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

- This work is released under a Creative Commons License with the following terms:
- **Attribution**
  - The licensor permits others to copy, distribute, display, and perform the work. In return, licensees must give the original authors credit.
- **Non-Commercial**
  - The licensor permits others to copy, distribute, display, and perform the work. In return, licensees may not use the work for commercial purposes – unless they get the licensor's permission.
- **Share Alike**
  - The licensor permits others to distribute derivative works only under a license identical to the one that governs the licensor's work.
- Full Text of the License

Outline

- Today
  1. Convex constrained optimization
     - Problem setting, common structures and basis assumptions
     - Solutions and approximate solutions
     - Motivating examples
  2. Optimality and duality
     - Optimality condition
     - Lagrange dualization
     - Min-max formulation
     - Equivalent interpretations of optimality condition.
     - Dual decomposition ability
  3. Classical solution methods
     - Convex problem with equality constraints and null space method.
     - Projected gradient method
     - Frank-Wolfe method
     - Quadratic penalty methods
     - Augmented Lagrangian methods
     - Alternating minimization algorithm (AMA)
     - Alternating direction method of multipliers (ADMM)
  4. Next week

  1. Nonsmooth constrained optimization

Reading material

1. S. Boyd and L. Vandenberghe, "*Convex Optimization*", University Press, Cambridge, 2004.
   - Chapter 4 – Convex optimization problems
   - Chapter 5 – Duality
   - Section 10.1-Chapter 10 – Equality constrained minimization.
2. J. Nocedal and S. Wright, "*Numerical Optimization*", Springer-Verlag, 1999.
   - Chapter 17 – Penalty, Barrier and augmented Lagrangian methods, Section 17.4.
3. S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "*Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*", Foundations and Trends in Machine Learning, 3(1):1–122, 2011.

Motivation

Motivation

- Unknown **parameters** in a model are constrained in practice.
- **Constrained convex optimization formulations** naturally encode these constraints.
- Hence, this lecture develops **numerical methods** for constrained convex optimization.

Mathematical form of constrained convex optimization

General setting of constrained convex optimization problems

$$f^\star := \begin{cases} \min_{\mathbf{x}\in\mathbb{R}^p} & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \in \mathcal{X}. \end{cases} \quad (1)$$

- $f \in \mathcal{F}(\mathbb{R}^p)$ is a convex function
- $\mathbf{A} \in \mathbb{R}^{n\times p}$, $\mathbf{b} \in \mathbb{R}^n$
- $\mathcal{X}$ is a nonempty, closed convex set.

Problem sources

- Many real-world **applications** (e.g., linear inverse problems, matrix completion) can be directly formulated as (1).
- Often times, computational considerations lead to (1) by reformulations of **existing unconstrained problems** (e.g., composite convex minimization, consensus optimization, and convex splitting).
- Many standard convex optimization formulations naturally fall under (1), such as *linear programming, convex quadratic programming, second order cone programming, semidefinite programming and geometric programming*.

Structures of constrained convex optimization

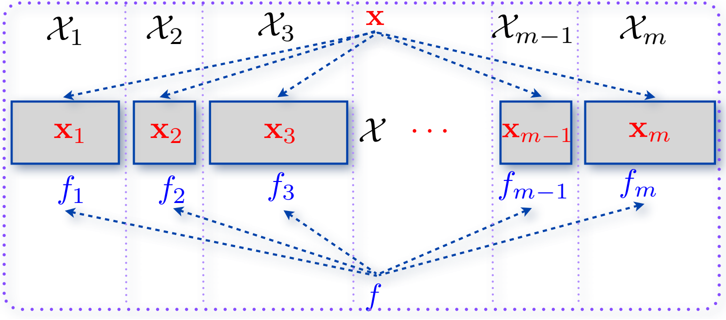
Common structures

When designing a **numerical solution method** for solving **problem** (1), we must rely on individual structures of f and $\mathcal{X}$.

In this lecture, we mainly rely on the following two structures:

- **Decomposability** of f and $\mathcal{X}$.
- **Tractable proximity**

Decomposability illustration



Decomposability and tractable proximity

Decomposable structure

The function f and the feasible set $\mathcal{X}$ have the following structure

$$f(\mathbf{x}) := \sum_{i=1}^{m} f_i(\mathbf{x}_i), \quad \text{and} \quad \mathcal{X} := \mathcal{X}_1 \times \cdots \times \mathcal{X}_m.$$

where $m \geq 1$ is the number of components, $\mathbf{x}_i$ is a sub-vector (component) of $\mathbf{x}$, $f_i : \mathbb{R}^{p_i} \to \mathbb{R} \cup \{+\infty\}$ is convex and $\sum_{i=1}^{m} p_i = p$.

Decomposability and tractable proximity

Decomposable structure

The function f and the feasible set $\mathcal{X}$ have the following structure

$$f(\mathbf{x}) := \sum_{i=1}^{m} f_i(\mathbf{x}_i), \quad \text{and} \quad \mathcal{X} := \mathcal{X}_1 \times \cdots \times \mathcal{X}_m.$$

where $m \geq 1$ is the number of components, $\mathbf{x}_i$ is a sub-vector (component) of $\mathbf{x}$, $f_i : \mathbb{R}^{p_i} \to \mathbb{R} \cup \{+\infty\}$ is convex and $\sum_{i=1}^{m} p_i = p$.

Tractable proximity

- Each **component** f_i has a **'tractable proximal operator"** $(i = 1, \ldots, m)$.
- The component **feasible set** $\mathcal{X}_i$ has **simple projection** ("tractable proximity" of the indicator function of $\mathcal{X}_i$).

Solutions and solution set

Definition (Feasible set)

The set

$$\mathcal{D} := \{\mathbf{x} \in \mathbb{R}^p \ : \ \mathbf{x} \in \mathcal{X}, \ \mathbf{A}\mathbf{x} = \mathbf{b}\} \tag{2}$$

is called the **feasible set** of (1). Any point $\mathbf{x} \in \mathcal{D}$ is called a **feasible point**.

Note: It is important to exclude the following trivial and pathalogical cases:

- $\mathcal{D} = \emptyset$, which leads to no solution of (1).
- $\mathcal{D} = \{\hat{\mathbf{x}}\}$, which leads to the unique solution $\mathbf{x}^\star = \hat{\mathbf{x}}$ of (1).

Solutions and solution set

Definition (Feasible set)

The set

$$\mathcal{D} := \{\mathbf{x} \in \mathbb{R}^p \ : \ \mathbf{x} \in \mathcal{X}, \ \mathbf{A}\mathbf{x} = \mathbf{b}\} \tag{2}$$

is called the **feasible set** of (1). Any point $\mathbf{x} \in \mathcal{D}$ is called a **feasible point**.

Note: It is important to exclude the following trivial and pathalogical cases:

- $\mathcal{D} = \emptyset$, which leads to no solution of (1).
- $\mathcal{D} = \{\hat{\mathbf{x}}\}$, which leads to the unique solution $\mathbf{x}^\star = \hat{\mathbf{x}}$ of (1).

Definition (Solution)

A feasible point $\mathbf{x}^\star \in \mathcal{D}$ is called a globally optimal solution (or solution) of (1) if

$$f(\mathbf{x}^\star) \leq f(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{D}.$$

All solutions of (1) forms the solution set $\mathcal{S}^\star$ of (1).

Note:

- The solution set $\mathcal{S}^\star$ is closed and convex.
- If $\mathbf{x}$ is not feasible, one may have $f(\mathbf{x}) \leq f^\star$ in the constrained setting case.

Approximate solution

Solution certification

- Computing an **exact solution** $\mathbf{x}^\star \in \mathcal{S}^\star$ is **impracticable** unless problem has a closed form solution (which is very limited in reality).
- We can only compute a point $\mathbf{x}^\star_\epsilon$ that approximates $\mathbf{x}^\star$ up to a given accuracy ϵ in a **given sense** by using **numerical optimization algorithms**.

There are **several ways** of certifying an approximate solution. We use the following definition.

Approximate solution

Solution certification

- Computing an **exact solution** $\mathbf{x}^\star \in \mathcal{S}^\star$ is **impracticable** unless problem has a closed form solution (which is very limited in reality).
- We can only compute a point $\mathbf{x}^\star_\epsilon$ that approximates $\mathbf{x}^\star$ up to a given accuracy ϵ in a **given sense** by using **numerical optimization algorithms**.

There are **several ways** of certifying an approximate solution. We use the following definition.

Definition (Approximate solution)

Given a tolerance $\epsilon \geq 0$, a point $\mathbf{x}^\star_\epsilon \in \mathbb{R}^p$ is called an ϵ-solution of (1) if

$$\begin{cases} |f(\mathbf{x}^\star_\epsilon) - f^\star| \leq \epsilon & \text{(objective residual)}, \\ \|\mathbf{A}\mathbf{x}^\star_\epsilon - \mathbf{b}\| \leq \epsilon & \text{(feasibility gap)}, \\ \mathbf{x}^\star_\epsilon \in \mathcal{X} & \text{(exact feasibility)}. \end{cases}$$

Very often, $\mathcal{X}$ is a "simple set." Hence, checking $\mathbf{x}^\star_\epsilon \in \mathcal{X}$ is acceptable in practice.

Motivating example: Composite convex minimization

Composite convex minimization

With a slight change in notation, let us recall the **composite convex minimization** problem in Lecture 5:

$$F^\star := \min_{\mathbf{u}\in\mathbb{R}^p} \left\{ F(\mathbf{u}) := h(\mathbf{u}) + g(\mathbf{u}) \right\}, \tag{3}$$

where both g and h are closed and convex.

Motivating example: Composite convex minimization

Composite convex minimization

With a slight change in notation, let us recall the **composite convex minimization** problem in Lecture 5:

$$F^\star := \min_{\mathbf{u}\in\mathbb{R}^p} \left\{ F(\mathbf{u}) := h(\mathbf{u}) + g(\mathbf{u}) \right\}, \tag{3}$$

where both g and h are closed and convex.

Optimization reformulation

By duplicating the variable $\mathbf{v} = \mathbf{u}$, we can reformulate (3) as

$$\boxed{\begin{array}{ll} \min\limits_{\mathbf{x}:=[\mathbf{u},\mathbf{v}]\in\mathbb{R}^{2p}} & \{f(\mathbf{x}) := h(\mathbf{v}) + g(\mathbf{u})\} \\ \text{s.t.} & \mathbf{u} - \mathbf{v} = 0. \end{array}} \tag{4}$$

This problem falls into the form (1) with separable objective function f and $\mathcal{X} = \mathbb{R}^{2p}$.
The methods studied in this lecture can also be used to solve the **composite convex problem** (3).

Image denoising/debluring

Problem (Imaging denoising/deblurring)

Given an observed image $\mathbf{b} \in \mathbb{R}^{n \times p}$*, the aim is to recover the clean image* $\mathbf{u}$ *via* $\mathbf{b} = \mathcal{A}(\mathbf{u}) + \mathbf{w}$*, where* $\mathcal{A}$ *is a linear operator and* $\mathbf{w}$ *is a Gaussian noise.*

Optimization formulation

$$\boxed{\min_{\mathbf{u} \in \mathbb{R}^{n \times p}} \left\{ (1/2)\|\mathcal{A}(\mathbf{u}) - \mathbf{b}\|_F^2 + \rho\|\mathbf{D}\mathbf{u}\|_1 \right\}} \tag{5}$$

where $\rho > 0$ is a regularization parameter and $\mathbf{D}$ is given matrix.
By reformulating (5) as

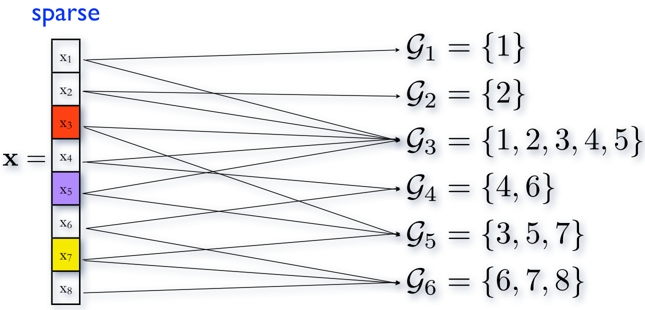
$$\boxed{\begin{array}{ll} \displaystyle\min_{\mathbf{u} \in \mathbb{R}^{n \times p}} & \left\{ (1/2)\|\mathcal{A}(\mathbf{u}) - \mathbf{b}\|_F^2 + \rho\|\mathbf{v}\|_1 \right\} \\ \text{s.t.} & \mathbf{D}\mathbf{u} - \mathbf{v} = 0. \end{array}} \tag{6}$$

This problem is of the form (1) with $\mathbf{x} := (\mathbf{u}^T, \mathbf{v}^T)^T$, $\mathcal{X} = \mathbb{R}^{np + n_D p}$ and $f(\mathbf{x}) := (1/2)\|\mathcal{A}(\mathbf{u}) - \mathbf{b}\|_F^2 + \rho\|\mathbf{v}\|_1$.

Group sparse recovery

Sparse recovery

- Let $\mathcal{I} := \{1, \ldots, p\}$ be the set of indices. Let $\mathfrak{G} := \{\mathcal{G}_1, \ldots, \mathcal{G}_m\}$ be the set of m groups $\mathcal{G}_i \subseteq \mathcal{I}$ and $\mathcal{I} \subseteq \cup_{i=1}^m \mathcal{U}_i$.
- For given group $\mathcal{G}_i$, and a vector $\mathbf{x} \in \mathbb{R}^p$, we use $\mathbf{x}_{\mathcal{G}_i} = \{x_j \ : \ j \in \mathcal{G}_i\}$.
- For fixed group structure $\mathfrak{G}$, $\mathbf{x} \in \mathbb{R}^p$ is called group sparse vector if the number of groups in $\mathcal{G}$ is small.
- Given a **linear operator** $\mathbf{A}$ and an **observed/measurement** vector $\mathbf{b} \in \mathbb{R}^n$. We want to recover the **group sparse** input vector $\mathbf{x} \in \mathbb{R}^p$ such that $\mathbf{b} = \mathbf{A}\mathbf{x}$.



Group sparse recovery

Sparse recovery

- Let $\mathcal{I} := \{1, \ldots, p\}$ be the set of indices. Let $\mathfrak{G} := \{\mathcal{G}_1, \ldots, \mathcal{G}_m\}$ be the set of m groups $\mathcal{G}_i \subseteq \mathcal{I}$ and $\mathcal{I} \subseteq \cup_{i=1}^m \mathcal{U}_i$.
- For given group $\mathcal{G}_i$, and a vector $\mathbf{x} \in \mathbb{R}^p$, we use $\mathbf{x}_{\mathcal{G}_i} = \{x_j \ : \ j \in \mathcal{G}_i\}$.
- For fixed group structure $\mathfrak{G}$, $\mathbf{x} \in \mathbb{R}^p$ is called group sparse vector if the number of groups in $\mathcal{G}$ is **small**.
- Given a **linear operator** $\mathbf{A}$ and an **observed/measurement** vector $\mathbf{b} \in \mathbb{R}^n$. We want to recover the **group sparse** input vector $\mathbf{x} \in \mathbb{R}^p$ such that $\mathbf{b} = \mathbf{A}\mathbf{x}$.

Optimization formulation

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^p} \quad & \textstyle\sum_{\mathcal{G}_i\in\mathfrak{G}} \|\mathbf{x}_{\mathcal{G}_i}\|_2 \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b}. \end{aligned} \tag{7}$$

Here, $f(\mathbf{x}) := \sum_{\mathcal{G}_i\in\mathfrak{G}} \|\mathbf{x}_{\mathcal{G}_i}\|_2$ and $\mathcal{X} := \mathbb{R}^p$. This problem possesses two common structures: decomposability and tractable proximity.
When $m = p$ and $\mathcal{G}_i = \{i\}$, (7) reduces to the well-known linear sparse recovery problem (basis pursuit):

$$\min_{\mathbf{x}\in\mathbb{R}^p} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{b}. \tag{8}$$

Robust principle component analysis

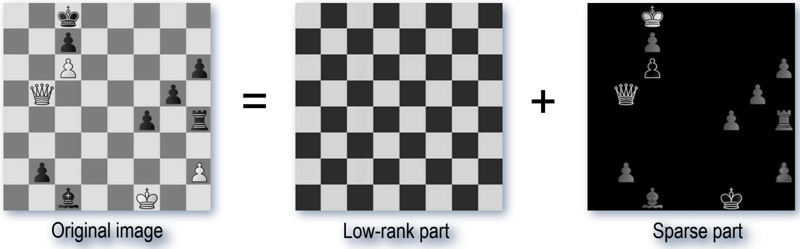
Robust principle component analysis (RPCA)

Assume that we are given a large-scale input matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$, which can be decomposed as $\mathbf{M} = \mathbf{L}_0 + \mathbf{S}_0$, where $\mathbf{L}_0$ has low-rank and $\mathbf{S}_0$ is sparse. We do not know $\mathbf{L}_0$ and $\mathbf{S}_0$ and want to recover them given that they are low-rank and sparse, respectively.

