbol{X} = \boldsymbol{C} - \operatorname{diag}(\boldsymbol{C})$

$$\min_{\boldsymbol{C},\boldsymbol{X},\boldsymbol{E}} \|\boldsymbol{C}\|_1 + \lambda_e \|\boldsymbol{E}\|_1 + \frac{\lambda_z}{2} \|\boldsymbol{Y} - \boldsymbol{Y}\boldsymbol{X} - \boldsymbol{E}\|_F^2 ,$$

$$\text{s.t.} \quad \boldsymbol{X} = \boldsymbol{C} - \operatorname{diag}(\boldsymbol{C}) .$$

- Form the augmented Lagrangian function:

$$\mathcal{L}_\mu(\boldsymbol{C}, \boldsymbol{X}, \boldsymbol{E}, \boldsymbol{\Lambda}) = \|\boldsymbol{C}\|_1 + \lambda_e \|\boldsymbol{E}\|_1 + \frac{\lambda_z}{2} \|\boldsymbol{Y} - \boldsymbol{Y}\boldsymbol{X} - \boldsymbol{E}\|_F^2$$

$$+ \langle \boldsymbol{\Lambda}, \boldsymbol{X} - \boldsymbol{C} + \operatorname{diag}(\boldsymbol{C}) \rangle + \frac{\mu}{2} \|\boldsymbol{X} - \boldsymbol{C} + \operatorname{diag}(\boldsymbol{C})\|_F^2$$

Example III: Sparse Subspace Clustering

- Fix $\boldsymbol{C}_k$, $\boldsymbol{E}_k$, and $\boldsymbol{\Lambda}_k$, find $\boldsymbol{X}_{k+1}$ via

$$\left(\lambda_z \boldsymbol{Y}^\top \boldsymbol{Y} + \mu \boldsymbol{I}\right) \boldsymbol{X} \;=\; \boldsymbol{Y}^\top \left(\boldsymbol{Y} - \boldsymbol{E}_k\right) + \mu \left(\boldsymbol{C}_k - \operatorname{diag}\left(\boldsymbol{C}_k\right)\right).$$

- Fix $\boldsymbol{X}_{k+1}$, $\boldsymbol{E}_k$, and $\boldsymbol{\Lambda}_k$, find $\boldsymbol{C}_{k+1}$ via

$$\boldsymbol{C}_{k+1} \;=\; \boldsymbol{J} - \operatorname{diag}\left(\boldsymbol{J}\right), \quad \boldsymbol{J} \;=\; \mathcal{S}_{\mu^{-1}}\left(\boldsymbol{X}_{k+1} + \mu^{-1} \boldsymbol{\Lambda}\right)$$

- Fix $\boldsymbol{X}_{k+1}$, $\boldsymbol{C}_{k+1}$, and $\boldsymbol{\Lambda}_k$, find $\boldsymbol{E}_{k+1}$ via

$$\boldsymbol{E}_{k+1} \;=\; \mathcal{S}_{\lambda_e/\lambda_z}\left(\boldsymbol{Y} - \boldsymbol{Y}\boldsymbol{X}\right)$$

- Fix $\boldsymbol{X}_{k+1}$, $\boldsymbol{C}_{k+1}$, and $\boldsymbol{E}_{k+1}$, find $\boldsymbol{\Lambda}_{k+1}$ via

$$\boldsymbol{\Lambda}_{k+1} \;=\; \boldsymbol{\Lambda}_k + \mu \left(\boldsymbol{X}_{k+1} - \boldsymbol{C}_k + \operatorname{diag}\left(\boldsymbol{C}_k\right)\right)$$

Example IV: Consensus Optimization

Consider solving the following problem:

$$\min_{\boldsymbol{x}} \sum_{i=1}^{N} f_i(\boldsymbol{x}),$$

which is equivalent to

$$\min_{\{\boldsymbol{x}_i\}_{i\geq 1},\, \boldsymbol{z}} \sum_{i=1}^{N} f_i(\boldsymbol{x}_i), \quad \text{s.t.} \quad \boldsymbol{x}_i = \boldsymbol{z} \quad (1 \leq i \leq N).$$

