Dual First-Order and ADMM Methods for LP

Yinyu Ye

Department of Management Science and Engineering

Stanford University

Stanford, CA 94305, U.S.A.

http://www.stanford.edu/~yyye

(LY: 14.5-14.6)

The Lagrangian Function and Method

We consider

$$f^* := \min \quad f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{h}(\mathbf{x}) = \mathbf{0},\ \mathbf{x} \in \mathbf{X}. \tag{1}$$

Recall that the Lagrangian function:

$$\underline{L(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) - \mathbf{y}^T \mathbf{h}(\mathbf{x}).} \quad \sum y_i h_i(x)$$

and the dual function:

$$\phi(\mathbf{y}) = \min_{\mathbf{x} \in X} L(\mathbf{x}, \mathbf{y}); \tag{2}$$

and the dual problem

$$\underline{(f^* \geq) \phi^* := \max \quad \phi(\mathbf{y}).} \tag{3}$$

In many cases, one can find $\mathbf{y}^*$ of dual problem (3), a unconstrained optimization problem; then go ahead to find $\mathbf{x}^*$ using (2).

The Local Duality Theorem

Suppose $\mathbf{x}^*$ is a local minimizer, and consider the localized (convex) problem

$$f(\mathbf{x}^*) := \min \quad f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{h}(\mathbf{x}) = \mathbf{0},\ \mathbf{x} \in \mathbf{X},\ \|\mathbf{x} - \mathbf{x}^*\|^{\mathbf{2}} \le \epsilon. \tag{4}$$

Then, the localized Lagrangian function:

$$L_{\mathbf{x}^*}(\mathbf{x}, \mathbf{y}, \mu(\le 0)) = f(\mathbf{x}) - \mathbf{y}^T\mathbf{h}(\mathbf{x}) - \mu(\|\mathbf{x} - \mathbf{x}^*\|^{\mathbf{2}} - \epsilon).$$

and the localized dual function:

$$\phi_{\mathbf{x}^*}(\mathbf{y}, \mu) = \min_{\mathbf{x} \in X,\ \|\mathbf{x} - \mathbf{x}^*\|^2 \le \epsilon} L_{\mathbf{x}^*}(\mathbf{x}, \mathbf{y}, \mu); \tag{5}$$

and the localized dual problem

$$\max \quad \phi(\mathbf{y}, \mu \le 0). \tag{6}$$

Under certain constraint qualification and local convexity conditions, we must have $\underline{f(\mathbf{x}^*) = \phi(\mathbf{y}^*, \mu^* = 0)}$ where the localization constraint becomes inactive.

The gradient and Hessian of ϕ

Let $\mathbf{x}(\mathbf{y})$ be a minimizer of (2). Then

$$x(y) = \arg\min_{x} L(x, y)$$

$$\phi(\mathbf{y}) = \underline{f(\mathbf{x}(\mathbf{y})) - \mathbf{y}^T\mathbf{h}(\mathbf{x}(\mathbf{y}))}$$

Thus,

$$\begin{aligned}\boxed{\nabla\phi(\mathbf{y})} &= \nabla f(\mathbf{x}(\mathbf{y}))^T\nabla\mathbf{x}(\mathbf{y}) - \mathbf{y}^T\nabla\mathbf{h}(\mathbf{x}(\mathbf{y}))\nabla\mathbf{x}(\mathbf{y}) - \mathbf{h}(\mathbf{x}(\mathbf{y}))\\ &= (\nabla f(\mathbf{x}(\mathbf{y}))^T - \mathbf{y}^T\nabla\mathbf{h}(\mathbf{x}(\mathbf{y})))\nabla\mathbf{x}(\mathbf{y}) - \mathbf{h}(\mathbf{x}(\mathbf{y}))\\ &\boxed{= -\mathbf{h}(\mathbf{x}(\mathbf{y})).}\end{aligned}$$

Similarly, we can derive

$$\nabla^2\phi(\mathbf{y}) = -\nabla\mathbf{h}(\mathbf{x}(\mathbf{y}))\left(\nabla^2_{\mathbf{x}}\mathbf{L}(\mathbf{x}(\mathbf{y}),\mathbf{y})\right)^{-1}\nabla\mathbf{h}(\mathbf{x}(\mathbf{y}))^{\mathbf{T}},$$

where $\nabla^2_{\mathbf{x}}L(\mathbf{x}(\mathbf{y}),\mathbf{y})$ is the Hessian of the Lagrangian function that is assumed to be positive definite at any (local) minimizer.

