Mathematics of Data: From Theory to Computation

Prof. Volkan Cevher
volkan.cevher@epfl.ch

Lecture 12: Constrained convex minimization II

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

EE-556 (Fall 2015)

License Information for Mathematics of Data Slides

Outline

- This class:
  1. Frank-Wolfe method
  2. Universal primal-dual gradient methods
  3. ADMM
- Next class
  1. Disciplined convex programming

Recommended reading material

- Martin Jaggi, *Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization* http://jmlr.org/proceedings/papers/v28/jaggi13-supp.pdf, 2013.
- Alp Yurtsever, Quoc Tran-Dinh and Volkan Cevher, *A universal primal-dual convex optimization framework* http://infoscience.epfl.ch/record/205073/files/PDUGA_MAIN_TEX.pdf, 2015.
- S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers* https://web.stanford.edu/~boyd/papers/pdf/admm_distr_stats.pdf, 2011.

Motivation

Motivation

- Evaluating the *proximal operator is costly* for many real world constrained optimization problems. This lecture covers the basics of the proximal-free numerical methods for constrained convex minimization, which use *cheaper Fenchel-type oracles* as a building block.

Swiss army knife of convex formulations

A **primal problem** prototype

$$f^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) : \mathbf{A}\mathbf{x} - \mathbf{b} \in \mathcal{K},\ \mathbf{x} \in \mathcal{X} \right\}, \tag{1}$$

- f is a proper, closed and convex function
- $\mathcal{X}$ and $\mathcal{K}$ are nonempty, closed convex sets
- $\mathbf{A} \in \mathbb{R}^{n\times p}$ and $\mathbf{b} \in \mathbb{R}^n$ are known
- An optimal solution $\mathbf{x}^\star$ to (1) satisfies $f(\mathbf{x}^\star) = f^\star$, $\mathbf{A}\mathbf{x}^\star = \mathbf{b}$ and $\mathbf{x}^\star \in \mathcal{X}$

Recall the prox-operator

Prox-operator helps us process nonsmooth terms "efficiently"

$$\mathrm{prox}_g^{\mathcal{X}}(\mathbf{x}) := \arg\min_{\mathbf{z}\in\mathcal{X}} \{g(\mathbf{z}) + (1/2)\|\mathbf{z}-\mathbf{x}\|^2\}.$$

Often efficient & has closed form expression:

- if $g(\mathbf{z}) = \|\mathbf{z}\|_1$ and $\mathcal{X} = \mathbb{R}^p$, then prox-operator $\Leftrightarrow$ soft-thresholding

Recall the prox-operator

Prox-operator helps us process nonsmooth terms "efficiently"

$$\mathrm{prox}_g^{\mathcal{X}}(\mathbf{x}) := \arg\min_{\mathbf{z}\in\mathcal{X}}\{g(\mathbf{z}) + (1/2)\|\mathbf{z}-\mathbf{x}\|^2\}.$$

Often efficient & has closed form expression:

- if $g(\mathbf{z}) = \|\mathbf{z}\|_1$ and $\mathcal{X} = \mathbb{R}^p$, then prox-operator $\Leftrightarrow$ soft-thresholding

Not all nonsmooth functions are proximal-friendly!

If $g(\mathbf{z}) = \|\mathbf{z}\|_\star$ (i.e., the nuclear norm of $\mathbf{z}$) and $\mathcal{X} = \mathbb{R}^p$, then

- prox-operator $\Leftrightarrow$ full singular value decomposition!
- **rules out all primal-dual proximal methods for our template**

Recall the prox-operator

Prox-operator helps us process nonsmooth terms "efficiently"

$$\mathrm{prox}_g^{\mathcal{X}}(\mathbf{x}) := \arg\min_{\mathbf{z}\in\mathcal{X}}\{g(\mathbf{z}) + (1/2)\|\mathbf{z}-\mathbf{x}\|^2\}.$$

Often efficient & has closed form expression:

- if $g(\mathbf{z}) = \|\mathbf{z}\|_1$ and $\mathcal{X} = \mathbb{R}^p$, then prox-operator $\Leftrightarrow$ soft-thresholding

Not all nonsmooth functions are proximal-friendly!

If $g(\mathbf{z}) = \|\mathbf{z}\|_\star$ (i.e., the nuclear norm of $\mathbf{z}$) and $\mathcal{X} = \mathbb{R}^p$, then

- prox-operator $\Leftrightarrow$ full singular value decomposition!
- **rules out all primal-dual proximal methods for our template**

Can we avoid the prox-operator for something cheaper as a building block?

Frank-Wolfe's method: Earliest example

Problem setting

$$\boxed{f^\star := \min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})} \qquad (2)$$

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).
- Note also that $\mathbf{A}\mathbf{x} - \mathbf{b} \in \mathcal{K}$ is missing from our prototype problem.

Frank-Wolfe's method (see [3] for a review)

Conditional gradient method (CGA)
1. Choose $\mathbf{x}^0 \in \mathcal{X}$. **2.** For $k = 0, 1, \ldots$ 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.

Frank-Wolfe's method: Earliest example

Problem setting

$$\boxed{f^\star := \min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})} \tag{2}$$

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).
- Note also that $\mathbf{A}\mathbf{x} - \mathbf{b} \in \mathcal{K}$ is missing from our prototype problem.

Frank-Wolfe's method (see [3] for a review)

Conditional gradient method (CGA)

1. Choose $\mathbf{x}^0 \in \mathcal{X}$.

2. For $k = 0, 1, \ldots$ 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.

When $\mathcal{X} := \{\mathbf{x} \in \mathbb{R}^{n\times p} : \|\mathbf{x}\|_\star \leq 1\}$, $(*)$ corresponds to rank-1 updates!

CGA is a special instance of dual averaging subgradient method

Problem setting

$$f^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) : \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \in \mathcal{X} \right\},$$

Assumptions

- $\mathcal{X}$ is nonempty, convex, closed and bounded.
- Note that this is a special case of our prototype where $\mathcal{K} = \{\mathbf{0}\}$.

Dual averaging subgradient method [6, 7]

Dual averaging subgradient method (DSM)
1. Choose $\mathbf{x}^0 = \mathbf{0}$.   **2.** For $k = 0, 1, \ldots$ perform: $\begin{cases} \mathbf{x}^{k+1} & := \mathbf{x}^k + \gamma_k \nabla d(\lambda^k), \\ \lambda^{k+1} & := \pi_{\beta_k}(\mathbf{x}^{k+1}), \end{cases}$ where $\gamma_k := 1$ and $\beta_{k+1} := \beta_k + \beta_0^2 \beta_k^{-1}$ for some $\beta_0 > 0$.

d is the dual function associated to the equality constraint and the mapping π_β is defined as:

$$\pi_\beta(\mathbf{x}) := \arg\min_\lambda \{\beta p(\lambda) - \langle \mathbf{x}, \lambda \rangle\}$$

where $p : \mathbb{R}^n \to \mathbb{R}_+$ is a proximity function, which is strongly convex.

Conjugation of functions

We need the definition of **Fenchel conjugation** and its basic properties to show the correspondence between CGA and DSM.

Definition

Let $\mathcal{Q}$ be a predefined Euclidean space and $\mathcal{Q}^*$ be its dual space. Given a proper, closed and convex function $f : \mathcal{Q} \to \mathbb{R} \cup \{+\infty\}$, the function $f^* : \mathcal{Q}^* \to \mathbb{R} \cup \{+\infty\}$ such that

$$f^*(\mathbf{y}) = \sup_{\mathbf{x} \in \mathrm{dom}(f)} \left\{ \mathbf{y}^T\mathbf{x} - f(\mathbf{x}) \right\}$$

is called the Fenchel conjugate (or conjugate) of f.

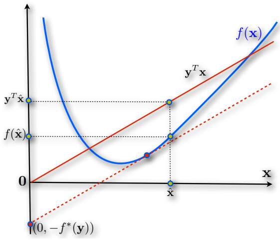


Figure: The conjugate function $f^*(\mathbf{y})$ is the maximum gap between the linear function $\mathbf{x}^T\mathbf{y}$ (red line) and $f(\mathbf{x})$.

- f^* is a convex and lower, semicontinuous function by construction (as the supremum of affine functions of $\mathbf{y}$).
- The conjugate of the conjugate of a convex function f is ... the same function f; i.e., $f^{**} = f$ for $f \in \mathcal{F}(\mathcal{Q})$.

