Dual and primal-dual methods

Ye Shi

ShanghaiTech University

Outline

- Dual proximal gradient method
- Primal-dual proximal gradient method

Dual proximal gradient method

Constrained convex optimization

$$\begin{array}{ll} \text{minimize}_{\boldsymbol{x}} & f(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} \in \mathcal{C} \end{array}$$

where f is convex, and $\mathcal{C}$ is convex set

- projection onto such a feasible set could sometimes be highly nontrivial (even when projection onto $\mathcal{C}$ is easy)

Constrained convex optimization

More generally, consider

$$\text{minimize}_{\boldsymbol{x}} \quad f(\boldsymbol{x}) + h(\boldsymbol{A}\boldsymbol{x})$$

where f and h are convex

- computing the proximal operator w.r.t. $\tilde{h}(\boldsymbol{x}) := h(\boldsymbol{A}\boldsymbol{x})$ could be difficult (even when prox_h is inexpensive)

A possible route: dual formulation

$$\text{minimize}_{\boldsymbol{x}} \quad f(\boldsymbol{x}) + h(\boldsymbol{A}\boldsymbol{x})$$

$$\Updownarrow \quad \text{add auxiliary variable } \boldsymbol{z}$$

$$\begin{aligned} \text{minimize}_{\boldsymbol{x},\boldsymbol{z}} \quad & f(\boldsymbol{x}) + h(\boldsymbol{z}) \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{z} \end{aligned}$$

dual formulation:

$$\text{maximize}_{\boldsymbol{\lambda}} \quad \min_{\boldsymbol{x},\boldsymbol{z}} \underbrace{f(\boldsymbol{x}) + h(\boldsymbol{z}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{z} \rangle}_{=: \mathcal{L}(\boldsymbol{x},\boldsymbol{z},\boldsymbol{\lambda}) \text{ (Lagrangian)}}$$

A possible route: dual formulation

$$\min f(x) = -\max \{-f(x)\}$$

$$\text{maximize}_{\boldsymbol{\lambda}} \quad \min_{\boldsymbol{x},\boldsymbol{z}} \; f(\boldsymbol{x}) + h(\boldsymbol{z}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{z} \rangle$$

$$\Updownarrow \quad \text{decouple } \boldsymbol{x} \text{ and } \boldsymbol{z}$$

$$\text{maximize}_{\boldsymbol{\lambda}} \quad \min_{\boldsymbol{x}} \left\{ \langle \boldsymbol{A}^\top \boldsymbol{\lambda}, \boldsymbol{x} \rangle + f(\boldsymbol{x}) \right\} + \min_{\boldsymbol{z}} \left\{ h(\boldsymbol{z}) - \langle \boldsymbol{\lambda}, \boldsymbol{z} \rangle \right\}$$

$$\Updownarrow$$

$$\text{maximize}_{\boldsymbol{\lambda}} \quad - f^*(-\boldsymbol{A}^\top \boldsymbol{\lambda}) - h^*(\boldsymbol{\lambda})$$

where f^* (resp. h^*) is the Fenchel conjugate of f (resp. h)

$$f^*(x) = \sup_z \{\langle x, z \rangle - f(z)\} = -\min_z \{-\langle x, z \rangle + f(z)\}$$

Primal vs. dual problems

$$\begin{array}{rll} \text{(primal)} & \text{minimize}_{\boldsymbol{x}} & f(\boldsymbol{x}) + h(\boldsymbol{A}\boldsymbol{x}) \\ \text{(dual)} & \text{minimize}_{\boldsymbol{\lambda}} & f^*(-\boldsymbol{A}^\top \boldsymbol{\lambda}) + h^*(\boldsymbol{\lambda}) \end{array}$$

Dual formulation is useful if

- the proximal operator w.r.t. h is cheap (then we can use the Moreau decomposition $\text{prox}_{h^*}(\boldsymbol{x}) = \boldsymbol{x} - \text{prox}_h(\boldsymbol{x})$)
- f^* is smooth (or if f is strongly convex)

$$\min \underbrace{f^*(-A^T\lambda)}_{\text{linear approximation}} + h^*(\lambda)$$

$$\lambda^{t+1} = \arg\min_{\lambda} \underbrace{f^*(-A^T\lambda^t)}_{\text{constant}} + \langle -A \cdot \nabla f^*(-A^T\lambda), \lambda - \lambda^t \rangle + h^*(\lambda) + \frac{1}{2\eta_t}\|\lambda - \lambda^t\|_2^2$$

$$= \arg\min \frac{1}{2}\|\lambda - (\lambda^t + \eta_t A \nabla f^*(-A^T\lambda^t))\|_2^2 + \eta_t h^*(\lambda)$$

$$= \mathrm{prox}_{\eta_t h^*}(\lambda^t + \eta_t A \nabla f^*(-A^T\lambda^t))$$

Dual proximal gradient methods

Apply proximal gradient methods to the dual problem:

Algorithm 9.1 Dual proximal gradient algorithm

1: **for** $t = 0, 1, \cdots$ **do**

2: $\quad \boldsymbol{\lambda}^{t+1} = \mathsf{prox}_{\eta_t h^*}\Big(\boldsymbol{\lambda}^t + \eta_t \boldsymbol{A} \nabla f^*\big(- \boldsymbol{A}^\top \boldsymbol{\lambda}^t\big)\Big)$

- let $Q(\boldsymbol{\lambda}) := -f^*(-\boldsymbol{A}^\top \boldsymbol{\lambda}) - h^*(\boldsymbol{\lambda})$ and $Q^{\mathsf{opt}} = \max_{\boldsymbol{\lambda}} Q(\boldsymbol{\lambda})$, then

$$Q^{\mathsf{opt}} - Q(\boldsymbol{\lambda}^t) \lesssim \frac{1}{t} \tag{9.1}$$

Primal representation of dual proximal gradient methods

Algorithm 9.1 admits a more explicit primal representation

Algorithm 9.2 Dual proximal gradient algorithm (primal representation)

for $t = 0, 1, \cdots$ **do**

$\quad \boldsymbol{x}^t = \arg\min_{\boldsymbol{x}} \ \{f(\boldsymbol{x}) + \langle \boldsymbol{A}^\top \boldsymbol{\lambda}^t, \boldsymbol{x} \rangle\}$

$\quad \boldsymbol{\lambda}^{t+1} = \boldsymbol{\lambda}^t + \eta_t \boldsymbol{A}\boldsymbol{x}^t - \eta_t \mathsf{prox}_{\eta_t^{-1} h}(\eta_t^{-1}\boldsymbol{\lambda}^t + \boldsymbol{A}\boldsymbol{x}^t)$

- $\{\boldsymbol{x}^t\}$ is a primal sequence, which is nonetheless *not always feasible*

Justification of the primal representation

By definition of $\boldsymbol{x}^t$,

$$-\boldsymbol{A}^\top \boldsymbol{\lambda}^t \in \partial f(\boldsymbol{x}^t)$$

This together with the conjugate subgradient theorem and the smoothness of f^* yields

$$\boldsymbol{x}^t = \nabla f^*(-\boldsymbol{A}^\top \boldsymbol{\lambda}^t)$$

Therefore, the dual proximal gradient update rule can be rewritten as

$$\boldsymbol{\lambda}^{t+1} = \mathrm{prox}_{\eta_t h^*}(\boldsymbol{\lambda}^t + \eta_t \boldsymbol{A}\boldsymbol{x}^t) \tag{9.2}$$

Justification of primal representation (cont.)

$$x = \text{prox}_{\lambda f}(x) + \lambda \text{prox}_{\lambda^{-1} f^*}(x/\lambda)$$

Moreover, from the extended Moreau decomposition, we know

$$\text{prox}_{\eta_t h^*}(\boldsymbol{\lambda}^t + \eta_t \boldsymbol{A}\boldsymbol{x}^t) = \boldsymbol{\lambda}^t + \eta_t \boldsymbol{A}\boldsymbol{x}^t - \eta_t \text{prox}_{\eta_t^{-1} h}(\eta_t^{-1}\boldsymbol{\lambda}^t + \boldsymbol{A}\boldsymbol{x}^t)$$

$$\Longrightarrow \qquad \boldsymbol{\lambda}^{t+1} = \boldsymbol{\lambda}^t + \eta_t \boldsymbol{A}\boldsymbol{x}^t - \eta_t \text{prox}_{\eta_t^{-1} h}(\eta_t^{-1}\boldsymbol{\lambda}^t + \boldsymbol{A}\boldsymbol{x}^t)$$

Accuracy of the primal sequence

One can control the primal accuracy via the dual accuracy:

Lemma 9.1

Let $\boldsymbol{x}_{\boldsymbol{\lambda}} := \arg\min_{\boldsymbol{x}} \left\{ f(\boldsymbol{x}) + \langle \boldsymbol{A}^\top \boldsymbol{\lambda}, \boldsymbol{x} \rangle \right\}$. *Suppose* f *is* μ*-strongly convex. Then*

$$\|\boldsymbol{x}^* - \boldsymbol{x}_{\boldsymbol{\lambda}}\|_2^2 \leq \frac{2\big(Q^{\mathsf{opt}} - Q(\boldsymbol{\lambda})\big)}{\mu}$$