Robust principle component analysis

Robust principle component analysis (RPCA)

Assume that we are given a large-scale input matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$, which can be decomposed as $\mathbf{M} = \mathbf{L}_0 + \mathbf{S}_0$, where $\mathbf{L}_0$ has low-rank and $\mathbf{S}_0$ is sparse. We do not know $\mathbf{L}_0$ and $\mathbf{S}_0$ and want to recover them given that they are low-rank and sparse, respectively.



Motivating example: Robust principle component analysis

Robust principle component analysis (RPCA)

Assume that we are given a large-scale input matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$, which can be decomposed as $\mathbf{M} = \mathbf{L}_0 + \mathbf{S}_0$, where $\mathbf{L}_0$ has low-rank and $\mathbf{S}_0$ is sparse. We do not know $\mathbf{L}_0$ and $\mathbf{S}_0$ and want to recover them given that they are low-rank and sparse, respectively.

Motivating example: Robust principle component analysis

Robust principle component analysis (RPCA)

Assume that we are given a large-scale input matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$, which can be decomposed as $\mathbf{M} = \mathbf{L}_0 + \mathbf{S}_0$, where $\mathbf{L}_0$ has low-rank and $\mathbf{S}_0$ is sparse. We do not know $\mathbf{L}_0$ and $\mathbf{S}_0$ and want to recover them given that they are low-rank and sparse, respectively.

Optimization formulation

$$\begin{aligned} \min_{\mathbf{L},\mathbf{S} \in \mathbb{R}^{m \times n}} \quad & \|\mathrm{vec}(\mathbf{S})\|_1 + \rho \|\mathbf{L}\|_*, \\ \text{s.t.} \quad & \mathbf{S} + \mathbf{L} = \mathbf{M}. \end{aligned} \tag{9}$$

Here $\rho > 0$ is a weighted parameter to trade-off between the sparse and low-rank terms, vex is the vectorization operator and $\|\cdot\|_*$ is the nuclear norm.

By letting

- $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2] := [\mathrm{vec}(\mathbf{S}), \mathrm{vec}(\mathbf{L})]$
- $f(\mathbf{x}) = f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) := \|\mathrm{vec}(\mathbf{S})\|_1 + \rho\|\mathbf{L}\|_*$
- $\mathbf{A} = [\mathbb{I}, \mathbb{I}]$, $\mathbf{b} := \mathrm{vec}(\mathbf{M})$ and
- $\mathcal{X} := \mathbb{R}^{mn}$.

Then, (9) can be transformed into (1).

Motivating example: Robust principle component analysis (cont)

Example - RPCA for object separation from video

Let $\mathbf{M}$ be the matrix extracted from a video clip. Our aim is to separate objects (e.g., humans) and backgrounds by solving (9).

Motivating example: Robust principle component analysis (cont)

Example - RPCA for object separation from video

Let $\mathbf{M}$ be the matrix extracted from a video clip. Our aim is to separate objects (e.g., humans) and backgrounds by solving (9).

Result: One frame from the solution of (9)

One original image $\mathbf{M}$ | The low-rank part $\mathbf{L}$ | The sparse part $\mathbf{S}$

Matrix completion

Matrix completion

Aim: Recover the unknown entries of a matrix $\mathbf{M} \in \mathbf{C}^{m \times n}$, when we only observe a **few** $q < m \times n$ entries at a given locations $(i, j) \in \Omega$.

Low-rankness: Since this is an underdetermined problem, there exist many matrix $\mathbf{X}$ such that $\mathbf{X}_{ij} = \mathbf{M}_{ij}$ for all $(i, j) \in \Omega$. We would like to recover a low-rank matrix $\mathbf{X}$ such that $\mathbf{X}_{ij} = \mathbf{M}_{ij}$ for all $(i, j) \in \Omega$.

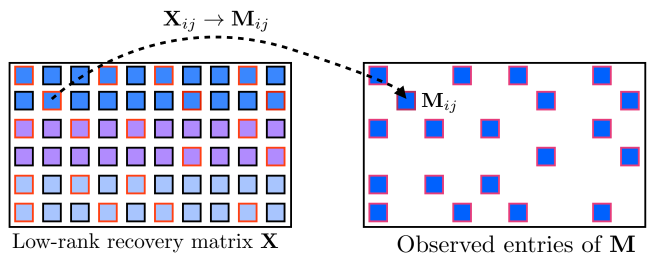
Matrix completion

Matrix completion

Aim: Recover the unknown entries of a matrix $\mathbf{M} \in \mathbf{C}^{m \times n}$, when we only observe a **few** $q < m \times n$ entries at a given locations $(i, j) \in \Omega$.

Low-rankness: Since this is an underdetermined problem, there exist many matrix $\mathbf{X}$ such that $\mathbf{X}_{ij} = \mathbf{M}_{ij}$ for all $(i, j) \in \Omega$. We would like to recover a low-rank matrix $\mathbf{X}$ such that $\mathbf{X}_{ij} = \mathbf{M}_{ij}$ for all $(i, j) \in \Omega$.

Illustration



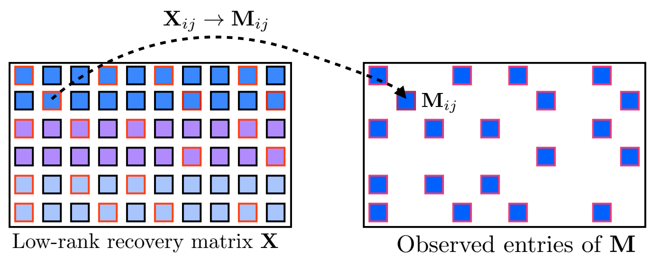
Matrix completion

Matrix completion

Aim: Recover the unknown entries of a matrix $\mathbf{M} \in \mathbb{C}^{m\times n}$, when we only observe a **few** $q < m \times n$ entries at a given locations $(i,j) \in \Omega$.

Low-rankness: Since this is an underdetermined problem, there exist many matrix $\mathbf{X}$ such that $\mathbf{X}_{ij} = \mathbf{M}_{ij}$ for all $(i,j) \in \Omega$. We would like to recover a low-rank matrix $\mathbf{X}$ such that $\mathbf{X}_{ij} = \mathbf{M}_{ij}$ for all $(i,j) \in \Omega$.

Illustration



Convex relaxation of matrix completion

$$\begin{aligned} \min_{\mathbf{X}\in\mathbb{C}^{m\times n}} \quad & \|\mathbf{X}\|_* \\ \text{s.t.} \quad & \mathbf{X}_{ij} = \mathbf{M}_{ij},\ \forall (i,j) \in \Omega. \end{aligned} \tag{10}$$

Outline

- Today
  1. Convex constrained optimization
     - Problem setting, common structures and basis assumptions
     - Solutions and approximate solutions
     - Motivating examples
  2. Optimality and duality
     - Optimality condition
     - Lagrange dualization
     - Min-max formulation
     - Equivalent interpretations of optimality condition.
     - Dual decomposition ability
  3. Classical solution methods
     - Convex problem with equality constraints and null space method.
     - Projected gradient method
     - Frank-Wolfe method
     - Quadratic penalty methods
     - Augmented Lagrangian methods
     - Alternating minimization algorithm (AMA)
     - Alternating direction method of multipliers (ADMM)
  4. Next week

  1. Nonsmooth constrained optimization

Optimality condition

Lagrange function

$$\mathcal{L}(\mathbf{x}, \lambda) := f(\mathbf{x}) + \lambda^T(\mathbf{A}\mathbf{x} - \mathbf{b}).$$

Here, $\lambda \in \mathbb{R}^n$ is the vector of Lagrange multipliers (or dual variables) w.r.t. $\mathbf{A}\mathbf{x} = \mathbf{b}$.

Optimality condition

Lagrange function

$$\mathcal{L}(\mathbf{x}, \lambda) := f(\mathbf{x}) + \lambda^T(\mathbf{A}\mathbf{x} - \mathbf{b}).$$

Here, $\lambda \in \mathbb{R}^n$ is the vector of Lagrange multipliers (or dual variables) w.r.t. $\mathbf{A}\mathbf{x} = \mathbf{b}$.

Optimality condition

The **optimality condition** of (1) can be written as

$$\begin{cases} 0 & \in \mathbf{A}^T\lambda^\star + \partial f(\mathbf{x}^\star) + \mathcal{N}_{\mathcal{X}}(\mathbf{x}^\star), \\ 0 & = \mathbf{A}\mathbf{x}^\star - \mathbf{b}. \end{cases} \tag{11}$$

Here:

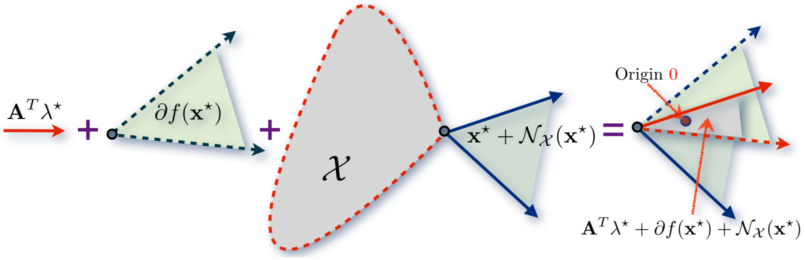
- $\partial f(\mathbf{x}) := \{\mathbf{z} \in \mathbb{R}^p \ : \ f(\mathbf{y}) \geq f(\mathbf{x}) + \mathbf{z}^T(\mathbf{y} - \mathbf{x}), \ \forall \mathbf{y} \in \mathbb{R}^p\}$ is the subdifferential of f at $\mathbf{x}$ (see Lecture 2).
- $\mathcal{N}_{\mathcal{X}}$ is the normal cone of $\mathcal{X}$ at $\mathbf{x}$ defined as

$$\mathcal{N}_{\mathcal{X}}(\mathbf{x}) := \begin{cases} \{\mathbf{z} \in \mathbb{R}^p \ : \ \mathbf{z}^T(\mathbf{x} - \mathbf{y}) \geq 0, \ \forall \mathbf{y} \in \mathcal{X}\} & \text{if } \mathbf{x} \in \mathcal{X}, \\ \emptyset, & \text{if } \mathbf{x} \notin \mathcal{X}. \end{cases}$$

The condition (11) can be considered as the KKT (Karush-Kuhn-Tuchker) condition. Any point $(\mathbf{x}^\star, \lambda^\star)$ satisfying (11) is called a KKT point. $\mathbf{x}^\star$ is called a stationary point and $\lambda^\star$ is the corresponding multipliers.

Example: Illustration

- This figure illustrates the first condition $0 \in \mathbf{A}^T\lambda^\star + \partial f(\mathbf{x}^\star) + \mathcal{N}_{\mathcal{X}}(\mathbf{x}^\star)$.



Example: Basis pursuit

Example (Basis pursuit)

$$\min_{\mathbf{x}\in\mathbb{R}^p} \|\mathbf{x}\|_1 \ \text{ s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}.$$

Note:

- $f(\mathbf{x}) := \|\mathbf{x}\|_1$ is nonsmooth, for any $\mathbf{v} \in \partial f(\mathbf{x})$ we have $v_i = +1$ if $x_i > 0$, $v_i = -1$ if $x_i < 0$ and $v_i \in (-1, 1)$ if $x_i = 0$.
- Since $\mathcal{X} \equiv \mathbb{R}^p$, we have $\mathcal{N}_{\mathcal{X}}(\mathbf{x}) = \{0\}$ for all $\mathbf{x}$.

Example: Basis pursuit

Example (Basis pursuit)

$$\min_{\mathbf{x}\in\mathbb{R}^p} \|\mathbf{x}\|_1 \;\text{ s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}.$$

Note:

- $f(\mathbf{x}) := \|\mathbf{x}\|_1$ is nonsmooth, for any $\mathbf{v} \in \partial f(\mathbf{x})$ we have $v_i = +1$ if $x_i > 0$, $v_i = -1$ if $x_i < 0$ and $v_i \in (-1, 1)$ if $x_i = 0$.
- Since $\mathcal{X} \equiv \mathbb{R}^p$, we have $\mathcal{N}_{\mathcal{X}}(\mathbf{x}) = \{0\}$ for all $\mathbf{x}$.

Optimality condition

The **optimality condition** of (11) becomes

$$\begin{cases} 0 \in \partial f(\mathbf{x}^\star) + \mathbf{A}^T\lambda^\star \\ 0 = \mathbf{A}\mathbf{x}^\star - \mathbf{b}. \end{cases} \Leftrightarrow \begin{cases} (\mathbf{A}^T\lambda^\star)_i = -1 & \text{if } x_i^\star > 0,\; 1 \le i \le p \\ (\mathbf{A}^T\lambda^\star)_i = +1 & \text{if } x_i^\star < 0,\; 1 \le i \le p \\ (\mathbf{A}^T\lambda^\star)_i \in (-1, 1) & \text{if } x_i^\star = 0,\; 1 \le i \le p \\ \mathbf{A}\mathbf{x}^\star = \mathbf{b}. \end{cases}$$

Min-max formulation and dual problem

Dual function and Dual problem

- **Dual function:**

$$d(\lambda) := \min_{\mathbf{x}\in\mathcal{X}}\{\mathcal{L}(\mathbf{x},\lambda) := f(\mathbf{x}) + \lambda^T(\mathbf{A}\mathbf{x} - \mathbf{b})\}. \quad (12)$$

  Let $\mathbf{x}^\star(\lambda)$ be a solution of (12) then $d(\lambda)$ is finite if $x^\star(\lambda)$ exists. $d(\cdot)$ is concave and possibly nonsmooth.

- **Dual problem**: The following dual problem is convex

$$\boxed{d^\star := \max_{\mathbf{x}\in\mathbb{R}^n} d(\lambda)} \quad (13)$$

Min-max formulation and dual problem

Dual function and Dual problem

- **Dual function:**

$$d(\lambda) := \min_{\mathbf{x}\in\mathcal{X}} \{\mathcal{L}(\mathbf{x},\lambda) := f(\mathbf{x}) + \lambda^T(\mathbf{A}\mathbf{x} - \mathbf{b})\}. \quad (12)$$

  Let $\mathbf{x}^\star(\lambda)$ be a solution of (12) then $d(\lambda)$ is finite if $x^\star(\lambda)$ exists. $d(\cdot)$ is concave and possibly nonsmooth.

- **Dual problem**: The following dual problem is convex

$$\boxed{d^\star := \max_{\mathbf{x}\in\mathbb{R}^n} d(\lambda)} \quad (13)$$

Min-max formulation

$$\begin{aligned} d^\star &= \max_{\lambda\in\mathbb{R}^n} d(\lambda) = \max_{\lambda\in\mathbb{R}^n} \min_{\mathbf{x}\in\mathcal{X}} \{f(\mathbf{x}) + \lambda^T(\mathbf{A}\mathbf{x} - \mathbf{b})\} \\ &\leq \min_{\mathbf{x}\in\mathcal{X}} \max_{\lambda\in\mathbb{R}^n} \{f(\mathbf{x}) + \lambda^T(\mathbf{A}\mathbf{x} - \mathbf{b})\} = \begin{cases} \min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) & \text{if } \mathbf{A}\mathbf{x} = \mathbf{b}, \\ +\infty & \text{otherwise} \end{cases} \end{aligned} \quad (14)$$

Here, the inequality is due to **the max-min theorem** [6].

Example: Strictly convex quadratic programming

Strictly convex quadratic programming

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^p} \quad & (1/2)\mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{h}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b}. \end{aligned}$$

where $\mathbf{H}$ is symmetric positive definite.

Example: Strictly convex quadratic programming

Strictly convex quadratic programming

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^p} \quad & (1/2)\mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{h}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b}. \end{aligned}$$

where $\mathbf{H}$ is symmetric positive definite.