Example IV: Consensus Optimization

Consider solving the following problem:

$$\min_{\boldsymbol{x}} \sum_{i=1}^{N} f_i(\boldsymbol{x}),$$

which is equivalent to

$$\min_{\boldsymbol{u},\, \boldsymbol{z}} \sum_{i=1}^{N} f_i(\boldsymbol{x}_i), \quad \text{s.t.} \quad \boldsymbol{u} := \begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} \\ \vdots \\ \boldsymbol{I} \end{bmatrix} \boldsymbol{z}.$$

Example IV: Consensus Optimization

$$\min_{\boldsymbol{u},\,\boldsymbol{z}} \sum_{i=1}^{N} f_i(\boldsymbol{x}_i), \quad \text{s.t.} \quad \boldsymbol{u} := \boldsymbol{B}\boldsymbol{z}, \quad \boldsymbol{u} = \begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_N \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} \boldsymbol{I} \\ \vdots \\ \boldsymbol{I} \end{bmatrix}$$

Form the augmented Lagrangian function:

$$\mathscr{L}_\mu(\boldsymbol{u}, \boldsymbol{z}, \boldsymbol{\lambda}) = \sum_{i=1}^{N} f_i(\boldsymbol{x}_i) + \langle \boldsymbol{\lambda}, \boldsymbol{u} - \boldsymbol{B}\boldsymbol{z} \rangle + \frac{\mu}{2} \|\boldsymbol{u} - \boldsymbol{B}\boldsymbol{z}\|_2^2 .$$

Example IV: Consensus Optimization

- Fix $\boldsymbol{z}_k$ and $\boldsymbol{\lambda}_k$, find $\boldsymbol{u}_{k+1}$ via

$$\boldsymbol{u}_{k+1} = \arg \min_{\boldsymbol{u}=[\boldsymbol{x}_i]_{1\le i\le N}} \left\{ \sum_{i=1}^{N} f_i(\boldsymbol{x}_i) + \frac{\mu}{2} \left\| \boldsymbol{u} - \boldsymbol{B}\boldsymbol{z}_k + \frac{1}{\mu}\boldsymbol{\lambda}_k \right\|_2^2 \right\}$$

- Fix $\boldsymbol{u}_{k+1}$ and $\boldsymbol{\lambda}_k$, find $\boldsymbol{z}_{k+1}$ via

$$\boldsymbol{z}_{k+1} = \arg\min_{\boldsymbol{z}} \left\{ \frac{\mu}{2} \left\| \boldsymbol{u}_{k+1} - \boldsymbol{B}\boldsymbol{z} + \frac{1}{\mu}\boldsymbol{\lambda}_k \right\|_2^2 \right\}$$

- Fix $\boldsymbol{u}_{k+1}$ and $\boldsymbol{z}_{k+1}$, find $\boldsymbol{\lambda}_{k+1}$ via

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \mu \left(\boldsymbol{u}_{k+1} - \boldsymbol{B}\boldsymbol{z}_{k+1} \right)$$

Example IV: Consensus Optimization

- Fix $\boldsymbol{z}_k$ and $\boldsymbol{\lambda}_k$, find $\boldsymbol{u}_{k+1}$ via

$$\boldsymbol{u}_{k+1} = \arg \min_{\boldsymbol{u}=[\boldsymbol{x}_i]_{1\le i\le N}} \sum_{i=1}^{N} \left\{ f_i(\boldsymbol{x}_i) + \frac{\mu}{2} \left\| \boldsymbol{x}_i - \boldsymbol{z}_k + \frac{1}{\mu}\boldsymbol{\lambda}_{i,k} \right\|_2^2 \right\}$$

- Fix $\boldsymbol{u}_{k+1}$ and $\boldsymbol{\lambda}_k$, find $\boldsymbol{z}_{k+1}$ via

$$\boldsymbol{z}_{k+1} = \arg\min_{\boldsymbol{z}} \left\{ \frac{\mu}{2} \sum_{i=1}^{N} \left\| \boldsymbol{x}_{i,k+1} - \boldsymbol{z} + \frac{1}{\mu}\boldsymbol{\lambda}_{i,k} \right\|_2^2 \right\}$$

- Fix $\boldsymbol{u}_{k+1}$ and $\boldsymbol{z}_{k+1}$, find $\boldsymbol{\lambda}_{k+1}$ via

$$\boldsymbol{\lambda}_{i,k+1} = \boldsymbol{\lambda}_{i,k} + \mu\left(\boldsymbol{x}_{i,k+1} - \boldsymbol{z}_{k+1}\right), \quad 1 \le i \le N.$$

