Linearized Alternating Direction Method: Two Blocks and Multiple Blocks

Zhouchen Lin
林宙辰
北京大学
Dec. 23, 2014
Outline

• Alternating Direction Method (ADM)
• Linearized Alternating Direction Method (LADM) – Two Blocks
• LADM – Multiple Blocks
• Proximal LADM – Multiple Blocks
• Conclusions

Lin et al., Linearized Alternating Direction Method with Adaptive Penalty for Low-Rank Representation, NIPS 2011.
Background

• Optimization is everywhere

Compressed Sensing: \( \min_{x} \|x\|_1, \quad \text{s.t.} \quad Ax = b. \)

RPCA w/ Missing Value: \( \min ||A||_* + \lambda ||E||_1, \quad \text{s.t.} \quad \pi_{\Lambda}(A + E) = d. \)

LASSO: \( \min_{x} \|Ax - b\|_2, \quad \text{s.t.} \quad \|x\|_1 \leq \varepsilon. \)

Image Restoration: \( \min_{x} \|Ax - b\|_2^2 + \lambda \|\nabla x\|_1, \quad \text{s.t.} \quad 0 \leq x \leq 255. \)

Covariance Selection: \( \min_{X} \text{tr}(\Sigma X) - \log(\det(X)) + \rho e^T |X| e, \)
\( s.t. \quad X \in S_{\Lambda}^m, \quad \text{where} \quad S_{\Lambda}^m = \{ X \succ 0 | \lambda_{\min} I \preceq X \preceq \lambda_{\max} I \} \)

Pose Estimation: \( \min_{Q} \text{tr}(WQ), \quad \text{s.t.} \quad \text{tr}(A_i Q) = 0, i = 1, \cdots, m, \)
\( Q \succ 0, \text{rank}(Q) \leq 1. \)
Alternating Direction Method (ADM)

Model Problem:

\[ \min_{x_1, x_2} f_1(x_1) + f_2(x_2), \]
\[ s.t. \quad A_1(x_1) + A_2(x_2) = b, \]

where \( f_i \) are convex functions and \( A_i \) are linear mappings.

\[ \tilde{L}(x_1, x_2, \lambda) = f_1(x_1) + f_2(x_2) + \langle \lambda, A_1(x_1) + A_2(x_2) - b \rangle \]
\[ + \frac{\beta}{2} \| A_1(x_1) + A_2(x_2) - b \|_F^2, \]

\[ x_1^{k+1} = \arg \min_{x_1} \tilde{L}(x_1, x_2^k, \lambda^k), \]
\[ x_2^{k+1} = \arg \min_{x_2} \tilde{L}(x_1^{k+1}, x_2, \lambda^k), \]
\[ \lambda^{k+1} = \lambda^k + \beta_k [ A_1(x_1^{k+1}) + A_2(x_2^{k+1}) - b ]. \]

Update \( \beta_k \)
Linearized Alternating Direction Method (LADM)

\[ \begin{align*}
\mathbf{x}_1^{k+1} &= \arg \min_{\mathbf{x}_1} f_1(\mathbf{x}_1) + \frac{\beta_k}{2} \| \mathbf{A}_1(\mathbf{x}_1) + \mathbf{A}_2(\mathbf{x}_2^k) - \mathbf{b} + \frac{\lambda_k}{\beta_k} \|^2, \\
\mathbf{x}_2^{k+1} &= \arg \min_{\mathbf{x}_2} f_2(\mathbf{x}_2) + \frac{\beta_k}{2} \| \mathbf{A}_2(\mathbf{x}_1^{k+1}) + \mathbf{A}_2(\mathbf{x}_2) - \mathbf{b} + \frac{\lambda_k}{\beta_k} \|^2.
\end{align*} \]

Proximal Operation

\[ \begin{align*}
\min_{\mathbf{x}} f_1(\mathbf{x}) + \frac{\beta}{2} \| \mathbf{x} - \mathbf{w} \|_F^2 \\
\min_{\mathbf{x}} f_2(\mathbf{x}) + \frac{\beta}{2} \| \mathbf{x} - \mathbf{w} \|_F^2
\end{align*} \]

\[ \arg \min_{\mathbf{x}} \| \mathbf{x} \|_1 + \frac{\beta}{2} \| \mathbf{x} - \mathbf{w} \|^2 = \mathcal{T}_{\beta}^{-1}(\mathbf{w}), \]

\[ \mathcal{T}_\varepsilon(\mathbf{x}) = \text{sgn}(\mathbf{x}) \max(|\mathbf{x}| - \varepsilon, 0). \]

\[ \arg \min_{\mathbf{X}} \| \mathbf{X} \|_* + \frac{\varepsilon}{2} \| \mathbf{X} - \mathbf{W} \|_F^2 = \Theta_{\varepsilon}^{-1}(\mathbf{W}) = U \mathcal{T}_{\varepsilon}^{-1}(S)V^T, \]

where \( \mathbf{W} = USV^T \) is the singular value decomposition (SVD) of \( \mathbf{W} \).