Dual problem is also a strictly convex quadratic program

- Lagrange function $\mathcal{L}(\mathbf{x}, \lambda) := (1/2)\mathbf{x}^T\mathbf{H}\mathbf{x} + (\mathbf{A}^T\lambda + \mathbf{h})^T\mathbf{x} - \mathbf{b}^T\lambda$.
- Dual function:

$$d(\lambda) = \min_{\mathbf{x}\in\mathbb{R}^p}\{(1/2)\mathbf{x}^T\mathbf{H}\mathbf{x} + (\mathbf{A}^T\lambda + \mathbf{h})^T\mathbf{x} - \mathbf{b}^T\lambda\}$$

- Since $\mathbf{x}^\star(\lambda) = -\mathbf{H}^{-1}(\mathbf{A}^T\lambda + \mathbf{h})$, we can obtain $d(\lambda)$ explicitly as

$$d(\lambda) = -(1/2)\lambda^T(\mathbf{A}\mathbf{H}^{-1}\mathbf{A}^T)\lambda - (\mathbf{b} + \mathbf{A}\mathbf{H}^{-1}\mathbf{h})^T\lambda.$$

- Dual problem (unconstrained):

$$d^\star := \max_{\lambda\in\mathbb{R}^n} d(\lambda) \quad \Leftrightarrow \quad \min_{\lambda\in\mathbb{R}^n} \frac{1}{2}\lambda^T(\mathbf{A}\mathbf{H}^{-1}\mathbf{A}^T)\lambda + (\mathbf{b} + \mathbf{A}\mathbf{H}^{-1}\mathbf{h})^T\lambda.$$

Example: Nonsmoothness of the dual function

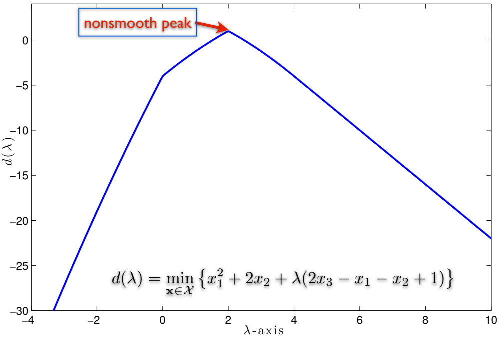
Consider a constrained convex problem:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^3} \quad & \{f(\mathbf{x}) := x_1^2 + 2x_2\}, \\ \text{s.t.} \quad & 2x_3 - x_1 - x_2 = 1, \\ & \mathbf{x} \in \mathcal{X} := [-2,2]\times[-2,2]\times[0,2]. \end{aligned}$$

The **dual function** is defined as

$$d(\lambda) := \min_{\mathbf{x}\in\mathcal{X}}\{x_1^2 + 2x_2 + \lambda(2x_3 - x_1 - x_2 + 1)\}$$

is concave and nonsmooth as illustrated in the figure below.



Saddle point

Definition (Saddle point)

A point $(\mathbf{x}^\star, \lambda^\star) \in \mathcal{X} \times \mathbb{R}^n$ is called a saddle point of the Lagrange function $\mathcal{L}$ if

$$\mathcal{L}(\mathbf{x}^\star, \lambda) \leq \mathcal{L}(\mathbf{x}^\star, \lambda^\star) \leq \mathcal{L}(\mathbf{x}, \lambda^\star), \ \forall \mathbf{x} \in \mathcal{X}, \ \lambda \in \mathbb{R}^n.$$

Recall the minmax form:

$$\max_{\lambda} \min_{\mathbf{x} \in \mathcal{X}} \{\mathcal{L}(\mathbf{x}, \lambda) := f(\mathbf{x}) + \lambda^T(\mathbf{A}\mathbf{x} - \mathbf{b})\}. \qquad ((12))$$

Saddle point

Definition (Saddle point)

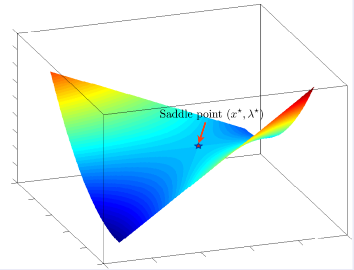
A point $(\mathbf{x}^\star, \lambda^\star) \in \mathcal{X} \times \mathbb{R}^n$ is called a saddle point of the Lagrange function $\mathcal{L}$ if

$$\mathcal{L}(\mathbf{x}^\star, \lambda) \le \mathcal{L}(\mathbf{x}^\star, \lambda^\star) \le \mathcal{L}(\mathbf{x}, \lambda^\star), \ \forall \mathbf{x} \in \mathcal{X}, \ \lambda \in \mathbb{R}^n.$$

Recall the minmax form:

$$\max_{\lambda} \min_{\mathbf{x} \in \mathcal{X}} \{\mathcal{L}(\mathbf{x}, \lambda) := f(\mathbf{x}) + \lambda^T(\mathbf{A}\mathbf{x} - \mathbf{b})\}. \quad ((12))$$

Illustration of saddle point: $\mathcal{L}(x, \lambda) := (1/2)x^2 + \lambda(x - 1)$ in $\mathbb{R}^2$



Slater's qualification condition

Slater's qualification condition

Recall $\mathrm{relint}(\mathcal{X})$ the relative interior of the **feasible set** $\mathcal{X}$. The Slater condition requires

$$\boxed{\mathrm{relint}(\mathcal{X}) \cap \{\mathbf{x} \; : \; \mathbf{A}\mathbf{x} = \mathbf{b}\} \neq \emptyset.} \qquad (15)$$

Slater's qualification condition

Slater's qualification condition

Recall $\text{relint}(\mathcal{X})$ the relative interior of the feasible set $\mathcal{X}$. The Slater condition requires

$$\boxed{\text{relint}(\mathcal{X}) \cap \{\mathbf{x} \ : \ \mathbf{A}\mathbf{x} = \mathbf{b}\} \neq \emptyset.} \quad (15)$$

Special cases

- If $\mathcal{X}$ is absent, then (15) $\Leftrightarrow$ $\boxed{\exists \bar{\mathbf{x}} \ : \ \mathbf{A}\bar{\mathbf{x}} = \mathbf{b}}$.
- If $\mathbf{A}\mathbf{x} = \mathbf{b}$ is absent, then (15) $\Leftrightarrow$ $\boxed{\text{relint}(\mathcal{X}) \neq \emptyset}$.
- If $\mathbf{A}\mathbf{x} = \mathbf{b}$ is absent and $\mathcal{X} := \{\mathbf{x} : h(\mathbf{x}) \leq 0\}$, where h is $\mathbb{R}^p \to R^q$ is convex, then

$$(15) \Leftrightarrow \boxed{\exists \bar{\mathbf{x}} \ : \ h(\bar{\mathbf{x}}) < 0.}$$

Example: Slater's condition

Example

Let us consider the feasible set $\mathcal{D}_\alpha := \mathcal{X} \cap \mathcal{A}_\alpha$ as

$$\mathcal{X} := \{\mathbf{x} \in \mathbb{R}^2 \ : \ x_1^2 + x_2^2 \leq 1\} \ \mathcal{A}_\alpha := \{\mathbf{x} \in \mathbb{R}^2 \ : \ x_1 + x_2 = \alpha\},$$

where $\alpha \in \mathbb{R}$.

Example: Slater's condition

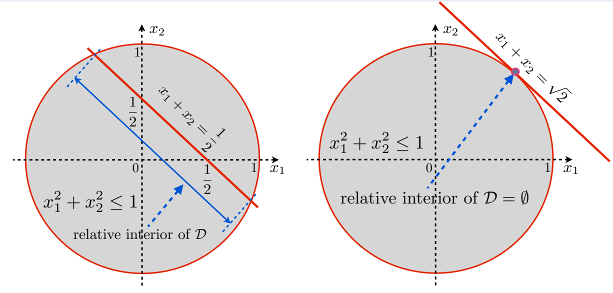
Example

Let us consider the feasible set $\mathcal{D}_\alpha := \mathcal{X} \cap \mathcal{A}_\alpha$ as

$$\mathcal{X} := \{\mathbf{x} \in \mathbb{R}^2 \ : \ x_1^2 + x_2^2 \leq 1\} \ \mathcal{A}_\alpha := \{\mathbf{x} \in \mathbb{R}^2 \ : \ x_1 + x_2 = \alpha\},$$

where $\alpha \in \mathbb{R}$.

Slater's condition holds and does not hold



$\mathcal{D}_{1/2}$ satisfies Slater's condition – $\mathcal{D}_{\sqrt{2}}$-does not satisfy Slater's condition

Necessary and sufficient condition

Theorem (Necessary and sufficient optimality condition)

Under Slater's condition (15)*:* $\mathrm{relint}(\mathcal{X}) \cap \{\mathbf{x} \ : \ \mathbf{A}\mathbf{x} = \mathbf{b}\} \neq \emptyset$*, the* ***KKT condition*** (11)

$$\begin{cases} 0 & \in \mathbf{A}^T\lambda^\star + \partial f(\mathbf{x}^\star) + \mathcal{N}_{\mathcal{X}}(\mathbf{x}^\star), \\ 0 & = \mathbf{A}\mathbf{x}^\star - \mathbf{b}. \end{cases}$$

is necessary and sufficient for a point $(\mathbf{x}^\star, \lambda^\star) \in \mathcal{X} \times \mathbb{R}^n$ *being an optimal solution for the primal problem* (1) *and dual problem* (13)*:*

$$f^\star := \begin{cases} \min\limits_{\mathbf{x}\in\mathbb{R}^p} & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b}, \ \mathbf{x} \in \mathcal{X}, \end{cases} \qquad \textit{and} \quad d^\star := \max_{\mathbf{x}\in\mathbb{R}^n} d(\lambda).$$

Necessary and sufficient condition

Theorem (Necessary and sufficient optimality condition)

Under Slater's condition (15)*:* $\mathrm{relint}(\mathcal{X}) \cap \{\mathbf{x} \ : \ \mathbf{A}\mathbf{x} = \mathbf{b}\} \neq \emptyset$*, the* ***KKT condition*** (11)

$$\begin{cases} 0 & \in \mathbf{A}^T\lambda^\star + \partial f(\mathbf{x}^\star) + \mathcal{N}_{\mathcal{X}}(\mathbf{x}^\star), \\ 0 & = \mathbf{A}\mathbf{x}^\star - \mathbf{b}. \end{cases}$$

is necessary and sufficient for a point $(\mathbf{x}^\star, \lambda^\star) \in \mathcal{X} \times \mathbb{R}^n$ *being an optimal solution for the primal problem* (1) *and dual problem* (13)*:*

$$f^\star := \begin{cases} \min\limits_{\mathbf{x}\in\mathbb{R}^p} & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b}, \ \mathbf{x} \in \mathcal{X}, \end{cases} \qquad \textit{and} \quad d^\star := \max_{\mathbf{x}\in\mathbb{R}^n} d(\lambda).$$

Strong duality

- By definition of $f^\star$ and $d^\star$, we always have $\boxed{d^\star \leq f^\star}$ (**weak duality**).
- Under Slater's condition and $\mathcal{X}^\star \neq \emptyset$, we have $\boxed{d^\star = f^\star}$ (**strong duality**).
- Any solution $(\mathbf{x}^\star, \lambda^\star)$ of the KKT condition (11) is also a saddle point.

What happens if Slater's condition does not hold?

Without Slater's condition, **KKT condition** is only sufficient but not necessary, i.e., if $(\mathbf{x}^\star, \lambda^\star)$ satisfies the KKT condition, then $\mathbf{x}^\star$ is a global solution of (1) but not vice versa.

Example (Violating Slater's condition)

Consider the following **constrained convex** problem:

$$\min_{\mathbf{x}\in\mathbb{R}^2} \{x_1 \ : \ x_2 = 0, x_1^2 - x_2 \leq 0\}$$

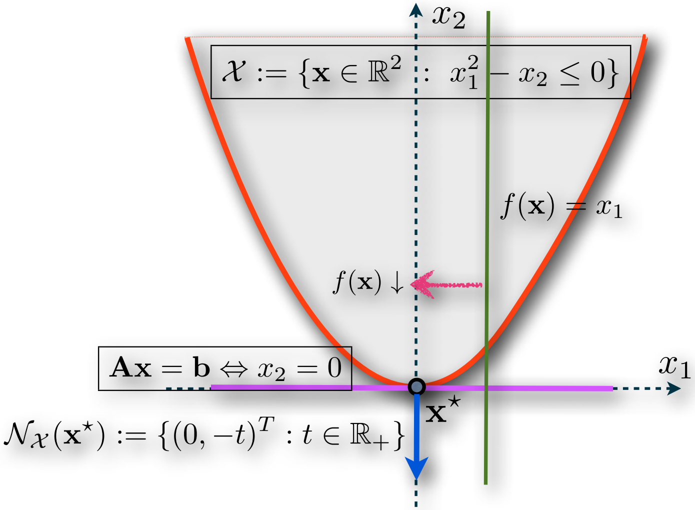
In the setting (1), we have $\mathbf{A} := [0, 1]$, $\mathbf{b} = 0$, $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^2 \ : \ x_1^2 - x_2 \leq 0\}$. The feasible set $\mathcal{D} := \{\mathbf{x} \in \mathbb{R}^2 \ : \ x_2 = 0, x_1^2 - x_2 \leq 0\} = \{(0,0)^T\}$ contains only one point, which is also the optimal solution of the problem, i.e., $\mathbf{x}^\star := (0,0)^T$.

In this case, Slater's condition is definitely violated. Let us check the **KKT condition**. Since $\mathcal{N}_{\mathcal{X}}(\mathbf{x}^\star) = \{(0, -t)^T \ : \ t \geq 0\}$, we can write the KKT condition as

$$\begin{bmatrix}1\\0\end{bmatrix} + \begin{bmatrix}0\\1\end{bmatrix}\lambda + \begin{bmatrix}0\\-t\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix}, \quad \lambda \in \mathbb{R}, \ t \in \mathbb{R}_+.$$

Since this linear system has no solution due to the first equation $1 = 0$, the **KKT condition** is inconsistent.

Violating Slater's condition



Variational inequality (VI) formulation

Primal-dual mapping

For simplicity, we assume that f is smooth. We introduce $\mathbf{z} := (\mathbf{x}^T, \lambda^T)^T \in \mathbb{R}^{p+n}$ and two mappings:

$$M(\mathbf{z}) := \begin{bmatrix} \nabla f(\mathbf{x}) + \mathbf{A}^T\lambda \\ \mathbf{A}\mathbf{x} - \mathbf{b} \end{bmatrix} \text{ and } \mathcal{T}(\mathbf{z}) := \mathcal{N}_{\mathcal{X}}(\mathbf{x}) \times \{0^n\}. \tag{16}$$

Then $M : \mathbb{R}^{p+n} \to \mathbb{R}^{p+n}$ is a single-valued mapping and $\mathcal{T} : \mathbb{R}^{p+n} \rightrightarrows \mathbb{R}^{p+n}$ is a set-valued mapping.

Variational inequality (VI) formulation

Primal-dual mapping

For simplicity, we assume that f is smooth. We introduce $\mathbf{z} := (\mathbf{x}^T, \lambda^T)^T \in \mathbb{R}^{p+n}$ and two mappings:

$$M(\mathbf{z}) := \begin{bmatrix} \nabla f(\mathbf{x}) + \mathbf{A}^T\lambda \\ \mathbf{A}\mathbf{x} - \mathbf{b} \end{bmatrix} \quad \text{and} \quad \mathcal{T}(\mathbf{z}) := \mathcal{N}_{\mathcal{X}}(\mathbf{x}) \times \{0^n\}. \tag{16}$$

Then $M : \mathbb{R}^{p+n} \to \mathbb{R}^{p+n}$ is a single-valued mapping and $\mathcal{T} : \mathbb{R}^{p+n} \rightrightarrows \mathbb{R}^{p+n}$ is a set-valued mapping.

Inclusion and VI formulation

- The **optimality condition** (11) can be written as an **inclusion**:

$$0 \in \mathcal{R}(\mathbf{z}) := M(\mathbf{z}) + \mathcal{T}(\mathbf{z}).$$

- (11) can also be expressed as a **variational inequality**:

$$M(\mathbf{z}^\star)^T(\mathbf{z} - \mathbf{z}^\star) \geq 0, \quad \forall \mathbf{z} \in \mathcal{Z} := \mathcal{X} \times \mathbb{R}^n. \tag{17}$$

Dual decomposition ability

Roles of strong duality

- **Strong duality** is a key property in convex optimization, which creates a connection between primal problem (1) and dual problem (13).
- Under Slater's condition, **strong duality** holds, i.e., $f^\star = d^\star$.
- Principally, by solving dual problem (13), we can recover a solution of primal problem (1) and vice versa.

Dual decomposition ability

Roles of strong duality

- **Strong duality** is a key property in convex optimization, which creates a connection between primal problem (1) and dual problem (13).
- Under Slater's condition, **strong duality** holds, i.e., $f^\star = d^\star$.
- Principally, by solving dual problem (13), we can recover a solution of primal problem (1) and vice versa.