Basic properties of the function and its conjugation

Property 1: Fenchel-Young inequality

Let $f : \mathcal{Q} \to \mathbb{R} \cup \{+\infty\}$ and $f^* : Q^* \to \mathbb{R} \cup \{+\infty\}$ be a function and its conjugation; here Q^* be the dual space of $\mathcal{Q}$. Then, the following inequality holds true:

$$f(\mathbf{x}) + f^*(\mathbf{y}) \geq \mathbf{x}^T\mathbf{y}, \quad \forall \mathbf{x} \in Q, \mathbf{y} \in Q^*.$$

Property 2: Subgradient property

Let $\mathbf{y} \in \partial f(\mathbf{x})$ for some $\mathbf{x} \in \mathrm{dom}(f)$. Then $\mathbf{y} \in \mathrm{dom}(f^*)$ and vise versa. Moreover, we have

$$\mathbf{u} \in \partial f(\mathbf{x}) \Leftrightarrow \mathbf{x} \in \partial f^*(\mathbf{u}).$$

Property 3: Duality of strong convexity and Lipschitz smoothness [5]

Let f be a convex and lower semi-continuos function. Then, strong convexity and Lipschitz gradients are dual in the following sense:

$$f \textit{ has Lipschitz continuos gradients } \Longleftrightarrow f^* \textit{ is strongly convex}$$

$$f \textit{ is strongly convex } \Longleftrightarrow f^* \textit{ has Lipschitz continuos gradients}$$

Frank-Wolfe's algorithm vs dual averaging subgradient method [10]

Consider the problem setting

$$f^\star := \min_{\mathbf{x},\mathbf{r}\in\mathbb{R}^p} \left\{ f(\mathbf{r}) : \mathbf{x} = \mathbf{r},\ \mathbf{x} \in \mathcal{X} \right\},$$

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).

- The dual function associated to the equality constraint and its gradient are:

$$\begin{cases} d(\lambda) & := \inf_{\mathbf{x}\in\mathcal{X}} \langle \lambda, \mathbf{x} \rangle - f^*(\lambda) \\ \nabla d(\lambda) & := \mathbf{x}^*(\lambda) - \nabla f^*(\lambda) \end{cases} \quad \textit{where} \quad \mathbf{x}^*(\lambda) \in \arg\min_{\mathbf{x}\in\mathcal{X}} \langle \mathbf{x}, \lambda \rangle.$$

- Let us define $\mathbf{x}^k := \nabla f^*(\lambda^k)$, then $\lambda^k := \nabla f(\mathbf{x}^k)$ by subgradient property. Hence;

$$\mathbf{x}^*(\lambda^k) = \arg\min_{\mathbf{x}\in\mathcal{X}} \langle \nabla f(\mathbf{x}^k), \mathbf{x} \rangle \quad \textit{and} \quad \nabla d(\lambda^k) = \mathbf{x}^*(\lambda^k) - \mathbf{x}^k.$$

- f^* is strongly convex by property 3. Choosing $p = f^*$, we get:

$$\pi_1(\mathbf{x}) := \arg\max_{\lambda\in\mathbb{R}^n} \{ \langle \mathbf{x}, \lambda \rangle - f^*(\lambda) \} = \nabla f^*(\lambda),$$

or equivalently $\lambda := \nabla f(\pi_1(\mathbf{x}))$ by the subgradient property.

$\Longrightarrow$ CGA is equivalent to DSM with $\beta = 1$, $\gamma_k = \frac{2}{k+2}$ and $p = f^*$.

Towards Fenchel-type operators

Generalized sharp operators [10]

We define the (generalized) sharp operator of a convex function g over $\mathcal{X}$ as follows:

$$\boxed{[\mathbf{x}]^{\sharp}_{\mathcal{X},g} := \operatorname*{argmin}_{\mathbf{z}\in\mathcal{X}} \left\{ g(\mathbf{z}) - \langle \mathbf{x}, \mathbf{z} \rangle \right\}.}$$

Important special cases:

1. If $g = 0$, then we obtain the so-called linear minimization oracle.
2. If $\mathcal{X} = \mathrm{dom}(g)$, then $[\mathbf{x}]^{\sharp}_{g} = \nabla g^*(\mathbf{x})$, where g^* is the Fenchel conjugate of g.

Towards Fenchel-type operators

Generalized sharp operators [10]

We define the (generalized) sharp operator of a convex function g over $\mathcal{X}$ as follows:

$$\boxed{[\mathbf{x}]^{\sharp}_{\mathcal{X},g} := \operatorname*{argmin}_{\mathbf{z}\in\mathcal{X}} \left\{ g(\mathbf{z}) - \langle \mathbf{x}, \mathbf{z} \rangle \right\}.}$$

Important special cases:

1. If $g = 0$, then we obtain the so-called linear minimization oracle.
2. If $\mathcal{X} = \mathrm{dom}(g)$, then $[\mathbf{x}]^{\sharp}_{g} = \nabla g^*(\mathbf{x})$, where g^* is the Fenchel conjugate of g.

Example (Nuclear norm)

Two examples with essentially the same computation:

	$g(\mathbf{x})$	$\mathcal{X}$	$[\mathbf{x}]^{\sharp}_{\mathcal{X},g}$
1.	0	$\{\mathbf{x} \in \mathbb{R}^{n\times p} : \|\mathbf{x}\|_\star \leq \kappa\}$	$\kappa\mathbf{u}\mathbf{v}^T$
2.	$\frac{1}{2}\|\mathbf{x}\|_\star^2$	$\mathbb{R}^{n\times p}$	$\|\mathbf{x}\|\mathbf{u}\mathbf{v}^T$

- $\|\cdot\|$ is the spectral norm
- $\mathbf{u}$ and $\mathbf{v}$ are the left and right principal singular vectors of $\mathbf{x}$

Revisiting Frank-Wolfe's method

Problem setting

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

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).
- Note that $\mathbf{A}\mathbf{x} - \mathbf{b} \in \mathcal{K}$ is missing from our prototype problem

Frank-Wolfe's method (see [3] for a review)

Conditional gradient method (CGA)

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

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

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

Conditional gradient method replaces the indicator function $\delta_{\mathcal{X}}$ with g:

$$\hat{\mathbf{x}}^k := \arg\min_{\mathbf{x}}\{g(\mathbf{x}) + \nabla f(\mathbf{x}^k)^T\mathbf{x}\} = [-\nabla f(\mathbf{x}^k)]^\sharp_g.$$

Revisiting Frank-Wolfe's method

Problem setting

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

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).

Next: Constrained problem $\underline{\mathbf{A}\mathbf{x} - \mathbf{b} \in \mathcal{K}}$ and $\underline{\text{nonsmooth } f(\mathbf{x})}$ with the sharp-operator

Frank-Wolfe's method (see [3] for a review)

Conditional gradient method (CGA)

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

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

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

Conditional gradient method replaces the indicator function $\delta_{\mathcal{X}}$ with g:

$$\hat{\mathbf{x}}^k := \arg\min_{\mathbf{x}}\{g(\mathbf{x}) + \nabla f(\mathbf{x}^k)^T\mathbf{x}\} = [-\nabla f(\mathbf{x}^k)]^\sharp_g.$$

Finding an optimal solution

A plausible algorithmic strategy for $\min_{\mathbf{x}\in\mathcal{X}}\{f(\mathbf{x}) : \mathbf{A}\mathbf{x} = \mathbf{b}\}$:

A natural minimax formulation:

$$(\mathbf{x}^\star, \lambda^\star) \in \arg\max_{\lambda} \min_{\mathbf{x}\in\mathcal{X}} \{\mathcal{L}(\mathbf{x}, \lambda) := f(\mathbf{x}) + \langle \lambda, \mathbf{A}\mathbf{x} - \mathbf{b}\rangle\}.$$

Lagrangian subproblem: $\mathbf{x}^*(\lambda) \in \arg\min_{\mathbf{x}\in\mathcal{X}} \mathcal{L}(\mathbf{x}, \lambda)$

Dual problem: $\lambda^\star \in \arg\max_{\lambda} \{d(\lambda) := \mathcal{L}(\mathbf{x}^*(\lambda), \lambda)\}$

- λ is called the Lagrange multiplier.
- The function $d(\lambda)$ is called the dual function, and it is concave!
- The optimal dual objective value is $d^\star = d(\lambda^\star)$.