Example IV: Consensus Optimization

This is equivalent to

$$\boldsymbol{x}_{i,k+1} = \arg\min_{\boldsymbol{x}_i} \left\{ f_i(\boldsymbol{x}_i) + \frac{\mu}{2} \left\| \boldsymbol{x}_i - \boldsymbol{z}_k + \frac{1}{\mu} \boldsymbol{\lambda}_{i,k} \right\|_2^2 \right\}$$

(can be computed in parallel on local workers)

$$\boldsymbol{z}_{k+1} = \frac{1}{N} \sum_{i=1}^{N} \left(\boldsymbol{x}_{i,k+1} + \frac{1}{\mu} \boldsymbol{\lambda}_{i,k} \right)$$

(gather all local iterates)

$$\boldsymbol{\lambda}_{i,k+1} = \boldsymbol{\lambda}_{i,k} + \mu \left(\boldsymbol{x}_{i,k+1} - \boldsymbol{z}_{k+1} \right)$$

("broadcast" $\boldsymbol{z}_{k+1}$ to update all local multipliers)

Example IV: Consensus Optimization

$$\boldsymbol{x}_{i,k+1} = \arg\min_{\boldsymbol{x}_i} \left\{ f_i(\boldsymbol{x}_i) + \frac{\mu}{2} \left\| \boldsymbol{x}_i - \boldsymbol{z}_k + \frac{1}{\mu}\boldsymbol{\lambda}_{i,k} \right\|_2^2 \right\}$$

(can be computed in parallel on local workers)

$$\boldsymbol{z}_{k+1} = \frac{1}{N}\sum_{i=1}^{N} \left(\boldsymbol{x}_{i,k+1} + \frac{1}{\mu}\boldsymbol{\lambda}_{i,k} \right)$$

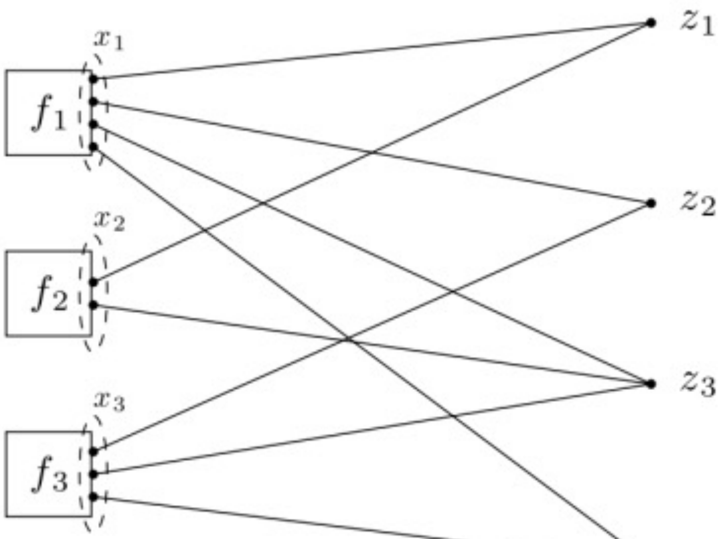
(gather all local iterates)

$$\boldsymbol{\lambda}_{i,k+1} = \boldsymbol{\lambda}_{i,k} + \mu \left(\boldsymbol{x}_{i,k+1} - \boldsymbol{z}_{k+1} \right)$$

("broadcast" $\boldsymbol{z}_{k+1}$ to update all local multipliers)

ADMM is well suited for **distributed optimization!**

Example IV: Consensus Optimization



ADMM is well suited for **distributed optimization!**

Convergence of Symmetric ADMM

$$\min_{\boldsymbol{x},\boldsymbol{z}} \; F(\boldsymbol{x},\boldsymbol{z}) \; := \; g(\boldsymbol{x}) + h(\boldsymbol{z}), \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} \; = \; \boldsymbol{y}.$$

- fix $\boldsymbol{z}$ and $\boldsymbol{\lambda}$, minimize $\boldsymbol{x}$:

$$\boldsymbol{x}_{k+1} \; = \; \arg\min_{\boldsymbol{x}} \; \left\{ g(\boldsymbol{x}) + \frac{\mu}{2} \left\| \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z}_k - \boldsymbol{y} + \frac{1}{\mu}\boldsymbol{\lambda}_k \right\|_2^2 \right\},$$