Lin et al., Linearized Alternating Direction Method with Adaptive Penalty for Low-Rank Representation, NIPS 2011.
Linearized Alternating Direction Method (LADM)

Introducing auxiliary variables:

\[
\min_{x_1, x_2, x_3, x_4} f_1(x_1) + f_2(x_2),
\]
\[
s.t. \quad x_1 = x_3, x_2 = x_4, A_1(x_3) + A_2(x_4) = b.
\]

\[
\tilde{\mathcal{L}}(x_1, x_2, x_3, x_4, \lambda_1, \lambda_2, \lambda_3)
= f_1(x_1) + f_2(x_2) + \langle \lambda_1, x_1 - x_3 \rangle + \langle \lambda_2, x_2 - x_4 \rangle + \langle \lambda_3, A_1(x_3) + A_2(x_4) - b \rangle + \frac{\beta}{2} \left( \|x_1 - x_3\|_F^2 + \|x_2 - x_4\|_F^2 + \|A_1(x_3) + A_2(x_4) - b\|_F^2 \right),
\]

Three drawbacks:

1. More blocks \(\rightarrow\) more memory & slower convergence.

2. Matrix inversion is expensive.

3. Convergence is NOT guaranteed!

Lin et al., Linearized Alternating Direction Method with Adaptive Penalty for Low-Rank Representation, NIPS 2011.
Linearized Alternating Direction Method (LADM)

\[
\begin{align*}
\mathbf{x}_1^{k+1} &= \arg\min_{\mathbf{x}_1} f_1(\mathbf{x}_1) + \frac{\beta_k}{2} \| \mathbf{A}_1(\mathbf{x}_1) + \mathbf{A}_2(\mathbf{x}_2^k) - \mathbf{b} + \lambda_k / \beta_k \|^2, \\
\mathbf{x}_2^{k+1} &= \arg\min_{\mathbf{x}_2} f_2(\mathbf{x}_2) + \frac{\beta_k}{2} \| \mathbf{A}_2(\mathbf{x}_1^k + 1) + \mathbf{A}_2(\mathbf{x}_2) - \mathbf{b} + \lambda_k / \beta_k \|^2
\end{align*}
\]

- Linearize the quadratic term

\[
\begin{align*}
\mathbf{x}_1^{k+1} &= \arg\min_{\mathbf{x}_1} f_1(\mathbf{x}_1) + \langle \mathbf{A}_1^*(\lambda_k) + \beta_k \mathbf{A}_1^*(\mathbf{A}_1(\mathbf{x}_1^k) + \mathbf{A}_2(\mathbf{x}_2^k) - \mathbf{b}), \mathbf{x}_1 - \mathbf{x}_1^k \rangle \\
& \quad + \frac{\beta_k \eta_1}{2} \| \mathbf{x}_1 - \mathbf{x}_1^k \|^2 \\
&= \arg\min_{\mathbf{x}_1} f_1(\mathbf{x}_1) + \frac{\beta_k \eta_1}{2} \| \mathbf{x}_1 - \mathbf{x}_1^k + \mathbf{A}_1^*(\lambda_k + \beta_k (\mathbf{A}_1(\mathbf{x}_1^k) + \mathbf{A}_2(\mathbf{x}_1^k) - \mathbf{b})) \| / \| \beta_k \eta_1 \|^2,
\end{align*}
\]

\[
\begin{align*}
\mathbf{x}_2^{k+1} &= \arg\min_{\mathbf{x}_2} f_2(\mathbf{x}_2) + \frac{\beta_k \eta_2}{2} \| \mathbf{x}_2 - \mathbf{x}_2^k + \mathbf{A}_2^*(\lambda_k + \beta_k (\mathbf{A}_1(\mathbf{x}_1^{k+1}) + \mathbf{A}_2(\mathbf{x}_2^k) - \mathbf{b})) \| / \| \beta_k \eta_2 \|^2.
\end{align*}
\]

Lin et al., Linearized Alternating Direction Method with Adaptive Penalty for Low-Rank Representation, NIPS 2011.
LADM with Adaptive Penalty (LADMAP)

**Theorem:** If \( \{\beta_k\} \) is non-decreasing and upper bounded, \( \eta_i > \|A_i\|^2, i = 1, 2 \), then the sequence \( \{(x^k_1, x^k_2, \lambda_k)\} \) converges to a KKT point of the model problem.