- **consequence:** $\|\boldsymbol{x}^* - \boldsymbol{x}^t\|_2^2 \lesssim 1/t$ (using (9.1))

Proof of Lemma 9.1

Recall that Lagrangian is given by

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda}) := \underbrace{f(\boldsymbol{x}) + \langle \boldsymbol{A}^\top \boldsymbol{\lambda}, \boldsymbol{x} \rangle}_{=: \tilde{f}(\boldsymbol{x}, \boldsymbol{\lambda})} + \underbrace{h(\boldsymbol{z}) - \langle \boldsymbol{\lambda}, \boldsymbol{z} \rangle}_{=: \tilde{h}(\boldsymbol{z}, \boldsymbol{\lambda})}$$

For any $\boldsymbol{\lambda}$, define $\boldsymbol{x}_{\boldsymbol{\lambda}} := \arg\min_{\boldsymbol{x}} \tilde{f}(\boldsymbol{x}, \boldsymbol{\lambda})$ and $\boldsymbol{z}_{\boldsymbol{\lambda}} := \arg\min_{\boldsymbol{z}} \tilde{h}(\boldsymbol{z}, \boldsymbol{\lambda})$ (non-rigorous). Then by strong convexity,

$\nabla \tilde{f}(x_\lambda, \lambda) = 0$

$$\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{\lambda}) - \mathcal{L}(\boldsymbol{x}_{\boldsymbol{\lambda}}, \boldsymbol{z}_{\boldsymbol{\lambda}}, \boldsymbol{\lambda}) \geq \tilde{f}(\boldsymbol{x}^*, \boldsymbol{\lambda}) - \tilde{f}(\boldsymbol{x}_{\boldsymbol{\lambda}}, \boldsymbol{\lambda}) \geq \frac{1}{2}\mu \|\boldsymbol{x}^* - \boldsymbol{x}_{\boldsymbol{\lambda}}\|_2^2$$

In addition, since $\boldsymbol{A}\boldsymbol{x}^* = \boldsymbol{z}^*$, one has

$$\begin{aligned}\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{\lambda}) &= f(\boldsymbol{x}^*) + h(\boldsymbol{z}^*) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{z}^* \rangle = f(\boldsymbol{x}^*) + h(\boldsymbol{A}\boldsymbol{x}^*) \\ &= F^{\mathsf{opt}} \overset{\text{duality}}{=} Q^{\mathsf{opt}}\end{aligned}$$

This combined with $\mathcal{L}(\boldsymbol{x}_{\boldsymbol{\lambda}}, \boldsymbol{z}_{\boldsymbol{\lambda}}, \boldsymbol{\lambda}) = Q(\boldsymbol{\lambda})$ gives

$$Q^{\mathsf{opt}} - Q(\boldsymbol{\lambda}) \geq \frac{1}{2}\mu \|\boldsymbol{x}^* - \boldsymbol{x}_{\boldsymbol{\lambda}}\|_2^2$$

as claimed

$$L(x^*, z^*, \lambda) - L(x_\lambda, z_\lambda, \lambda)$$

$$= \tilde{f}(x^*, \lambda) + \tilde{h}(z^*, \lambda) - \tilde{f}(x_\lambda, \lambda) - \tilde{h}(z_\lambda, \lambda)$$

$$\geq \tilde{f}(x^*, \lambda) - \tilde{f}(x_\lambda, \lambda) \qquad (\tilde{h}(z^*, \lambda) - \tilde{h}(z_\lambda, \lambda) \geq 0)$$

Accelerated dual proximal gradient methods

One can apply FISTA to dual problem to improve convergence:

Algorithm 9.3 Accelerated dual proximal gradient algorithm

1: **for** $t = 0, 1, \cdots$ **do**

2: $\quad \boldsymbol{\lambda}^{t+1} = \mathsf{prox}_{\eta_t h^*}\Big(\boldsymbol{w}^t + \eta_t \boldsymbol{A} \nabla f^*(-\boldsymbol{A}^\top \boldsymbol{w}^t)\Big)$

3: $\quad \theta_{t+1} = \frac{1+\sqrt{1+4\theta_t^2}}{2}$

4: $\quad \boldsymbol{w}^{t+1} = \boldsymbol{\lambda}^{t+1} + \frac{\theta_t - 1}{\theta_{t+1}}(\boldsymbol{\lambda}^{t+1} - \boldsymbol{\lambda}^t)$

- apply FISTA theory and Lemma 9.1 to get

$$Q^{\mathsf{opt}} - Q(\boldsymbol{\lambda}^t) \lesssim \frac{1}{t^2} \quad \text{and} \quad \|\boldsymbol{x}^* - \boldsymbol{x}^t\|_2^2 \lesssim \frac{1}{t^2}$$