Decomposability is a key property for parallel algorithms

- Under the **decomposable assumption**, the dual function d can be decomposed as

$$d(\lambda) = \sum_{i=1}^{g} d_i(\lambda) - \mathbf{b}^T\lambda.$$

where

$$d_i(\lambda) = \min_{\mathbf{x}_i \in \mathcal{X}_i} \left\{ f_i(\mathbf{x}_i) + \lambda^T \mathbf{A}_i \mathbf{x}_i \right\}, \quad i = 1, \ldots, g.$$

- Evaluating function $d_i(\cdot)$ and its [sub]gradients can be computed in **parallel**

Outline

- Today
  1. Convex constrained optimization
     - Problem setting, common structures and basis assumptions
     - Solutions and approximate solutions
     - Motivating examples
  2. Optimality and duality
     - Optimality condition
     - Lagrange dualization
     - Min-max formulation
     - Equivalent interpretations of optimality condition.
     - Dual decomposition ability
  3. Classical solution methods
     - Convex problem with equality constraints and null space method.
     - Projected gradient method
     - Frank-Wolfe method
     - Quadratic penalty methods
     - Augmented Lagrangian methods
     - Alternating minimization algorithm (AMA)
     - Alternating direction method of multipliers (ADMM)
  4. Next week

  1. Nonsmooth constrained optimization

Null space method for convex programs with equality constraints

Convex problems with equality constraints

We consider the case $\mathcal{X} \equiv \mathbf{R}^p$. Then (1) reduces to

$$f^\star := \begin{cases} \min_{\mathbf{x}\in\mathbb{R}^p} & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b}. \end{cases} \quad (18)$$

Null space method for convex programs with equality constraints

Convex problems with equality constraints

We consider the case $\mathcal{X} \equiv \mathbf{R}^p$. Then (1) reduces to

$$f^\star := \begin{cases} \min_{\mathbf{x}\in\mathbb{R}^p} & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b}. \end{cases} \tag{18}$$

Dimensional reduction

- Assume that $\mathrm{rank}(\mathbf{A}) = m < p$, then the dimension of the null space $\dim(\mathrm{null}(\mathbf{A})) = p - n$.
- By eliminating the equality constraints $\mathbf{A}\mathbf{x} = \mathbf{b}$, we can reduce the problem dimension from p to $p - n$.
- This elimination can be done via projection onto the null space $\mathrm{null}(\mathbf{A})$ of $\mathbf{A}$, (e.g., by QR factorization of $\mathbf{A}$).
- Problem (18) can be transformed into an unconstrained problem with dimension $p - n$.

Null space method

Null space representation of the equality constraint $\mathbf{A}\mathbf{x} = \mathbf{b}$

- Any vector $\mathbf{x} \in \mathbb{R}^p$ can be represented as

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{x}_{\mathcal{N}} = \bar{\mathbf{x}} + \mathbf{U}\mathbf{z},$$

  where $\mathbf{x}_{\mathcal{N}} \in \mathrm{null}(\mathbf{A})$, $\mathbf{U}$ is a basis of $\mathrm{null}(\mathbf{A})$ and $\bar{\mathbf{x}}$ satisfies $\mathbf{A}\bar{\mathbf{x}} = \mathbf{b}$.
- For any feasible point $\bar{\mathbf{x}}$ (i.e., $\mathbf{A}\bar{\mathbf{x}} = \mathbf{b}$), the point $\mathbf{x} := \bar{\mathbf{x}} + \mathbf{U}\mathbf{z}$ is also feasible to $\mathbf{A}\mathbf{x} = \mathbf{b}$, since

$$\mathbf{A}\mathbf{x} = \mathbf{A}\bar{\mathbf{x}} + \mathbf{A}\mathbf{U}\mathbf{z} = \mathbf{A}\bar{\mathbf{x}} = \mathbf{b}, \text{ since } \mathbf{A}\mathbf{U} = 0.$$

- $\mathbf{U}$ can be computed via the QR-factorization of $\mathbf{A}^T$, and $\bar{\mathbf{x}}$ can be obtained by solving a triangular linear system.

Null space method

Null space representation of the equality constraint $\mathbf{A}\mathbf{x} = \mathbf{b}$

- Any vector $\mathbf{x} \in \mathbb{R}^p$ can be represented as

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{x}_{\mathcal{N}} = \bar{\mathbf{x}} + \mathbf{U}\mathbf{z},$$

  where $\mathbf{x}_{\mathcal{N}} \in \mathrm{null}(\mathbf{A})$, $\mathbf{U}$ is a basis of $\mathrm{null}(\mathbf{A})$ and $\bar{\mathbf{x}}$ satisfies $\mathbf{A}\bar{\mathbf{x}} = \mathbf{b}$.

- For any feasible point $\bar{\mathbf{x}}$ (i.e., $\mathbf{A}\bar{\mathbf{x}} = \mathbf{b}$), the point $\mathbf{x} := \bar{\mathbf{x}} + \mathbf{U}\mathbf{z}$ is also feasible to $\mathbf{A}\mathbf{x} = \mathbf{b}$, since

$$\mathbf{A}\mathbf{x} = \mathbf{A}\bar{\mathbf{x}} + \mathbf{A}\mathbf{U}\mathbf{z} = \mathbf{A}\bar{\mathbf{x}} = \mathbf{b}, \text{ since } \mathbf{A}\mathbf{U} = 0.$$

- $\mathbf{U}$ can be computed via the QR-factorization of $\mathbf{A}^T$, and $\bar{\mathbf{x}}$ can be obtained by solving a triangular linear system.

Unconstrained formulation

By using the null space representation $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{U}\mathbf{z}$, (18) can be transformed into the following unconstrained formulation:

$$\min_{\mathbf{z} \in \mathbb{R}^{p-n}} \left\{ \tilde{f}(\mathbf{z}) := f(\bar{\mathbf{x}} + \mathbf{U}\mathbf{z}) \right\}.$$

Example of null space representation

Problem

Given $\mathbf{s} \in \mathbb{R}^3$, we want to compute the **projection** of $\mathbf{s}$ onto an affine space as:

$$\min_{\mathbf{x} \in \mathbb{R}^3} (1/2)\|\mathbf{x} - \mathbf{s}\|_2^2 \quad \text{s.t.} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \ \mathbf{x} \in \mathbb{R}^3. \tag{19}$$

Example of null space representation

Problem

Given $\mathbf{s} \in \mathbb{R}^3$, we want to compute the **projection** of $\mathbf{s}$ onto an affine space as:

$$\min_{\mathbf{x}\in\mathbb{R}^3} (1/2)\|\mathbf{x}-\mathbf{s}\|_2^2 \ \text{ s.t. } \begin{bmatrix}1 & 1 & 1\\ 1 & 1 & -1\end{bmatrix}\mathbf{x} = \begin{bmatrix}2\\1\end{bmatrix}, \ \mathbf{x}\in\mathbb{R}^3. \tag{19}$$

Null-space representation

- By computing the QR factorization of $\mathbf{A}^T$ we obtain a 3×3 orthonormal matrix $\mathbf{Z}$ and a 1×1 triangle matrix $\mathbf{R}$.
- Since $\mathrm{rank}(\mathbf{A}) = 2$, $\dim(\mathrm{null}(\mathbf{A})) = 3-2 = 1$, we take the last column of $\mathbf{Z}$ to form a basis $\mathbf{U}$ of $\mathrm{null}(\mathbf{A})$, which is $\mathbf{U} := \begin{bmatrix}-\sqrt{2}/2\\ \sqrt{2}/2\\ 0\end{bmatrix}$.
- The two first columns of $\mathbf{Z}$ forms the basis of the range space of $\mathbf{A}^T$ called $\mathbf{V}$.
- By solving $\mathbf{R}^T\mathbf{y} = \mathbf{b}$ we obtain $\mathbf{y} \approx (-1.15470, -0.20412)^T$. Therefore

$$\bar{\mathbf{x}} := \mathbf{V}\mathbf{y} = (3/4, 3/4, 1/2)^T.$$

- We finally obtain $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{U}\mathbf{z}$, where $\mathbf{z}\in\mathbb{R}^2$ such that $\mathbf{A}\mathbf{x} = \mathbf{b}$.

From constrained to unconstrained formulation

The projection of $\mathbf{s}$ onto the affine space $\mathbf{A}\mathbf{x} = \mathbf{b}$

Problem (19) can be transformed into the unconstrained problem:

$$\min_{\mathbf{z}\in\mathbb{R}} (1/2)\|\mathbf{U}\mathbf{z} + \bar{\mathbf{x}} - \mathbf{s}\|_2^2.$$

This problem has a closed form solution $\mathbf{z}^\star = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T(\mathbf{s} - \bar{\mathbf{x}}) = \mathbf{U}^T(\mathbf{s} - \bar{\mathbf{x}})$.

From constrained to unconstrained formulation

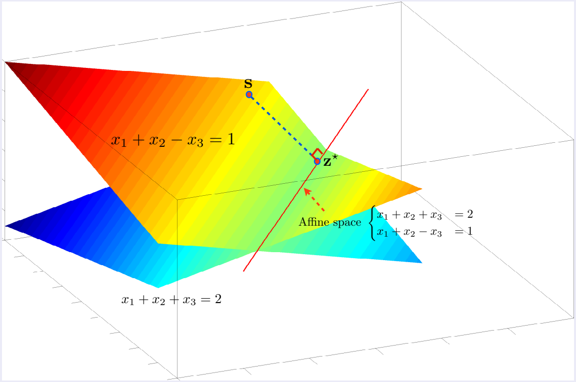
The projection of s onto the affine space $\mathbf{Ax} = \mathbf{b}$

Problem (19) can be transformed into the unconstrained problem:

$$\min_{\mathbf{z} \in \mathbb{R}} (1/2)\|\mathbf{U}\mathbf{z} + \bar{\mathbf{x}} - \mathbf{s}\|_2^2.$$

This problem has a closed form solution $\mathbf{z}^\star = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T(\mathbf{s} - \bar{\mathbf{x}}) = \mathbf{U}^T(\mathbf{s} - \bar{\mathbf{x}})$.

Illustration



Limitations of the null-space method

Limitations of the null space approach

- Require matrix factorization (e.g., QR factorization) to compute a basis $\mathbf{U}$ of the **null space** of $\mathbf{A}$ and a feasible point $\bar{\mathbf{x}}$, which is computational demand in high-dimension ($\mathcal{O}(n^2p)$).
- If matrix $\mathbf{A}$ is given implicitly (e.g., by linear operator), then computing $\mathbf{U}$ is impractical.
- Null space method destroys the original structure of the objective function f due to the affine transformation $\mathbf{U}\mathbf{z} + \bar{\mathbf{x}}$. For instance, $f(\mathbf{x}) := \|\mathbf{x}\|_1$, which is component-wise decomposable.

Convex problems with simple constraints

Convex problems with simple constraints

When $\mathbf{A}\mathbf{x} = \mathbf{b}$ is absent, problem (1) reduces to:

$$f^\star := \min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) \tag{20}$$

Convex problems with simple constraints

Convex problems with simple constraints

When $\mathbf{A}\mathbf{x} = \mathbf{b}$ is absent, problem (1) reduces to:

$$f^\star := \min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) \tag{20}$$

Assumption (Simplicity)

*$\mathcal{X}$ is "simple" so that the **projection** $\pi_{\mathcal{X}}$ of any point $\mathbf{s} \in \mathbb{R}^p$ onto $\mathcal{X}$ can be computed efficiently, i.e.:*

$$\pi_{\mathcal{X}}(\mathbf{s}) := \arg\min_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x} - \mathbf{s}\|_2,$$

can be solved efficiently (e.g., closed form solution or polynomial time).

Note: Let $\iota_{\mathcal{X}}$ be the **indicator function** of $\mathcal{X}$. Then

$$\pi_{\mathcal{X}}(\mathbf{s}) = \mathrm{prox}_{\iota_{\mathcal{X}}}(\mathbf{s}).$$

Examples can be found in Lectures 4 and 5.

Projected-gradient method

Assumption A.1

- $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^p)$
- $\pi_{\mathcal{X}}$ can be computed exactly.

Projected-gradient method

Assumption A.1

- $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^p)$
- $\pi_{\mathcal{X}}$ can be computed exactly.

Projected gradient method (ProjGA)
1. Choose $\mathbf{x}^0 \in \mathbb{R}^p$. **2.** For $k = 0, 1, \cdots$, perform: $$\mathbf{x}^{k+1} := \pi_{\mathcal{X}}(\mathbf{x}^k - (1/L_f)\nabla f(\mathbf{x}^k)).$$

Projected-gradient method

Assumption A.1

- $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^p)$
- $\pi_{\mathcal{X}}$ can be computed exactly.

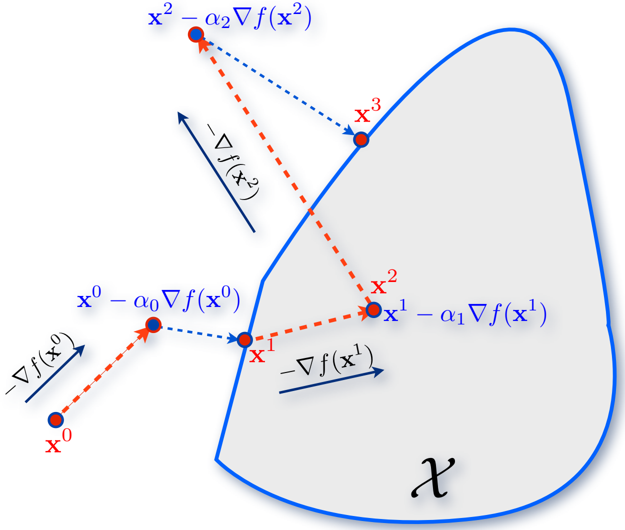
Projected gradient method (ProjGA)
1. Choose $\mathbf{x}^0 \in \mathbb{R}^p$. **2.** For $k = 0, 1, \cdots,$ perform: $$\mathbf{x}^{k+1} := \pi_{\mathcal{X}}(\mathbf{x}^k - (1/L_f)\nabla f(\mathbf{x}^k)).$$

Properties

- ProjGA can be enhanced by performing a line-search for approximating L_f.
- **Convergence**: The convergence of ProjGA remains the same as in standard gradient method, i.e.:

$$f(\mathbf{x}^k) - f^\star \le \frac{L_f\|\mathbf{x}^0 - \mathbf{x}^\star\|_2^2}{2(k+1)}, \ k \ge 0.$$

Illustration of the projected gradient method



Three iterations of the **projected gradient method**.

Fast projected-gradient method

Assumption

Under **Assumption A.1.**, ProjGA can be accelerated by using Nesterov's optimal method.

Fast projected gradient method (FastProjGA)
1. Choose $\mathbf{x}^0 \in \mathbb{R}^p$. Set $\mathbf{y}^0 := \mathbf{x}^0$ and $t_0 := 1$
2. For $k = 0, 1, \cdots$, perform:
$$\begin{cases} \mathbf{x}^{k+1} & := \pi_{\mathcal{X}}(\mathbf{y}^k - (1/L_f)\nabla f(\mathbf{y}^k)), \\ \mathbf{y}^{k+1} & := \mathbf{x}^{k+1} + ((t_k - 1)/t_{k+1})(\mathbf{x}^{k+1} - \mathbf{x}^k), \\ t_{k+1} & := (1 + \sqrt{1 + 4t_k^2})/2. \end{cases}$$

Fast projected-gradient method

Assumption

Under **Assumption A.1.**, ProjGA can be accelerated by using Nesterov's optimal method.

Fast projected gradient method (FastProjGA)

1. Choose $\mathbf{x}^0 \in \mathbb{R}^p$. Set $\mathbf{y}^0 := \mathbf{x}^0$ and $t_0 := 1$
2. For $k = 0, 1, \cdots$, perform:

$$\begin{cases} \mathbf{x}^{k+1} & := \pi_{\mathcal{X}}(\mathbf{y}^k - (1/L_f)\nabla f(\mathbf{y}^k)), \\ \mathbf{y}^{k+1} & := \mathbf{x}^{k+1} + ((t_k - 1)/t_{k+1})(\mathbf{x}^{k+1} - \mathbf{x}^k), \\ t_{k+1} & := (1 + \sqrt{1 + 4t_k^2})/2. \end{cases}$$

Convergence

The convergence of FastProjGA remains the same as in fast gradient method, i.e.:

$$f(\mathbf{x}^k) - f^\star \leq \frac{2L_f\|\mathbf{x}^0 - \mathbf{x}^\star\|_2^2}{(k+1)^2}, \ \ k \geq 0.$$

Frank-Wolfe's method

Problem setting and assumption

$$f^\star := \min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \tag{21}$$

Assumptions

- $\mathcal{X}$ is nonempty, convex, closed and bounded.
- $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^p)$ (i.e., convex with Lipschitz gradient).
- For given $c \in \mathbb{R}^p$, $\hat{\mathbf{x}} := \arg\min_{\mathbf{x}\in\mathcal{X}} c^T\mathbf{x}$ can be solved efficiently.

