ADMM for Multi-Block Problems

Hailiang ZHAO @ ZJU-CS

http://hliangzhao.me

November 2, 2022

Outline

Outline

Multi-Block Problem

We try to extend ADMM to solve the following problem with multi-blocks:

$$\min_{\boldsymbol{x}_i} \sum_{i=1}^{m} f_i(\boldsymbol{x}_i), \quad s.t. \quad \sum_{i=1}^{m} \mathbf{A}_i \boldsymbol{x}_i = \boldsymbol{b}. \tag{1}$$

We define

$$f(\boldsymbol{x}) = \sum_{i=1}^{m} f_i(\boldsymbol{x}_i) \tag{2}$$

$$\mathbf{A} = [\mathbf{A}_1, ..., \mathbf{A}_m] \tag{3}$$

$$\boldsymbol{x} = (\boldsymbol{x}_1^T, ..., \boldsymbol{x}_m^T)^T \tag{4}$$

for simplification.

Direct Extension of ADMM

The straightforward extension of the vanilla ADMM:

$$\tilde{\boldsymbol{x}}_i^{k+1} = \operatorname*{argmin}_{\boldsymbol{x}_i} L_\beta\big(\tilde{\boldsymbol{x}}_1^{k+1}, \ldots, \tilde{\boldsymbol{x}}_{i-1}^{k+1}, \boldsymbol{x}_i, \boldsymbol{x}_{i+1}^k, \ldots, \boldsymbol{x}_m^k, \boldsymbol{\lambda}^k\big), \forall i, \tag{5}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \beta\Big(\sum_{i=1}^m \mathbf{A}_i \tilde{\boldsymbol{x}}_i^{k+1} - \boldsymbol{b}\Big), \tag{6}$$

with $\boldsymbol{x}_i^k = \tilde{\boldsymbol{x}}_i^k$ (*no correction here*), where

$$L_\beta(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m, \boldsymbol{\lambda}) = \sum_i f_i(\boldsymbol{x}_i) + \langle \boldsymbol{\lambda}, \sum_i \mathbf{A}_i \boldsymbol{x}_i - \boldsymbol{b} \rangle + \frac{\beta}{2} \| \sum_i \mathbf{A}_i \boldsymbol{x}_i - \boldsymbol{b} \|^2. \tag{7}$$

However, it is proved that the above method might not convergent. Thus, we should make several modifications on the original ADMM for convergence guarantees.

Outline

ADMM-GBS

Now we introduce the Gaussian Back Substitution scheme. It first predicts $\tilde{x}_i^{k+1}$ for all $i \in [m]$, and then corrects $\boldsymbol{x}_i^{k+1}$ from $\tilde{x}_i^{k+1}$. Denote

$$\mathbf{M} = \begin{bmatrix} \mathbf{A}_1^T\mathbf{A}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_2^T\mathbf{A}_1 & \mathbf{A}_2^T\mathbf{A}_2 & \ddots & \mathbf{0} \\ \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{A}_m^T\mathbf{A}_1 & \mathbf{A}_m^T\mathbf{A}_2 & \cdots & \mathbf{A}_m^T\mathbf{A}_m \end{bmatrix} \tag{8}$$

and

$$\mathbf{H} = \mathrm{diag}\left(\mathbf{A}_1^T\mathbf{A}_1, ..., \mathbf{A}_m^T\mathbf{A}_m\right), \tag{9}$$

$\mathbf{M}$ and $\mathbf{H}$ will be used in the following proofs frequently.

Lemma I.1

Suppose that $i \in [m]\left[f_i(\boldsymbol{x}_i) \text{ is convex}\right]$. Then with (5) and (6), we have

$$f(\tilde{\boldsymbol{x}}^{k+1}) - f(\boldsymbol{x}^*) + \langle \boldsymbol{\lambda}^*, \sum_{i=1}^{m} \mathbf{A}_i \tilde{\boldsymbol{x}}_i^{k+1} - \boldsymbol{b} \rangle$$
$$\leq \frac{1}{2\beta}\left(\|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*\|^2 - \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\|^2\right) - \frac{\beta}{2}\|\tilde{\boldsymbol{x}}^{k+1} - \boldsymbol{x}^k\|_{\mathbf{H}}^2$$
$$- \beta(\boldsymbol{x}^k - \boldsymbol{x}^*)^T \mathbf{M}(\tilde{\boldsymbol{x}}^{k+1} - \boldsymbol{x}^k). \tag{10}$$

We want to make the RHS of the inequality in a form of $\phi^k - \phi^{k+1}$ such that a recursion can be established.

