ADMM and Primal-Dual Methods

Wednesday, April 21, 2021 10:12 AM

Recall: $(I+\partial f)^{-1}y = \text{prox}_f(y)$. Why? let $x = \text{prox}_f(y) = \arg\min_x f(x) + \frac{1}{2}\|x-y\|^2$

so optimality condition

$$0 \in \partial f(x) + x - y$$

i.e., $y \in (I+\partial f)(x)$

i.e., $x = (I+\partial f)^{-1}(y)$

since $\partial(f + \frac{1}{2}\|\cdot - y\|^2) = \partial f + \nabla \frac{1}{2}\|\cdot - y\|^2$ since CQ hold because $\frac{1}{2}\|\cdot - y\|^2$ has full domain

ADMM (recap from last time)

motivation: $\min F(x) + G(x)$

$$\min_{x,z} F(x) + G(z) \quad \text{s.t. } Ax + Bz = c$$

make augmented Lagrangian

$$\mathcal{L}_\rho(\underbrace{x, z}_{\text{primal}}, \underbrace{y}_{\text{dual}}) = F(x) + G(z) + \langle y, Ax + Bz - c\rangle + \underbrace{\rho/2 \|Ax + Bz - c\|^2}_{\text{augmented term}}$$

ADMM algo

$$x_{k+1} \in \arg\min_x \mathcal{L}_\rho(x, z_k, y_k)$$
$$z_{k+1} \in \arg\min_z \mathcal{L}_\rho(x_{k+1}, z, y_k)$$

} if we jointly minimized, this is Aug. Lagr.

$$y_{k+1} = y_k + \rho(Ax_{k+1} + Bz_{k+1} - c)$$ dual gradient ascent

What about $\min_x \sum_{i=1}^n f_i(x)$

one idea: $\min_{x_i, i=1,\dots,n} \sum f_i(x_i)$ s.t. linear constraints enforcing $x_i = x_j$

but solving via Gauss-Seidel ... doesn't work

instead, use consensus ideas

let $\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$, and z (same size as x_i),

$$\min_{\vec{x}, z} \underbrace{F(\vec{x})}_{\sum f_i(x_i)} + \underbrace{G(z)}_{0}$$

$$\text{s.t. } \underbrace{\begin{bmatrix} I & & & -I \\ & I & & -I \\ & & \ddots & \vdots \\ & & I & -I \end{bmatrix}}_{n} \cdot \begin{bmatrix} \vec{x} \\ z \end{bmatrix} = 0$$

$A = I$

$B = \begin{bmatrix} -I \\ \vdots \\ -I \end{bmatrix}$

enforces $x_i = z \Rightarrow x_i = x_j$

$$\vec{x}_{k+1} \in \arg\min_{\vec{x}} \mathcal{L}_\rho(\vec{x}, z_k, y_k)$$ } decouples, solve for each x_i independently

$$\vec{z}_{k+1} \in \arg\min_z \mathcal{L}_\rho(x_{k+1}, z, y_k) = \frac{1}{n}\sum_{i=1}^n x_i$$ } averaging, "consensus"

y_{k+1} ... as usual

Douglas-Rachford (equivalent to ADMM in certain senses)

ref. see Bauschke, Combettes '17 §28.3

(P) $\min_x f(x) + g(x)$ (D) $\min_u f^*(-u) + g^*(u)$

Algo: $0<\lambda<2$, $\rho>0$, y_0

- $x_k = \mathrm{prox}_{\rho g}(y_k)$
- $z_k = \mathrm{prox}_{\rho f}(2x_k - y_k)$
- $y_{k+1} = y_k + \lambda(z_k - y_k)$

Thm $f, g \in \Gamma_0(\mathbb{R}^n)$, assume optimal sol'n exists and CQ hold, then $y_k \to y$, $x = \mathrm{prox}_{\rho g}(y)$ is primal optimal, $u = \rho^{-1}(y-x)$ is dual optimal.