The Toy Example

$$\begin{array}{ll} \text{minimize} & \underline{(x_1-1)}^2+\underline{(x_2-1)}^2 \\ \text{subject to} & \underline{x_1+2x_2-1}=0, \quad \underline{2x_1+x_2-1}=0. \end{array}$$

$$L(\mathbf{x},\mathbf{y})=(x_1-1)^2+(x_2-1)^2-y_1(x_1+2x_2-1)-y_2(2x_1+x_2-1).$$

$$x_1=0.5y_1+y_2+1, \quad x_2=y_1+0.5y_2+1.$$

$$\phi(\mathbf{y})=\underline{-1.25y_1^2-1.25y_2^2-2y_1y_2-2y_1-2y_2.}$$

$$\nabla\phi(\mathbf{y})=\begin{pmatrix} 2.5y_1+2y_2+2 \\ 2y_1+2.5y_21+2 \end{pmatrix},$$

$$\boxed{\nabla^2\phi(\mathbf{y})}=-\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}^{-1}\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}^T=-\begin{pmatrix} 2.5 & 2 \\ 2 & 2.5 \end{pmatrix}$$

The Augmented Lagrangian Function

$\min f(x)$

$\text{s.t. } h(x)=0$

In both theory and practice, we actually consider an augmented Lagrangian function (ALF)

$$\phi_\beta(y) = \min_x \quad L_a(\mathbf{x},\mathbf{y}) = f(\mathbf{x}) - \mathbf{y}^T\mathbf{h}(\mathbf{x}) + \frac{\beta}{2}\|\mathbf{h}(\mathbf{x})\|^2,$$

which corresponds to an equivalent problem of (1):

$$f^* := \min \quad \boxed{f(\mathbf{x}) + \frac{\beta}{2}\|\mathbf{h}(\mathbf{x})\|^2} \quad \text{s.t.} \quad \underline{\mathbf{h}(\mathbf{x}) = \mathbf{0}}, \ \mathbf{x} \in \mathbf{X}.$$

Note that, although at feasibility the additional square term in objective is redundant, it helps to improve strict convexity of the Lagrangian function.

The Augmented Lagrangian Dual

Now the dual function:

$$\phi_{\overset{\beta}{a}}(\mathbf{y}) = \min_{\mathbf{x}\in X} L_{\overset{\beta}{a}}(\mathbf{x},\mathbf{y}); \tag{7}$$

and the dual problem

$$(f^* \geq)\phi_a^* := \max \quad \phi_{\overset{\beta}{a}}(\mathbf{y}). \tag{8}$$

Note that the dual function satisfies $\frac{1}{\beta}$-Lipschitz condition (see Chapter 14 of L&Y).

For the convex optimization case, say $\mathbf{h}(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}$, we have

$$\nabla^2 L_a(\mathbf{x},\mathbf{y}) = \nabla^2 f(\mathbf{x}) + \beta(A^T A).$$

$L(x,y) = 5\ln(2x_1+x_2) + 8\ln(3x_3+x_4) - y_1(x_1+x_3-1) - y_2(x_2+x_4-1)$

$x_1, x_2, x_3, x_4 \geq 0$

$\frac{10}{2x_1+x_2} - y_1 \leq 0 \quad \wedge \quad x_1$

$\frac{5}{2x_1+x_2} - y_2 \leq 0 \quad \wedge \quad x_2$

The Augmented Lagrangian Method

The augmented Lagrangian method (ALM) is:

Start from any $(\mathbf{x}^0 \in X, \mathbf{y}^0)$, we compute a new iterate pair

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}\in X} L_a(\mathbf{x}, \mathbf{y}^k), \text{ and } \mathbf{y}^{k+1} = \mathbf{y}^k - \beta \mathbf{h}(\mathbf{x^{k+1}}).$$

The calculation of $\mathbf{x}$ is used to compute the gradient vector of $\phi_{\beta}(\mathbf{y})$, which is a steepest ascent direction.

The method converges just like the SDM, because the dual function satisfies $\frac{1}{\beta}$-Lipschitz condition.

Other SDM strategies may be adapted to update $\mathbf{y}$ (the BB, ASDM, Conjugate, Quasi-Newton ...).

Analysis of the Augmented Lagrangian Method

Consider the convex optimization case $\mathbf{h}(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}$. Since $\mathbf{x}^{k+1}$ makes KKT condition:

$$\begin{aligned} \mathbf{0} &= \nabla f(\mathbf{x}^{k+1}) - A^T\mathbf{y}^k + \beta A^T(A\mathbf{x}^{k+1} - \mathbf{b}) \\ &= \nabla f(\mathbf{x}^{k+1}) - A^T(\mathbf{y}^k - \beta(A\mathbf{x}^{k+1} - \mathbf{b})) \\ &= \nabla f(\mathbf{x}^{k+1}) - A^T\mathbf{y}^{k+1}, \end{aligned}$$

we only need to be concerned about whether or not $\|A\mathbf{x}^k - \mathbf{b}\|$ converges to zero and how fast it converges. First, from the convexity of $f(\mathbf{x})$, we have