To this end, we need to define $\boldsymbol{x}^{k+1}$ and find some $\mathbf{G} \succeq \mathbf{0}$ so

$$-\frac{\beta}{2}\|\tilde{\boldsymbol{x}}^{k+1} - \boldsymbol{x}^k\|_{\mathbf{H}}^2 - \beta(\boldsymbol{x}^k - \boldsymbol{x}^*)^T\mathbf{M}(\tilde{\boldsymbol{x}}^{k+1} - \boldsymbol{x}^k)$$
$$= \frac{\beta}{2}(\|\boldsymbol{x}^k - \boldsymbol{x}^*\|_{\mathbf{G}}^2 - \|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\|_{\mathbf{G}}^2) \tag{11}$$

If we let

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \mathbf{D}(\tilde{x}^{k+1} - \boldsymbol{x}^k), \tag{12}$$

then the RHS of (11) becomes

$$- (\tilde{\boldsymbol{x}}^{k+1} - \boldsymbol{x}^k)^T\mathbf{D}^T\mathbf{G}\mathbf{D}(\tilde{\boldsymbol{x}}^{k+1} - \boldsymbol{x}^k)$$
$$- 2(\boldsymbol{x}^k - \boldsymbol{x}^*)^T\mathbf{G}\mathbf{D}(\boldsymbol{x}^{\widetilde{k+1}} - \boldsymbol{x}^k). \tag{13}$$

It means $\mathbf{D}$ and $\mathbf{G}$ should satisfy that

$$\mathbf{D}^T\mathbf{G}\mathbf{D} = \mathbf{H} \text{ and } \mathbf{G}\mathbf{D} = \mathbf{M}. \tag{14}$$

ADMM-GBS

Assume that $\mathbf{A}_i$'s are all of full column rank. Then $\mathbf{M}$ and $\mathbf{H}$ are invertible. Let $\mathbf{D} = \mathbf{M}^{-T}\mathbf{H}$ and $\mathbf{G} = \mathbf{M}\mathbf{H}^{-1}\mathbf{M}^T$, then (12) becomes

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \mathbf{M}^{-T}\mathbf{H}(\tilde{\boldsymbol{x}}^{k+1} - \boldsymbol{x}^k), \tag{15}$$

which can be computed by the famous Gaussian back substitution efficiently due to the lower-block-triangular structure of $\mathbf{M}$.

We call this method, i.e., (5), (6), and (15), *ADMM-GBS*. Note that $\tilde{\boldsymbol{x}}_i$ and $\boldsymbol{x}_i$ are updated sequentially.

Ergodic Convergenece Rate of ADMM-GBS

Theorem I.1

Suppose that $i \in [m] \left[f_i(\boldsymbol{x}_i) \text{ is convex}\right]$. Then for ADMM-GBS we have

$$|f(\boldsymbol{x}^{\hat{K}+1}) - f(\boldsymbol{x}^*)| \leq \frac{C}{2(K+1)} + \frac{2\sqrt{C}\|\boldsymbol{\lambda}^*\|}{\sqrt{\beta}(K+1)} \tag{16}$$

$$\|\mathbf{A}\hat{\boldsymbol{x}}^{K+1} - \boldsymbol{b}\| \leq \frac{2\sqrt{C}}{\sqrt{\beta}(K+1)}, \tag{17}$$

where

$$\hat{\boldsymbol{x}}^{K+1} = \frac{1}{K+1} \sum_{k=0}^{K} \tilde{\boldsymbol{x}}^{k+1} \tag{18}$$

$$C = \frac{1}{\beta}\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|^2 + \beta\|\boldsymbol{x}^0 - \boldsymbol{x}^*\|^2_{\mathbf{G}}. \tag{19}$$

Outline

ADMM-PC

ADMM-GBS is not simple enough since it needs to solve m small-scale linear systems. In the following, we give an improved strategy, which is based on the intuition that, *when solving* (5)*, we actually only need* $\mathbf{A}_i\boldsymbol{x}_i^k$ *rather than* $\boldsymbol{x}_i^k$. Thus, we don't have to compute $\boldsymbol{x}_i^k$ explicitly.