Our strategy $\Rightarrow$ Make progress on the dual and obtain the primal solution

Finding an optimal solution

A plausible algorithmic strategy for $\min_{\mathbf{x}\in\mathcal{X}}\{f(\mathbf{x}) : \mathbf{A}\mathbf{x} = \mathbf{b}\}$:

A natural minimax formulation:

$$(\mathbf{x}^\star, \lambda^\star) \in \arg\max_{\lambda} \min_{\mathbf{x}\in\mathcal{X}} \{\mathcal{L}(\mathbf{x}, \lambda) := f(\mathbf{x}) + \langle \lambda, \mathbf{A}\mathbf{x} - \mathbf{b}\rangle\}.$$

Lagrangian subproblem: $\mathbf{x}^*(\lambda) \in \arg\min_{\mathbf{x}\in\mathcal{X}} \mathcal{L}(\mathbf{x}, \lambda)$

Dual problem: $\lambda^\star \in \arg\max_{\lambda} \{d(\lambda) := \mathcal{L}(\mathbf{x}^*(\lambda), \lambda)\}$

- λ is called the Lagrange multiplier.
- The function $d(\lambda)$ is called the dual function, and it is concave!
- The optimal dual objective value is $d^\star = d(\lambda^\star)$.

Our strategy $\Rightarrow$ Make progress on the dual and obtain the primal solution

Challenges for the plausible strategy above

1. Establishing its correctness
2. Computational efficiency of finding an $\bar{\epsilon}$-approximate optimal dual solution $\lambda^\star_{\bar{\epsilon}}$
3. Mapping $\lambda^\star_{\bar{\epsilon}} \to \mathbf{x}^\star_{\bar{\epsilon}}$

Finding an optimal solution

A plausible algorithmic strategy for $\min_{\mathbf{x}\in\mathcal{X}}\{f(\mathbf{x}) : \mathbf{A}\mathbf{x} = \mathbf{b}\}$:

A natural minimax formulation:

$$(\mathbf{x}^\star, \lambda^\star) \in \arg\max_{\lambda} \min_{\mathbf{x}\in\mathcal{X}} \{\mathcal{L}(\mathbf{x}, \lambda) := f(\mathbf{x}) + \langle \lambda, \mathbf{A}\mathbf{x} - \mathbf{b}\rangle\}.$$

Lagrangian subproblem: $\mathbf{x}^*(\lambda) \in \arg\min_{\mathbf{x}\in\mathcal{X}} \mathcal{L}(\mathbf{x}, \lambda)$

Dual problem: $\lambda^\star \in \arg\max_\lambda \{d(\lambda) := \mathcal{L}(\mathbf{x}^*(\lambda), \lambda)\}$

- λ is called the Lagrange multiplier.
- The function $d(\lambda)$ is called the dual function, and it is concave!
- The optimal dual objective value is $d^\star = d(\lambda^\star)$.

Our strategy $\Rightarrow$ Make progress on the dual and obtain the primal solution

Challenges for the plausible strategy above

1. Establishing its correctness: Assume $f^\star > -\infty$ and Slater's condition for $f^\star = d^\star$
2. Computational efficiency of finding an $\bar{\epsilon}$-approximate optimal dual solution $\lambda^\star_{\bar{\epsilon}}$
3. Mapping $\lambda^\star_{\bar{\epsilon}} \to \mathbf{x}^\star_{\epsilon}$

Efficiency considerations for the dual problem

Nonsmooth

Assumption: Bounded subgradients, i.e.,

$$\|\mathbf{v}\|_2 \leq G, \quad \forall \mathbf{v} \in \partial d(\lambda),\ \lambda \in \mathbb{R}^n.$$

Method: Subgradient method with worst case complexity $\mathcal{O}\left(\frac{1}{\epsilon^2}\right)$.

Efficiency considerations for the dual problem

Nonsmooth

Assumption: Bounded subgradients, i.e.,

$$\|\mathbf{v}\|_2 \leq G, \quad \forall \mathbf{v} \in \partial d(\lambda),\ \lambda \in \mathbb{R}^n.$$

Method: Subgradient method with worst case complexity $\mathcal{O}\left(\frac{1}{\epsilon^2}\right)$.

Lipschitz smoothness

Assumption: Lipschitz continuous gradients, i.e.,

$$\|\nabla d(\lambda) - \nabla d(\eta)\|_2 \leq L\|\lambda - \eta\|_2, \quad \forall \lambda,\ \eta \in \mathbb{R}^n.$$

Method: Accelerated gradient method with worst case complexity $\mathcal{O}\left(\frac{1}{\sqrt{\epsilon}}\right)$.

Efficiency considerations for the dual problem

Nonsmooth

Assumption: Bounded subgradients, i.e.,

$$\|\mathbf{v}\|_2 \leq G, \quad \forall \mathbf{v} \in \partial d(\lambda),\ \lambda \in \mathbb{R}^n.$$

Method: Subgradient method with worst case complexity $\mathcal{O}\left(\frac{1}{\epsilon^2}\right)$.

Hölder smoothness

Assumption: Hölder continuous gradient for some $\nu \in [0, 1]$, i.e.,

$$M_\nu(d) := \sup_{\lambda \neq \eta} \frac{\|\nabla d(\lambda) - \nabla d(\eta)\|_2}{\|\lambda - \eta\|_2^\nu}, \quad M_d^* := \inf_{0 \leq \nu \leq 1} M_\nu(d) < +\infty.$$

Method: Universal gradient method [8] with worst case complexity in the sequel

Lipschitz smoothness

Assumption: Lipschitz continuous gradients, i.e.,

$$\|\nabla d(\lambda) - \nabla d(\eta)\|_2 \leq L\|\lambda - \eta\|_2, \quad \forall \lambda,\ \eta \in \mathbb{R}^n.$$

Method: Accelerated gradient method with worst case complexity $\mathcal{O}\left(\frac{1}{\sqrt{\epsilon}}\right)$.

Brief detour: Exploring the smoothness in depth

Consider the following unconstrained setup in the sequel

$$\min_{\mathbf{x}\in\mathbb{R}^p} g(\mathbf{x})$$

Definition (Hölder continuity [4])

g is ν-Hölder continuous ($\nu \in [0,1]$) with Hölder constant $M_\nu < \infty$ when

$$\boxed{\|\nabla g(\mathbf{x}) - \nabla g(\mathbf{y})\|_2 \leq M_\nu \|\mathbf{x} - \mathbf{y}\|_2^\nu}$$

where, with some abuse of notation, $\nabla g(\lambda)$ is a (sub)gradient of g.

Brief detour: Exploring the smoothness in depth

Consider the following unconstrained setup in the sequel

$$\min_{\mathbf{x}\in\mathbb{R}^p} g(\mathbf{x})$$

Definition (Hölder continuity [4])

g is ν-Hölder continuous ($\nu \in [0,1]$) with Hölder constant $M_\nu < \infty$ when

$$\boxed{\|\nabla g(\mathbf{x}) - \nabla g(\mathbf{y})\|_2 \leq M_\nu \|\mathbf{x}-\mathbf{y}\|_2^\nu}$$

where, with some abuse of notation, $\nabla g(\lambda)$ is a (sub)gradient of g.

Highlights

1. $\nu = 0$ is the bounded subgradient assumption.
2. $\nu = 1$ is the Lipschitz continuous gradients case where $L = M_\nu$.
3. Iteration lowerbound for the Hölder class: $\mathcal{O}\left(\left(\frac{M_\nu\|\mathbf{x}^0-\mathbf{x}^\star\|^{1+\nu}}{\epsilon}\right)^{\frac{2}{1+3\nu}}\right)$.

Nesterov's universal gradient methods

The Hölder continuity assumption: The challenge

Hölder continuous (sub)gradients ensures the following surrogate for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$:

$$g(\mathbf{y}) \leq g(\mathbf{x}) + \langle \nabla g(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{M_\nu}{1+\nu} \|\mathbf{x} - \mathbf{y}\|^{1+\nu} \tag{3}$$

In practice, smoothness parameters ν **and** M_ν **are not known**.

Nesterov's universal gradient methods

The Hölder continuity assumption: The challenge

Hölder continuous (sub)gradients ensures the following surrogate for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$:

$$g(\mathbf{y}) \leq g(\mathbf{x}) + \langle \nabla g(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{M_\nu}{1+\nu} \|\mathbf{x} - \mathbf{y}\|^{1+\nu} \tag{3}$$

In practice, smoothness parameters ν **and** M_ν **are not known**.