$$\begin{aligned} \mathbf{0} &\le (\nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k))^T(\mathbf{x}^{k+1} - \mathbf{x}^k) \\ &= (-A^T\mathbf{y}^{k+1} + A^T\mathbf{y}^k)^T(\mathbf{x}^{k+1} - \mathbf{x}^k) \\ &= (\mathbf{y}^{k+1} - \mathbf{y}^k)^T(A\mathbf{x}^{k+1} - A\mathbf{x}^k) \\ &= -\beta(A\mathbf{x}^{k+1} - \mathbf{b})(A\mathbf{x}^{k+1} - \mathbf{b} - (A\mathbf{x}^k - \mathbf{b})), \end{aligned}$$

which implies that $\|A\mathbf{x}^{k+1} - \mathbf{b}\| \le \|A\mathbf{x}^k - \mathbf{b}\|$, that is, the error is non-increasing.

Again, from the convexity, we have

$$\begin{aligned}
\mathbf{0} &\le (\nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^*))^T(\mathbf{x}^{k+1} - \mathbf{x}^*) \\
&= (A^T\mathbf{y}^{k+1} - A^T\mathbf{y}^*)^T(\mathbf{x}^{k+1} - \mathbf{x}^*) \\
&= (\mathbf{y}^{k+1} - \mathbf{y}^*)^T(A\mathbf{x}^{k+1} - A\mathbf{x}^*) = (\mathbf{y}^{k+1} - \mathbf{y}^*)^T(A\mathbf{x}^{k+1} - \mathbf{b}) \\
&= \frac{1}{\beta}(\mathbf{y}^{k+1} - \mathbf{y}^*)^T(\mathbf{y}^k - \mathbf{y}^{k+1}).
\end{aligned}$$

Thus, from the positivity of the cross product, we have

$$\begin{aligned}
\|\mathbf{y}^k - \mathbf{y}^*\|^2 &= \|\mathbf{y}^k - \mathbf{y}^{k+1} + \mathbf{y}^{k+1} - \mathbf{y}^*\|^2 \\
&\ge \|\mathbf{y}^k - \mathbf{y}^{k+1}\|^2 + \|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2 \\
&= \beta\|A\mathbf{x}^{k+1} - \mathbf{b}\|^2 + \|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2.
\end{aligned}$$

Sum up from 0 to k of the inequality we have

$$\begin{aligned}
\|\mathbf{y}^0 - \mathbf{y}^*\|^2 &\ge \|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2 + \beta\sum_{l=0}^{k}\|A\mathbf{x}^{l+1} - \mathbf{b}\|^2 \\
&\ge \beta\sum_{l=0}^{k}\|A\mathbf{x}^{l+1} - \mathbf{b}\|^2 \\
&\ge (k+1)\beta\|A\mathbf{x}^{k+1} - \mathbf{b}\|^2.
\end{aligned}$$

$$\frac{1}{K+1} \lesssim \varepsilon$$

$$K = \frac{1}{\varepsilon}$$

Two-Block Alternating Direction Method with Multipliers

ADMM

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \hline \mathbf{x}_2 \end{pmatrix}$$

For the ADMM method, we consider structured problem $[A_1 \;\; A_2]\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \mathbf{b}$

$$\min \quad \overset{c_1^T \mathbf{x}_1}{\underline{f_1(\mathbf{x}_1)}} + \overset{c_2^T \mathbf{x}_2}{\underline{f_2(\mathbf{x}_2)}} \quad \text{s.t.} \quad \boxed{A_1\mathbf{x}_1 + A_2\mathbf{x}_2 = \mathbf{b}}, \; \underset{?.}{\mathbf{x}_1 \in X_1}, \; \mathbf{x}_2 \in X_2.$$

Consider

$$L(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}) = f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) - \mathbf{y}^T(A_1\mathbf{x}_1 + A_2\mathbf{x}_2 - \mathbf{b}) + \frac{\beta}{2}\|A_1\mathbf{x}_1 + A_2\mathbf{x}_2 - \mathbf{b}\|_2^2.$$

Then, for any given $(\mathbf{x}_1^k, \mathbf{x}_2^k, \mathbf{y}^k)$, we compute a new iterate

$$\left\{\begin{array}{ll} \mathbf{x}_1^{k+1} & = \arg\min_{\mathbf{x}_1 \in X_1} L(\mathbf{x}_1, \mathbf{x}_2^k, \mathbf{y}^k), \\ \mathbf{x}_2^{k+1} & = \arg\min_{\mathbf{x}_2 \in X_2} L(\mathbf{x}_1^{k+1}, \mathbf{x}_2, \mathbf{y}^k), \end{array}\right.$$
$$\mathbf{y}^{k+1} = \mathbf{y}^k - \beta(\underline{A_1\mathbf{x}_1^{k+1} + A_2\mathbf{x}_2^{k+1} - \mathbf{b}}).$$

Again, we can prove that the iterates converge with the same speed.