Denote $\mathbf{P} = \text{diag}(\mathbf{A}_1, ..., \mathbf{A}_m)$ and

$$\mathbf{L} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{I} & \mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I} & \mathbf{I} & \cdots & \mathbf{I} \end{bmatrix} \tag{20}$$

Then $\mathbf{M} = \mathbf{P}^T\mathbf{L}\mathbf{P}$, $\mathbf{H} = \mathbf{P}^T\mathbf{P}$.

ADMM-PC

Similar to (11), we need to define $\boldsymbol{x}^{k+1}$ and find $\mathbf{G}' \succeq \mathbf{0}$ such that:

$$
\begin{aligned}
&-\frac{\beta}{2}\|\tilde{\boldsymbol{x}}^{k+1} - \boldsymbol{x}^k\|_{\mathbf{H}}^2 - \beta(\boldsymbol{x}^k - \boldsymbol{x}^*)^T\mathbf{M}(\tilde{\boldsymbol{x}}^{k+1} - \boldsymbol{x}^k) \\
&\qquad = \frac{\beta}{2}(\|\mathbf{P}\boldsymbol{x}^k - \mathbf{P}\boldsymbol{x}^*\|_{\mathbf{G}'}^2 - \|\mathbf{P}\boldsymbol{x}^{k+1} - \mathbf{P}\boldsymbol{x}^*\|_{\mathbf{G}'}^2). \qquad (21)
\end{aligned}
$$

If we let

$$
\mathbf{P}\boldsymbol{x}^{k+1} = \mathbf{P}\boldsymbol{x}^k + \mathbf{D}'(\mathbf{P}\tilde{\boldsymbol{x}}^{k+1} - \mathbf{P}\boldsymbol{x}^k), \qquad (22)
$$

then the RHS of (21) becomes

$$
\begin{aligned}
&-(\mathbf{P}\tilde{\boldsymbol{x}}^{k+1} - \mathbf{P}\boldsymbol{x}^k)^T(\mathbf{D}')^T\mathbf{G}'\mathbf{D}'(\mathbf{P}\tilde{\boldsymbol{x}}^{k+1} - \mathbf{P}\boldsymbol{x}^k) \\
&\quad -2(\mathbf{P}\boldsymbol{x}^k - \mathbf{P}\boldsymbol{x}^*)^T\mathbf{G}'\mathbf{D}'(\mathbf{P}\tilde{\boldsymbol{x}}^{k+1} - \mathbf{P}\boldsymbol{x}^k). \qquad (23)
\end{aligned}
$$

ADMM-PC

Note that the LHS of (21) equals to

$$-\frac{\beta}{2}\|\mathbf{P}\tilde{x}^{k+1}-\mathbf{P}\boldsymbol{x}^{k}\|^{2}-\beta(\mathbf{P}\boldsymbol{x}^{k}-\mathbf{P}\boldsymbol{x}^{*})^{T}\mathbf{L}(\mathbf{P}\tilde{x}^{k+1}-\mathbf{P}\boldsymbol{x}^{k}). \quad (24)$$

Therefore, we should choose

$$\mathbf{D}'=\mathbf{L}^{-T} \text{ and } \mathbf{G}'=\mathbf{L}\mathbf{L}^{T}. \quad (25)$$

ADMM-PC

Now (22) can be explicitly re-written as

$$
\begin{pmatrix} \mathbf{A}_1 \boldsymbol{x}_1^{k+1} \\ \mathbf{A}_2 \boldsymbol{x}_2^{k+1} \\ \vdots \\ \mathbf{A}_{m-1} \boldsymbol{x}_{m-1}^{k+1} \\ \mathbf{A}_m \boldsymbol{x}_m^{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{A}_1 \boldsymbol{x}_1^{k} \\ \mathbf{A}_2 \boldsymbol{x}_2^{k} \\ \vdots \\ \mathbf{A}_{m-1} \boldsymbol{x}_{m-1}^{k} \\ \mathbf{A}_m \boldsymbol{x}_m^{k} \end{pmatrix}
$$

$$
+ \begin{pmatrix} \mathbf{I} & -\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & -\mathbf{I} & \cdots & \mathbf{0} & \mathbf{0} \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{A}_1 \tilde{\boldsymbol{x}}_1^{k+1} - \mathbf{A}_1 \boldsymbol{x}_1^{k} \\ \mathbf{A}_2 \tilde{\boldsymbol{x}}_2^{k+1} - \mathbf{A}_2 \boldsymbol{x}_1^{2} \\ \ddots \\ \mathbf{A}_{m-1} \tilde{\boldsymbol{x}}_{m-1}^{k+1} - \mathbf{A}_{m-1} \boldsymbol{x}_{m-1}^{k} \\ \mathbf{A}_m \tilde{\boldsymbol{x}}_m^{k+1} - \mathbf{A}_m \boldsymbol{x}_m^{k} \end{pmatrix}. \tag{26}
$$

