Fast Proximal Linearized Alternating Direction Method of Multiplier with Parallel Splitting

Canyi Lu[1], Huan Li[2], Zhouchen Lin[2,], Shuicheng Yan[1]

[1]National University of Singapore, [2]Peking University

Problem and Contributions

- **Problem:** to solve the following linearly constrained separable convex problem with n blocks of variables

$$\min_{\mathbf{x}_1,\cdots,\mathbf{x}_n} f(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x}_i) = \sum_{i=1}^{n} \left(g_i(\mathbf{x}_i) + h_i(\mathbf{x}_i)\right), \quad \text{s.t.} \quad \mathcal{A}(\mathbf{x}) = \sum_{i=1}^{n} \mathcal{A}_i(\mathbf{x}_i) = \mathbf{b}, \tag{1}$$

- g_i is a convex smooth function with Lipschitz continuous gradient, i.e.,
$$\|\nabla g_i(\mathbf{x}) - \nabla g_i(\mathbf{y})\|_F \leq L_i\|\mathbf{x} - \mathbf{y}\|_F, \ \forall\, \mathbf{x}, \mathbf{y}.$$
- h_i is simple, i.e., $\min_{\mathbf{x}} h_i(\mathbf{x}) + \frac{\alpha}{2}\|\mathbf{x} - \mathbf{a}\|^2$ $(\alpha > 0)$ can be cheaply solved.
- **Contributions:** faster solvers for (1) with bettter convergence rates:

PALM	Fast PALM	PL-ADMM-PS	Fast PL-ADMM-PS
$O\left(\frac{D_{\mathbf{x}^*}^2+D_{\lambda^*}^2}{K}\right)$	$O\left(\frac{D_{\mathbf{x}^*}^2+D_{\lambda^*}^2}{K^2}\right)$	$O\left(\frac{D_{\mathbf{x}^*}^2}{K}+\frac{D_{\mathbf{x}^*}^2}{K}+\frac{D_{\lambda^*}^2}{K}\right)$	$O\left(\frac{D_{\mathbf{x}^*}^2}{K^2}+\frac{D_X^2}{K}+\frac{D_\Lambda^2}{K}\right)$

Fast PALM for (1) with $n = 1$

- Solve problem (1) with $n = 1$ by **Proximal Augmented Lagrangian Method (PALM)**:

$$\begin{cases} \mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} g(\mathbf{x}^k) + \langle \nabla g(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k\rangle + h(\mathbf{x}) \\ \qquad + \langle \boldsymbol{\lambda}^k, \mathcal{A}(\mathbf{x}) - \mathbf{b}\rangle + \frac{\beta^{(k)}}{2}\|\mathcal{A}(\mathbf{x}) - \mathbf{b}\|^2 + \frac{L}{2}\|\mathbf{x} - \mathbf{x}^k\|^2, \\ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \beta^{(k)}(\mathcal{A}(\mathbf{x}^{k+1}) - \mathbf{b}), \end{cases} \tag{2}$$

- Solve problem (1) with $n = 1$ by our proposed **Fast PALM**

Initialize: $\mathbf{x}^0$, $\mathbf{z}^0$, $\boldsymbol{\lambda}^0$, $\beta^{(0)} = \theta^{(0)} = 1$.
for $k = 0, 1, 2, \cdots$ **do**

$$\mathbf{y}^{k+1} = (1-\theta^{(k)})\mathbf{x}^k + \theta^{(k)}\mathbf{z}^k; \tag{6}$$

$$\mathbf{z}^{k+1} = \operatorname*{argmin}_{\mathbf{x}} \langle \nabla g(\mathbf{y}^{k+1}), \mathbf{x}\rangle + h(\mathbf{x}) + \langle \boldsymbol{\lambda}^k, \mathcal{A}(\mathbf{x})\rangle + \frac{\beta^{(k)}}{2}\|\mathcal{A}(\mathbf{x}) - \mathbf{b}\|^2 + \frac{L\theta^{(k)}}{2}\|\mathbf{x} - \mathbf{z}^k\|^2; \tag{7}$$

$$\mathbf{x}^{k+1} = (1-\theta^{(k)})\mathbf{x}^k + \theta^{(k)}\mathbf{z}^{k+1}; \tag{8}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \beta^{(k)}(\mathcal{A}(\mathbf{z}^{k+1}) - \mathbf{b}); \tag{9}$$

$$\theta^{(k+1)} = \frac{-(\theta^{(k)})^2 + \sqrt{(\theta^{(k)})^4 + 4(\theta^{(k)})^2}}{2}; \tag{10}$$

$$\beta^{(k+1)} = \frac{1}{\theta^{(k+1)}}. \tag{11}$$

end

Algorithm 1: Fast PALM Algorithm

- **Theorem 1** *In Algorithm 1, for any $K > 0$, we have*

$$f(\mathbf{x}^{K+1}) - f(\mathbf{x}^*) + \left\langle \boldsymbol{\lambda}^*, \mathcal{A}(\mathbf{x}^{K+1}) - \mathbf{b}\right\rangle + \frac{1}{2}\|\mathcal{A}(\mathbf{x}^{K+1}) - \mathbf{b}\|^2 \leq \frac{2}{(K+2)^2}\left(LD_{\mathbf{x}^*}^2 + D_{\boldsymbol{\lambda}^*}^2\right). \tag{3}$$