The ADMM method resembles the Block Coordinate Descent (BCD) Method ...

Direct Application of ADMM to Linear Programming I

Consider the standard-form LP

$$\begin{array}{lll} \text{minimize}_{\mathbf{x}} & \mathbf{c}^T\mathbf{x} & \\ \text{s.t.} & \underline{\boxed{A\mathbf{x} = \mathbf{b},}} & \\ & \boxed{\mathbf{x} \geq \mathbf{0}.} & \end{array} \quad \rightleftarrows \quad \begin{array}{lll} \text{minimize}_{(\mathbf{x}_1, x_2)} & \mathbf{c}^T\mathbf{x}_1 & \\ \text{s.t.} \quad y & \underline{A\mathbf{x}_1 = \mathbf{b},} & \\ s \rightarrow & \underline{\boxed{\mathbf{x}_1 - \mathbf{x}_2 = \mathbf{0},}} & \underline{\mathbf{x}_2 \geq \mathbf{0}.} \end{array}$$

$$L(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}) = \underline{\mathbf{c}^T\mathbf{x}_1} - \mathbf{y}^T\underline{(A\mathbf{x}_1 - \mathbf{b})} - \mathbf{s}^T(\mathbf{x}_1 - \mathbf{x}_2) + \underline{\frac{\beta}{2}\left(\cancel{\|A\mathbf{x}_1 - \mathbf{b}\|^2} + \|\mathbf{x}_1 - \mathbf{x}_2\|^2\right)}.$$

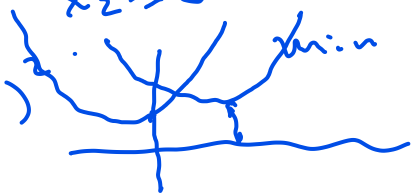
where $\mathbf{y}$ and $\mathbf{s}$ are the multiplier vectors of first and second equality constraints in the reformulation.

The advantage of such splitting reformulation is that the update of either $\mathbf{x}_1$ or $\mathbf{x}_2$ has a simple close form solution.

$\min \; L(x_1, x_2^0, y^0, s^0)$, $\qquad \min \; L(x_1^1, x_2, y^0, s^0)$

$x_2 \geq 0$

$$\min_{x \geq 0} \; \underline{s^T x + \frac{\beta}{2}\|x - c\|^2} \qquad \sum_j s_j x_j + \frac{\beta}{2}(x_j - c_j) \qquad x_j \geq 0$$



$$s^T x_2 + \frac{\beta}{2}\|x_2 - x_1^1\|^2$$

Direct Application of ADMM to Dual Linear Programming I

Consider the dual LP

$$\begin{array}{ll} \text{maximize}_{(\mathbf{y},\mathbf{s})} & \mathbf{b}^T\mathbf{y} \\ \text{s.t.} & \underline{A^T\boxed{\mathbf{y}}+\boxed{\mathbf{s}}=\mathbf{c}}, \ \mathbf{s}\geq \mathbf{0}. \end{array}$$

The augmented Lagrangian function would be

$$L(\underset{s\geq 0}{\mathbf{y},\mathbf{s},\mathbf{x}}) = \underline{-\mathbf{b}^T\mathbf{y}-\mathbf{x}^T(A^T\mathbf{y}+\mathbf{s}-\mathbf{c})+\frac{\beta}{2}\|A^T\underbrace{\mathbf{y}+\mathbf{s}-\mathbf{c}}\|^2,}$$

where β is a positive parameter, and $\mathbf{x}$ is the multiplier vector.

Direct Application of ADMM to Dual Linear Programming II

The ADMM for the dual is straightforward: starting from any $\mathbf{y}^0$, $\mathbf{s}^0 \geq \mathbf{0}$, and multiplier $\mathbf{x}^0$,

- Update variable $\mathbf{y}$:

$$\mathbf{y}^{k+1} = \arg\min_{\mathbf{y}} L(\mathbf{y}, \mathbf{s}^k, \mathbf{x}^k);$$

ⓜ ← y, QP-min

- Update slack variable $\mathbf{s}$:

$$\mathbf{s}^{k+1} = \arg\min_{\mathbf{s}\geq\mathbf{0}} L(\mathbf{y}^{k+1}, \mathbf{s}, \mathbf{x}^k);$$

$s \geq 0$, separable QP

- Update multipliers $\mathbf{x}$:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \beta(A^T\mathbf{y}^{k+1} + \mathbf{s}^{k+1} - \mathbf{c}).$$

Note that the updates of $\mathbf{y}$ is a least-squares problem with constant matrix, and the update of $\mathbf{s}$ has a simple close form. (Also note that $\mathbf{x}$ would be non-positive at the end, since we changed maximization to minimization of the dual.)

To split $\mathbf{y}$ into multi blocks and update cyclically in random order?