ADMM-PC

The RHS of (26) equals to

$$\begin{pmatrix} \mathbf{A}_1\tilde{\boldsymbol{x}}_1^{k+1} + \mathbf{A}_2\boldsymbol{x}_2^k - \mathbf{A}_2\tilde{\boldsymbol{x}}_2^{k+1} \\ \mathbf{A}_2\tilde{\boldsymbol{x}}_2^{k+1} + \mathbf{A}_3\boldsymbol{x}_3^k - \mathbf{A}_3\tilde{\boldsymbol{x}}_3^{k+1} \\ \vdots \\ \mathbf{A}_{m-1}\tilde{\boldsymbol{x}}_{m-1}^{k+1} + \mathbf{A}_m\boldsymbol{x}_m^k - \mathbf{A}_m\tilde{\boldsymbol{x}}_m^{k+1} \\ \mathbf{A}_m\tilde{\boldsymbol{x}}_m^{k+1} \end{pmatrix}. \tag{27}$$

The result shows that we can obtain $\mathbf{A}_i\boldsymbol{x}_i^{k+1}$ conveniently without solving linear systems.

However, $\boldsymbol{x}_i^{k+1}$ may not exist given $\mathbf{A}_i\boldsymbol{x}_i^{k+1}$, making the next round iteration invalid. Thus we need to revise the iterations accordingly.

ADMM-PC

Introducing variables $\boldsymbol{\xi}_i^k$, which play the role of $\mathbf{A}_i \boldsymbol{x}_i^k$, we iterate as follows:

$$\tilde{\boldsymbol{x}}_i^{k+1} = \underset{\boldsymbol{x}_i}{\operatorname{argmin}} \tilde{L}_i(\underbrace{\tilde{\boldsymbol{x}}_1^{k+1}, \ldots, \tilde{\boldsymbol{x}}_{i-1}^{k+1}}_{\text{calculated in seq.}}, \boldsymbol{x}_i, \boldsymbol{\xi}_{i+1}^k, \ldots, \boldsymbol{\xi}_m^k, \boldsymbol{\lambda}^k), \forall \boldsymbol{i}, \tag{28}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \beta\Big(\sum_{i=1}^m \mathbf{A}_i \tilde{\boldsymbol{x}}_i^{k+1} - \boldsymbol{b}\Big) \tag{29}$$

$$\boldsymbol{\xi}^{k+1} = \boldsymbol{\xi}^k + \mathbf{L}^{-T}(\mathbf{P}\tilde{\boldsymbol{x}}^{k+1} - \boldsymbol{\xi}^k), \tag{30}$$

where $\tilde{L}_i(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i, \boldsymbol{\xi}_{i+1}, \ldots, \boldsymbol{\xi}_m, \boldsymbol{\lambda})$ is

$$\sum_{j=1}^i f_j(\boldsymbol{x}_j) + \langle \boldsymbol{\lambda}, \sum_{j=1}^i \mathbf{A}_j \boldsymbol{x}_j + \sum_{j=i+1}^m \boldsymbol{\xi}_j - \boldsymbol{b} \rangle$$

$$+ \frac{\beta}{2} \| \sum_{j=1}^i \mathbf{A}_j \boldsymbol{x}_j + \sum_{j=i+1}^m \boldsymbol{\xi}_j - \boldsymbol{b} \|^2, \tag{31}$$

ADMM-PC

We call (28), (29), and (30) ADMM with Prediction-Correction (*ADMM-PC*).

For ADMM-PC, we have a similar result:

Lemma I.2
Suppose that $i \in [m] \left[f_i(\boldsymbol{x}_i) \text{ is convex} \right]$. Then for ADMM-PC, we have

$$\begin{aligned} & f(\tilde{\boldsymbol{x}}^{k+1}) - f(\boldsymbol{x}^*) + \langle \boldsymbol{\lambda}^*, \sum_{i=1}^{m} \mathbf{A}_i \tilde{\boldsymbol{x}}_i^{k+1} - \boldsymbol{b} \rangle \\ & \leq \frac{1}{2\beta} \left(\|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*\|^2 - \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\|^2 \right) \\ & + \frac{\beta}{2} \left(\|\boldsymbol{\xi}^k - \mathbf{P}\boldsymbol{x}^*\|_{\mathbf{L}\mathbf{L}^T}^2 - \|\boldsymbol{\xi}^{k+1} - \mathbf{P}\boldsymbol{x}^*\|_{\mathbf{L}\mathbf{L}^T}^2 \right). \end{aligned} \tag{32}$$