- fix $\boldsymbol{x}$ and $\boldsymbol{z}$, take a dual ascent step on $\boldsymbol{\lambda}$:

$$\boldsymbol{\lambda}_{k+1} \; = \; \boldsymbol{\lambda}_k \; + \; \mu_k \left(\boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{z}_k - \boldsymbol{y} \right),$$

- fix $\boldsymbol{x}$ and $\boldsymbol{\lambda}$, minimize $\boldsymbol{z}$:

$$\boldsymbol{z}_{k+1} \; = \; \arg\min_{\boldsymbol{z}} \; \left\{ h(\boldsymbol{z}) + \frac{\mu}{2} \left\| \boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{y} + \frac{1}{\mu}\boldsymbol{\lambda}_{k+1} \right\|_2^2 \right\}.$$

Convergence of Symmetric ADMM

To show convergence, we define the following quantities:

$$\boldsymbol{w} := \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{z} \\ \boldsymbol{\lambda} \end{bmatrix}, \quad \boldsymbol{w}_k := \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{z}_k \\ \boldsymbol{\lambda}_k \end{bmatrix}, \quad \widetilde{\boldsymbol{x}}_k := \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{x}_i, \quad \widetilde{\boldsymbol{z}}_k := \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{z}_i$$

$$\mathcal{R}(\boldsymbol{w}) := \begin{bmatrix} \boldsymbol{A}^\top \boldsymbol{\lambda} \\ \boldsymbol{B}^\top \boldsymbol{\lambda} \\ \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x} - \boldsymbol{B}\boldsymbol{z} \end{bmatrix}, \quad \boldsymbol{Q} := \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \mu \boldsymbol{B}^\top \boldsymbol{B} & -\boldsymbol{B}^\top \\ \boldsymbol{0} & -\boldsymbol{B} & \frac{1}{\mu}\boldsymbol{I} \end{bmatrix} \succeq \boldsymbol{0}$$

Convergence of Symmetric ADMM

Theorem. (Convergence of ADMM)
Let $(\boldsymbol{x}_\star, \boldsymbol{z}_\star, \boldsymbol{\lambda}_\star)$ be an optimal solution. Suppose $g(\cdot)$ and $h(\cdot)$ are closed convex functions, and let $\rho \geq \|\boldsymbol{\lambda}_\star\|_2$. Then

$$\|\boldsymbol{A}\widetilde{\boldsymbol{x}}_k + \boldsymbol{B}\widetilde{\boldsymbol{z}}_k - \boldsymbol{y}\|_2 \leq \frac{1}{2\left(\rho - \|\boldsymbol{\lambda}_\star\|_2\right) k} \|\boldsymbol{w}_\star - \boldsymbol{w}_0\|_{\boldsymbol{Q}}^2$$

$$F(\widetilde{\boldsymbol{x}}_k, \widetilde{\boldsymbol{z}}_k) - F(\boldsymbol{x}_\star, \boldsymbol{z}_\star) \leq \frac{1}{2k} \|\boldsymbol{w}_\star - \boldsymbol{w}_0\|_{\boldsymbol{Q}}^2$$

where $\|\boldsymbol{v}\|_{\boldsymbol{Q}}^2 = \boldsymbol{v}^\top \boldsymbol{Q} \boldsymbol{v}$ for $\forall\ \boldsymbol{v}$, and $\boldsymbol{w}_\star = \begin{bmatrix} \boldsymbol{x}_\star \\ \boldsymbol{z}_\star \\ \frac{\rho(\boldsymbol{A}\widetilde{\boldsymbol{x}}_k + \boldsymbol{B}\widetilde{\boldsymbol{z}}_k - \boldsymbol{y})}{\|\boldsymbol{A}\widetilde{\boldsymbol{x}}_k + \boldsymbol{B}\widetilde{\boldsymbol{z}}_k - \boldsymbol{y}\|_2} \end{bmatrix}$.

Convergence of ADMM

- The convergence rate of ADMM is with $O(1/k)$ iteration complexity $O(1/\varepsilon)$.