Direct Application of ADMM to Dual Linear Programming III

One can also consider to reformulate the dual as

$$\begin{array}{lll} \text{maximize}_{\mathbf{y},\mathbf{s},\mathbf{u}_1,\mathbf{u}_2} & \mathbf{b}^T\mathbf{y} & \\ \text{s.t.} & A_1^T\mathbf{y}_1 - \mathbf{u}_1 = \mathbf{0}, & (\mathbf{v}_1) \\ & A_2^T\mathbf{y}_2 - \mathbf{u}_2 = \mathbf{0}, & (\mathbf{v}_2) \\ & \mathbf{u}_1 + \mathbf{u}_2 + \mathbf{s} = \mathbf{c}, & (\mathbf{x}) \\ & \mathbf{s} \geq \mathbf{0}; & \end{array} \tag{9}$$

with the multiplier $\mathbf{v}_1,\ \mathbf{v}_2$ and $\mathbf{x}$ for the three sets of the equality constraints.

$$\begin{array}{l} L^d(\mathbf{y}_1,\mathbf{y}_2,\mathbf{u}_1,\mathbf{u}_2,\mathbf{s},\mathbf{v}_1,\mathbf{v}_2,\mathbf{x}) \\ = -\mathbf{b}_1^T\mathbf{y}_1 - \mathbf{b}_2^T\mathbf{y}_2 - \mathbf{v}_1^T(A_1^T\mathbf{y}_1 - \mathbf{u}_1) - \mathbf{v}_2^T(A_2^T\mathbf{y}_2 - \mathbf{u}_2) - \mathbf{x}^T(\mathbf{u}_1 + \mathbf{u}_2 + \mathbf{s} - \mathbf{c}) \\ \quad + \frac{\beta}{2}\left(\|A_1^T\mathbf{y}_1 - \mathbf{u}_1\|^2 + \|A_2^T\mathbf{y}_2 - \mathbf{u}_2\|^2 + \|\mathbf{u}_1 + \mathbf{u}_2 + \mathbf{s} - \mathbf{c}\|^2\right). \end{array} \tag{10}$$

Note that $\mathbf{y}_i$, $i = 1, 2$, and $\mathbf{s} \geq \mathbf{0}$ can be updated independently and in parallel, and $\mathbf{u}_i,\ i = 1, 2$, can be updated jointly with a close form(?). This is essentially a two-block ADMM!

Direct Application of ADMM to Dual LP IV: Barrier Regularization

Now consider dual linear program with the logarithmic barrier function

$$
\begin{array}{ll}
\text{maximize}_{\mathbf{y},\mathbf{s}} & \mathbf{b}^T\mathbf{y} + \mu \sum_j \ln(s_j) \\
\text{s.t.} & A^T\mathbf{y} + \mathbf{s} = \mathbf{c}, \quad (x)
\end{array}
\qquad (11)
$$

where μ is a fixed small positive constant.

The augmented Lagrangian function would be

a

$$L_\mu(\mathbf{y}, \mathbf{s}, \mathbf{x}) = -\mathbf{b}^T\mathbf{y} - \mu \sum_j \ln(s_j) - \mathbf{x}^T(A^T\mathbf{y} + \mathbf{s} - \mathbf{c}) + \frac{\beta}{2}\|\underbrace{A^T\mathbf{y} + \mathbf{s} - \mathbf{c}}\|^2,$$

$$\min_{s \geq 0} \underline{-\mu \ln s \cancel{+} xs + (s-a)^2 \frac{1}{2}}$$

Apply the path-following idea to the Dual ADMM with barrier.

$$-\frac{\mu}{s} - \overset{s}{x} \overset{s}{+} (s-a) = 0$$

"An ADMM-Based Interior-Point Method for Large-Scale Linear Programming," (https://arxiv.org/abs/1805.12344).

$$x_1 + x_2 = u_i$$

$$u_i \geq 0$$

ADMM for Multi-block Convex Minimization

Why not consider convex minimization problems with *three blocks*:

$$
\begin{array}{ll}
\min & \underline{\overset{c_1^T x_1}{f_1(\mathbf{x}_1)} + \overset{c_2^T x_2}{f_2(\mathbf{x}_2)} + \overset{c_3^T x_3}{f_3(\mathbf{x}_3)}} \\
\text{s.t.} & A_1\mathbf{x}_1 + A_2\mathbf{x}_2 + A_3\mathbf{x}_3 = \mathbf{b} \\
 & \mathbf{x}_1 \in \mathcal{X}_1,\ \mathbf{x}_2 \in \mathcal{X}_2,\ \mathbf{x}_3 \in \mathcal{X}_3
\end{array}
$$

The direct and natural extension of ADMM with null objectives:

$\left(\frac{n}{3}\right)^3$

70

1: ALM

$\boxed{2: \text{ADMM}}$

$\boxed{3+: \text{MBADMM}}$?