Frank-Wolfe's method

Problem setting and assumption

$$f^\star := \min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \tag{21}$$

Assumptions

- $\mathcal{X}$ is nonempty, convex, closed and bounded.
- $f \in \mathcal{F}_L^{1,1}(\mathbb{R}^p)$ (i.e., convex with Lipschitz gradient).
- For given $c \in \mathbb{R}^p$, $\hat{\mathbf{x}} := \arg\min_{\mathbf{x}\in\mathcal{X}} c^T\mathbf{x}$ can be solved efficiently.

Frank-Wolfe's method [5]

Conditional gradient method (CGA)

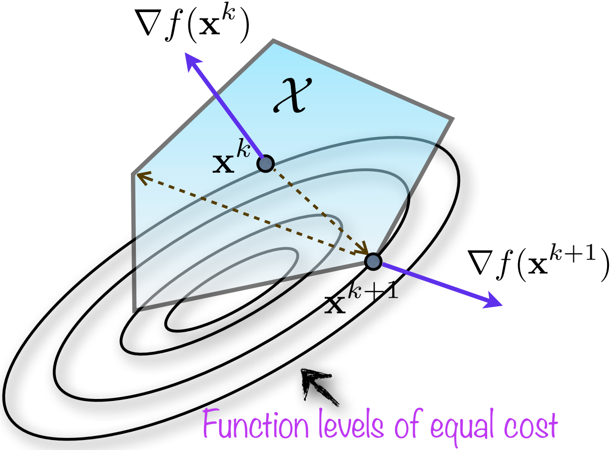
1. Choose $\mathbf{x}^0 \in \mathcal{X}$.
2. For $k = 0, 1, \cdots$, perform:

$$\begin{cases} \hat{\mathbf{x}}^k & := \arg\min_{\mathbf{x}\in\mathcal{X}} \nabla f(\mathbf{x}^k)^T\mathbf{x}, \\ \mathbf{x}^{k+1} & := (1-\gamma_k)\mathbf{x}^k + \gamma_k\hat{\mathbf{x}}^k, \end{cases}$$

where $\gamma_k := \frac{2}{k+2}$ is a given relaxation parameter.

Geometric interpretation of Frank-Wolfe's method

- Most straightforward way to generate a feasible *descent direction*: find $\hat{\mathbf{x}}^k$ that satisfies $\nabla f(\mathbf{x}^k)^T(\hat{\mathbf{x}}^k - \mathbf{x}^k) < 0$.
- We assume that the constraint set $\mathcal{X}$ is compact so that the direction finding problem has a solution.



Properties and convergence of Frank-Wolfe's method

Properties

- Since $\mathcal{X}$ is bounded, $\hat{x}^k$ is well-defined.
- CGA is a "norm-free" method
- $\hat{x}^k$ attains at the boundary of $\mathcal{X}$, which preserves sparsity.
- When $\mathcal{X}$ is a polytope, computing $\hat{x}^k$ is equivalent to solving a linear program.
- Allows inexactness in computing $\hat{\mathbf{x}}^k$
- γ_k can be estimated by a line-search procedure.

Properties and convergence of Frank-Wolfe's method

Properties

- Since $\mathcal{X}$ is bounded, $\hat{x}^k$ is well-defined.
- CGA is a "norm-free" method
- $\hat{x}^k$ attains at the boundary of $\mathcal{X}$, which preserves sparsity.
- When $\mathcal{X}$ is a polytope, computing $\hat{x}^k$ is equivalent to solving a linear program.
- Allows inexactness in computing $\hat{\mathbf{x}}^k$
- γ_k can be estimated by a line-search procedure.

Theorem (Convergence [5])

Let $\{\mathbf{x}^k\}$ be the sequence generated by CGA. Then

$$\boxed{f(\mathbf{x}^k) - f^\star \leq \frac{2L_f}{k+1} D_{\mathcal{X}}^2,}$$

where $D_{\mathcal{X}} := \max_{\mathbf{x},\mathbf{y}\in\mathcal{X}} \|\mathbf{x}-\mathbf{y}\|$, the diameter of $\mathcal{X}$ w.r.t. $\|\cdot\|$.

The convergence rate of **CGA** is $\mathcal{O}(1/k)$ which is the same order as **ProjGA**. However, the diameter $\mathcal{D}_{\mathcal{X}}$ is in general worse than $\|\mathbf{x}^0 - \mathbf{x}^\star\|_2$ in **ProjGA** in the ℓ_2-norm.

Dual subgradient method

Dual problem (13) is in general nonsmooth and convex. **Subgradient ascent method** can be applied to solve it.

Properties of dual function

- d is **concave**, but **not necessary differentiable**.
- **Subgradient:** $\mathbf{A}\mathbf{x}^\star(\lambda) - \mathbf{b} \in \partial d(\lambda)$, where $\mathbf{x}^\star(\lambda)$ is a solution of (12).

Dual subgradient method

Dual problem (13) is in general nonsmooth and convex. **Subgradient ascent method** can be applied to solve it.

Properties of dual function

- d is **concave**, but **not necessary differentiable**.
- **Subgradient:** $\mathbf{A}\mathbf{x}^\star(\lambda) - \mathbf{b} \in \partial d(\lambda)$, where $\mathbf{x}^\star(\lambda)$ is a solution of (12).

Dual subgradient ascent method

Dual subgradient method (DSGM):

1. Choose $\lambda^0 \in \mathbb{R}^p$.
2. For $k = 0, 1, \cdots$, perform:
 2.a. Solve (12) to obtain $\mathbf{x}^\star(\lambda)$.
 2.b. Compute the subgradient $\nabla d(\lambda^k) := \mathbf{A}\mathbf{x}^\star(\lambda^k) - \mathbf{b}$.
 2.c. Update $\boxed{\lambda^{k+1} := \lambda^k + \frac{R}{\sqrt{k+1}}\nabla d(\lambda^k)}$, where R is a given constant.

Convergence of DSGM

Well-definedness

- Problem (12) may not have solution $\mathbf{x}^\star(\lambda)$ for any λ. Then DSGM is not well-defined except $\mathcal{X}$ is bounded.
- Impractical to evaluate $R_\star := \|\lambda^0 - \lambda^\star\|_2$, use an upper bound R of $R_\star$.

Convergence of DSGM

Well-definedness

- Problem (12) may not have solution $\mathbf{x}^\star(\lambda)$ for any λ. Then DSGM is not well-defined except $\mathcal{X}$ is bounded.
- Impractical to evaluate $R_\star := \|\lambda^0 - \lambda^\star\|_2$, use an upper bound R of $R_\star$.

Theorem (Convergence)

Assume that $\|\mathbf{A}\mathbf{x}^\star(\lambda^k) - \mathbf{b}\| \leq M_d$ *for all* $k \geq 0$. *Then* $\{\lambda^k\}$ *generated by DSGM satisfies*

$$\boxed{d^\star - d(\lambda^k) \leq \frac{M_d R_\star}{\sqrt{k+1}}, \forall k \geq 0,}$$

where $R_\star := \min_{\lambda^\star} \|\lambda^0 - \lambda^\star\|_2$. *Convergence rate of DSGM is* $\mathcal{O}(1/\sqrt{k})$.

Convergence of DSGM

Well-definedness

- Problem (12) may not have solution $\mathbf{x}^\star(\lambda)$ for any λ. Then DSGM is not well-defined except $\mathcal{X}$ is bounded.
- Impractical to evaluate $R_\star := \|\lambda^0 - \lambda^\star\|_2$, use an upper bound R of $R_\star$.

Theorem (Convergence)

Assume that $\|\mathbf{A}\mathbf{x}^\star(\lambda^k) - \mathbf{b}\| \leq M_d$ for all $k \geq 0$. Then $\{\lambda^k\}$ generated by DSGM satisfies

$$\boxed{d^\star - d(\lambda^k) \leq \frac{M_d R_\star}{\sqrt{k+1}}, \forall k \geq 0,}$$

where $R_\star := \min_{\lambda^\star} \|\lambda^0 - \lambda^\star\|_2$. Convergence rate of DSGM is $\mathcal{O}(1/\sqrt{k})$.

Special cases

1. If both f is strongly convex, then d is smooth and its gradient is Lipschitz continuous., $d \in \mathcal{F}_L^{1,1}(\mathbb{R}^p)$. **Gradient** and **fast gradient methods** in Lecture 3 can be used to solve the dual problem.
2. **Smoothing techniques** in Lecture 5 can be used to smooth the dual function d.

Augmented Lagrangian method

Dual problem (13) is convex but generally nonsmooth. By augmenting $\mathcal{L}$ with $(\kappa/2)\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2$, we obtain augmented dual function d_κ, which maintains basic properties of d but smooth and Lipschitz gradient.

Augmented Lagrangian method

Dual problem (13) is convex but generally nonsmooth. By augmenting $\mathcal{L}$ with $(\kappa/2)\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$, we obtain augmented dual function d_κ, which maintains basic properties of d but smooth and Lipschitz gradient.

Augmented Lagrangian and augmented dual function

- **Augmented Lagrangian:** $\mathcal{L}_\kappa(\mathbf{x}, \lambda) := \mathcal{L}(\mathbf{x}, \lambda) + (\kappa/2)\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$, where $\rho > 0$ is a penalty parameter.
- **Augmented dual function:**

$$d_\kappa(\lambda) := \min_{\mathbf{x} \in \mathcal{X}} \left\{ \mathcal{L}_\kappa(\mathbf{x}, \lambda) := f(\mathbf{x}) + \lambda^T(\mathbf{A}\mathbf{x} - \mathbf{b}) + (\kappa/2)\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 \right\}. \quad (22)$$

Augmented Lagrangian method

Dual problem (13) is convex but generally nonsmooth. By augmenting $\mathcal{L}$ with $(\kappa/2)\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$, we obtain augmented dual function d_κ, which maintains basic properties of d but smooth and Lipschitz gradient.

Augmented Lagrangian and augmented dual function

- **Augmented Lagrangian:** $\mathcal{L}_\kappa(\mathbf{x}, \lambda) := \mathcal{L}(\mathbf{x}, \lambda) + (\kappa/2)\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$, where $\rho > 0$ is a penalty parameter.
- **Augmented dual function:**

$$d_\kappa(\lambda) := \min_{\mathbf{x}\in\mathcal{X}} \left\{ \mathcal{L}_\kappa(\mathbf{x}, \lambda) := f(\mathbf{x}) + \lambda^T(\mathbf{A}\mathbf{x} - \mathbf{b}) + (\kappa/2)\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 \right\}. \quad (22)$$

Key properties of d_κ

- d_κ is concave and smooth and

$$\nabla d_\kappa(\lambda) = \mathbf{A}\mathbf{x}^\star_\kappa(\lambda) - \mathbf{b},$$

  where $\mathbf{x}^\star_\kappa(\lambda)$ is the solution of (22).
- ∇d_κ is Lipschitz continuous with a Lipschitz constant $L_d := \kappa^{-1}$, i.e.:

$$\|\nabla d_\kappa(\lambda) - \nabla d_\kappa(\hat{\lambda})\| \le \kappa^{-1}\|\lambda - \hat{\lambda}\|, \ \forall \lambda, \hat{\lambda} \in \mathbb{R}^n.$$

Example: Behavior of the augmented Lagrangian dual function

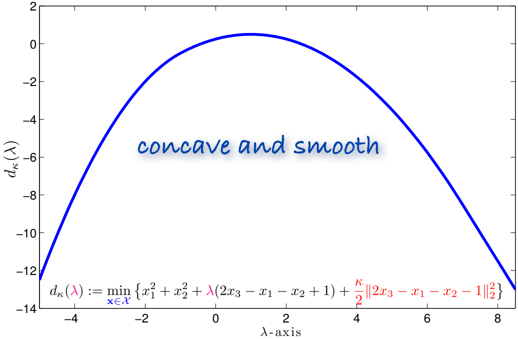
Consider a constrained convex problem:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^3} \quad & \{f(\mathbf{x}) := x_1^2 + x_2^2\}, \\ \text{s.t.} \quad & 2x_3 - x_1 - x_2 = 1, \\ & \mathbf{x} \in \mathcal{X} := [-2,2]\times[-2,2]\times[0,2]. \end{aligned}$$

The **augmented Lagrangian dual function** is defined as

$$d_\kappa(\lambda) := \min_{\mathbf{x}\in\mathcal{X}}\{x_1^2 + x_2^2 + \lambda(2x_3 - x_1 - x_2 + 1) + (\kappa/2)\|2x_3 - x_1 - x_2 - 1\|_2^2\}$$

is concave and nonsmooth as illustrated in the figure below.



Augmented dual problem

Augmented dual problem

$$\boxed{d_\kappa^\star := \max_{\lambda \in \mathbb{R}^n} d_\kappa(\lambda), \quad \kappa > 0.} \tag{23}$$

Augmented dual problem

Augmented dual problem

$$\boxed{d_\kappa^\star := \max_{\lambda\in\mathbb{R}^n} d_\kappa(\lambda), \quad \kappa > 0.} \tag{23}$$

Relation to the dual problem (13)

Under Slater's condition and $\mathcal{X}^\star \neq \emptyset$, we have

- The dual solution set of (23) is coincided with the one of the dual problem (13).
- $f^\star = d^\star = d_\kappa^\star$ for any $\kappa > 0$.

The augmented dual problem (23) is smooth and convex $\Rightarrow$ **Gradient and Fast gradient methods** can be applied to solve it.

Augmented Lagrangian method

Augmented Lagrangian method (ALM):

1. Choose $\lambda^0 \in \mathbb{R}^p$ and $\kappa > 0$.
2. For $k = 0, 1, \cdots,$ perform:
   - **2.a**. Solve (22) to compute $\nabla d_\kappa(\lambda^k) := \mathbf{A}\mathbf{x}^\star_\kappa(\lambda^k) - \mathbf{b}$.
   - **2.b**. Update $\boxed{\lambda^{k+1} := \lambda^k + \kappa \nabla d_\kappa(\lambda^k).}$

Augmented Lagrangian method

Augmented Lagrangian method (ALM):
1. Choose $\lambda^0 \in \mathbb{R}^p$ and $\kappa > 0$.
2. For $k = 0, 1, \cdots,$ perform:
2.a. Solve (22) to compute $\nabla d_\kappa(\lambda^k) := \mathbf{A}\mathbf{x}^\star_\kappa(\lambda^k) - \mathbf{b}$.
2.b. Update $\boxed{\lambda^{k+1} := \lambda^k + \kappa\nabla d_\kappa(\lambda^k).}$

ALM can be accelerated by Nesterov's optimal method.

Fast augmented Lagrangian method (FALM)

1. Choose $\lambda^0 \in \mathbb{R}^p$ and $\kappa > 0$. Set $\tilde{\lambda}^0 := \lambda^0$ and $t_0 := 1$
2. For $k = 0, 1, \cdots,$ perform:
   - 2.a. Solve (22) to compute $\nabla d_\kappa(\tilde{\lambda}^k) := \mathbf{A}\mathbf{x}^\star_\kappa(\tilde{\lambda}^k) - \mathbf{b}$.
   - 2.b. Update

$$\begin{cases} \lambda^{k+1} & := \tilde{\lambda}^k + \kappa\nabla d_\kappa(\tilde{\lambda}^k), \\ \tilde{\lambda}^{k+1} & := \lambda^{k+1} + ((t_k - 1)/t_{k+1})(\lambda^{k+1} - \lambda^k), \\ t_{k+1} & := (1 + \sqrt{1 + 4t_k^2})/2. \end{cases}$$

Convergence of ALM and FALM

Theorem (Convergence)

- *Let $\{\lambda^k\}$ be the sequence generated by ALM. Then*

$$\boxed{d^\star - d_\kappa(\lambda^k) \le \frac{\|\lambda^0 - \lambda^\star\|_2^2}{2\kappa(k+1)},\ k \ge 0.}$$

- *Let $\{\lambda^k\}$ be the sequence generated by FALM. Then*

$$\boxed{d^\star - d_\kappa(\lambda^k) \le \frac{2\|\lambda^0 - \lambda^\star\|_2^2}{\kappa(k+2)^2},\ k \ge 0.}$$

- The convergence rate of ALM is $\mathcal{O}(1/k)$ w.r.t. the augmented dual function d_κ.
- The convergence rate of FALM is $\mathcal{O}(1/k^2)$ w.r.t. the augmented dual function d_κ.
- Important observation: The right-hand side of both estimates depends on κ. When κ is getting large, the right-hand side is decreasing.