- ADMM is **slow** to converge to high accuracy.

- ADMM often converges to modest accuracy within a few tens of iterations, which is sufficient for many large-scale applications.

Accelerated ADMM

$$\min_{\boldsymbol{x},\boldsymbol{z}} \; g(\boldsymbol{x}) + h(\boldsymbol{z}), \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} \;=\; \boldsymbol{y}.$$

- fix $\boldsymbol{z}$ and $\boldsymbol{\lambda}$, minimize $\boldsymbol{x}$:

$$\boldsymbol{x}_{k+1} \;=\; \arg\min_{\boldsymbol{x}} \left\{ g(\boldsymbol{x}) + \frac{\mu}{2} \left\| \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\widehat{\boldsymbol{z}}_k - \boldsymbol{y} + \frac{1}{\mu}\widehat{\boldsymbol{\lambda}}_k \right\|_2^2 \right\},$$

- fix $\boldsymbol{x}$ and $\boldsymbol{\lambda}$, minimize $\boldsymbol{z}$:

$$\boldsymbol{z}_{k+1} \;=\; \arg\min_{\boldsymbol{z}} \left\{ h(\boldsymbol{z}) + \frac{\mu}{2} \left\| \boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{y} + \frac{1}{\mu}\widehat{\boldsymbol{\lambda}}_k \right\|_2^2 \right\},$$

Accelerated ADMM

$$\min_{\boldsymbol{x},\boldsymbol{z}} \; g(\boldsymbol{x}) + h(\boldsymbol{z}), \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} = \boldsymbol{y}.$$

- **Original ADMM:** fix $\boldsymbol{x}$ and $\boldsymbol{z}$, take a dual ascent step on $\boldsymbol{\lambda}$,

$$\boldsymbol{\lambda}_{k+1} = \widehat{\boldsymbol{\lambda}}_k + \mu_k \left(\boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{z}_{k+1} - \boldsymbol{y}\right).$$

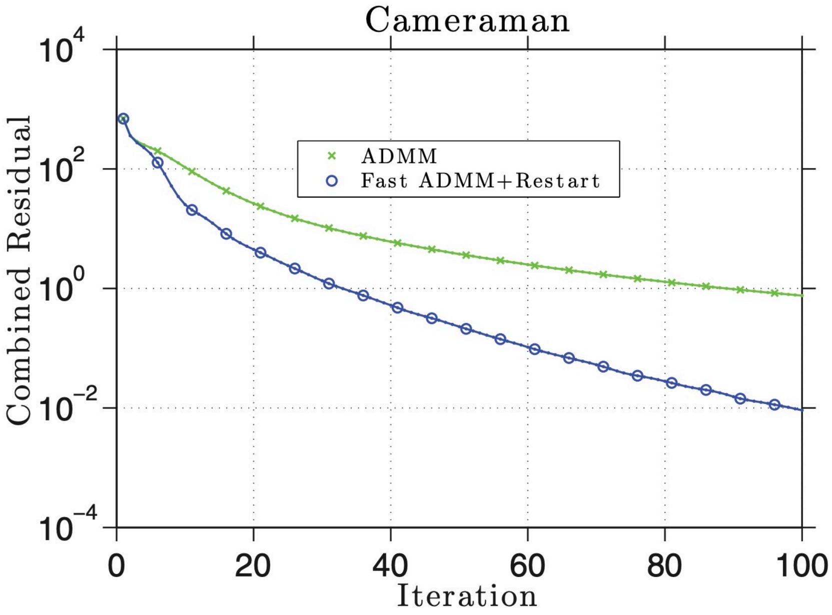
- **Accelerated ADMM: further** update the auxiliary variables,