Ergodic Convergenece Rate of ADMM-PC

ADMM-PC have the same convergence rate to ADMM-GBS:

Theorem I.2

Suppose that $i \in [m] \left[f_i(\boldsymbol{x}_i) \text{ is convex} \right]$. Then for ADMM-PC (16) and (17) holds, but C changes to

$$\frac{1}{\beta} \|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|^2 + \beta \|\boldsymbol{\xi}^0 - \mathbf{P}\boldsymbol{x}^*\|^2_{\mathbf{L}\mathbf{L}^T} \tag{33}$$

Outline

LADMM-PS

ADMM-GBS and ADMM-PC have the following drawbacks:

- It requires to maintain two groups of variables, the predicted and the corrected, thus increasing the memory cost (at least doubled)
- Neither $\boldsymbol{x}_i^k$ (ADMM-GBS) nor $\boldsymbol{\xi}_i^k$ (ADMM-PC) could be sparse or low-rank, thus the memory consumption can be more

Nevertheless, we can use the linearized augmented Lagrangian method.

LADMM-PS

Similar to (46) and (47) of *ADMM Slide: Part 2*, we use the following iterations:

$$
\begin{aligned}
\boldsymbol{x}^{k+1} &= \operatorname*{argmin}_{\boldsymbol{x}} \Big(f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}^k, \mathbf{A}\boldsymbol{x} - \boldsymbol{b} \rangle \\
&\qquad\qquad + \frac{\beta}{2} \|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 + D_{\Psi}(\boldsymbol{x}, \boldsymbol{x}^k) \Big), \qquad (34) \\
\boldsymbol{\lambda}^{k+1} &= \boldsymbol{\lambda}^k + \beta \Big(\sum_{i=1}^{m} \mathbf{A}_i \boldsymbol{x}_i^{k+1} - \boldsymbol{b} \Big), \qquad (35)
\end{aligned}
$$

where $\mathbf{L} = \operatorname{diag}(L_1 \mathbf{I}_{n_1}, \ldots, L_m \mathbf{I}_{n_m})$, in which n_i is the dimension of $\boldsymbol{x}_i$ and L_i is to be determined.

Note that we hope each $\boldsymbol{x}_i^k$ can be calculated separately for all $i \in [m]$.

LADMM-PS

If we set

$$\Psi(\boldsymbol{x}) = \frac{\beta}{2}\|\boldsymbol{x}\|_{\mathbf{L}}^2 - \frac{\beta}{2}\|\mathbf{A}\boldsymbol{x}\|^2, \tag{36}$$

then with $\mathbf{A}\boldsymbol{x} - \boldsymbol{b} = (\mathbf{A}\boldsymbol{x}^k - \boldsymbol{b}) + (\mathbf{A}\boldsymbol{x} - \mathbf{A}\boldsymbol{x}^k)$, we have

$$\begin{aligned} &\frac{\beta}{2}\|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 + D_{\Psi}(\boldsymbol{x}, \boldsymbol{x}^k) \\ = &\frac{\beta}{2}\|\mathbf{A}\boldsymbol{x}^k - \boldsymbol{b}\|^2 + \beta\langle \mathbf{A}^T(\mathbf{A}\boldsymbol{x}^k - \boldsymbol{b}), \boldsymbol{x} - \boldsymbol{x}^k\rangle + \frac{\beta}{2}\|\boldsymbol{x} - \boldsymbol{x}^k\|_{\mathbf{L}}^2. \end{aligned}$$

If we choose $L_i \geq m\|\mathbf{A}_i\|^2$, then $\forall \boldsymbol{x}$,

$$\begin{aligned} \|\boldsymbol{x}\|_{\mathbf{L}}^2 = \sum_i L_i \boldsymbol{x}_i^T \boldsymbol{x}_i \geq m \sum_i \|\mathbf{A}_i\|^2 \|\boldsymbol{x}_i\|^2 \geq m \sum_i \|\mathbf{A}_i \boldsymbol{x}_i\|^2 \\ \geq \|\sum_i \mathbf{A}_i \boldsymbol{x}_i\|^2 \geq \|\mathbf{A}\boldsymbol{x}\|^2. \end{aligned} \tag{37}$$