- Remark: Fast PALM improves the rate of PALM from $O\left(\frac{D_{\mathbf{x}^*}^2+D_{\lambda^*}^2}{K}\right)$ to $O\left(\frac{D_{\mathbf{x}^*}^2+D_{\lambda^*}^2}{K^2}\right)$.

Fast PL-ADMM-PS for (1) with $n > 1$

- Solve problem (1) with $n > 1$ by **Proximal Linearized Alternating Direction Method of Multiplier (PL-ADMM-PS)**: updates each $\mathbf{x}_i$ in parallel by

$$\begin{cases} \mathbf{x}_i^{k+1} = \operatorname*{argmin}_{\mathbf{x}_i} g_i(\mathbf{x}_i^k) + \left\langle \nabla g_i(\mathbf{x}_i^k), \mathbf{x}_i - \mathbf{x}_i^k\right\rangle + h_i(\mathbf{x}_i) + \left\langle \boldsymbol{\lambda}^k, \mathcal{A}_i(\mathbf{x}_i)\right\rangle \\ \qquad + \left\langle \beta^{(k)}\mathcal{A}_i^T\left(\mathcal{A}(\mathbf{x}^k) - \mathbf{b}\right), \mathbf{x}_i - \mathbf{x}_i^k\right\rangle + \frac{L_i + \beta^{(k)}\eta_i}{2}\|\mathbf{x}_i - \mathbf{x}_i^k\|^2, \\ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \beta^{(k)}(\mathcal{A}(\mathbf{x}^{k+1}) - \mathbf{b}), \end{cases} \tag{4}$$

where $\eta_i > n\|\mathcal{A}_i\|^2$ and $\beta^{(k)} > 0$.

- Solve problem (1) with $n = 1$ by our proposed **Fast PL-ADMM-PS**

Initialize: $\mathbf{x}^0$, $\mathbf{z}^0$, $\boldsymbol{\lambda}^0$, $\theta^{(0)} = 1$, fix $\beta^{(k)} = \beta$ for $k \geq 0$, $\eta_i > n\|\mathcal{A}_i\|^2$, $i = 1, \cdots, n$,
for $k = 0, 1, 2, \cdots$ **do**

// Update $\mathbf{y}_i$, $\mathbf{z}_i$, $\mathbf{x}_i$, $i = 1, \cdots, n$, in parallel by

$$\mathbf{y}_i^{k+1} = (1-\theta^{(k)})\mathbf{x}_i^k + \theta^{(k)}\mathbf{z}_i^k; \tag{17}$$

$$\mathbf{z}_i^{k+1} = \operatorname*{argmin}_{\mathbf{x}_i} \langle \nabla g_i(\mathbf{y}_i^{k+1}), \mathbf{x}_i\rangle + h_i(\mathbf{x}_i) + \langle \boldsymbol{\lambda}^k, \mathcal{A}_i(\mathbf{x}_i)\rangle + \left\langle \beta^{(k)}\mathcal{A}_i^T\left(\mathcal{A}(\mathbf{z}^k) - \mathbf{b}\right), \mathbf{x}_i\right\rangle + \frac{L(g_i)\theta^{(k)} + \beta^{(k)}\eta_i}{2}\|\mathbf{x}_i - \mathbf{z}_i^k\|^2; \tag{18}$$

$$\mathbf{x}_i^{k+1} = (1-\theta^{(k)})\mathbf{x}_i^k + \theta^{(k)}\mathbf{z}_i^{k+1}; \tag{19}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \beta^k\left(\mathcal{A}(\mathbf{z}^{k+1}) - \mathbf{b}\right); \tag{20}$$

$$\theta^{(k+1)} = \frac{-(\theta^{(k)})^2 + \sqrt{(\theta^{(k)})^4 + 4(\theta^{(k)})^2}}{2}. \tag{21}$$