$$
\left\{
\begin{array}{l}
\mathbf{x}_1^{k+1} = \arg\min\{L(\mathbf{x}_1, \mathbf{x}_2^k, \mathbf{x}_3^k, \mathbf{y}^k) \mid \mathbf{x}_1 \in \mathcal{X}_1\} \\
\mathbf{x}_2^{k+1} = \arg\min\{L(\mathbf{x}_1^{k+1}, \mathbf{x}_2, \mathbf{x}_3^k, \mathbf{y}^k) \mid \mathbf{x}_2 \in \mathcal{X}_2\} \\
\mathbf{x}_3^{k+1} = \arg\min\{L(\mathbf{x}_1^{k+1}, \mathbf{x}_2^{k+1}, \mathbf{x}_3, \mathbf{y}^k) \mid \mathbf{x}_3 \in \mathcal{X}_3\} \\
\mathbf{y}^{k+1} = \mathbf{y}^k - \underline{\beta(A_1\mathbf{x}_1^{k+1} + A_2\mathbf{x}_2^{k+1} + A_3\mathbf{x}_3^{k+1} - \mathbf{b})}
\end{array}
\right.
$$

$$
L(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{y}) = \underline{\sum_{i=1}^{3} f_i(\mathbf{x}_i) - \mathbf{y}^T\Big(\sum_{i=1}^{3} A_i\mathbf{x}_i - \mathbf{b}\Big) + \frac{\beta}{2}\Big\|\sum_{i=1}^{3} A_i\mathbf{x}_i - \mathbf{b}\Big\|^2}
$$

2014

Divergent Example of the Extended ADMM I

Should it converge? (Not easy to analyze the convergence of ADMM with more than two blocks; or the proving operator theory of two-block cannot be directly extended to the ADMM with three blocks.)

Consider the system of homogeneous linear equations with three variables and null objective functions:

$$\min \quad \underbrace{0x_1 + 0x_2 + 0x_3}$$

$$\text{s.t.} \quad A_1x_1 + A_2x_2 + A_3x_3 = \mathbf{0}, \text{ where } A = (A_1, A_2, A_3) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 2 \end{pmatrix}.$$

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = 0$$

$\boxed{x^{k+1} = M x^{k}}$ $x^{k+1} = M^{k} x^{0}$

Then the extended ADMM with $\beta = 1$ can be specified as a linear map

$$\begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 4 & 6 & 0 & 0 & 0 & 0 \\ 5 & 7 & 9 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 2 & 0 & 1 & 0 \\ 1 & 2 & 2 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1^{k+1} \\ x_2^{k+1} \\ x_3^{k+1} \\ \mathbf{y}^{k+1} \end{pmatrix} = \begin{pmatrix} 0 & -4 & -5 & 1 & 1 & 1 \\ 0 & 0 & -7 & 1 & 1 & 2 \\ 0 & 0 & 0 & 1 & 2 & 2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1^{k} \\ x_2^{k} \\ x_3^{k} \\ \mathbf{y}^{k} \end{pmatrix}.$$

Divergent Example of the Extended ADMM II

Or equivalently,

$$\begin{pmatrix} x_2^{k+1} \\ x_3^{k+1} \\ \mathbf{y}^{k+1} \end{pmatrix} = M \begin{pmatrix} x_2^{k} \\ x_3^{k} \\ \mathbf{y}^{k} \end{pmatrix},$$

where

$$M = \frac{1}{162}\begin{pmatrix} 144 & -9 & -9 & -9 & 18 \\ 8 & 157 & -5 & 13 & -8 \\ 64 & 122 & 122 & -58 & -64 \\ 56 & -35 & -35 & 91 & -56 \\ -88 & -26 & -26 & -62 & 88 \end{pmatrix}.$$

Divergent Example of the Extended ADMM III

The matrix $M = V\mathrm{Diag}(\mathrm{d})\mathrm{V}^{-1}$, where $d = \begin{pmatrix} 0.9836 + 0.2984i \\ 0.9836 - 0.2984i \\ 0.8744 + 0.2310i \\ 0.8744 - 0.2310i \\ 0 \end{pmatrix}$. Note that $\rho(M) = |d_1| = \underline{|d_2| > 1}.$

Theorem 1 *There existing an example where the direct extension of ADMM of three blocks with a real number initial point is not necessarily convergent for any choice of* β*. Moreover, for any randomly generated initial point, ADMM diverges with probability* 1*.*

Multi-block problems and ADMM

In general, consider a convex optimization problem

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in R^N} & f_1(\mathbf{x}_1)+\cdots+f_n(\mathbf{x}_n), \\ \text{subject to} & A\mathbf{x} \triangleq A_1\mathbf{x}_1+\cdots+A_n\mathbf{x}_n=\mathbf{b}, \\ & \mathbf{x}_i\in\mathcal{X}_i\subset R^{d_i},\ i=1,\ldots,n. \end{array} \tag{12}$$

$$L(\mathbf{x}_1,\ldots,\mathbf{x}_n;\mathbf{y})=\sum_i f_i(x_i)-\mathbf{y}^T(\sum_i A_i\mathbf{x}_i-\mathbf{b})+\frac{\beta}{2}\|\sum_i A_i\mathbf{x}_i-\mathbf{b}\|^2$$