Primal representation of accelerated dual proximal gradient methods

Algorithm 9.3 admits more explicit primal representation

Algorithm 9.4 Accelerated dual proximal gradient algorithm (primal representation)

1: **for** $t = 0, 1, \cdots$ **do**
2: $\quad \boldsymbol{x}^t = \arg\min_{\boldsymbol{x}} \ f(\boldsymbol{x}) + \langle \boldsymbol{A}^\top \boldsymbol{w}^t, \boldsymbol{x} \rangle$
3: $\quad \boldsymbol{\lambda}^{t+1} = \boldsymbol{w}^t + \eta_t \boldsymbol{A}\boldsymbol{x}^t - \eta_t \mathsf{prox}_{\eta_t^{-1} h}(\eta_t^{-1} \boldsymbol{w}^t + \boldsymbol{A}\boldsymbol{x}^t)$
4: $\quad \theta_{t+1} = \frac{1+\sqrt{1+4\theta_t^2}}{2}$
5: $\quad \boldsymbol{w}^{t+1} = \boldsymbol{\lambda}^{t+1} + \frac{\theta_t - 1}{\theta_{t+1}}(\boldsymbol{\lambda}^{t+1} - \boldsymbol{\lambda}^t)$

Primal-dual proximal gradient method

Nonsmooth optimization

$$\text{minimize}_{\boldsymbol{x}} \quad f(\boldsymbol{x}) + h(\boldsymbol{A}\boldsymbol{x})$$

where f and h are closed and convex

- both f and h might be non-smooth
- both f and h might have inexpensive proximal operators

Primal-dual approaches?

$$\text{minimize}_{\boldsymbol{x}} \quad f(\boldsymbol{x}) + h(\boldsymbol{A}\boldsymbol{x})$$

So far we have discussed proximal methods (resp. dual proximal methods), which essentially updates only primal (resp. dual) variables

Question: can we update both primal and dual variables simultaneously and take advantage of both prox_f and prox_h?

A saddle-point formulation

To this end, we first derive a saddle-point formulation that includes both primal and dual variables

$$\text{minimize}_{\boldsymbol{x}} \quad f(\boldsymbol{x}) + h(\boldsymbol{A}\boldsymbol{x})$$

$$\Updownarrow \quad \text{add an auxiliary variable } \boldsymbol{z}$$

$$\text{minimize}_{\boldsymbol{x},\boldsymbol{z}} \quad f(\boldsymbol{x}) + h(\boldsymbol{z}) \quad \text{subject to } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{z}$$

$$\Updownarrow$$

$$\text{maximize}_{\boldsymbol{\lambda}} \ \text{min}_{\boldsymbol{x},\boldsymbol{z}} \ f(\boldsymbol{x}) + h(\boldsymbol{z}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{z} \rangle$$

$$\Updownarrow$$

$$\text{maximize}_{\boldsymbol{\lambda}} \ \text{min}_{\boldsymbol{x}} \ f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} \rangle - h^*(\boldsymbol{\lambda})$$

$$\Updownarrow$$

$$\text{minimize}_{\boldsymbol{x}} \ \text{max}_{\boldsymbol{\lambda}} \ f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} \rangle - h^*(\boldsymbol{\lambda}) \quad \text{(saddle-point problem)}$$

A saddle-point formulation

$$\text{minimize}_{\boldsymbol{x}} \ \max_{\boldsymbol{\lambda}} \ f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} \rangle - h^*(\boldsymbol{\lambda}) \tag{9.3}$$

- one can then consider updating the primal variable $\boldsymbol{x}$ and the dual variable $\boldsymbol{\lambda}$ simultaneously
- we'll first examine the optimality condition for (9.3), which in turn gives ideas about how to jointly update primal and dual variables

Optimality condition

$$\text{minimize}_{\boldsymbol{x}} \ \text{max}_{\boldsymbol{\lambda}} \ f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} \rangle - h^*(\boldsymbol{\lambda})$$

optimality condition:

$$\begin{cases} \mathbf{0} \in & \partial f(\boldsymbol{x}) + \boldsymbol{A}^\top \boldsymbol{\lambda} \\ \mathbf{0} \in & -\boldsymbol{A}\boldsymbol{x} + \partial h^*(\boldsymbol{\lambda}) \end{cases}$$

$$\Longleftrightarrow \quad \mathbf{0} \in \begin{bmatrix} & \boldsymbol{A}^\top \\ -\boldsymbol{A} & \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix} + \begin{bmatrix} \partial f(\boldsymbol{x}) \\ \partial h^*(\boldsymbol{\lambda}) \end{bmatrix} =: \mathcal{F}(\boldsymbol{x}, \boldsymbol{\lambda}) \quad (9.4)$$

key idea: iteratively update $(\boldsymbol{x}, \boldsymbol{\lambda})$ to reach a point obeying $\mathbf{0} \in \mathcal{F}(\boldsymbol{x}, \boldsymbol{\lambda})$

How to solve $\mathbf{0} \in \mathcal{F}(x)$ in general?