Derivation

Optimality: $0 \in \partial(f+g)(x) \overset{\text{via CQ}}{=} \partial f(x) + \partial g(x)$

or equiv., $0 \in \rho \cdot \partial f(x) + \rho\partial g(x)$ for a fixed $\rho > 0$

$$-\rho \cdot \partial g(x) \in \rho \cdot \partial f(x)$$
$$x - \rho \cdot \partial g(x) \in x + \rho\partial f(x)$$
$$2x - x - \rho \cdot \partial g(x) \in x + \rho\partial f(x)$$
$$2x - (I + \rho\partial g)(x) \in (I + \rho\partial f)(x)$$

invert:

$$x = \underbrace{(I+\rho\partial f)^{-1}\big(2x - \underbrace{(I+\rho\partial g)(x)}_{y}\big)}_{z}$$

so $x = \mathrm{prox}_{\rho g}(y)$

$z = \mathrm{prox}_{\rho f}(2x - y)$

solve $x = z$

i.e. $0 = z - x$ $\lambda > 0$

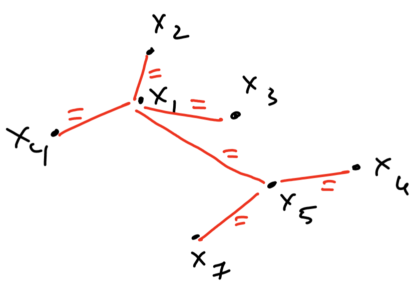
$0 = \lambda \cdot (z - x)$ i.e., $y = y + \lambda \cdot (z - x)$ Fixed pt. iteration

Similar consensus tricks

$$\min \sum f_i(x) \iff \min \sum_i f_i(x_i) \quad \text{s.t. } x_i = x_j \ \forall i,j$$

different ways to encode

$x_i = x_{i+1}$ OR $x_i = x_n$



History/extensions due to Lions + Mercier '79, motivated by PDEs

use "resolvents" $J_A := (I + A)^{-1}$ which is just the prox if $A = \partial f$

A: operator, not necessarily a matrix

see Ernest Ryu's arXiv 1802.07534

"Uniqueness of Douglas-Rachford Splitting as the Two Operator Resolvent Splitting and Impossibility of Three operator resolvent splitting"

we're about to see a 3 operator splitting, but it will involve 2 resolvents and 1 forward (gradient) step.

Primal Dual methods

issue w/ ADMM: $\overbrace{\min_x \; g(x) + \tilde{h}(A\cdot x)}^{\text{D-R}}$, $h(x) = \tilde{h}(A\cdot x)$ $\quad \overbrace{\min_{x,z} \; g(x) + \tilde{h}(z)}^{\text{ADMM}}$ $\quad Ax - z = 0$

issue: need $\text{prox}_h(\ldots)$, often hard due to A

(since $\text{prox}_{\tilde{h}}$ easy $\not\Rightarrow$ prox_h is easy)

want an algo involving $\text{prox}_{\tilde{h}}$ not prox_h

motivate Aug. Lagr.

Trick: use ADMM w/ a scaled norm: $\quad \min_{Ax-z=0} \; g(x) + \tilde{h}(z) + \rho/2 \underbrace{\|Ax - z\|^2}_{\text{use a diff't norm!}}$

Clever choice: z update uses

$$\|z\|_M^2 = \langle z | M | z \rangle, \quad M = \sigma^{-1} I - A^T A$$

$$\sigma < 1/\|A\|^2 \Rightarrow M \succ 0$$

introduced in 2000's

Chambolle and Pock, "primal-dual hybrid gradient", preconditioned ADMM

refs: $\underbrace{\text{Chambolle and Pock}}_{\text{paper}}$

- Cevher, Becker, Schmidt '14 review paper, see more refs on course page.

general primal-dual method (Condat '11) $\qquad f, g, h$ convex, proper, lsc

$$\min_x \; \underset{\substack{\text{smooth}\\ \nabla f}}{f(x)} + \underset{\substack{\text{easy prox}\\ \text{prox}_g}}{g(x)} + \underset{\substack{\text{easy prox}\\ \text{prox}_h}}{h(Ax)}$$

Avoid $\text{prox}_{h\circ A}$!

Note: $x = \text{prox}_h(x) + \text{prox}_{h^*}(x)$

one view: $\quad h(w) = h^{**}(w) = \sup_y \; \langle w, y \rangle - h^*(y)$

So Solve

$$\min_x \max_y \; f(x) + g(x) + \underbrace{\langle Ax, y \rangle}_{\text{links primal and dual variables}} - h^*(y) \qquad (\text{saddle-pt. problem})$$