Drawbacks and enhancements

Drawbacks

1. Drawback 1: The quadratic term $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$ in (22) destroys the separability as well as the tractable proximity of f.
2. Drawback 2: Solving (22) exactly is impractical.
3. Drawback 3: No theoretical guarantee for choosing appropriate values of κ.

Drawbacks and enhancements

Drawbacks

1. Drawback 1: The quadratic term $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$ in (22) destroys the separability as well as the tractable proximity of f.
2. Drawback 2: Solving (22) exactly is impractical.
3. Drawback 3: No theoretical guarantee for choosing appropriate values of κ.

Enhancements

1. Allow inexactness of solving (22), while guaranteeing the same convergence rate.
2. Update the penalty parameter κ
   - Increasing ρ: Lead to the increase of ill-condition in (22).
   - Adaptively update κ: Often heuristic
3. Process the quadratic term $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$ by linearization, alternating, etc.

Example: Group basis pursuit

Group basis pursuit

Given a linear operator $\mathbf{A}$, a measurement vector $\mathbf{b}$ and a group structure $\mathcal{G} := \{\mathcal{G}_1, \ldots, \mathcal{G}_g\}$. The aim is to solve:

$$\min_{\mathbf{x}\in\mathbb{R}^p} \sum_{i=1}^{g} \|\mathbf{x}_{\mathcal{G}_i}\|_2 \ \text{ s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}. \tag{24}$$

Example: Group basis pursuit

Group basis pursuit

Given a linear operator $\mathbf{A}$, a measurement vector $\mathbf{b}$ and a group structure $\mathcal{G} := \{\mathcal{G}_1, \ldots, \mathcal{G}_g\}$. The aim is to solve:

$$\min_{\mathbf{x}\in\mathbb{R}^p} \sum_{i=1}^{g} \|\mathbf{x}_{\mathcal{G}_i}\|_2 \ \text{ s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}. \tag{24}$$

Applying ALM and FALM

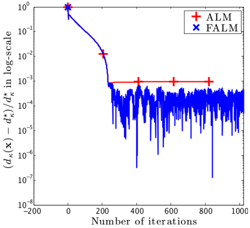
The main computation:

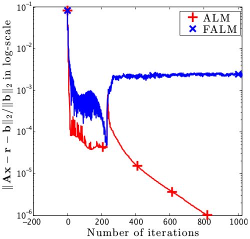
- Solving the subproblem (22), which is

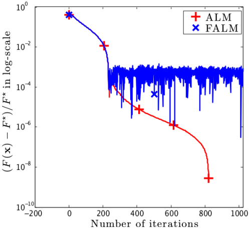
$$\mathbf{x}^\star_\kappa(\lambda) := \arg\min_{\mathbf{x}\in\mathcal{X}} \left\{ \sum_{i=1}^{g} \|\mathbf{x}_{\mathcal{G}_i}\|_2 + \lambda^T(\mathbf{A}\mathbf{x} - \mathbf{b}) + (\kappa/2)\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 \right\},$$

  by applying, e.g., FISTA (Lecture 5).
- Updating κ by increasing it as $\kappa_{k+1} := \eta\kappa_k$ for given $\eta > 1$.

Numerical results



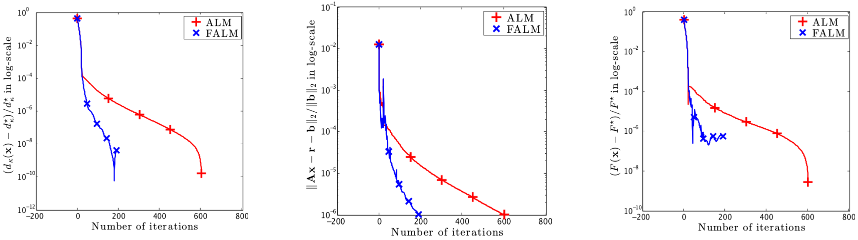




	ALM	FALM
Primal Obj. Value	47.145	47.187
Feas. Gap	0.99×10^{-6}	0.23×10^{-2}
Dual Obj. Value	33.196	33.165
Iterations	821	2000
CPU time (s)	2.656	6.513
Calls A/A^T	9031/8210	22000/20000
Recovery error	0.04%	0.4%

- Parameters: $\kappa = 0.5, \ \eta = 1$
- Input: $n = 341, \ p = 1024, \ g = 85, \ \mathrm{nzg} = 11; \ \min|\mathcal{G}_i| = 5, \ \max|\mathcal{G}_i| = 23, \ \mathrm{mean}|\mathcal{G}_i| = 12.04$
- Proximal operations (FISTA): max iterations 10, stop criteria 10^{-9} relative change, warm start
- Stopping criteria: $\|\mathbf{Ax}^k - \mathbf{r}^k - b\| \leq 10^{-6}\|\mathbf{b}\|$ and $\|(\mathbf{x}^k, \mathbf{r}^k) - (\mathbf{x}^{k-1}, \mathbf{r}^{k-1})\| \leq 10^{-6}\|(\mathbf{x}^k, \mathbf{r}^k)\|$

Numerical results



	ALM	FALM
Primal Obj. Value	47.1451	47.1452
Feas. Gap	0.99 $\times 10^{-6}$	0.99 $\times 10^{-6}$
Dual Obj. Value	33.196	33.196
Iterations	605	192
CPU time (s)	10.647	4.920
Calls A/A^T	38348/37743	17420/17228
Recovery error	0.04%	0.04%

- Parameters: $\kappa = 0.5, \ \eta = 1$
- Input: $n = 341, \ p = 1024, \ g = 85, \ \text{nzg} = 11; \ \min|\mathcal{G}_i| = 5, \ \max|\mathcal{G}_i| = 23, \ \text{mean}|\mathcal{G}_i| = 12.04$
- Proximal operations (FISTA): max iterations 100, stop criteria 10^{-9} relative change, warm start
- Stopping criteria: $\|\mathbf{Ax}^k - \mathbf{r}^k - b\| \leq 10^{-6}\|\mathbf{b}\|$ and $\|(\mathbf{x}^k, \mathbf{r}^k) - (\mathbf{x}^{k-1}, \mathbf{r}^{k-1})\| \leq 10^{-6}\|(\mathbf{x}^k, \mathbf{r}^k)\|$

Remarks

Remarks

- The FALM method is sensitive to the inexactness of the solution of (22)

$$\mathbf{x}^\star_\kappa(\lambda) := \arg\min_{\mathbf{x}\in\mathcal{X}} \left\{ \sum_{i=1}^{g} \|\mathbf{x}_{\mathcal{G}_i}\|_2 + \lambda^T(\mathbf{A}\mathbf{x}-\mathbf{b}) + (\kappa/2)\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 \right\}$$

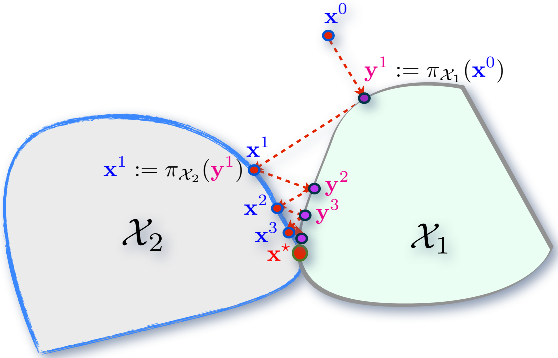
- "Fast" updates of the dual variable λ^k influence the primal updates
  - warm-start strategy - at iteration k choose initial solution of (22) $x^\star_\kappa(\lambda^{k-1})$
  - increase iterations number to achieve convergence of the primal (also tolerance)
  - keep η small (FALM more sensitive to large values of η)
- Guarantes are given only for the dual problem, not for the primal

Alternating idea to overcome the non-separability

- **Problem:** Given two nonempty, closed and convex sets $\mathcal{X}_1$ and $\mathcal{X}_2$. **Find** a point $\boxed{\mathbf{x}^\star \in \mathcal{X}_1 \cap \mathcal{X}_2}$.
- **Strategy:** Start from $\mathbf{x}^0$ and **iterate alternatively**:

$$\begin{cases} \mathbf{y}^{k+1} & := \pi_{\mathcal{X}_1}(\mathbf{x}^k) \\ \mathbf{x}^{k+1} & := \pi_{\mathcal{X}_2}(\mathbf{y}^{k+1}) \end{cases}$$

where $\pi_{\mathcal{X}}$ is the projection on the convex set $\mathcal{X}$.



Alternating minimization algorithm (AMA)

Assumptions

- Problem (1) has a separable structure with $p = 2$, i.e.:

$$f^\star := \begin{cases} \min_{\mathbf{x}\in\mathbb{R}^p} & \left\{ f(\mathbf{x}) := f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) \right\}, \\ \text{s.t.} & \mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 = \mathbf{b},\ \mathbf{x}_1 \in \mathcal{X}_1, \mathbf{x}_2 \in \mathcal{X}_2. \end{cases} \tag{25}$$

- f_1 is strongly convex with parameter $\mu_1 > 0$.

Alternating minimization algorithm (AMA)

Assumptions

- Problem (1) has a separable structure with $p = 2$, i.e.:

$$f^\star := \begin{cases} \min_{\mathbf{x}\in\mathbb{R}^p} & \left\{ f(\mathbf{x}) := f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) \right\}, \\ \text{s.t.} & \mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 = \mathbf{b},\ \mathbf{x}_1 \in \mathcal{X}_1, \mathbf{x}_2 \in \mathcal{X}_2. \end{cases} \tag{25}$$

- f_1 is strongly convex with parameter $\mu_1 > 0$.

The idea of AMA [7]

- Alternating between variables $\mathbf{x}_1$ and $\mathbf{x}_2$ in:

$$\min_{\mathbf{x}_1\in\mathcal{X}_1, \mathbf{x}_2\in\mathcal{X}_2} \left\{ f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) + \lambda^T\mathbf{A}_1\mathbf{x}_1 + \lambda^T\mathbf{A}_2\mathbf{x}_2 + (\kappa/2)\|\mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 - \mathbf{b}\|_2^2 \right\}.$$

- Since f_1 is convex, neglects the augmented term. Then, this step becomes

$$\begin{cases} \mathbf{x}_1^{k+1} & := \arg\min_{\mathbf{x}_1\in\mathcal{X}_1} \left\{ f_1(\mathbf{x}_1) + (\lambda^k)^T\mathbf{A}_1\mathbf{x}_1 \right\}, \\ \mathbf{x}_2^{k+1} & := \arg\min_{\mathbf{x}_2\in\mathcal{X}_2} \left\{ f_2(\mathbf{x}_2) + (\lambda^k)^T\mathbf{A}_2\mathbf{x}_2 + \frac{\kappa}{2}\|\mathbf{A}_1\mathbf{x}_1^{k+1} + \mathbf{A}_2\mathbf{x}_2 - \mathbf{b}\|_2^2 \right\}. \end{cases}$$

AMA: Alternating minimization algorithm

Alternating minimization algorithm (AMA):

1. Choose $\lambda^0 \in \mathbb{R}^p$ and $\kappa > 0$.
2. For $k = 0, 1, \cdots$, perform:

$$\begin{cases} \mathbf{x}_1^{k+1} & := \arg\min_{\mathbf{x}_1 \in \mathcal{X}_1} \left\{ f_1(\mathbf{x}_1) + (\lambda^k)^T \mathbf{A}_1 \mathbf{x}_1 \right\} \\ \mathbf{x}_2^{k+1} & := \arg\min_{\mathbf{x}_2 \in \mathcal{X}_2} \left\{ f_2(\mathbf{x}_2) + (\lambda^k)^T \mathbf{A}_2 \mathbf{x}_2 + \frac{\kappa}{2} \|\mathbf{A}_1 \mathbf{x}_1^{k+1} + \mathbf{A}_2 \mathbf{x}_2 - \mathbf{b}\|_2^2 \right\} \\ \lambda^{k+1} & := \lambda^k + \kappa(\mathbf{A}_1 \mathbf{x}_1^{k+1} + \mathbf{A}_2 \mathbf{x}_2^{k+1} - \mathbf{b}). \end{cases}$$

AMA: Alternating minimization algorithm

Alternating minimization algorithm (AMA):

1. Choose $\lambda^0 \in \mathbb{R}^p$ and $\kappa > 0$.
2. For $k = 0, 1, \cdots$, perform:

$$
\begin{cases}
\mathbf{x}_1^{k+1} & := \arg\min_{\mathbf{x}_1 \in \mathcal{X}_1} \left\{ f_1(\mathbf{x}_1) + (\lambda^k)^T \mathbf{A}_1 \mathbf{x}_1 \right\} \\
\mathbf{x}_2^{k+1} & := \arg\min_{\mathbf{x}_2 \in \mathcal{X}_2} \left\{ f_2(\mathbf{x}_2) + (\lambda^k)^T \mathbf{A}_2 \mathbf{x}_2 + \frac{\kappa}{2} \|\mathbf{A}_1 \mathbf{x}_1^{k+1} + \mathbf{A}_2 \mathbf{x}_2 - \mathbf{b}\|_2^2 \right\} \\
\lambda^{k+1} & := \lambda^k + \kappa(\mathbf{A}_1 \mathbf{x}_1^{k+1} + \mathbf{A}_2 \mathbf{x}_2^{k+1} - \mathbf{b}).
\end{cases}
$$

Implementation remarks

- **Main computation:** Solving two subproblems to compute $\mathbf{x}_1^{k+1}$ and $\mathbf{x}_2^{k+1}$.
- $\mathbf{A}_2$ prevents the tractable proximity from f_2.
- When $\mathbf{A}_2^T \mathbf{A}_2 = \mathbf{I}$, we have $\mathbf{x}_2^{k+1} = \mathrm{prox}_{\kappa^{-1} f_2}(\mathbf{A}_2^T(\mathbf{b} - \mathbf{A}_1 \mathbf{x}_1^{k+1}) - \kappa^{-1} \mathbf{A}_2^T \lambda^k)$.
- When $\mathbf{A}_2^T \mathbf{A}_2 \neq \mathbf{I}$, we can approximate $\mathbf{x}_2^{k+1}$ by linearizing the quadratic term.
- The penalty parameter κ can be updated.

Convergence of AMA

Observations

- AMA is a **proximal-gradient method** applying to the Frenchel dual problem:

$$\tilde{d}^\star := \max_{\lambda\in\mathbb{R}^p} \left\{ \tilde{d}(\lambda) := -f_1^*(-\mathbf{A}_1^T\lambda) - f_2^*(-\mathbf{A}_2^T\lambda) - \mathbf{b}^T\lambda \right\}. \tag{26}$$

  where f_1^* and f_2^* are the Fenchel conjugate of f_1 and f_2, respectively.
- Since f_1 is strongly convex, the conjugate f_1^* is Lipschitz gradient with Lipschitz constant $L_{f_1^*} := \mu_1^{-1}$.
- AMA can be accelerated by using Nesterov's optimal gradient method (see [3]).

Convergence of AMA

Observations

- AMA is a **proximal-gradient method** applying to the Frenchel dual problem:

$$\tilde{d}^\star := \max_{\lambda \in \mathbb{R}^p} \left\{ \tilde{d}(\lambda) := -f_1^*(-\mathbf{A}_1^T\lambda) - f_2^*(-\mathbf{A}_2^T\lambda) - \mathbf{b}^T\lambda \right\}. \qquad (26)$$

  where f_1^* and f_2^* are the Fenchel conjugate of f_1 and f_2, respectively.
- Since f_1 is strongly convex, the conjugate f_1^* is Lipschitz gradient with Lipschitz constant $L_{f_1^*} := \mu_1^{-1}$.
- AMA can be accelerated by using Nesterov's optimal gradient method (see [3]).

Theorem (Convergence theorem [3])

Let $\{(\mathbf{x}_1^k, \mathbf{x}_2^k, \lambda^k)\}$ be the sequence generated by AMA. Assume that $\rho < 2\mu_1/\lambda_{\max}(\mathbf{A}_1^T\mathbf{A}_1)$. Then

$$\boxed{\tilde{d}^\star - \tilde{d}(\lambda^k) \le \frac{\lambda_{\max}(\mathbf{A}_1^T\mathbf{A}_1)}{2\mu_1(k+1)}\|\lambda^0 - \lambda^\star\|_2^2,}$$

where $\lambda_{\max}(\mathbf{A}_1^T\mathbf{A}_1)$ is the maximum eigenvalue of $\mathbf{A}_1^T\mathbf{A}_1$.

Example: ℓ_1-regularized least squares

Problem (ℓ_1-regularized least squares)

$$\min_{\mathbf{x}\in\mathbb{R}^p} (1/2)\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 + \rho\|\mathbf{x}\|_1, \tag{27}$$

where $\rho > 0$ is a regularization parameter.