In general, finding solution to

$$\underbrace{\mathbf{0} \in \mathcal{F}(\boldsymbol{x})}_{\text{called "monotone inclusion problem" if } \mathcal{F} \text{ is maximal monotone}}$$

$$\Longleftrightarrow \qquad \boldsymbol{x} \in (\mathcal{I} + \mathcal{F})(\boldsymbol{x})$$

is equivalent to finding fixed points of $\underbrace{(\mathcal{I} + \eta\mathcal{F})^{-1}}_{\text{resolvent of } \mathcal{F}}$, i.e. solutions to

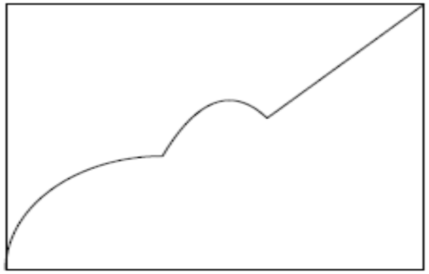
$$\boldsymbol{x} = (\mathcal{I} + \eta\mathcal{F})^{-1}(\boldsymbol{x})$$

This suggests a natural fixed-point iteration / resolvent iteration:

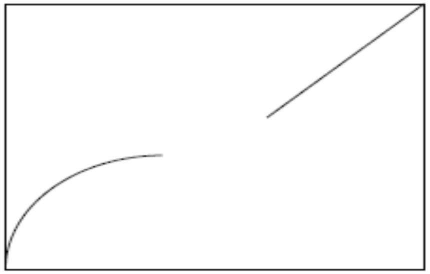
$$\boldsymbol{x}^{t+1} = (\mathcal{I} + \eta\mathcal{F})^{-1}(\boldsymbol{x}^t), \qquad t = 0, 1, \cdots$$

Aside: monotone operators

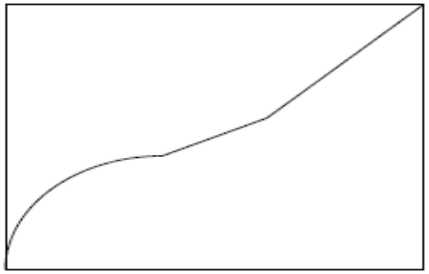
— Ryu, Boyd '16



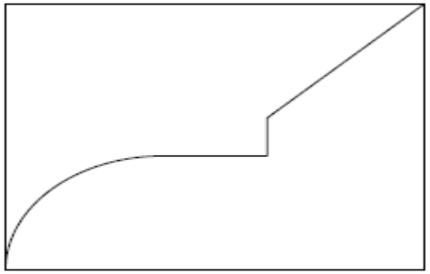
(A) Not monotone.



(B) Monotone but not maximal.



(C) Maximal monotone function.



(D) Maximal monotone but not a function.

- a relation $\mathcal{F}$ is called *monotone* if

$$\langle \boldsymbol{u} - \boldsymbol{v}, \boldsymbol{x} - \boldsymbol{y} \rangle \geq 0, \quad \forall (\boldsymbol{x}, \boldsymbol{u}), (\boldsymbol{y}, \boldsymbol{v}) \in \mathcal{F}$$

- relation $\mathcal{F}$ is called *maximal monotone* if there is no monotone operator that contains it

Proximal point method

$$\boldsymbol{x}^{t+1} = (\mathcal{I} + \eta_t \mathcal{F})^{-1}(\boldsymbol{x}^t), \qquad t = 0, 1, \cdots$$

If $\mathcal{F} = \partial f$ for some convex function f, then this proximal point method becomes

$$\boldsymbol{x}^{t+1} = \mathsf{prox}_{\eta_t f}(\boldsymbol{x}^t), \qquad t = 0, 1, \cdots$$

- useful when $\mathsf{prox}_{\eta_t f}$ is cheap

Back to primal-dual approaches

Recall that we want to solve

$$\mathbf{0} \in \begin{bmatrix} & \boldsymbol{A}^\top \\ -\boldsymbol{A} & \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix} + \begin{bmatrix} \partial f(\boldsymbol{x}) \\ \partial h^*(\boldsymbol{\lambda}) \end{bmatrix} =: \mathcal{F}(\boldsymbol{x}, \boldsymbol{\lambda})$$

the issue of proximal point methods: computing $(\mathcal{I} + \eta \mathcal{F})^{-1}$ is in general difficult