Nesterov's solution: The basic idea [8]

Suppose that g satisfies (3). Then, for any $\delta > 0$ and

$$M \geq \left[\frac{1-\nu}{1+\nu} \cdot \frac{1}{\delta} \right]^{\frac{1-\nu}{1+\nu}} M_\nu^{\frac{2}{1+\nu}}$$

we can use the following basic inexact majorization bound

$$g(\mathbf{y}) \leq g(\mathbf{x}) + \langle \nabla g(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{M}{2} \|\mathbf{x} - \mathbf{y}\|^2 + \frac{\delta}{2}.$$

Nesterov's universal gradient methods

Universal primal gradient method (PGM)[1]

1. Choose $\mathbf{x}^0 \in \mathcal{X}$, $M_{-1} > 0$ and accuracy $\epsilon > 0$.
2. For $k = 0, 1, \ldots$ perform:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - M_k^{-1} \nabla g(\mathbf{x}^k)$$

using line-search to find $M_k \geq 0.5 M_{k-1}$ that satisfies:

$$g(\mathbf{x}^{k+1}) \leq g(\mathbf{x}^k) + \langle \nabla g(\mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x}^k \rangle + \frac{M_k}{2} \|\mathbf{x}^k - \mathbf{x}^{k+1}\|^2 + \frac{\epsilon}{2}$$

Nesterov's universal gradient method [8]

- Adapt to the unknown ν via an line-search strategy
- Universal since they ensure the best possible rate of convergence for each ν

[1] PGM in [8] uses the Bregman / prox setup.

Nesterov's universal gradient methods

Universal primal gradient method (PGM)[1]
1. Choose $\mathbf{x}^0 \in \mathcal{X}$, $M_{-1} > 0$ and accuracy $\epsilon > 0$. **2.** For $k = 0, 1, \ldots$ perform: $$\mathbf{x}^{k+1} = \mathbf{x}^k - M_k^{-1}\nabla g(\mathbf{x}^k)$$ using line-search to find $M_k \geq 0.5M_{k-1}$ that satisfies: $$g(\mathbf{x}^{k+1}) \leq g(\mathbf{x}^k) + \langle \nabla g(\mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x}^k \rangle + \frac{M_k}{2}\|\mathbf{x}^k - \mathbf{x}^{k+1}\|^2 + \frac{\epsilon}{2}$$

Nesterov's universal gradient method [8]

- Adapt to the unknown ν via an line-search strategy
- Universal since they ensure the best possible rate of convergence for each ν

Yes, there is an accelerated version [8].

[1] PGM in [8] uses the Bregman / prox setup.

Nesterov's universal gradient methods

Universal primal gradient method (PGM)[1]
1. Choose $\mathbf{x}^0 \in \mathcal{X}$, $M_{-1} > 0$ and accuracy $\epsilon > 0$. **2.** For $k = 0, 1, \ldots$ perform: $$\mathbf{x}^{k+1} = \mathbf{x}^k - M_k^{-1}\nabla g(\mathbf{x}^k)$$ using line-search to find $M_k \geq 0.5 M_{k-1}$ that satisfies: $$g(\mathbf{x}^{k+1}) \leq g(\mathbf{x}^k) + \langle \nabla g(\mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x}^k \rangle + \frac{M_k}{2}\|\mathbf{x}^k - \mathbf{x}^{k+1}\|^2 + \frac{\epsilon}{2}$$

Nesterov's universal gradient method [8]

- Adapt to the unknown ν via an line-search strategy
- Universal since they ensure the best possible rate of convergence for each ν

Yes, there is an accelerated version [8].

New: Our FISTA variant.

[1] PGM in [8] uses the Bregman / prox setup.

Our universal primal-dual gradient methods: The dual steps

$$\boxed{[\mathbf{x}]^{\sharp}_{\mathcal{X},g} := \arg\min_{\mathbf{z}\in\mathcal{X}} \{g(\mathbf{z}) - \langle \mathbf{x}, \mathbf{z}\rangle\}}$$

Dual steps: The level of inexactness

$$\boxed{\mathbf{x}^*(\lambda^k) := \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{f(\mathbf{x}) + \langle \mathbf{A}^T\lambda^k, \mathbf{x}\rangle\right\} \equiv \left[-\mathbf{A}^T\lambda^k\right]^{\sharp}_f}$$

- **(UniPDGrad)** requires 2 linesearch steps on the average with ϵ:

$$\lambda^{k+1} := \lambda^k + \frac{1}{M_k}\nabla d(\lambda^k) = \lambda_k + \frac{1}{M_k}\left(\mathbf{A}\mathbf{x}^*(\lambda^k) - \mathbf{b}\right).$$

- **(AccUniPDGrad)** requires 1 linesearch step on the average with ϵ/t_k:

$$\begin{cases} t_k & := 0.5\left(1 + \sqrt{1 + 4t_{k-1}^2}\right) \\ \hat{\lambda}^k & := \lambda^k + \frac{t_{k-1}-1}{t_k}\left(\lambda^k - \hat{\lambda}^{k-1}\right) \\ \lambda^{k+1} & := \hat{\lambda}^k + \frac{1}{M_k}\left(\mathbf{A}\mathbf{x}^*(\hat{\lambda}^k) - \mathbf{b}\right). \end{cases}$$

Our universal primal-dual gradient methods: The primal steps

Primal steps: Characterized by weighted averaging

$$\boxed{\mathbf{x}^*(\lambda^k) := \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) + \langle \mathbf{A}^T\lambda^k, \mathbf{x}\rangle \right\} \equiv \left[-\mathbf{A}^T\lambda^k\right]_f^\sharp}$$

(UniPDGrad): $$\bar{\mathbf{x}}^k := \left(\sum_{i=0}^k \frac{1}{M_i}\right)^{-1} \sum_{i=0}^k \frac{1}{M_i}\mathbf{x}^*(\lambda^i).$$

(AccUniPDGrad): $$\bar{\mathbf{x}}^k := \left(\sum_{i=0}^k \frac{t_i}{M_i}\right)^{-1} \sum_{i=0}^k \frac{t_i}{M_i}\mathbf{x}^*(\lambda^i).$$

Summary of the algorithms and convergence guarantees - I

Universal primal-dual gradient method (UniPDGrad)
Initialization: Choose $\lambda^0 \in \mathbb{R}^n$ and $\epsilon > 0$. Estimate a value $M_{-1} < 2M_\epsilon$.
Iteration: For $k = 0, 1, \ldots$ perform:
1. *Primal step:* $\mathbf{x}^*(\lambda^k) \ = [-\mathbf{A}^T\lambda^k]^\sharp_f$
2. *Dual gradient:* $\nabla d(\lambda^k) = \mathbf{A}^T\mathbf{x}^*(\lambda^k) - \mathbf{b}$
3. Line-search: Find $M_k \in [0.5M_{k-1}, 2M_\epsilon]$ from line-search condition and: $\lambda^{k+1} = \lambda^k + M_k^{-1}\nabla d(\lambda^k)$
4. *Primal averaging:* $\bar{\mathbf{x}}^k := S_k^{-1}\sum_{j=0}^k M_j^{-1}\mathbf{x}^*(\lambda^j)$ where $S_k = \sum_{j=0}^k M_j^{-1}$.

Theorem [10]

$\bar{\mathbf{x}}^k$ obtained by **UniPDGrad** satisfy:

$$\begin{cases} -\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|\|\lambda^\star\| \leq & f(\bar{\mathbf{x}}^k) - f^\star & \leq \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|\|\lambda^0\| + \frac{\epsilon}{2}, \\ & \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| & \leq \frac{4M_\epsilon\|\lambda^0 - \lambda^\star\|}{k+1} + \sqrt{\frac{2M_\epsilon\epsilon}{k+1}}. \end{cases}$$

Summary of the algorithms and convergence guarantees - II

Accelerated universal primal-dual gradient method (AccUniPDGrad)
Initialization: Choose $\lambda^0 \in \mathbb{R}^n$, $\epsilon > 0$. Set $t_0 = 1$. Estimate a value $M_{-1} < 2M_\epsilon$.
Iteration: For $k = 0, 1, \ldots$ perform:
1. *Primal step:* $\mathbf{x}^*(\hat{\lambda}^k) \ = [-\mathbf{A}^T\hat{\lambda}^k]^\sharp_f$,
2. *Dual gradient:* $\nabla d(\hat{\lambda}^k) = \mathbf{A}^T\mathbf{x}^*(\hat{\lambda}^k) - \mathbf{b}$,
3. *Line-search:* Find $M_k \in [M_{k-1}, 2M_\epsilon]$ from line-search condition and: $\lambda^{k+1} = \hat{\lambda}^k + M_k^{-1}\nabla d(\hat{\lambda}^k)$,
4. $t_{k+1} = 0.5[1 + \sqrt{1 + 4t_k^2}]$,
5. $\hat{\lambda}_{k+1} = \lambda_{k+1} + \frac{t_k - 1}{t_{k+1}}(\lambda_{k+1} - \lambda_k)$,
6. *Primal averaging:* $\bar{\mathbf{x}}^k := S_k^{-1}\sum_{j=0}^k t_j M_j^{-1}\mathbf{x}^*(\lambda^j)$ where $S_k = \sum_{j=0}^k t_j M_j^{-1}$.