Example: ℓ_1-regularized least squares

Problem (ℓ_1-regularized least squares)

$$\min_{\mathbf{x}\in\mathbb{R}^p} (1/2)\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 + \rho\|\mathbf{x}\|_1, \tag{27}$$

where $\rho > 0$ is a regularization parameter.

Applying AMA

Introducing a slack variable $\mathbf{r} = \mathbf{A}\mathbf{x} - \mathbf{b}$, we can reformulate (27) as

$$\min_{\mathbf{x}\in\mathbb{R}^p, \mathbf{r}\in\mathbb{R}^n} \quad (1/2)\|\mathbf{r}\|_2^2 + \rho\|\mathbf{x}\|_1, \quad \text{s.t. } \mathbf{A}\mathbf{x} - \mathbf{r} = \mathbf{b}.$$

The main steps of AMA becomes

$$\begin{cases} \mathbf{r}^{k+1} & := \arg\min_{\mathbf{r}\in\mathbb{R}^n} \left\{(1/2)\|\mathbf{r}\|_2^2 - (\lambda^k)^T\mathbf{r}\right\} \equiv \lambda^k \\ \mathbf{x}^{k+1} & := \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{\rho\|\mathbf{x}\|_1 + (\lambda^k)^T\mathbf{A}\mathbf{x} + \frac{\kappa}{2}\|\mathbf{A}\mathbf{x} - \mathbf{r}^{k+1} - \mathbf{b}\|_2^2\right\}, \\ \lambda^{k+1} & := \lambda^k + \kappa(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{r}^{k+1} - \mathbf{b}). \end{cases}$$

For $\mathbf{A}^T\mathbf{A} = \mathbb{I}$, the $\mathbf{x}$-step reduces to:

$$\mathbf{x}^{k+1} := \mathrm{prox}_{\kappa^{-1}\rho\|\mathbf{x}\|_1}\left(\mathbf{A}^T(\mathbf{b} + \lambda^k) - \kappa^{-1}\mathbf{A}^T\lambda^k\right).$$

Approaches to solving the subproblem

Problem

- *The main computation of AMA is the solution of:*

$$\mathbf{x}^{k+1} := \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ \rho\|\mathbf{x}\|_1 + (\lambda^k)^T\mathbf{A}\mathbf{x} + \frac{\kappa}{2}\|\mathbf{A}\mathbf{x} - \mathbf{r}^{k+1} - \mathbf{b}\|_2^2 \right\} \tag{28}$$

- (28) *has no closed form solution (except for* $\mathbf{A}^T\mathbf{A} = \mathbb{I}$ *).*

Solution

- There are two ways to overcome this drawback:
  - Applying FISTA.
  - Linearize the quadratic term: $q(\mathbf{x}) := q(\mathbf{x}^k) + \nabla q(\mathbf{x}^k)^T(\mathbf{x} - x^k) + \frac{L}{2}\|\mathbf{x} - \mathbf{x}^k\|_2^2$ where L is teh Lipschitz constant equal to $\|\mathbf{A}\|_2^2$

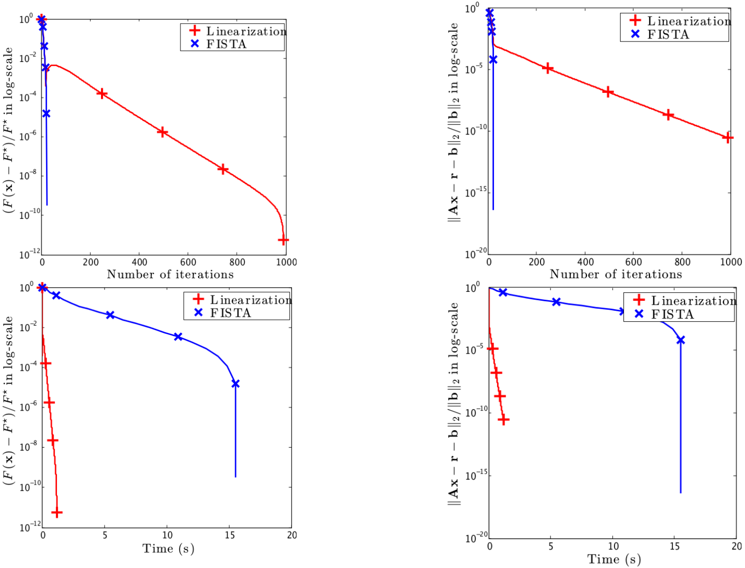
    Note: Is equivalent to applying FISTA with 1 iteration

Numerical results - High accuracy

	Linearization	FISTA
Primal Obj. Value	14.241	14.241
Feas. Gap	0.3 $\times 10^{-10}$	0.3 $\times 10^{-17}$
Iterations	991	23
Inner Iterations	991	13835
CPU time (s)	1.187	15.555
Calls A/A^T	992/991	13859/13835

- Parameters: $\rho = 0.1,\ \kappa = 0.01,\ \eta = 1.25$
- Input: $n = 750,\ p = 2000,\ k = 200,$ Noise ~ $\mathcal{N}(0, \sigma^2 \mathcal{I})$ with $\sigma = 10^{-3}$
- FISTA: max iterations 1000, stop criteria 10^{-10} relative change, warm start
- Stopping criteria: $\|\mathbf{A}\mathbf{x}^k - \mathbf{r}^k - b\| \leq 10^{-10}\|\mathbf{b}\|$ and $\|\mathbf{x}^k - \mathbf{x}^{k-1}\| \leq 10^{-10}\|\mathbf{x}^k\|$

Convergence plots

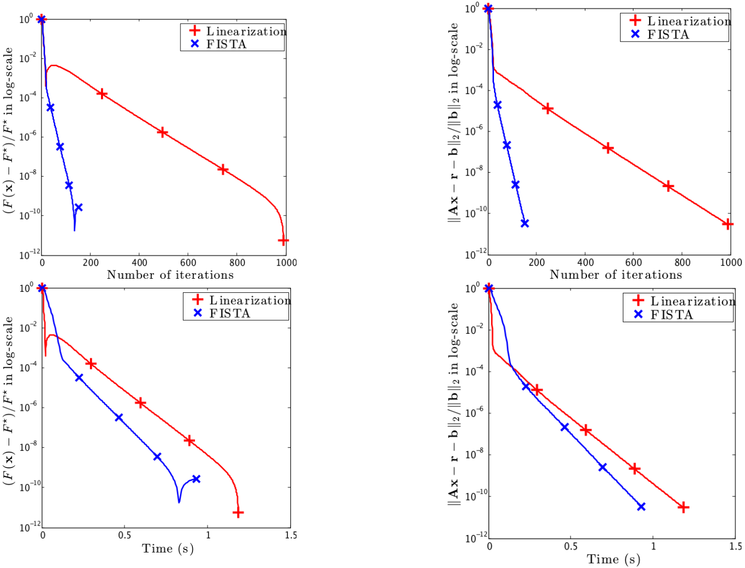


Numerical results - Low accuracy

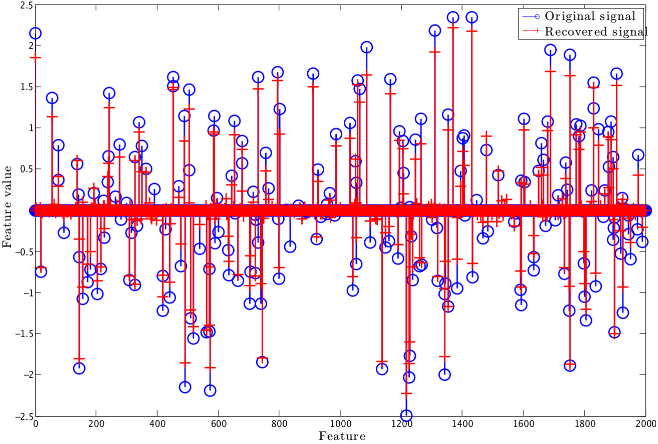
	Linearization	FISTA
Primal Obj. Value	14.241	14.241
Feas. Gap	0.3 $\times 10^{-10}$	0.29 $\times 10^{-10}$
Iterations	991	154
Inner Iterations	991	758
CPU time (s)	1.187	0.938
Calls A/A^T	992/991	913/758

- Parameters: $\rho = 0.1,\ \kappa = 0.01,\ \eta = 1.25$
- Input: $n = 750,\ p = 2000,\ k = 200,$ Noise $\sim \mathcal{N}(0, \sigma^2 \mathcal{I})$ with $\sigma = 10^{-3}$
- FISTA: max iterations 5, stop criteria 10^{-10} relative change, warm start
- Stopping criteria: $\|\mathbf{A}\mathbf{x}^k - \mathbf{r}^k - b\| \leq 10^{-10}\|\mathbf{b}\|$ and $\|\mathbf{x}^k - \mathbf{x}^{k-1}\| \leq 10^{-10}\|\mathbf{x}^k\|$

Convergence plots



Recovery error



- $\|\mathbf{x}^\star - \mathbf{x}^\natural\| / \|\mathbf{x}^\natural\|$
  - Linearization: 18.88%
  - FISTA: 18.88%
- $\|\mathbf{x}^\star_{\text{Lin}} - \mathbf{x}^\star_{\text{FISTA}}\| / \|\mathbf{x}^\natural\| = 0.43 \times 10^{-8}$

Alternating direction method of multipliers (ADMM)

The idea

When f_1 is not strongly convex, to overcome the drawback of ALM, by alternating solving (22).

Alternating direction method of multipliers (ADMM)

The idea

When f_1 is not strongly convex, to overcome the drawback of ALM, by alternating solving (22).

ADMM

Alternating direction method of multipliers (ADMM):

1. Choose $\lambda^0 \in \mathbb{R}^p$, $\mathbf{x}_2^0 \in \mathbb{R}^p$, $\gamma \geq 0$ and $\kappa > 0$.
2. For $k = 0, 1, \cdots$, perform:

$$\begin{cases} \mathbf{x}_1^{k+1} := \underset{\mathbf{x}_1 \in \mathcal{X}_1}{\mathrm{argmin}} \Big\{ f_1(\mathbf{x}_1) + \frac{\kappa}{2} \|\mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2^k - \mathbf{b} - \kappa^{-1}\mathbf{A}_1^T\lambda^k\|_2^2 + \frac{\gamma}{2}\|\mathbf{x}_1 - \mathbf{x}_1^k\|_2^2 \Big\}, \\ \mathbf{x}_2^{k+1} := \underset{\mathbf{x}_2 \in \mathcal{X}_2}{\mathrm{argmin}} \Big\{ f_2(\mathbf{x}_2) + \frac{\kappa}{2} \|\mathbf{A}_1\mathbf{x}_1^{k+1} + \mathbf{A}_2\mathbf{x}_2 - \mathbf{b} - \kappa^{-1}\mathbf{A}_2^T\lambda^k\|_2^2 \Big\}, \\ \lambda^{k+1} := \lambda^k + \kappa(\mathbf{A}_1\mathbf{x}_1^{k+1} + \mathbf{A}_2\mathbf{x}_2^{k+1} - \mathbf{b}). \end{cases}$$

In the original ADMM version, the proximal term $(\gamma/2)\|\mathbf{x}_1 - \mathbf{x}_1^k\|_2^2$ is neglected.

Enhancements

Update the parameter κ

- Constant step-size: We can fix $\kappa_k = \kappa > 0$.
- Increasing step-size: κ_k can be increased as $\kappa_{k+1} := \eta\kappa_k$, for $k \geq 0$ and $\eta > 1$.
- Adaptive step size: κ_k can be updated adaptively based on the primal and dual residuals (see [2]).

Enhancements

Update the parameter κ

- Constant step-size: We can fix $\kappa_k = \kappa > 0$.
- Increasing step-size: κ_k can be increased as $\kappa_{k+1} := \eta\kappa_k$, for $k \geq 0$ and $\eta > 1$.
- Adaptive step size: κ_k can be updated adaptively based on the primal and dual residuals (see [2]).

Preconditioned ADMM

- **Drawback:** When $\mathcal{X}_1$ and $\mathcal{X}_2$ are absent, f_1 and f_2 possess a tractable prox-operator, if $\mathbf{A}_1$ and $\mathbf{A}_2$ are not column orthogonal, then we can not exploit the proximal tractability of f_1 and f_2.
- **Overcome:** Linearize the quadratic terms and using the gradient step to approximate $\mathbf{x}_1^{k+1}$ and $\mathbf{x}_2^{k+1}$:

$$\begin{cases} \mathbf{g}_1^k & := \mathbf{x}_1^k - \alpha_k^1 \mathbf{A}_1^T(\mathbf{A}_1\mathbf{x}_1^k + \mathbf{A}_2\mathbf{x}_2^k - \mathbf{b}) & \text{(gradient step for } \mathbf{x}_1\text{)} \\ \mathbf{x}_1^{k+1} & := \mathrm{prox}_{\alpha_k^1\kappa^{-1}f_1}\left(\mathbf{g}_1^k + \kappa^{-1}\mathbf{A}_1^T\lambda^k\right) & \text{(proximal step for } \mathbf{x}_1\text{)} \\ \mathbf{g}_2^k & := \mathbf{x}_2^k - \alpha_k^2 \mathbf{A}_2^T(\mathbf{A}_1\mathbf{x}_1^{k+1} + \mathbf{A}_2\mathbf{x}_2^k - \mathbf{b}) & \text{(gradient step for } \mathbf{x}_2\text{)} \\ \mathbf{x}_2^{k+1} & := \mathrm{prox}_{\alpha_k^2\kappa^{-1}f_2}\left(\mathbf{g}_2^k + \kappa^{-1}\mathbf{A}_2^T\lambda^k\right) & \text{(proximal step for } \mathbf{x}_2\text{).} \end{cases}$$

where α_k^1 and α_k^2 can be chosen proportionally to $\|\mathbf{A}_1\|^2$ and $\|\mathbf{A}_2\|^2$, respectively.

Convergence of ADMM

Theorem (Convergence of ADMM [2])

Assume that f_1 and f_2 are proper, closed and convex and $\mathcal{L}$ has a saddle point $(\mathbf{x}^\star, \lambda^\star)$. For $\gamma = 0$, we have

- ***Residual convergence:*** $\{r_k\}$ *converges to zero, where*

$$r_k := \|\mathbf{A}_1\mathbf{x}_1^k + \mathbf{A}_2\mathbf{x}_2^k - \mathbf{b}\|_2.$$

- ***Objective convergence:*** $\{f(\mathbf{x}^k)\}$ *converges to* $f^\star$.
- ***Dual variable convergence:*** $\{\lambda^k\}$ *converges to* $\lambda^\star$.

Convergence of ADMM

Theorem (Convergence of ADMM [2])

Assume that f_1 and f_2 are proper, closed and convex and $\mathcal{L}$ has a saddle point $(\mathbf{x}^\star, \lambda^\star)$. For $\gamma = 0$, we have

- ***Residual convergence:*** *$\{r_k\}$ converges to zero, where*

$$r_k := \|\mathbf{A}_1\mathbf{x}_1^k + \mathbf{A}_2\mathbf{x}_2^k - \mathbf{b}\|_2.$$

- ***Objective convergence:*** *$\{f(\mathbf{x}^k)\}$ converges to $f^\star$.*
- ***Dual variable convergence:*** *$\{\lambda^k\}$ converges to $\lambda^\star$.*

Theorem (Convergence rate of ADMM [4])

Let $\{\mathbf{w}^k\}$ be the sequence generated by ADMM, where $\mathbf{w}^k := (\mathbf{x}^k, \lambda^k)$ and $\mathbf{w}^\star := (\mathbf{x}^\star, \lambda^\star)$. Let $\bar{\mathbf{w}}^k := (k+1)^{-1}\sum_{j=0}^k \mathbf{w}^j$. Then $\{\bar{\mathbf{w}}^k\}$ satisfies

$$f(\bar{\mathbf{x}}^k) - f(\mathbf{x}^\star) + (\bar{\mathbf{w}}^k - \mathbf{w}^\star)^T M(\mathbf{w}^\star) \le \frac{1}{2(k+1)}\|\mathbf{w}^0 - \mathbf{w}^\star\|_{\mathbf{H}}^2, \quad \forall k \ge 0,$$

where $M(\mathbf{w}) := \begin{bmatrix} -\mathbf{A}^T\lambda \\ \mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 - \mathbf{b} \end{bmatrix}$ *and* $\mathbf{H} := \mathbf{diag}(\sqrt{\gamma}\mathbb{I}, \kappa\mathbf{A}_2^T\mathbf{A}_2, \kappa^{-1}\mathbb{I})$.
Consequently, $\{\mathbf{w}^k\}$ converges to $\mathbf{w}^\star$ at $\mathcal{O}(1/k)$ rate.