Back to primal-dual approaches

observation: practically we may often consider splitting $\mathcal{F}$ into two operators

$$\mathbf{0} \in \mathcal{A}(\boldsymbol{x}, \boldsymbol{\lambda}) + \mathcal{B}(\boldsymbol{x}, \boldsymbol{\lambda})$$

$$\text{with } \mathcal{A}(\boldsymbol{x}, \boldsymbol{\lambda}) = \begin{bmatrix} & \boldsymbol{A} \\ -\boldsymbol{A}^\top & \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix}, \; \mathcal{B}(\boldsymbol{x}, \boldsymbol{\lambda}) = \begin{bmatrix} \partial f(\boldsymbol{x}) \\ \partial h^*(\boldsymbol{\lambda}) \end{bmatrix} \tag{9.5}$$

- $(\mathcal{I} + \eta \mathcal{A})^{-1}$ can be computed by solving linear systems
- $(\mathcal{I} + \eta \mathcal{B})^{-1}$ is easy if prox_f and prox_{h^*} are both inexpensive

solution: design update rules based on $(\mathcal{I} + \eta \mathcal{A})^{-1}$ and $(\mathcal{I} + \eta \mathcal{B})^{-1}$ instead of $(\mathcal{I} + \eta \mathcal{F})^{-1}$

Operator splitting via Cayley operators

We now introduce a principled approach based on operator splitting

$$\text{find} \quad \boldsymbol{x} \qquad \text{s.t.} \ \ \mathbf{0} \in \mathcal{F}(\boldsymbol{x}) = \underbrace{\mathcal{A}(\boldsymbol{x}) + \mathcal{B}(\boldsymbol{x})}_{\text{operator splitting}}$$

let $\mathcal{R}_{\mathcal{A}} := (\mathcal{I} + \eta\mathcal{A})^{-1}$ and $\mathcal{R}_{\mathcal{B}} := (\mathcal{I} + \eta\mathcal{B})^{-1}$ be the resolvents, and $\mathcal{C}_{\mathcal{A}} := 2\mathcal{R}_{\mathcal{A}} - \mathcal{I}$ and $\mathcal{C}_{\mathcal{B}} := 2\mathcal{R}_{\mathcal{B}} - \mathcal{I}$ be the Cayley operators

Lemma 9.2

$$\underbrace{\mathbf{0} \in \mathcal{A}(\boldsymbol{x}) + \mathcal{B}(\boldsymbol{x})}_{\boldsymbol{x} \in \mathcal{R}_{\mathcal{A}+\mathcal{B}}(\boldsymbol{x})} \quad \Longleftrightarrow \quad \underbrace{\mathcal{C}_{\mathcal{A}}\mathcal{C}_{\mathcal{B}}(\boldsymbol{z}) = \boldsymbol{z} \text{ with } \boldsymbol{x} = \mathcal{R}_{\mathcal{B}}(\boldsymbol{z})}_{\text{it comes down to finding fixed points of } \mathcal{C}_{\mathcal{A}}\mathcal{C}_{\mathcal{B}}} \tag{9.6}$$

Operator splitting via Cayley operators

$$\boldsymbol{x} \in \mathcal{R}_{\mathcal{A}+\mathcal{B}}(\boldsymbol{x}) \quad \Longleftrightarrow \quad \mathcal{C}_{\mathcal{A}}\mathcal{C}_{\mathcal{B}}(\boldsymbol{z}) = \boldsymbol{z}$$

- **advantage:** allows us to apply $\mathcal{C}_{\mathcal{A}}$ (resp. $\mathcal{R}_{\mathcal{A}}$) and $\mathcal{C}_{\mathcal{B}}$ (resp. $\mathcal{R}_{\mathcal{B}}$) sequentially (instead of computing $\mathcal{R}_{\mathcal{A}+\mathcal{B}}$ directly)

Proof of Lemma 9.2

$$\mathcal{C}_{\mathcal{A}}\mathcal{C}_{\mathcal{B}}(\boldsymbol{z}) = \boldsymbol{z}$$

$$\begin{aligned} \boldsymbol{x} &= \mathcal{R}_{\mathcal{B}}(\boldsymbol{z}) && (9.7a) \\ \Longleftrightarrow \quad \tilde{\boldsymbol{z}} &= 2\boldsymbol{x} - \boldsymbol{z} && (9.7b) \\ \tilde{\boldsymbol{x}} &= \mathcal{R}_{\mathcal{A}}(\tilde{\boldsymbol{z}}) && (9.7c) \\ \boldsymbol{z} &= 2\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{z}} && (9.7d) \end{aligned}$$

From (9.7b) and (9.7d), we see that

$$\tilde{\boldsymbol{x}} = \boldsymbol{x}$$

which together with (9.7d) gives

$$2\boldsymbol{x} = \boldsymbol{z} + \tilde{\boldsymbol{z}} \tag{9.8}$$

Proof of Lemma 9.2 (cont.)