Theorem [10]

$\bar{\mathbf{x}}^k$ and λ^k obtained by **AccUniProx** satisfy:

$$\begin{cases} -\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|\|\lambda^\star\| \leq & f(\bar{\mathbf{x}}^k) - f^\star & \leq \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\|\|\lambda^0\| + \frac{\epsilon}{2}, \\ & \|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| & \leq \frac{16M_\epsilon\|\lambda^0 - \lambda^\star\|}{(k+2)^{\frac{1+3\nu}{1+\nu}}} + \sqrt{\frac{8M_\epsilon\epsilon}{(k+2)^{\frac{1+3\nu}{1+\nu}}}}. \end{cases}$$

Number of iterations to reach ϵ: Optimality

The worst-case iteration complexity [10]

To achieve $\bar{\mathbf{x}}^k$ such that $|f(\bar{\mathbf{x}}^k) - f^\star| \leq \epsilon$ and $\|\mathbf{A}\bar{\mathbf{x}}^k - \mathbf{b}\| \leq \epsilon$ is:

$$\begin{cases} \text{For } \textbf{(UniPDGrad):} & \mathcal{O}\left(D_{\Lambda^\star}^2 \inf_{0\leq\nu\leq 1} \left(\frac{M_\nu}{\epsilon}\right)^{\frac{2}{1+\nu}} \right), \quad \text{optimal for } \nu = 0 \\ \text{For } \textbf{(AccUniPDGrad):} & \mathcal{O}\left(D_{\Lambda^\star}^{\frac{2+2\nu}{1+3\nu}} \inf_{0\leq\nu\leq 1} \left(\frac{M_\nu}{\epsilon}\right)^{\frac{2}{1+3\nu}} \right), \quad \text{optimal for } \nu \in [0, 1] \end{cases}$$

where $D_{\Lambda^\star} := \|\lambda^0 - \lambda^\star\|$.

Scalability example: Quantum tomography with Pauli operators - I

Problem formulation

Let $\mathbf{X}^\natural \in \mathcal{S}_+^p$ be a density matrix which characterizes a $q-$qubit quantum system, where $p = 2^q$. Using Pauli operators $\mathcal{A}$ [2], we can deduce the state from $\mathbf{b} = \mathcal{A}(\mathbf{X}) \in \mathcal{C}^n$ based on the following convex optimization formulation:

$$\varphi^\star := \min_{\mathbf{X} \in \mathcal{S}_+^p} \left\{ \frac{1}{2} \|\mathcal{A}(\mathbf{X}) - \mathbf{b}\|_2^2 : \mathrm{tr}(\mathbf{X}) = 1 \right\}. \tag{4}$$

The recovery is also robust to noise.

Scalability example: Quantum tomography with Pauli operators - I

Problem formulation

Let $\mathbf{X}^\natural \in \mathcal{S}_+^p$ be a density matrix which characterizes a $q-$qubit quantum system, where $p = 2^q$. Using Pauli operators $\mathcal{A}$ [2], we can deduce the state from $\mathbf{b} = \mathcal{A}(\mathbf{X}) \in \mathcal{C}^n$ based on the following convex optimization formulation:

$$\varphi^\star := \min_{\mathbf{X} \in \mathcal{S}_+^p} \left\{ \frac{1}{2} \|\mathcal{A}(\mathbf{X}) - \mathbf{b}\|_2^2 : \mathrm{tr}(\mathbf{X}) = 1 \right\}. \tag{4}$$

The recovery is also robust to noise.
Perfect scalability test: tuning free constraint + Lipschitz continuous gradient

Setup

Synthetic random pure quantum state (e.g., rank-1 $\mathbf{X}^\natural$) with:

- $q = 14$ qubits, that corresponds to $2^{28} = 268'435'456$ dimensional problem.
- $n := 2p\log(p) = 138'099$ number of Pauli measurements.
- Input parameters $\lambda^0 = \mathbf{0}^n$ and $\epsilon = 2 \cdot 10^{-4}$.

Scalability example: Quantum tomography with Pauli operators - II

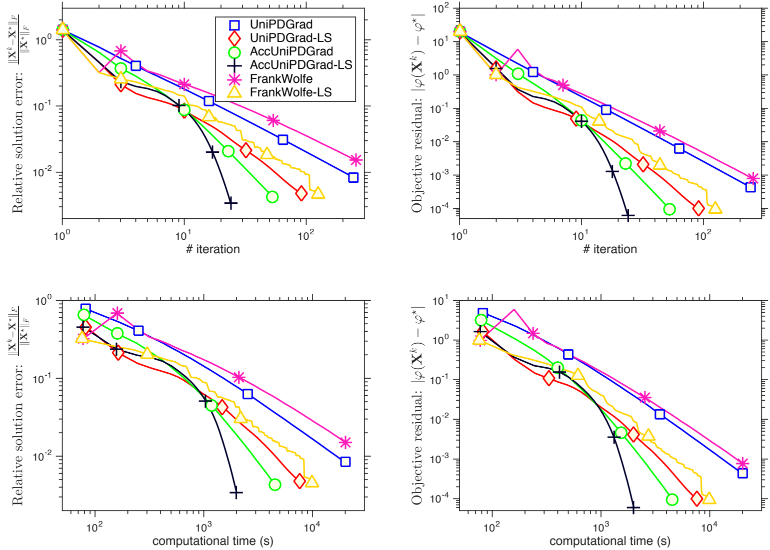


Figure: The performance of (Acc)UniPDGrad and Frank-Wolfe algorithms for (4).

Scalability example: Phase retrieval with matrix lifting - I

Phase retrieval

Phase retrieval problem aims to recover a signal $\mathbf{x}^\natural \in \mathcal{C}^p$ from n phaseless linear measurements where $\mathbf{a}_i \in \mathcal{C}^p$ are known vectors:

$$b_i = \left|\langle \mathbf{a}_i, \mathbf{x}^\natural \rangle\right|^2.$$

Scalability example: Phase retrieval with matrix lifting - I

Phase retrieval

Phase retrieval problem aims to recover a signal $\mathbf{x}^\natural \in \mathcal{C}^p$ from n phaseless linear measurements where $\mathbf{a}_i \in \mathcal{C}^p$ are known vectors:

$$b_i = \left|\langle \mathbf{a}_i, \mathbf{x}^\natural \rangle\right|^2.$$

Problem in the lifted dimensions [1]

We can equivalently express b_i as:

$$b_i = \text{trace}\left(\mathbf{a}_i \mathbf{X}^\natural \mathbf{a}_i^H\right), \quad \text{where } \mathbf{X}^\natural = \mathbf{x}^\natural (\mathbf{x}^\natural)^H.$$

This leads to the following linear observation model of the **lifted matrix** $\mathbf{X}^\natural$:

$$\mathbf{b} = \mathcal{A}(\mathbf{X}^\natural), \quad \text{where } \mathcal{A}(\mathbf{X}) = \text{diag}\left(\mathbf{A}\mathbf{X}\mathbf{A}^H\right) \text{ and } \mathcal{A}^H(\lambda) = \mathbf{A}^H \mathbf{D}(\lambda)\mathbf{A}.$$

Scalability example: Phase retrieval with matrix lifting - II

Problem formulation

$$f^\star := \min_{\mathbf{X} \in \mathcal{S}_+^{p^2}} \left\{ \frac{1}{2} \|\mathcal{A}(\mathbf{X}) - \mathbf{b}\|_2^2 : \|\mathbf{X}\|_* \leq \kappa \right\}. \quad (5)$$

Setup [9]

Real images of different size as input vector $\mathbf{x}^\natural$:

- EPFL campus 800×1280 pixels, $p \approx 10^6$, lifted dimension $p^2 \approx 10^{12}$.
- Milky Way 1080×1920 pixels, $p \approx 2 \cdot 10^6$, lifted dimension $p^2 \approx 4 \cdot 10^{12}$.
- We measure the magnitude of the diffraction pattern of the signal $\mathbf{x}^\natural \in \mathbb{R}^p$ modulated by 20 different random waveform $\mathbf{d}_l \in \mathbb{C}^p$, $1 \leq l \leq 20$:

$$(b_l)_i = \left| \sum_{i=1}^{p} x_i^\natural (d_l)_i^* \exp(-j2\pi ki/p) \right|.$$

- Input parameters $\lambda^0 = \mathbf{0}^n$, $\epsilon = 2 \cdot 10^{-2}$ and $\kappa = \text{mean}(\mathbf{b})$.