Example 1: Robust principle component analysis (RPCA)

Robust PCA

$$\begin{aligned} \min_{\mathbf{L},\mathbf{S}} \quad & \|\mathrm{vec}(\mathbf{S})\|_1 + \rho\|\mathbf{L}\|_*, \\ \text{s.t.} \quad & \mathbf{S} + \mathbf{L} = \mathbf{M}. \end{aligned} \tag{29}$$

Here $\rho > 0$ is a weighted parameter between the sparse and low-rank terms.

Example 1: Robust principle component analysis (RPCA)

Robust PCA

$$\begin{aligned} \min_{\mathbf{L},\mathbf{S}} \quad & \|\mathrm{vec}(\mathbf{S})\|_1 + \rho\|\mathbf{L}\|_*, \\ \text{s.t.} \quad & \mathbf{S} + \mathbf{L} = \mathbf{M}. \end{aligned} \tag{29}$$

Here $\rho > 0$ is a weighted parameter between the sparse and low-rank terms.

Applying ADMM

The main steps of ADMM applying to (29) become:

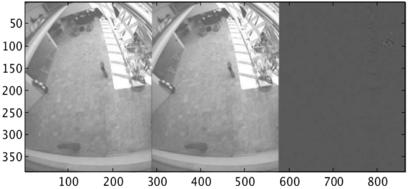
$$\begin{cases} \mathbf{S}^{k+1} & := \mathrm{prox}_{\kappa^{-1}\|\mathrm{vec}(\cdot)\|_1}\left(\mathbf{M} - \mathbf{L}^k + \kappa^{-1}\mathbf{W}^k\right), \\ \mathbf{L}^{k+1} & := \mathrm{prox}_{\beta\kappa^{-1}\|\cdot\|_*}\left(\mathbf{M} - \mathbf{S}^{k+1} + \kappa^{-1}\mathbf{W}^k\right), \\ \mathbf{W}^{k+1} & := \mathbf{W}^k + \kappa(\mathbf{S}^k + \mathbf{L}^k - \mathbf{M}). \end{cases}$$

These prox-operators are computed as

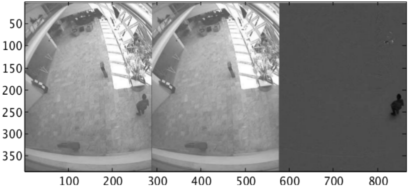
$$\begin{aligned} \mathrm{prox}_{\tau\|\mathrm{vec}(\cdot)\|_1}(\mathbf{S}) &= \mathrm{sign}(\mathbf{S}_1) \otimes \max\left\{|\mathbf{S}_1| - \tau, 0\right\}, \\ \mathrm{prox}_{\tau\|\cdot\|_*}(\mathbf{L}) &= \mathbf{U}\Sigma_\tau\mathbf{V}^T, \end{aligned}$$

where $\Sigma_\tau := \mathrm{sign}(\Sigma) \otimes \max\{|\Sigma| - \tau, 0\}$ and $\mathbf{U}\Sigma\mathbf{V}^T = \mathbf{L}$ is the SVD factorization of $\mathbf{L}$.

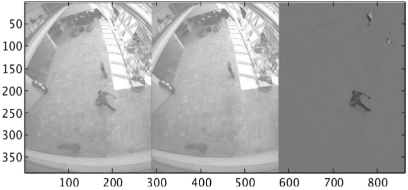
Video surveillance



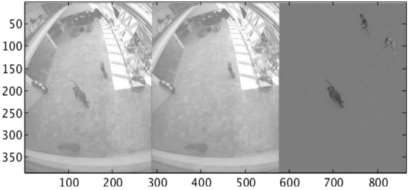
Frame 1



Frame 34



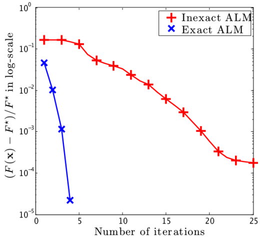
Frame 67

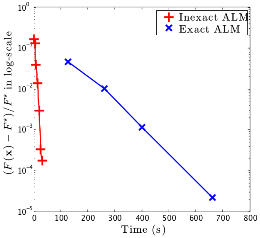


Frame 100

Unprocessed video from EC Funded CAVIAR project/IST 2001 37540, homepages.inf.ed.ac.uk/rbf/CAVIAR/.

Numerical test





	Exact ALM	Inexact ALM
Objective Value	553.5×10^3	553.6×10^3
Feas. Gap	0.33×10^{-5}	0.45×10^{-5}
$\\|\mathbf{L}\\|_*$	474.9×10^3	471.1×10^3
$\\|\text{vec}(\mathbf{S})\\|_1$	22.4616×10^6	23.556×10^6
Iterations	5	25
CPU time (s)	719.7	32.7
SVD Operations	644	25
Rank	1	1
Sparsity (%)	19.3	20.5

Algorithm

- Input
  - M is 110592×100: 100 frames of 288×384 pixels as columns
- Algorithm
  - $\rho = 0.35 \times 10^{-2}$ - tunnebale
  - Stopping criteria: $\|\mathbf{M} - \mathbf{L}^k - \mathbf{S}^k\| < 10^{-5}\|\mathbf{M}\|$

	Exact ADMM	Inexact ADMM
(tunneable)	$\kappa^1 = 0.5/\max\{\Sigma\}$	$\kappa^1 = 1.5/\max\{\Sigma\}$
(tunneable)	$\kappa^{k+1} = \kappa^k * 6$	$\kappa^{k+1} = \kappa^k * 1.5$
prox op.	Tolerance: $10^{-6}\|\mathbf{M}\|$	Iterations: 1

- Output
  - Numerical rounding $\Rightarrow$ threshold
  - $\mathbf{L}_{\text{output}} = \mathbf{U}\Sigma_{0.01\max\{\Sigma\}}\mathbf{V}^T$
  - $\mathbf{S}_{\text{output}} = \mathbf{S}_{0.01\max\{|\mathbf{S}|\}}$

Codes available at perception.csl.illinois.edu/matrix-rank/home.html

Example 2: Image deblurring

Image deblurring

The image deblurring presented previously can be written as:

$$\begin{array}{ll} \min\limits_{\mathbf{u}\in\mathbb{R}^{n\times p},\mathbf{v}} & \left\{(1/2)\|\mathbf{v}\|_F^2 + \rho\|\mathbf{u}\|_{\mathrm{TV}}\right\} \\ \text{s.t.} & \mathcal{A}(\mathbf{u}) - \mathbf{v} = \mathbf{b}. \end{array} \tag{30}$$

Example 2: Image deblurring

Image deblurring

The image deblurring presented previously can be written as:

$$\begin{array}{ll} \min\limits_{\mathbf{u}\in\mathbb{R}^{n\times p},\mathbf{v}} & \left\{(1/2)\|\mathbf{v}\|_F^2 + \rho\|\mathbf{u}\|_{\mathrm{TV}}\right\} \\ \text{s.t.} & \mathcal{A}(\mathbf{u}) - \mathbf{v} = \mathbf{b}. \end{array} \tag{30}$$

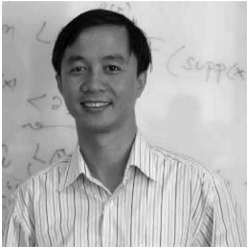
Applying ADMM

- We assume that $\mathcal{A}^*\mathcal{A} = \mathbb{I}$, where $\mathcal{A}^*$ is the adjoint operator of $\mathcal{A}$.
- The $\mathbf{v}$-step can be computed explicitly and the $\mathbf{u}$-step can be computed relying on the prox-operator of the TV-norm.
- The main steps of ADMM becomes

$$\left\{\begin{array}{ll} \mathbf{v}^{k+1} & := (\kappa+1)^{-1}\left(\lambda^k + \kappa(\mathcal{A}(\mathbf{u}^k) - \mathbf{b})\right), \\ \mathbf{u}^{k+1} & := \mathrm{prox}_{\rho\kappa^{-1}\|\cdot\|_{\mathrm{TV}}}\left(\mathcal{A}^*(\mathbf{b} + \mathbf{v}^{k+1} - \kappa^{-1}\lambda^k)\right), \\ \lambda^{k+1} & := \lambda^k + \kappa(\mathcal{A}(\mathbf{u}^{k+1}) - \mathbf{v}^{k+1} - \mathbf{b}). \end{array}\right.$$

Wrong regularization parameter

$$\rho = \pi^e$$

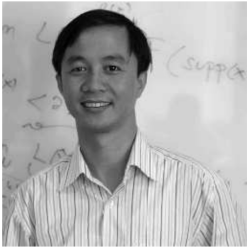


Original image

Blured image
$\mathrm{SNR} = 40\mathrm{dB}$

Wrong regularization parameter

$$\rho = \pi^e$$

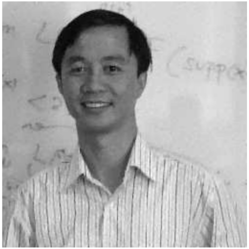


Original image

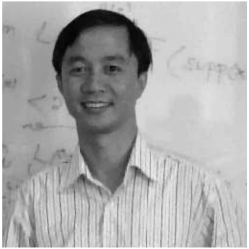
Blured image
$\mathrm{SNR} = 40\mathrm{dB}$

Recoverd image

Different values of regularization parameter



$\rho = 5 \times 10^{-3}$



$\rho = 1 \times 10^{-2}$

$\rho = 2.5 \times 10^{-2}$

Numerical results

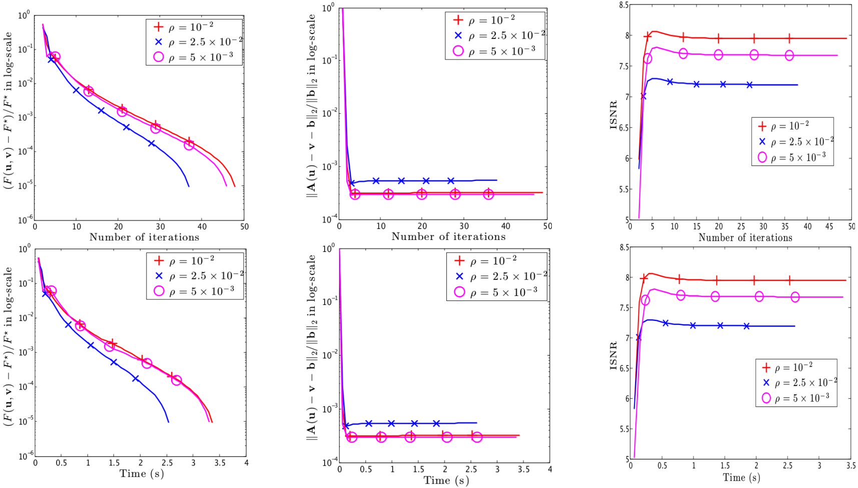
	$\rho = 5 \times 10^{-3}$	$\rho = 1 \times 10^{-2}$	$\rho = 2.5 \times 10^{-2}$
Objective Value	5317	7600	13344
MSE	24.1	22.8	27.2
ISNR (dB)	7.73	7.97	7.2
Feas. Gap ($\times 10^{-4}$)	3.01	3.38	5.45
Iterations	48	47	37
CPU time (s)	3.46	3.24	2.59
Linear Op. Calls*	99	97	77

- Algorithm
  - $\kappa = \rho/10$
  - Stopping criteria: $|F(\mathbf{u}^k, \mathbf{v}^k) - F(\mathbf{u}^{k-1}, \mathbf{v}^{k-1})| < 10^{-5} F(\mathbf{u}^k, \mathbf{v}^k)$
  - Maximum 5 iterations for TV prox-operator (with warmstart)
  - Input: $256\text{px} \times 256\text{px}$ image
- MSE(Mean Squared Error) $= \frac{\|\mathbf{u} - \mathbf{u}^\natural\|_2}{np}$
- ISNR(Improvement in Signal-to-Noise Ratio) $= \frac{\|\mathbf{b} - \mathbf{u}^\natural\|_2}{np\text{MSE}}$ [dB]

* number of applications of $\mathbf{A}$ and $\mathbf{A}^T$ operators

Convergence plots

Objective | Feasibility Gap | ISNR



$\text{ISNR}_0 = -20\text{dB}$

Summary

We have studied several methods for solving the following constrained convex problem:

$$f^\star := \min_{\mathbf{x}} \{ f(\mathbf{x}) \ : \ \mathbf{A}\mathbf{x} = \mathbf{b}, \ \mathbf{x} \in \mathcal{X} \}. \tag{1}$$

Under different assumptions, we have presented the following methods:

- Null-space, projected gradient and Frank-Wolf's methods.
- Dual subgradient and augmented Lagrangian methods
- Alternating minimization algorithm (AMA) and alternating direction methods of multipliers (ADMM).

Summary

We have studied several methods for solving the following constrained convex problem:

$$f^\star := \min_{\mathbf{x}} \{ f(\mathbf{x}) \ : \ \mathbf{A}\mathbf{x} = \mathbf{b}, \ \mathbf{x} \in \mathcal{X} \}. \quad (1)$$

Under different assumptions, we have presented the following methods:

- Null-space, projected gradient and Frank-Wolf's methods.
- Dual subgradient and augmented Lagrangian methods
- Alternating minimization algorithm (AMA) and alternating direction methods of multipliers (ADMM).

However, such methods still have limitations, few of them are listed below.

Methods	Limitations
Null-space method	require null-space representation (e.g., QR with $\mathcal{O}(n^2 p)$ complexity), destroy the original structure of f
Projected gradient	require tractability of the projection on $\mathcal{X}$, smooth f
Dual subgradient method	advantage for decomposable structure, but slow convergence rate $\mathcal{O}(1/\sqrt{k})$, sensitive with the choices of step-size
Augmented Lagrangian	non-separability of the quadratic term, high-computational cost for subproblems, no supporting theory for penalty parameter selection
AMA	only application for partly strongly convex objective, not using the tractable proximity of f due to linear operator, no supporting theory for penalty parameter selection
ADMM	not using the tractable proximity of f due to linear operator, no supporting theory for penalty parameter selection

Summary

We have studied several methods for solving the following constrained convex problem:

$$f^\star := \min_{\mathbf{x}}\{f(\mathbf{x}) \ : \ \mathbf{A}\mathbf{x} = \mathbf{b}, \ \mathbf{x} \in \mathcal{X}\}. \tag{1}$$

Under different assumptions, we have presented the following methods:

- Null-space, projected gradient and Frank-Wolf's methods.
- Dual subgradient and augmented Lagrangian methods
- Alternating minimization algorithm (AMA) and alternating direction methods of multipliers (ADMM).

However, such methods still have limitations, few of them are listed below.

Methods	Limitations
Null-space method	require null-space representation (e.g., QR with $\mathcal{O}(n^2p)$ complexity), destroy the original structure of f
Projected gradient	require tractability of the projection on $\mathcal{X}$, smooth f
Dual subgradient method	advantage for decomposable structure, but slow convergence rate $\mathcal{O}(1/\sqrt{k})$, sensitive with the choices of step-size
Augmented Lagrangian	non-separability of the quadratic term, high-computational cost for subproblems, no supporting theory for penalty parameter selection
AMA	only application for partly strongly convex objective, not using the tractable proximity of f due to linear operator, no supporting theory for penalty parameter selection
ADMM	not using the tractable proximity of f due to linear operator, no supporting theory for penalty parameter selection

In the next lecture, we will present other methods for solving (1) that either use different set of assumptions or overcome some of these limitations.

References

[1] N. Andreasson, A. Evgrafov, and M. Patriksson.
Introduction to Continuous Optimization: Foundations and Fundamental Algorithms.
Studentlitteratur AB, 2006.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein.
Distributed optimization and statistical learning via the alternating direction method of multipliers.
Foundations and Trends in Machine Learning, 3(1):1–122, 2011.

[3] T. Goldstein, B. ODonoghue, and S. Setzer.
Fast Alternating Direction Optimization Methods.
Tech. report., Department of Mathematics, University of California, Los Angeles, USA, May 2012.

[4] B.S. He and X.M. Yuan.
On the $O(1/n)$ convergence rate of the Douglas-Rachford alternating direction method.
SIAM J. Numer. Anal., 50:700–709, 2012.

[5] M. Jaggi.
Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization.
JMLR W&CP, 28(1):427–435, 2013.

References

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

[7] P. Tseng and D.P. Bertsekas.
Relaxation methods for problems with strictly convex cost and linear constraints.
Math. Oper. Research, 16(3):462–481, 1991.