end

Algorithm 2: Fast PL-ADMM-PS Algorithm

- **Theorem 2** *In Algorithm 2, for any $K > 0$, we have*

$$f(\mathbf{x}^{K+1}) - f(\mathbf{x}^*) + \left\langle \boldsymbol{\lambda}^*, \mathcal{A}(\mathbf{x}^{K+1}) - \mathbf{b}\right\rangle + \frac{\beta\alpha}{2}\left\|\mathcal{A}(\mathbf{x}^{K+1}) - \mathbf{b}\right\|^2 \leq \frac{2L_{\max}D_{\mathbf{x}^*}^2}{(K+2)^2} + \frac{2\beta\eta_{\max}D_X^2}{K+2} + \frac{2D_\Lambda^2}{\beta(K+2)}, \tag{5}$$

where $\alpha = \min\left\{\frac{1}{n+1}, \left\{\frac{\eta_i - n\|\mathcal{A}_i\|^2}{2(n+1)\|\mathcal{A}_i\|^2}, i = 1, \cdots, n\right\}\right\}$, $L_{\max} = \max\{L_i, i = 1, \cdots, n\}$ and $\eta_{\max} = \max\{\eta_i, i = 1, \cdots, n\}$.

- Remark: Fast PL-ADMM-PS improves the rate of PL-ADMM-PS from $O\left(\frac{D_{\mathbf{x}^*}^2}{K}+\frac{D_{\mathbf{x}^*}^2}{K}+\frac{D_{\lambda^*}^2}{K}\right)$ to $O\left(\frac{D_{\mathbf{x}^*}^2}{K^2}+\frac{D_X^2}{K}+\frac{D_\Lambda^2}{K}\right)$.

Experiments

Comparison of PALM and Fast PALM

- Consider the following problem

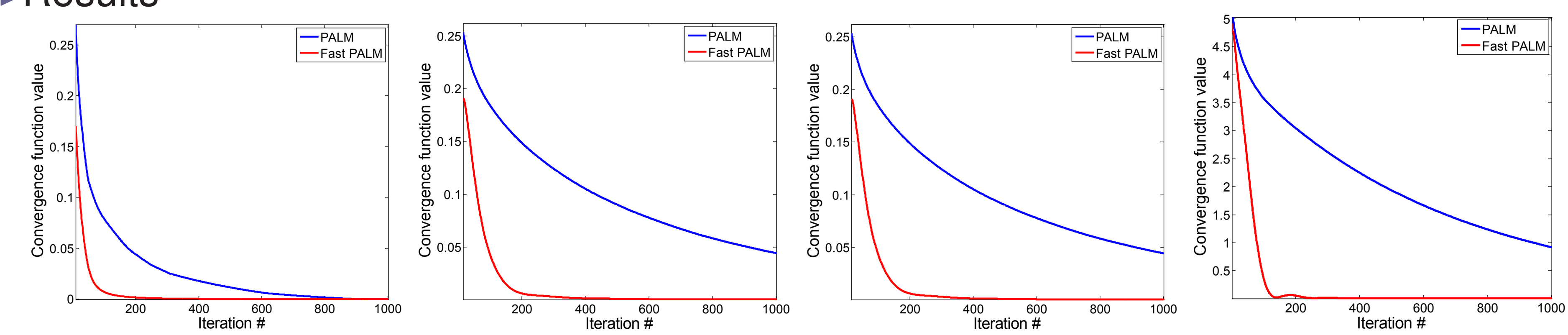
$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 + \frac{\alpha}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2, \quad \text{s.t.} \quad \mathbf{1}^T\mathbf{x} = 1, \tag{6}$$

where $\alpha > 0$, $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{1} \in \mathbb{R}^n$ is the all one vector.

- Generate different sizes of $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$ by the Matlab command `randn`.
- Use the left hand side of (3) as the convergence function to evaluate the convergence behaviors of PALM and Fast PALM.

Comparison of PALM and Fast PALM (Cont')

- Results



(a) (b) (c) (d)

Figure: Plots of the convergence function values of (3) in each iterations by using PALM and Fast PALM for (6) with different sizes of $\mathbf{A} \in \mathbb{R}^{m\times n}$.