Recall that

$$\boldsymbol{z} \in \boldsymbol{x} + \eta \mathcal{B}(\boldsymbol{x}) \qquad \text{and} \qquad \tilde{\boldsymbol{z}} \in \boldsymbol{x} + \eta \mathcal{A}(\boldsymbol{x})$$

Adding these two facts and using (9.8), we get

$$2\boldsymbol{x} = \boldsymbol{z} + \tilde{\boldsymbol{z}} \in 2\boldsymbol{x} + \eta \mathcal{B}(\boldsymbol{x}) + \eta \mathcal{A}(\boldsymbol{x})$$

$$\Longleftrightarrow \qquad \boldsymbol{0} \in \mathcal{A}(\boldsymbol{x}) + \mathcal{B}(\boldsymbol{x})$$

Douglas-Rachford splitting

How to find points obeying $\boldsymbol{x} = \mathcal{C}_{\mathcal{A}}\mathcal{C}_{\mathcal{B}}(\boldsymbol{x})$?

- First attempt: fixed-point iteration

$$\boldsymbol{z}^{t+1} = \mathcal{C}_{\mathcal{A}}\mathcal{C}_{\mathcal{B}}(\boldsymbol{z}^t)$$

  unfortunately, it may not converge in general

- **Douglas-Rachford splitting**: damped fixed-point iteration

$$\boldsymbol{z}^{t+1} = \frac{1}{2}\big(\mathcal{I} + \mathcal{C}_{\mathcal{A}}\mathcal{C}_{\mathcal{B}}\big)(\boldsymbol{z}^t)$$

  converges when a solution to $\mathbf{0} \in \mathcal{A}(\boldsymbol{x}) + \mathcal{B}(\boldsymbol{x})$ exists!

More explicit expression for D-R splitting

Douglas-Rachford splitting update rule $\boldsymbol{z}^{t+1} = \frac{1}{2}(\mathcal{I} + \mathcal{C}_{\mathcal{A}}\mathcal{C}_{\mathcal{B}})(\boldsymbol{z}^t)$ is essentially:

$$
\begin{aligned}
\boldsymbol{x}^{t+\frac{1}{2}} &= \mathcal{R}_{\mathcal{B}}(\boldsymbol{z}^t) \\
\boldsymbol{z}^{t+\frac{1}{2}} &= 2\boldsymbol{x}^{t+\frac{1}{2}} - \boldsymbol{z}^t \\
\boldsymbol{x}^{t+1} &= \mathcal{R}_{\mathcal{A}}(\boldsymbol{z}^{t+\frac{1}{2}}) \\
\boldsymbol{z}^{t+1} &= \frac{1}{2}(\boldsymbol{z}^t + 2\boldsymbol{x}^{t+1} - \boldsymbol{z}^{t+\frac{1}{2}}) \\
&= \boldsymbol{z}^t + \boldsymbol{x}^{t+1} - \boldsymbol{x}^{t+\frac{1}{2}}
\end{aligned}
$$

where $\boldsymbol{x}^{t+\frac{1}{2}}$ and $\boldsymbol{z}^{t+\frac{1}{2}}$ are auxiliary variables

More explicit expression for D-R splitting

or equivalently,

$$
\begin{aligned}
\boldsymbol{x}^{t+\frac{1}{2}} &= \mathcal{R}_{\mathcal{B}}(\boldsymbol{z}^t) \\
\boldsymbol{x}^{t+1} &= \mathcal{R}_{\mathcal{A}}\big(2\boldsymbol{x}^{t+\frac{1}{2}} - \boldsymbol{z}^t\big) \\
\boldsymbol{z}^{t+1} &= \boldsymbol{z}^t + \boldsymbol{x}^{t+1} - \boldsymbol{x}^{t+\frac{1}{2}}
\end{aligned}
$$

Douglas-Rachford primal-dual splitting

$$\text{minimize}_{\boldsymbol{x}} \ \max_{\boldsymbol{\lambda}} \ f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} \rangle - h^*(\boldsymbol{\lambda})$$