Optimality: use Fenchel-Rockafellar, or, rederive it:

$$0 \in \partial(f + g + h\circ A)(x), \text{ assume CQ hold so this is}$$

(CQ = constraint qualification)

$$0 \in \nabla f(x) + \partial g(x) + A^T \underbrace{\partial h(Ax)}_{y}$$

$y \in \partial h(Ax)$

or $Ax \in \partial h^{-1}(y)$

i.e. $Ax \in \partial h^*(y)$

2 optimality eq'n

$$0 \in \nabla f(x) + \partial g(x) + A^T y$$

$$Ax \in \partial h^*(y)$$

rewrite:

$$-\begin{bmatrix} \nabla f & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \in \begin{bmatrix} \partial g & A^T \\ -A & \partial h^* \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

$$0 \in T_1 \vec{x} + T_2 \vec{x}$$

$$\vec{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\underbrace{\qquad}_{T_2} \qquad \underbrace{\qquad}_{T_1}$$

Solve via forward-backward (= proximal descent)

assume T_2 is 1-Lipschitz

$$\vec{x}_{k+1} = \underbrace{(I + T_1)^{-1}}_{\text{backward / implicit}} \underbrace{(I - T_2)}_{\text{forward / explicit}} \cdot \vec{x}_k$$

Assuming

- $\underbrace{(I + \partial g)^{-1}}_{= \text{prox}_g}$ easy
- $(I + \partial h^*)^{-1}$ easy

Need a fix. Trick: derived by $-T_2 \vec{x} \in T_1 \vec{x}$

$$\vec{x} - T_2 \vec{x} \in \vec{x} + T_1 \vec{x}$$ ← Option 1 (standard, not helpful)

instead, do $$V \cdot \vec{x} - T_2 \vec{x} \in V \cdot \vec{x} + T_1 \cdot \vec{x}$$ ← Option 2 (helpful)

cleverly chosen

i.e., $\vec{x}_{k+1} = (V + T_1)^{-1} (V - T_2) \cdot \vec{x}_k \qquad V = \begin{bmatrix} \tau^{-1} \cdot I & -A^T \\ -A & \sigma^{-1} \cdot I \end{bmatrix}$

$$\left(\begin{bmatrix} \tau^{-1} I & -A^T \\ -A & \sigma^{-1} I \end{bmatrix} + \begin{bmatrix} \partial g & A^T \\ -A & \partial h^* \end{bmatrix} \right)^{-1} = \begin{bmatrix} \tau^{-1} I + \partial g & 0 \\ -2A & \sigma^{-1} I + \partial h^* \end{bmatrix}^{-1}$$

observe: it's triangular!

$$\begin{pmatrix} \tau^{-1} I + \partial g & 0 \\ -2A & \sigma^{-1} I + \partial h^* \end{pmatrix} \begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} v \\ w \end{pmatrix}$$

↑ solve for

Solve: $x_{k+1} = (\tau^{-1} I + \partial g)^{-1} v$ via $\text{prox}_{\partial g}$

then solve for y_{k+1} (forward substitution)

need $V \succ 0$, guaranteed if $\sigma \tau > \|A\|^{-2}$

Altogether

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} \tau^{-1} I + \partial g & 0 \\ -2A & \sigma^{-1} I + \partial h^* \end{bmatrix}^{-1} \underbrace{\begin{bmatrix} \tau^{-1} I - \nabla f & -A^T \\ -A & \sigma^{-1} I \end{bmatrix} \cdot \begin{bmatrix} x_k \\ y_k \end{bmatrix}}$$

$$= \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} \frac{1}{\tau} x_k - \nabla f(x_k) \\ -A x_k + 1/\sigma \, y_k \end{bmatrix}$$

$$\begin{array}{l} (1) \\ (2) \end{array} \begin{bmatrix} \tau^{-1} I + \partial g & 0 \\ -2A & \sigma^{-1} I + \partial h^* \end{bmatrix} \begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} v \\ w \end{bmatrix}$$

So (1) $(I + \tau \partial g)(x_{k+1}) = \tau \cdot v$

① $x_{k+1} = \text{prox}_{\tau g}(\tau \cdot v) = \text{prox}_{\tau g}\left(x_k - \tau \cdot \nabla f(x_k) \right)$

then (2) $(I + \sigma \partial h^*)(y_{k+1}) = \sigma \cdot w + 2 \sigma A \cdot x_{k+1}$

$$y_{k+1} = \text{prox}_{\sigma h^*}(\sigma \cdot w + 2 \sigma A x_{k+1})$$

② $$= \text{prox}_{\sigma h^*}\left(y_k + \sigma \cdot A (2 x_{k+1} - x_k) \right)$$