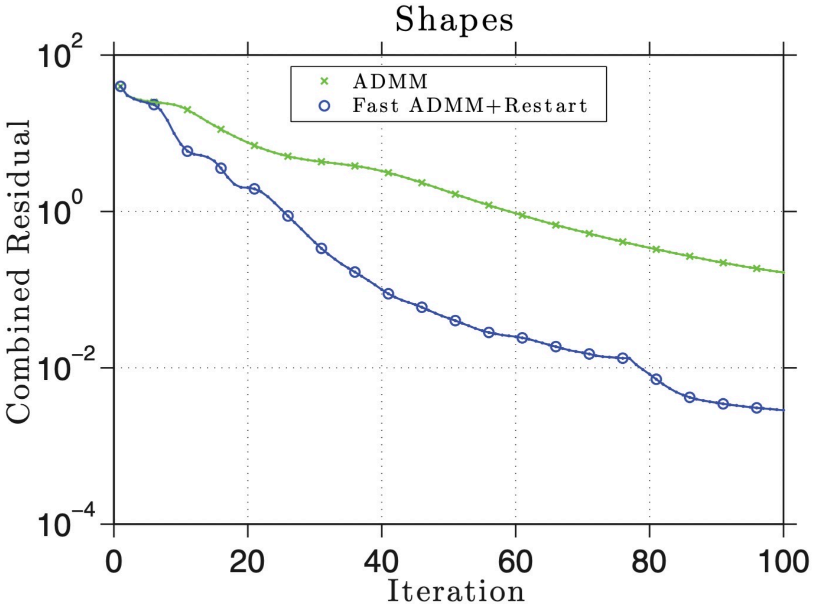
$$\alpha_{k+1} = \frac{1 + \sqrt{1 + 4\alpha_k^2}}{2}$$

$$\widehat{\boldsymbol{z}}_{k+1} = \boldsymbol{z}_k + \frac{\alpha_k - 1}{\alpha_{k+1}} \left(\boldsymbol{z}_k - \boldsymbol{z}_{k-1}\right)$$

$$\widehat{\boldsymbol{\lambda}}_{k+1} = \boldsymbol{\lambda}_{k+1} + \frac{\alpha_k - 1}{\alpha_{k+1}} \left(\boldsymbol{\lambda}_{k+1} - \boldsymbol{\lambda}_k\right)$$

Accelerated ADMM





Fast Alternating Direction Optimization Methods. Tom Goldstein, Brendan O'Donoghue, Simon Setzer, and Richard Baraniuk, SIAM Journal on Imaging Sciences, 2014.

Beyond Two-block Models

- Convergence is **not guaranteed** when there are **three** or **more blocks**, e.g.,

$$\min_{\boldsymbol{x} \in \mathbb{R}^3} 0.05(x_1^2 + x_2^2 + x_3^2), \quad \text{s.t.} \quad \boldsymbol{a}_1 x_1 + \boldsymbol{a}_2 x_2 + \boldsymbol{a}_3 x_3 = \mathbf{0},$$

where

$$\begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \boldsymbol{a}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 2 \end{bmatrix}.$$

Beyond Two-block Models

- Convergence is **not guaranteed** when there are **three** or **more blocks**, e.g.,

$$\min_{\boldsymbol{x} \in \mathbb{R}^3} 0.05(x_1^2 + x_2^2 + x_3^2), \quad \text{s.t.} \quad \boldsymbol{a}_1 x_1 + \boldsymbol{a}_2 x_2 + \boldsymbol{a}_3 x_3 = \boldsymbol{0},$$

where

$$\begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \boldsymbol{a}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 2 \end{bmatrix}.$$

For this example, we can show that 3-block ADMM is divergent (Chen et al.'16).

The Direct Extension of ADMM for Multi-Block Convex Minimization Problems is Not Necessarily Convergent," C. Chen, B. He, Y. Ye, X. Yuan, Mathematical Programming, 2016.

Further Readings

- *High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications.* John Wright, Yi Ma. (**Chapter 8.5**)

- *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers.* S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Foundations and Trends in Machine learning, 2011.

- *The Direct Extension of ADMM for Multi-Block Convex Minimization Problems is Not Necessarily Convergent,"* C. Chen, B. He, Y. Ye, X. Yuan, Mathematical Programming, 2016.

Further Readings

- *Augmented Lagrangian and Alternating Direction Methods for Convex Optimization: A Tutorial and Some Illustrative Computational Results.* Jonathan Eckstein, RUTCOR Research Report RRR 32-2012, 2012.

- *On the Global and Linear Convergence of the Generalized Alternating Direction Method of Multipliers.* Wei Deng and Wotao Yin, Journal of Scientific Computing, 2016.

- *Fast Alternating Direction Optimization Methods.* Tom Goldstein, Brendan O'Donoghue, Simon Setzer, and Richard Baraniuk, SIAM Journal on Imaging Sciences, 2014.