Scalability example: Phase retrieval with matrix lifting - III

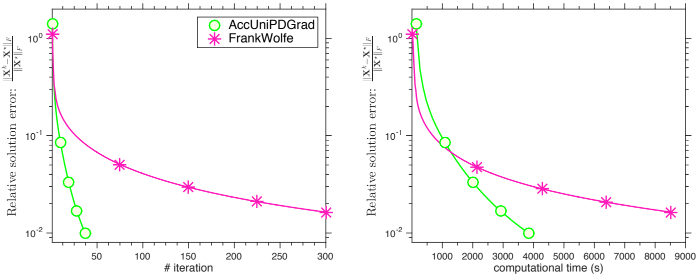


Figure: The performance of (Acc)UniPDGrad and Frank-Wolfe algorithms for (5).

Scalability example: Phase retrieval with matrix lifting - IV

Figure: EPFL campus 800×1280 estimate after 37 iterations of AccUniPDGrad.

Scalability example: Phase retrieval with matrix lifting - V

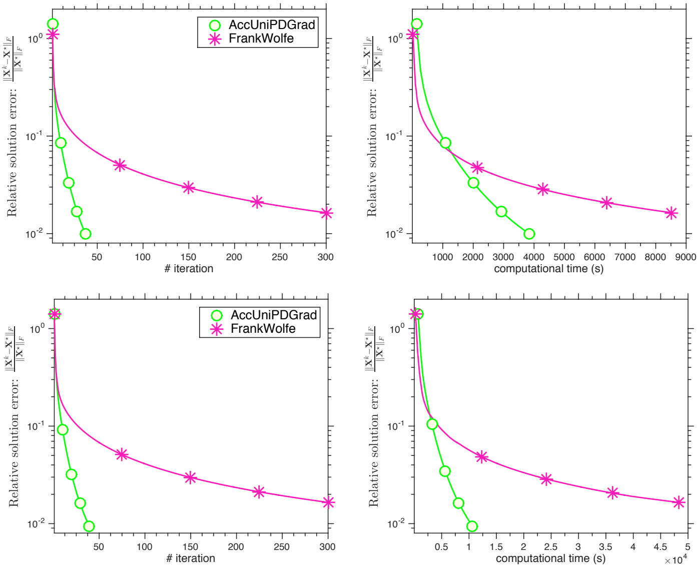


Figure: The performance of (Acc)UniPDGrad and Frank-Wolfe algorithms for (5).

Scalability example: Phase retrieval with matrix lifting - VI

Figure: Milky Way 1080×1920 estimate after 39 iterations of AccUniPDGrad.

Flexibility example: Matrix completion with MovieLens dataset - I

Problem formulation

Let $\Omega \subseteq \{1, \cdots, p\} \times \{1, \cdots, q\}$ be a subset of indexes and $\mathbf{M}_\Omega = (\mathbf{M}_{ij})_{(i,j)\in\Omega}$ be the observed entries of a missed matrix $\mathbf{M}$. $\mathcal{P}_\Omega$ is the projection on the subset Ω.

$$f^\star := \min_{\mathbf{X}\in\mathbb{R}^{p\times q}} \left\{ \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{X}) - \mathbf{M}_\Omega\|_2^2 : \|\mathbf{X}\|_\star \leq \kappa \right\} \tag{6}$$

We can also solve another robust version against outliers:

$$f^\star := \min_{\mathbf{X}\in\mathbb{R}^{p\times q}} \left\{ \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{X}) - \mathbf{M}_\Omega\|_1^2 : \|\mathbf{X}\|_\star \leq \kappa \right\}. \tag{7}$$

Note that Frank-Wolfe cannot solve (7).

Flexibility example: Matrix completion with MovieLens dataset - II

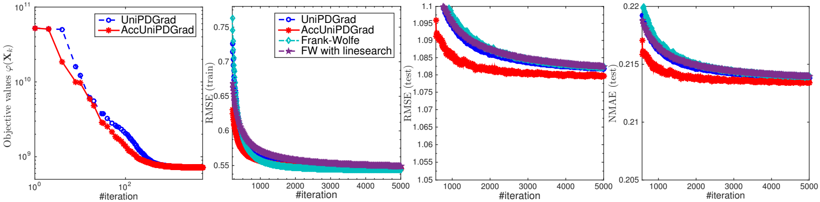


Figure: The performance of UniProx and AccUniProx algorithms for (6) and (7).

Setup [10]

- MovieLens $100k$ dataset: $100'000$ ratings from 943 users on 1682 movies
- Input parameters $\lambda^0 = \mathbf{0}^n$, $\epsilon = 2 \cdot 10^{-2}$ and $\kappa = 9975/2$.

Performance measures

$$\text{RMSE} = \frac{\|\mathcal{P}_\Omega(\mathbf{X}) - \mathbf{M}_\Omega\|_2}{\sqrt{n}} \qquad \textit{and} \qquad \text{NMAE} = \frac{\|\mathcal{P}_\Omega(\mathbf{X}) - \mathbf{M}_\Omega\|_1}{4n}$$

The general constraint case

Handling to the constraint $\mathbf{A}\mathbf{x} - \mathbf{b} \in \mathcal{K}$

the universal dual accelerated gradient method:

$$\begin{cases} t_k & := 0.5\left(1 + \sqrt{1 + 4t_{k-1}^2}\right) \\ \hat{\lambda}^k & := \bar{\lambda}^k + \frac{t_{k-1}-1}{t_k}\left(\bar{\lambda}^k - \hat{\lambda}^{k-1}\right) \\ \lambda^{k+1} & := \hat{\lambda}^k + \frac{1}{M_k}\Big(\mathbf{A}\mathbf{x}^*(\hat{\lambda}^k) - \mathbf{b}\Big). \end{cases}$$

The general constraint case

Handling to the constraint $\mathbf{A}\mathbf{x} - \mathbf{b} \in \mathcal{K}$

Only one prox change in the universal dual accelerated gradient method:

$$\begin{cases} t_k & := 0.5\left(1 + \sqrt{1 + 4t_{k-1}^2}\right) \\ \hat{\lambda}^k & := \bar{\lambda}^k + \frac{t_{k-1}-1}{t_k}\left(\bar{\lambda}^k - \hat{\lambda}^{k-1}\right) \\ \lambda^{k+1} & := \mathrm{prox}_{M_k^{-1}h}\left(\hat{\lambda}^k + \frac{1}{M_k}\left(\mathbf{A}\mathbf{x}^*(\hat{\lambda}^k) - \mathbf{b}\right)\right). \end{cases}$$

Here, h is defined by $h(\lambda) := \sup_{\mathbf{r}\in\mathcal{K}} \langle \lambda, \mathbf{r} \rangle$.

Flexibility example II: Matrix completion with MovieLens dataset

Problem formulation

Let $\Omega \subseteq \{1, \cdots, p\} \times \{1, \cdots, q\}$ be a subset of indexes and $\mathbf{M}_\Omega = (\mathbf{M}_{ij})_{(i,j)\in\Omega}$ be the observed entries of a missed matrix $\mathbf{M}$. $\mathcal{P}_\Omega$ is the projection on the subset Ω.

$$f^\star := \min_{\mathbf{X}\in\mathbb{R}^{p\times q}} \left\{ \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{X}) - \mathbf{M}_\Omega\|_2^2 : \|\mathbf{X}\|_\star \leq \kappa \right\} \tag{8}$$

We can also solve another robust version against outliers:

$$f^\star := \min_{\mathbf{X}\in\mathbb{R}^{p\times q}} \left\{ \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{X}) - \mathbf{M}_\Omega\|_1^2 : \|\mathbf{X}\|_\star \leq \kappa \right\}. \tag{9}$$

Note that Frank-Wolfe cannot solve (9).