Applying Douglas-Rachford splitting to (9.5) yields

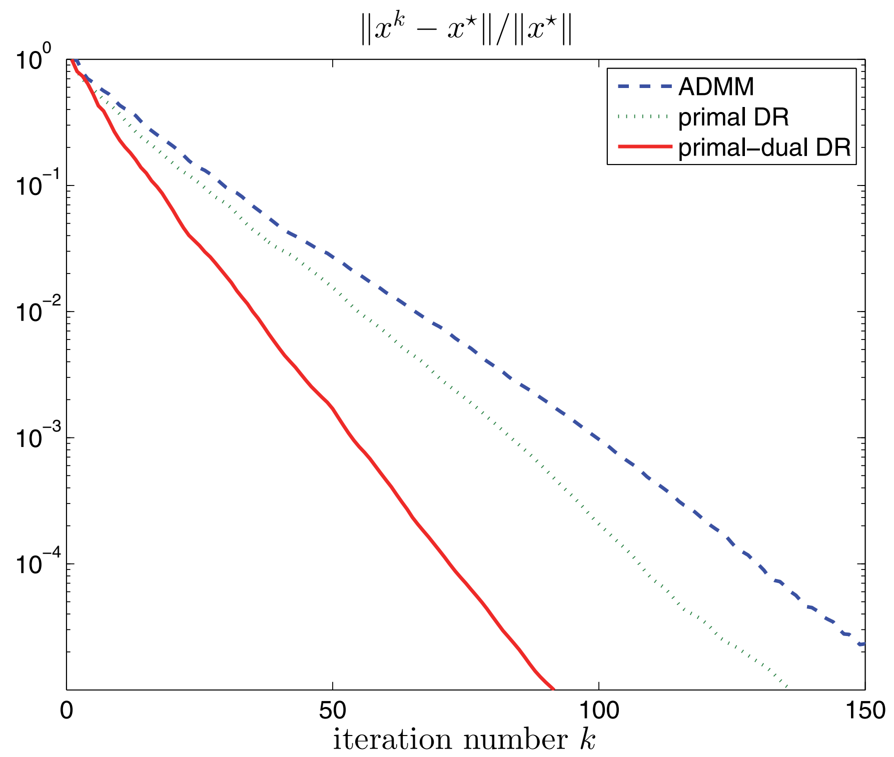
$$\begin{aligned}
\boldsymbol{x}^{t+\frac{1}{2}} &= \text{prox}_{\eta f}(\boldsymbol{p}^t) \\
\boldsymbol{\lambda}^{t+\frac{1}{2}} &= \text{prox}_{\eta h^*}(\boldsymbol{q}^t) \\
\begin{bmatrix} \boldsymbol{x}^{t+1} \\ \boldsymbol{\lambda}^{t+1} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{I} & \eta \boldsymbol{A}^\top \\ -\eta \boldsymbol{A} & \boldsymbol{I} \end{bmatrix}^{-1} \begin{bmatrix} 2\boldsymbol{x}^{t+\frac{1}{2}} - \boldsymbol{p}^t \\ 2\boldsymbol{\lambda}^{t+\frac{1}{2}} - \boldsymbol{q}^t \end{bmatrix} \\
\boldsymbol{p}^{t+1} &= \boldsymbol{p}^t + \boldsymbol{x}^{t+1} - \boldsymbol{x}^{t+\frac{1}{2}} \\
\boldsymbol{q}^{t+1} &= \boldsymbol{q}^t + \boldsymbol{\lambda}^{t+1} - \boldsymbol{\lambda}^{t+\frac{1}{2}}
\end{aligned}$$

Example

$$\text{minimize}_{\boldsymbol{x}} \quad \|\boldsymbol{x}\|_2 + \gamma \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_1$$

$$\Longleftrightarrow \qquad \text{minimize}_{\boldsymbol{x}} \quad f(\boldsymbol{x}) + g(\boldsymbol{A}\boldsymbol{x})$$

with $f(\boldsymbol{x}) := \|\boldsymbol{x}\|_2$ and $g(\boldsymbol{y}) := \gamma \|\boldsymbol{y} - \boldsymbol{b}\|_1$



— *Connor, Vandenberghe '14*

Reference

[1] "*A First Course in Convex Optimization Theory*," E. Ryu, W. Yin.

[2] "*Optimization methods for large-scale systems, EE236C lecture notes*," L. Vandenberghe, UCLA.

[3] "*Convex optimization, EE364B lecture notes*," S. Boyd, Stanford.

[4] "*Mathematical optimization, MATH301 lecture notes*," E. Candes, Stanford.

[5] "*First-order methods in optimization*," A. Beck, Vol. 25, SIAM, 2017.

[6] "*Primal-dual decomposition by operator splitting and applications to image deblurring*," D. O' Connor, L. Vandenberghe, *SIAM Journal on Imaging Sciences*, 2014.

[7] "*On the numerical solution of heat conduction problems in two and three space variables*," J. Douglas, H. Rachford, *Transactions of the American mathematical Society*, 1956.

Reference

[8] "*A primer on monotone operator methods*," E. Ryu, S. Boyd, *Appl. Comput. Math.*, 2016.