Lecture 19

Last time

- ADMM
- Examples

Today

- Preconditioning
- PDHG
- Primal-dual guarantee

Preconditioning operators

We go back to a template we considered for Fenchel duality

$$(P) \quad \inf_{x \in E} f(x) + g(Ax).$$

linear $A : E \to Y$

closed, convex, proper

We could apply ADMM, but last time we saw that this would imply solving a linear system involving A, which might be prohibitively expensive!

Instead, we take an alternative path. Recall that its dual was

$$\text{(D)} \quad \sup_{y \in Y} -f^*(-A^*y) - g^*(y).$$ changed a sign

Moreover, a pair $(\bar{x}, \bar{y})$ are primal-dual solutions iff

$$-A^*\bar{y} \in \partial f(x) \text{ and } \bar{y} \in \partial g(A\bar{x}),$$

$$\Longleftrightarrow$$

$$-A^*\bar{y} \in \partial f(x) \text{ and } A\bar{x} \in \partial g^*(\bar{y}),$$

which means, the solution is a zero

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \in \left(\begin{bmatrix} 0 & A^* \\ -A & 0 \end{bmatrix} + \begin{bmatrix} \partial f \\ \partial g^* \end{bmatrix} \right) \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix}.$$

We could aim to apply the KM

iteration by considering R_T

$$\underbrace{\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix}}_{z_{k+1}} \leftarrow R_T\left(\begin{bmatrix} x_k \\ y_k \end{bmatrix}\right)$$

or equivalently

$$\begin{bmatrix} x_k - x_{k+1} \\ y_k - y_{k+1} \end{bmatrix} \in \begin{bmatrix} A^* y_{k+1} + \partial f(x_{k+1}) \\ -A x_{k+1} + \partial g^*(y_{k+1}) \end{bmatrix}$$

This algorithm is mostly conceptual as it requires evaluating a resolvant depending on both primal and dual variables.

Instead we use the following simple, but remarkably useful observation.

Proposition: Suppose $T: E \rightrightarrows E$ is a monotone operator with a zero. Consider a sequence defined via

$\alpha \geq 0$

(⌂) $$H(z_k - z_{k+1}) \in \alpha T z_{k+1}$$

with $H : E \to E$ a linear, self-adjoint, positive definite. Then, the sequence converges $z_k \to z^*$ to a zero of T.

Proof: The map $\mathcal{H}$ has a square root, call it W, with $\mathcal{H} = W \circ W$. Make the change of variable $w \leftarrow Wz$. Then

$$W \circ W(z_k - z_{k+1}) \in \alpha T z_{k+1}$$

$$\Updownarrow$$

$$W(w_k - w_{k+1}) \in \alpha T W^{-1} z_{k+1}$$

$$\Updownarrow$$

$$(w_k - w_{k+1}) \in \alpha \overbrace{W^{-1} T W^{-1}}^{S} w_{k+1}.$$

Then, it is easy to see that S is monotone:

$$\begin{aligned}\langle S(w) - S(w'), w - w' \rangle &= \langle T(L^{-1}w) - T(L^{-1}w'), L^{-1}(w - w') \rangle \\ &\overset{\text{monotonicity of } T}{\geq} 0.\end{aligned}$$

Then, by the KM iteration we have $w_k \to w^*$ with $Sw^* = 0$ and $z_k = W^{-1} w_k \to W^{-1} w^* =: z^*$ with

$$0 = S w^* = W^{-1} T W^{-1} w^* = W^{-1} T z^*$$

$$\Longleftrightarrow$$

$$0 = W0 = Tz^*. \qquad \square$$

Thus, if we find a good $\mathcal{H}$ to make the algorithm (A) implementable.

Primal-Dual Hybrid Gradient

In 2011, Chambolle and Pock came up with a simple $\mathcal{H}$

$$\mathcal{H} = \begin{pmatrix} \frac{1}{\tau} I & -A^* \\ -A & \frac{1}{s} I \end{pmatrix} \quad \text{for } \tau, s > 0$$

Claim: $\mathcal{H}$ is positive definite as long as $\tau s \|A\|_{op}^2 < 1$.

$\sup_{x \in E} \|Ax\|$

This $\mathcal{H}$ induces the following update

$$\begin{bmatrix} \frac{1}{\tau} I & -A^* \\ -A & \frac{1}{s} I \end{bmatrix} \begin{bmatrix} x_k - x_{k+1} \\ y_k - y_{k+1} \end{bmatrix} \in \begin{bmatrix} A^* y_{k+1} + \partial f(x_{k+1}) \\ -A x_{k+1} + \partial g^*(y_{k+1}) \end{bmatrix}$$

equivalently

$$\frac{1}{\tau}(x_k - x_{k+1}) - A^*(y_k - y_{k+1}) \in A^* y_{k+1} + \partial f(x_{k+1})$$

$$-A(x_k - x_{k+1}) + \frac{1}{s}(y_k - y_{k+1}) \in -A x_{k+1} + \partial g^*(y_{k+1})$$

rearranging

$$x_k - \tau A^* y_k \in x_{k+1} + \tau \partial f(x_{k+1})$$
$$y_k + sA(2x_{k+1} - x_k) \in y_{k+1} + s\,\partial g(x_{k+1})$$

or equivalently

$$x_{k+1} \leftarrow \text{prox}_{\tau f}(x_k - \tau A^* y_k)$$
$$y_{k+1} \leftarrow \text{prox}_{sg^*}(y_k + sA(2x_{k+1} - x_k)).$$

Remarkably each step is now implementable, provided that we can compute the prox of f and g^*.

Example

If we take $f = \langle c, \cdot \rangle + \iota_{\mathbb{R}^n_+}(\cdot)$
$g = \iota_{\{\cdot = b\}}(\cdot)$. Then, we obtain

$$\text{prox}_{\tau f}(x) = \text{proj}_{\mathbb{R}^n_+}(x - \tau c)$$
$$\text{prox}_{sg^*}(y) = y - \text{prox}_{sg}(y/s)$$
$$= y - b.$$

This involves no linear systems!

Google, Nvidia, Gurobi and other big companies have implemented a solver known as PDLP based on this update (plus many other enhancements).

A primal-dual guarantee.

Our nonasymptotic convergence rate only ensured that

$$\min \|z_k - z_{k+1}\| = O\left(\frac{1}{\sqrt{k}}\right)$$

For primal-dual problems this might be a good progress metric.

Problems (P)-(D) can also be formulated as a minimax problem

$$\inf_x \sup_y \underbrace{f(x) + \langle Ax, y\rangle - h^*(y)}_{L(x,y)}$$

Indeed a simple computation yields

$$\underbrace{L(x^*, y) \leq L(x^*, y^*)}_{\text{primal-dual solution}} \leq L(x, y^*)$$

Thus, a natural measure of fitness is the primal-dual gap

$$L(x, y^*) - L(x^*, y).$$