Problem formulation

Following formulation may be easier to tune with an expected perturbation level τ:

$$f^\star := \min_{\mathbf{X}\in\mathbb{R}^{p\times q}} \left\{ \frac{1}{2}\|\mathbf{X}\|_\star^2 : \|\mathcal{P}_\Omega(\mathbf{X}) - \mathbf{M}_\Omega\|_1 \leq \tau \right\}. \tag{10}$$

Note that Frank-Wolfe cannot solve (10)

Flexibility example II: Matrix completion with MovieLens dataset

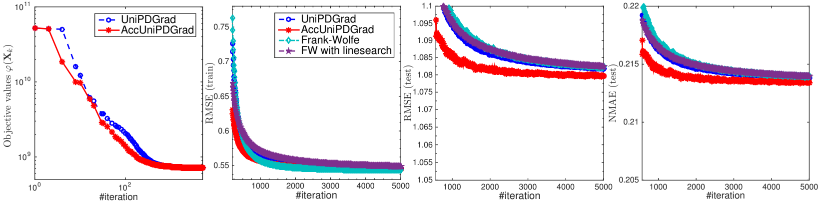


Figure: The performance of UniProx and AccUniProx algorithms for (8) and (9).

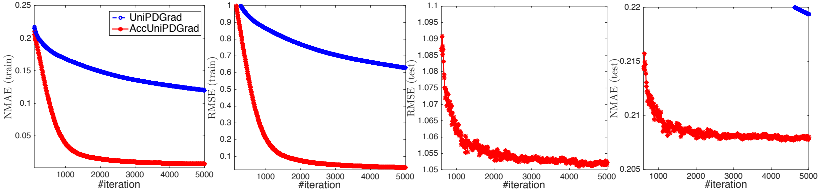


Figure: The performance of UniProx and AccUniProx algorithms for (10).

Setup [10]

- $\kappa = 9975/2$.
- $\tau = 4\times$ NMAE $\times\#$ of test samples

Outline

Yet another template from source separation

Bonus: ADMM[2]

Primal problem with a specific decomposition structure

$$f^\star := \min_{\mathbf{x}:=(\mathbf{u},\mathbf{v})} \{f(\mathbf{x}) := g(\mathbf{u}) + h(\mathbf{v}) : \mathbf{B}\mathbf{u} + \mathbf{C}\mathbf{v} = \mathbf{b},\ \mathbf{u} \in \mathcal{U},\ \mathbf{v} \in \mathcal{V}\}$$

- $\mathcal{X} := \mathcal{U} \times \mathcal{V}$ - nonempty, closed, convex and bounded.
- $\mathbf{A} := [\mathbf{B}, \mathbf{C}]$.

The Fenchel dual problem

$$d^\star := \max_{\lambda \in \mathbb{R}^n} \left\{ d(\lambda) := -g_{\mathcal{U}}^*(-\mathbf{B}^T\lambda) - h_{\mathcal{V}}^*(-\mathbf{C}^T\lambda) + \langle \mathbf{b}, \lambda \rangle \right\}$$

- $g_{\mathcal{U}}^*$ and $h_{\mathcal{U}}^*$ are the Fenchel conjugate of $g_{\mathcal{U}} := g + \delta_{\mathcal{U}}$ and $h_{\mathcal{V}} := h + \delta_{\mathcal{V}}$, resp.

The dual function

$$d(\lambda) := \underbrace{\min_{\mathbf{u} \in \mathcal{U}} \left\{ g(\mathbf{u}) + \langle \mathbf{B}^T\lambda, \mathbf{u} \rangle \right\}}_{d^1(\lambda)} + \underbrace{\min_{\mathbf{v} \in \mathcal{V}} \left\{ h(\mathbf{v}) + \langle \mathbf{C}^T\lambda, \mathbf{v} \rangle \right\}}_{d^2(\lambda)} - \langle \mathbf{b}, \lambda \rangle.$$

[2] Q. Tran-Dinh and V. Cevher, *Splitting the Smoothed Primal-dual Gap: Optimal Alternating Direction Methods* Tech. Report, 2015, (http://arxiv.org/pdf/1507.03734.pdf) / (http://lions.epfl.ch/publications)

Standard ADMM as the dual Douglas-Rachford method

We can derive ADMM via the Douglas-Rachford splitting on the dual:

$$0 \in \mathbf{B}\partial g^*_{\mathcal{U}}(-\mathbf{B}^T\lambda) + \mathbf{C}\partial h^*{}_{\mathcal{V}}(-\mathbf{C}^T\lambda) + \boldsymbol{c},$$

which is the optimality condition of the **dual problem**.

Douglas-Rachford splitting method

$$\begin{cases} \mathbf{z}_g^k & := \mathrm{prox}_{\eta_k^{-1} g^*_{\mathcal{U}}(-\mathbf{B}^T\cdot)}(\lambda^k) \\ \mathbf{z}_h^k & := \mathrm{prox}_{\eta_k^{-1} h^*_{\mathcal{V}}(-\mathbf{C}^T\cdot)}(2\mathbf{z}_g^k - \lambda^k) \\ \lambda^{k+1} & := \lambda^k + (\mathbf{z}_g^k - \mathbf{z}_h^k). \end{cases}$$

Standard ADMM

$$\begin{cases} \mathbf{u}^{k+1} & := \arg\min_{\mathbf{u}\in\mathcal{U}} \left\{ g(\mathbf{u}) + \langle \lambda^k, \mathbf{B}\mathbf{u}\rangle + \frac{\eta_k}{2}\|\mathbf{B}\mathbf{u} + \mathbf{C}\mathbf{v}^k - \mathbf{b}\|^2 \right\} \\ \mathbf{v}^{k+1} & := \arg\min_{\mathbf{v}\in\mathcal{V}} \left\{ h(\mathbf{v}) + \langle \lambda^k, \mathbf{C}\mathbf{v}\rangle + \frac{\eta_k}{2}\|\mathbf{B}\mathbf{u}^{k+1} + \mathbf{C}\mathbf{v} - \mathbf{b}\|^2 \right\} \\ \lambda^{k+1} & := \lambda^k + \eta_k\left(\mathbf{B}\mathbf{u}^{k+1} + \mathbf{C}\mathbf{v}^{k+1} - \mathbf{b}\right). \end{cases}$$

Here, $\eta_k > 0$ is a given penalty parameter.

Splitting the smoothed gap

Smoothing the gap

- The **dual components** d^1 and d^2 are nonsmooth. We smooth one, e.g., d^1, using:

$$d^1_\gamma(\lambda) := \min_{\mathbf{u}\in\mathcal{U}} \left\{ g(\mathbf{u}) + \frac{\gamma}{2}\|\mathbf{B}(\mathbf{u}-\mathbf{u}_c)\|^2 + \langle \lambda, \mathbf{B}\mathbf{u}\rangle \right\}$$

- Recall: We also approximate f by f_β as:

$$f_\beta(\mathbf{x}) := f(\mathbf{x}) + \frac{1}{2\beta}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2 \to f(\mathbf{x}) \quad \text{as } \mathbf{x} \text{ becomes feasible}$$

Three key properties of d^1_γ

- d^1_γ is concave and smooth.
- ∇d^1_γ is Lipschitz continuous with $L := \gamma^{-1}$.
- d^1_γ approximates d^1 as:

$$d^1_\gamma(\lambda) - \gamma D_{\mathcal{U}} \le d^1(\lambda) \le d^1_\gamma(\lambda),$$

where $D_{\mathcal{U}} := \max\left\{(1/2)\|\mathbf{B}(\mathbf{u}-\mathbf{u}_c)\|^2 : \mathbf{u}\in\mathcal{U}\right\}$.

Our ADMM scheme: D-R on the smoothed gap

- Our new ADMM scheme consists of three steps: ADMM step, acceleration step, and primal averaging.