LADMM-PS

In the above slide, we get $\|\boldsymbol{x}\|_L^2 \geq \|\mathbf{A}\boldsymbol{x}\|^2$. Now (34) becomes separable and the subproblem for each $\boldsymbol{x}_i$ is

$$\begin{aligned} \boldsymbol{x}_i^{k+1} = \underset{\boldsymbol{x}_i}{\operatorname{argmin}} \Big(& f_i(\boldsymbol{x}_i) + \langle \boldsymbol{\lambda}^k, \mathbf{A}_i \boldsymbol{x}_i \rangle \\ & + \beta \langle \mathbf{A}_i^T \Big(\sum_{j=1}^m \mathbf{A}_j \boldsymbol{x}_j^k - \boldsymbol{b} \Big), \boldsymbol{x}_i - \boldsymbol{x}_i^k \rangle \\ & + \frac{\beta L_i}{2} \|\boldsymbol{x}_i - \boldsymbol{x}_i^k\|^2 \Big), \end{aligned} \tag{38}$$

which can be solved in parallel for $i \in [m]$.

We call (34) and (38) linearized ADMM with parallel splitting (*LADMM-PS*). Similar to Bregman ADMM, LADMM-PS has the $\mathcal{O}(1/K)$ and the linear convergence rates under certain conditions.

LADMM-SPU

LADMM-PS works by replacing the serial update order in linearized ADMM by the parallel update order in the linearized augmented Lagrangian method.

Since for the two-block case, serial update order is faster than parallel update order, we may combine them for faster convergence when dealing with the multi-block case. That is, divide the m blocks into two partitions:

$$1, \ldots, m' \quad \text{and} \quad m' + 1, \ldots, m \tag{39}$$

and then *update the two partitions in serial, while updating the blocks in the same partition in parallel.*

LADMM-SPU

The detailed iterates are:

$$
\begin{aligned}
\boldsymbol{x}_i^{k+1} = \operatorname*{argmin}_{\boldsymbol{x}_i} \Bigg(& f_i(\boldsymbol{x}_i) + \langle \boldsymbol{\lambda}^k, \mathbf{A}_i \boldsymbol{x}_i \rangle \\
& + \beta \langle \mathbf{A}_i^T \Big(\sum_{t=1}^{m} \mathbf{A}_t \boldsymbol{x}_t^k - \boldsymbol{b} \Big), \boldsymbol{x}_i - \boldsymbol{x}_i^k \rangle \\
& + \frac{m' \beta \|\mathbf{A}_i\|^2}{2} \|\boldsymbol{x}_i - \boldsymbol{x}_i^k\|^2 \Bigg), \qquad \forall i \in [1, m'], \qquad (40) \\
\boldsymbol{x}_j^{k+1} = \operatorname*{argmin}_{\boldsymbol{x}_j} \Bigg(& f_j(\boldsymbol{x}_j) + \langle \boldsymbol{\lambda}^k, \mathbf{A}_j \boldsymbol{x}_j \rangle \\
& + \beta \langle \mathbf{A}_j^T \Big(\sum_{t=1}^{m} \mathbf{A}_t \boldsymbol{x}_t^{k+1} + \sum_{t=m'+1}^{m} \mathbf{A}_t \boldsymbol{x}_t^k - \boldsymbol{b} \Big), \boldsymbol{x}_j - \boldsymbol{x}_j^k \rangle \\
& + \frac{(m - m') \beta \|\mathbf{A}_j\|^2}{2} \|\boldsymbol{x}_j - \boldsymbol{x}_j^k\|^2 \Bigg), \forall i \in [m' + 1, m]. \quad (41)
\end{aligned}
$$

LADMM-SPU

The detailed iterates are (cont'd):

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^{k} + \beta\Big(\sum_{i=1}^{m} \mathbf{A}_i \boldsymbol{x}_i^{k+1} - \boldsymbol{b}\Big). \tag{42}$$

We call (40), (41), and (42) ADMM with the Serial and the Parallel Update Orders (LADMM-SPU).

LADMM-SPU has exactly the same form of Bregman ADMM. Thus we can also have the sublinear and the linear convergence rates under different conditions, which is faster than making the m blocks parallel.

Outline

References

1. Lin, Zhouchen, Huan Li, and Cong Fang. *Alternating Direction Method of Multipliers for Machine Learning.* Springer Nature, 2022.
2. Boyd, Stephen, Stephen P. Boyd, and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.