The direct Cyclic Extension Multi-block ADMM:

n=3

3!

$$\begin{cases} \mathbf{x}_1 \longleftarrow \arg\min_{\mathbf{x}_1\in\mathcal{X}_1} L(\mathbf{x}_1,\ldots,\mathbf{x}_n;\mathbf{y}), \\ \qquad\vdots \\ \mathbf{x}_n \longleftarrow \arg\min_{\mathbf{x}_n\in\mathcal{X}_n} L(\mathbf{x}_1,\ldots,\mathbf{x}_n;\mathbf{y}), \\ \mathbf{y} \longleftarrow \mathbf{y}-\beta(A\mathbf{x}-\mathbf{b}), \end{cases}$$

How to Make it Work?

There are many “correction” methods to deal with the problem, but ... Extra work

Is there a “simple way” to make the ADMM with the multi-block work?

Permute the updating order of $\mathbf{x}_i$ randomly, and it works for the example – the expected $\rho(M)$ equals 0.9723!

$$M = \begin{pmatrix} 1.5 & & 0 \\ & 0.5 & \\ 0 & & 0.5 \end{pmatrix}, \quad M = \begin{pmatrix} 0.5 & & \\ & 1.5 & \\ & & 0.5 \end{pmatrix}$$

$$\frac{1}{3}\begin{pmatrix} 2.5 & & 0 \\ & 2.5 & \\ 0 & & 2.5 \end{pmatrix} < 1 \qquad \begin{pmatrix} 0.5 & & \\ & 0.5 & \\ & & 1.5 \end{pmatrix} \rightarrow$$

Randomly Permuted ADMM

Random-Permuted ADMM (RP-ADMM): each round, draw a random permutation $\sigma = (\sigma(1), \ldots, \sigma(n))$ of $\{1, \ldots, n\}$, and

$$\text{Update } \mathbf{x}_{\sigma(1)} \to \mathbf{x}_{\sigma(2)} \to \cdots \to \mathbf{x}_{\sigma(n)} \to \mathbf{y}.$$

(This is sample without replacement.)

Interpretation: Force "absolute fairness" among blocks.

Simulation Test Result on solving linear equations: always converges!

1 2 3

Any theory behind the success?

$\rho(M_\sigma) > 1$

We produced a positive result for ADMM on solving the system of linear equations.

$x^{k+1} = M_\sigma x^k$

$\rho\left(\frac{1}{\sigma}\sum_\sigma M_\sigma\right) < 1$

Random Permuted ADMM for Linear Systems

- Consider solving any square system of linear equation ($f_i = 0,\ \forall i$).

$$\begin{array}{lll} \text{(S)} & \min_{\mathbf{x}\in R^N} & 0, \\ & \text{s.t.} & A_1\mathbf{x}_1 + \cdots + A_n\mathbf{x}_n = \mathbf{b}, \end{array}$$

where $A = [A_1, \ldots, A_n] \in R^{N\times N}$ is full-rank, $\mathbf{x}_i \in R^{d_i}$ and $\sum_i d_i = N$.

- RP-ADMM: Pick a permutation σ of $\{1, \ldots, n\}$ uniformly at random, then compute (with $\beta = 1$) $\mathbf{x}^{k+1}_{\sigma(1)}, \ldots, \mathbf{x}^{k+1}_{\sigma(n)}, \mathbf{y}^{k+1}$ by

$$\begin{cases} -A^T_{\sigma(1)}\mathbf{y}^k + A^T_{\sigma(1)}(A_{\sigma(1)}\mathbf{x}^{k+1}_{\sigma(1)} + \sum_{l=2}^n A_{\sigma(l)}\mathbf{x}^k_{\sigma(l)} - \mathbf{b}) = \mathbf{0}, \\ \ldots \\ -A^T_{\sigma(n)}\mathbf{y}^k + A^T_{\sigma(n)}(\sum_{j=1}^{n-1} A_{\sigma(j)}\mathbf{x}^{k+1}_{\sigma(j)} + A_{\sigma(n)}\mathbf{x}^{k+1}_{\sigma(n)} - \mathbf{b}) = \mathbf{0}, \\ \mathbf{y}^{k+1} = \mathbf{y}^k - (\sum_{i=1}^n A_i\mathbf{x}^{k+1}_i - \mathbf{b}) \end{cases}$$

- WOLG, assume $\mathbf{b} = \mathbf{0}$.