Step 1: The main ADMM steps

$$\begin{cases} \hat{\mathbf{u}}^{k+1} & := \arg\min_{\mathbf{u}\in\mathcal{U}} \left\{ g_{\gamma_{k+1}}(\mathbf{u}) + \langle \hat{\lambda}^k, \mathbf{B}\mathbf{u}\rangle + \frac{\rho_k}{2}\|\mathbf{B}\mathbf{u} + \mathbf{C}\hat{\mathbf{v}}^k - \mathbf{b}\|^2 \right\} \\ \hat{\mathbf{v}}^{k+1} & := \arg\min_{\mathbf{v}\in\mathcal{V}} \left\{ h(\mathbf{v}) + \langle \hat{\lambda}^k, \mathbf{C}\mathbf{v}\rangle + \frac{\eta_k}{2}\|\mathbf{B}\hat{\mathbf{u}}^{k+1} + \mathbf{C}\mathbf{v} - \mathbf{b}\|^2 \right\} \\ \lambda^{k+1} & := \hat{\lambda}^k + \eta_k\left(\mathbf{B}\hat{\mathbf{u}}^{k+1} + \mathbf{C}\hat{\mathbf{v}}^{k+1} - \mathbf{b}\right). \end{cases}$$

where $g_\gamma(\cdot) := g(\cdot) + \frac{\gamma}{2}\|\mathbf{B}(\cdot - \mathbf{u}_c)\|^2$.

The dual accelerated and primal averaging steps

- **Step 2:** [Dual acceleration] $\hat{\lambda}^k$ is computed as:

$$\hat{\lambda}^k := (1-\tau_k)\lambda_k + \frac{\tau_k}{\beta_k}(\mathbf{B}\mathbf{u}^k + \mathbf{C}\mathbf{v}^k - \mathbf{b}).$$

- **Step 3:** [Averaging] The primal iteration $\mathbf{x}^k := (\mathbf{u}^k, \mathbf{v}^k)$ is updated as:

$$\mathbf{u}^{k+1} := (1-\tau_k)\mathbf{u}^k + \tau_k\hat{\mathbf{u}}^{k+1} \quad \text{and} \quad \mathbf{v}^{k+1} := (1-\tau_k)\mathbf{v}^k + \tau_k\hat{\mathbf{v}}^{k+1}.$$

How do we update parameters?

Duality gap and smoothed gap functions

- The duality gap: $G(\mathbf{w}) := f(\mathbf{x}) - d(\lambda)$, where $\mathbf{w} := (\mathbf{x}, \lambda)$.
- The smoothed gap: $\boxed{G_{\gamma\beta}(\mathbf{w}) := f_\beta(\mathbf{x}) - d_\gamma(\lambda)}$ with $d_\gamma := d_\gamma^1 + d^2$.

Model-based gap reduction

The **gap reduction model** provides conditions to derive parameter update rules:

$$\boxed{G_{\gamma_{k+1}\beta_{k+1}}(\mathbf{w}^{k+1}) \leq (1-\tau_k)G_{\gamma_k\beta_k}(\mathbf{w}^k) + \tau_k(\eta_k + \rho_k)D_{\mathcal{X}}}$$

where $\gamma_{k+1} < \gamma_k$, $\beta_{k+1} < \beta_k$ and $D_{\mathcal{X}} := \max_{\mathbf{x}\in\mathcal{X}} \left\{(1/2)\|\mathbf{Bu} + \mathbf{Cv} - \mathbf{b}\|^2\right\}$.

Update rules

- **The smoothness parameters:** $\gamma_{k+1} := \frac{2\gamma_0}{k+3}$ and $\beta_k := \frac{9(k+3)}{\gamma_0(k+1)(k+7)}$.
- **The penalty parameters:** $\eta_k := \frac{\gamma_0}{k+3}$ and $\rho_k := \frac{3\gamma_0}{(k+3)(k+4)}$.
- **The step-size** $\tau_k := \frac{3}{k+4}$ $\Rightarrow$ $\mathcal{O}\left(\frac{1}{k}\right)$.

Convergence guarantee & Other cases of interest

Convergence rate guarantee

- Rate on the primal objective residual and constraint feasibility:

$$
\begin{aligned}
f(\mathbf{x}^k) - f^\star &\leq \frac{2\gamma_0 D_{\mathcal{U}}}{k+2} + \frac{3\gamma_0 D_{\mathcal{X}}}{2(k+3)}\left(1 + \frac{6}{k+2}\right) && \Rightarrow \quad \mathcal{O}\left(\frac{1}{k}\right) \\
\|\mathbf{A}\mathbf{x}^k - \mathbf{b}\| &\leq \frac{18 D_d^*}{\gamma_0 (k+2)} + \frac{6}{k+2}\sqrt{D_{\mathcal{U}} + \frac{3(k+8)}{2(k+3)} D_{\mathcal{X}}} && \Rightarrow \quad \mathcal{O}\left(\frac{1}{k}\right)
\end{aligned}
$$

  where D_d^* is the diameter of the **dual solution set** $\Lambda^\star$.

- Lower bound: $-D_d^* \|\mathbf{A}\mathbf{x}^k - \mathbf{b}\| \leq f(\mathbf{x}^k) - f^\star$.
- Rate on the dual objective residual:

$$
d^\star - d(\lambda^k) \leq \frac{18 (D_d^*)^2}{\gamma_0 (k+2)} + \frac{6 D_d^*}{k+2}\sqrt{D_{\mathcal{U}} + \frac{3(k+8)}{2(k+3)} D_{\mathcal{X}}} \quad \Rightarrow \quad \mathcal{O}\left(\frac{1}{k}\right).
$$

Special cases: *cf.*, `http://lions.epfl.ch/publications`

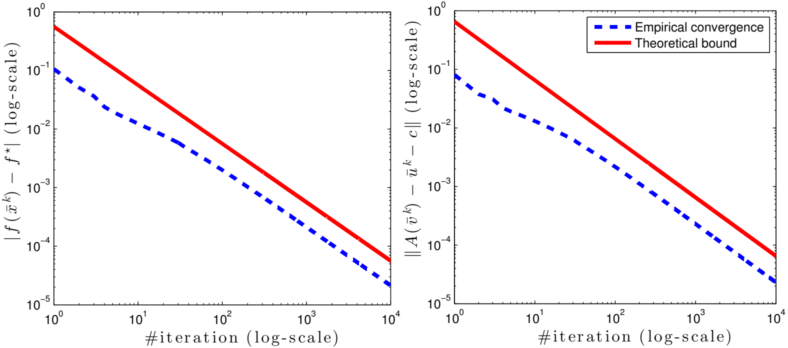
- Full-column rank or orthogonality of $\mathbf{A}$: Using smoothing term $(\gamma/2)\|\mathbf{u} - \mathbf{u}_c\|^2$.
- Strong convexity of g: We do not need to smooth d^1.
- Decomposability of g and $\mathcal{U}$: Using smoothing term

$$
(\gamma/2)\sum_{i=1}^{s} \|\mathbf{B}_i(\mathbf{u}_i - \mathbf{u}_{c,i})\|^2.
$$

A comparison to the theoretical bounds

A stylized example: Square-root LASSO

$$f^\star := \min_{\mathbf{u}\in\mathcal{U},\mathbf{v}\in\mathcal{V}} \left\{ f(\mathbf{x}) := \|\mathbf{u}\|_2 + \kappa \|\mathbf{v}\|_1 : \mathbf{B}(\mathbf{v}) - \mathbf{u} = \boldsymbol{c} \right\}.$$



- See the preprint for more examples, enhancements, ...

References I

[1] E.J. Candès, T. Strohmer, and V. Voroninski.
Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming.
IEE Trans. Signal Processing, 60(5):2422–2432, 2012.

[2] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert.
Quantum state tomography via compressed sensing.
Physical Review Letters, 105(15):150401(0–4), 2010.

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

[4] A. Nemirovskii and D. Yudin.
Problem Complexity and Method Efficiency in Optimization.
Wiley Interscience, 1983.

[5] Y. Nesterov.
Smooth minimization of non-smooth functions.
Math. Program., 103(1):127–152, 2005.

References II

[6] Y. Nesterov.
Dual extrapolation and its applications to solving variational inequalities and related problems.
Math. Program., 109(2–3):319–344, 2007.

[7] Y. Nesterov.
Primal-dual subgradient methods for convex problems.
Math. Program., 120(1, Ser. B):221–259, 2009.

[8] Y. Nesterov.
Universal gradient methods for convex optimization problems.
Math. Program., xx:1–24, 2014.

[9] A. Yurtsever, Y. P. Hsieh, and V. Cevher.
Scalable convex methods for phase retrieval.
In *under review*, 2015.

[10] A. Yurtsever, Q. Tran-Dinh, and V. Cevher.
Universal primal-dual proximal-gradient methods.
Tech. Report. (LIONS, EPFL), Available at:
http://arxiv.org/pdf/1502.03123.pdf., 2015.