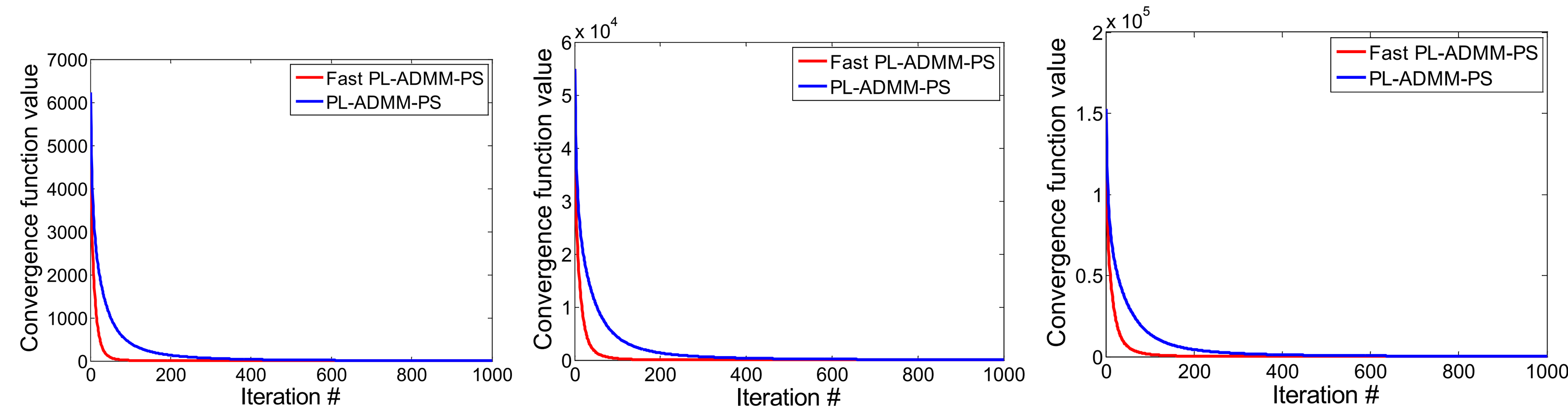
Comparison of PL-ADMM-PS and Fast PL-ADMM-PS

- Conduct a problem with three blocks of variables as follows

$$\min_{\mathbf{X}_1,\mathbf{X}_2,\mathbf{X}_3} \sum_{i=1}^{3}\left(\|\mathbf{X}_i\|_{\ell_i} + \frac{\alpha_i}{2}\|\mathbf{C}_i\mathbf{X}_i - \mathbf{D}_i\|_F^2\right), \ \text{s.t.} \ \sum_{i=1}^{3}\mathbf{A}_i\mathbf{X}_i = \mathbf{B}, \tag{7}$$

where $\|\cdot\|_{\ell_1}$ uses the ℓ_1-norm, $\|\cdot\|_{\ell_2}$ uses the nuclear norm, and $\|\cdot\|_{\ell_3}$ uses the $\ell_{2,1}$-norm defined as the sum of the ℓ_2-norm of each column of a matrix.

- Generate the matrices $\mathbf{A}_i, \mathbf{C}_i, \mathbf{D}_i, \mathbf{B}, \in \mathbb{R}^{m\times m}$ by the Matlab command `randn`.

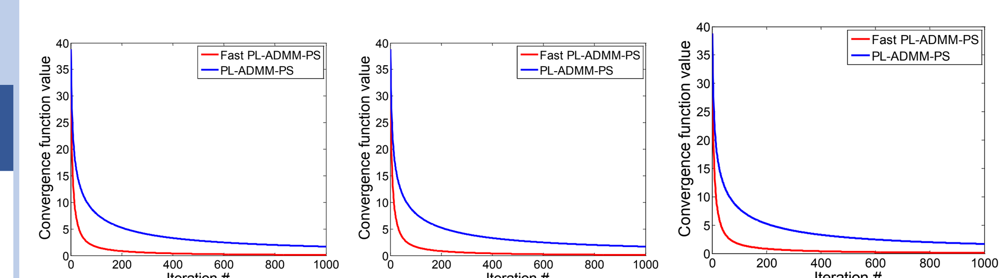


(a) (b) (c)

Figure: Plots of the convergence function values of (5) in each iterations by using PL-ADMM-PS and Fast PL-ADMM-PS for (7) with different sizes of $\mathbf{X} \in \mathbb{R}^{m\times m}$.

- Application to subspace clustering

$$\min_{\mathbf{Z}} \ \alpha_1\|\mathbf{Z}\|_* + \alpha_2\|\mathbf{Z}\|_1 + \frac{1}{2}\|\mathbf{XZ} - \mathbf{X}\|^2, \text{s.t.} \ \mathbf{1}^T\mathbf{Z} = \mathbf{1}^T, \tag{8}$$



(a) (b) (c)

Figure: Plots of the convergence function values of (5) in each iterations by using PL-ADMM-PS and Fast PL-ADMM-PS for (8) with different sizes of data $\mathbf{X}$ for subspace clustering.

Table: Comparision of subspace clustering accuracies (%) on the Extended Yale B database.

Methods	PL-ADMM-PS	Fast PL-ADMM-PS
5 subjects	94.06	96.88
8 subjects	85.94	90.82
10 subjects	75.31	75.47

Department of Electrical and Computer Engineering, National University of Singapore.

Email: canyilu@gmail.com