Main Result: Convergence in Expectation

- After k rounds, RP-ADMM generates $\mathbf{z}^k$, an r.v. depending on

$$\boldsymbol{\xi}_k = (\sigma_1, \ldots, \sigma_k), \quad \mathbf{z}^i = M_{\sigma_i}\mathbf{z}^{i-1},\ i = 1, ..., k,$$

  where σ_i is the picked permutation at i-th round.

- Denote the expected output $\boldsymbol{\phi}^k \triangleq E_{\xi_k}(\mathbf{z}^k)$

  Theorem 2 *The expected output converges to the unique solution, i.e.*

$$\{\boldsymbol{\phi}^k\}_{k\to\infty} \longrightarrow \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}.$$

- **Remark:** Expected convergence $\neq$ convergence, but is a strong evidence for convergence for solving most problems, e.g., when iterates are bounded.

The Average Mapping is a Contraction

- The update equation of RP-ADMM for (S) is

$$\mathbf{z}^{k+1} = M_\sigma \mathbf{z}^k,$$

where $M_\sigma \in R^{2N \times 2N}$ depend on σ.

- Define the expected update matrix as

$$M = E_\sigma(M_\sigma) = \frac{1}{n!}\sum_\sigma M_\sigma.$$

Theorem 3 *The spectral radius of M, $\rho(M)$, is strictly less than 1.*

- Remark: For A in the divergence example, $\rho(M_\sigma) > 1$ for any σ
  - Averaging Helps, a lot.

Math Problem of Theorem 3

- Define

$$Q \triangleq E(L_\sigma^{-1}) = \frac{1}{n!}\sum_\sigma L_\sigma^{-1}. \tag{13}$$

- Example:

$$L_{(231)} = \begin{bmatrix} 1 & A_1^T A_2 & A_1^T A_3 \\ 0 & 1 & 0 \\ 0 & A_3^T A_2 & 1 \end{bmatrix}.$$

- Need to prove that, for all A, $\rho(M) < 1$ where

$$M = \begin{bmatrix} I - QA^T A & QA^T \\ -A + AQA^T A & I - AQA^T \end{bmatrix}.$$

Difficulties of Proving Theorem 3

- **Difficulty 1**: Few tools deal with spectral radius of non-symmetric matrices.
  - E.g. $\rho(X+Y) \leq \rho(X)+\rho(Y)$ and $\rho(XY) \leq \rho(X)\rho(Y)$ don't hold.
  - Though $\rho(M) < \|M\|$, it turns out $\|M\| > 2.3$ for the counterexample.
- **Difficulty 2**: M is a complicated function of A.
  - $n=3$, let $(A^TA)_{k,l} = b_{kl}$, then $Q_{12} = -\frac{1}{2}b_{12} + \frac{1}{6}b_{13}b_{23}$.
  - $n=4$, $Q_{12} = -\frac{1}{2!}b_{12} + \frac{1}{3!}(b_{13}b_{32}+b_{14}b_{42}) - \frac{1}{4!}(b_{13}b_{34}b_{42}+b_{14}b_{43}b_{32})$.
- **Solution**: Symmetrization and Mathematical Induction.

Two Main Lemmas to Prove Theorem 3

- **Step 1**: Relate M to a symmetric matrix AQA^T.

  Lemma 1

  $$\mathbf{y} \in \textit{eig}(M) \Longleftrightarrow \frac{(1-\mathbf{y})^2}{1-2\mathbf{y}} \in \textit{eig}(AQA^T).$$

  Since Q defined by (13) *is symmetric, we have*

  $$\rho(M) < 1 \Longleftrightarrow \textit{eig}(AQA^T) \subseteq (0, \frac{4}{3}).$$

- **Step 2**: Bound eigenvalues of AQA^T - prove by math induction.

  Lemma 2

  $$\textit{eig}(AQA^T) \subseteq (0, \frac{4}{3}).$$

- Remark: $4/3$ is “almost” tight; for $n = 3$, maximum ≈ 1.18. Increase to $4/3$ as n increases.

More Randomization: Randomly Assembled ADMM

Randomly Assembled ADMM (RA-ADMM) – Random Variable Sampling without Replacement in Each ADMM round:

1. Set the initial set N_x as all (primal) decision variables.
2. Randomly select a subset of variables from N_x to optimize.
3. Remove this set of variables from N_x and return to Step 1 till N_x is empty.
4. Update the dual multipliers as usual.

Provide much better results for non-convex and discrete/combinatorial optimization (Mihic et al. 2020).

Extensions and Research Directions

- Non-square system of linear equations: resolved
- Non-separable convex quadratic minimization: resolved

$E[z^k] \to z^*$

- Theory: Convergence w.h.p.?
- Theory: Generalize to inequality systems or convex optimization at large?
- Theory: Overall complexity of Interior-Point ADMM for LP?
- Theory: More analyses on RA-ADMM?

ABIP

- Implementation and computation development!