LADM with Adaptive Penalty (LADMAP)

• Adaptive Penalty

\[
x_{1}^{k+1} = \arg \min_{x_{1}} f_{1}(x_{1}) + \frac{\beta_{k} \eta_{1}}{2} \|x_{1} - x_{1}^{k} + A_{1}^{*}(\lambda_{k} + \beta_{k}(A_{1}(x_{1}^{k}) + A_{2}(x_{1}^{k}) - b))/(\beta_{k} \eta_{1})\|^{2},
\]

\[
x_{2}^{k+1} = \arg \min_{x_{2}} f_{2}(x_{2}) + \frac{\beta_{k} \eta_{2}}{2} \|x_{2} - x_{2}^{k} + A_{2}^{*}(\lambda_{k} + \beta_{k}(A_{1}(x_{1}^{k+1}) + A_{2}(x_{2}^{k}) - b))/(\beta_{k} \eta_{2})\|^{2}.
\]

\[
\downarrow
\]

\[-\beta_{k} \eta_{1}(x_{1}^{k+1} - x_{1}^{k}) - A_{1}^{*}(\lambda_{k} + \beta_{k}(A_{1}(x_{1}^{k}) + A_{2}(x_{1}^{k}) - b)) \in \partial f_{1}(x_{1}^{k+1})
\]

\[-\beta_{k} \eta_{2}(x_{2}^{k+1} - x_{2}^{k}) - A_{2}^{*}(\lambda_{k} + \beta_{k}(A_{1}(x_{1}^{k+1}) + A_{2}(x_{2}^{k}) - b)) \in \partial f_{2}(x_{2}^{k+1})
\]

KKT condition: \(\exists(x^{*}, y^{*}, \lambda^{*})\) such that

\[
A_{1}(x_{1}^{*}) + A_{2}(x_{2}^{*}) - b = 0,
\]

\[-A_{1}^{*}(\lambda^{*}) \in \partial f_{1}(x_{1}^{*}), -A_{2}^{*}(\lambda^{*}) \in \partial f_{2}(x_{2}^{*}).\]
LADM with Adaptive Penalty (LADMAP)

Both $\beta_k \eta_1 \|x_1^{k+1} - x_1^k\|/\|A^*_1(b)\|$ and $\beta_k \eta_2 \|x_2^{k+1} - x_2^k\|/\|A^*_2(b)\|$ should be small.

$$\eta_i = \|A_i\|^2 \implies \text{Approximate } \|A^*_i(b)\| \text{ by } \sqrt{\eta_i} \|b\|$$

- **Adaptive Penalty**

$$\beta_{k+1} = \min(\beta_{\max}, \rho \beta_k),$$

$$\rho = \begin{cases} 
\rho_0, & \text{if } \beta_k \max(\sqrt{\eta_1}\|x_1^{k+1} - x_1^k\|_F, \sqrt{\eta_2}\|x_2^{k+1} - x_2^k\|_F)/\|b\|_F < \varepsilon_2, \\
1, & \text{otherwise}, 
\end{cases}$$

where $\rho_0 \geq 1$ is a constant.

- **Loop until**

$$\|A_1(x_1^{k+1}) + A_2(x_2^{k+1}) - b\|_F < \varepsilon_1,$$

$$\beta_k \max(\sqrt{\eta_1}\|x_1^{k+1} - x_1^k\|_F, \sqrt{\eta_2}\|x_2^{k+1} - x_2^k\|_F)/\|b\|_F < \varepsilon_2.$$
LADM with Adaptive Penalty
(LADMAP)

• Choice of parameters

1. $\beta_0 = \alpha \varepsilon_2$, where $\alpha \propto$ the size of $b$. $\beta_0$ should not be too large, so that $\beta_k$ increases in the first few iterations.

2. $\rho_0 \geq 1$ should be chosen such that $\beta_k$ increases steadily (but not necessarily every iteration).

Lin et al., Linearized Alternating Direction Method with Adaptive Penalty for Low-Rank Representation, NIPS 2011.
LADM with Adaptive Penalty (LADMAP)

• An example (LRR):

\[
\min_{Z,E} \|Z\|_* + \mu \|E\|_1, \quad s.t. \quad X = XZ + E.
\]

\[A_1(Z) = XZ, \quad A_2(E) = E.\]

\[A_1^*(Z) = X^T Z, \quad A_2^*(E) = E, \quad \eta_1 = \|X\|_2^2, \quad \eta_2 = 1.\]
Experiment

Table 1: Comparison among APG, ADM, LADM and LADMAP on the synthetic data. For each quadruple \((s, p, d, \tilde{r})\), the LRR problem, with \(\mu = 0.1\), was solved for the same data using different algorithms. We present typical running time (in \(\times 10^3\) seconds), iteration number, relative error (%) of output solution \((\hat{E}, \hat{Z})\) and the clustering accuracy (%) of tested algorithms, respectively.

<table>
<thead>
<tr>
<th>Size ((s, p, d, \tilde{r}))</th>
<th>Method</th>
<th>Time</th>
<th>Iter.</th>
<th>(\frac{|Z - Z_0|}{|Z_0|})</th>
<th>(\frac{|E - E_0|}{|E_0|})</th>
<th>Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>((10, 20, 200, 5))</td>
<td>APG</td>
<td>0.0332</td>
<td>110</td>
<td>2.2079</td>
<td>1.5096</td>
<td>81.5</td>
</tr>
<tr>
<td></td>
<td>ADM</td>
<td>0.0529</td>
<td>176</td>
<td>0.5491</td>
<td>0.5093</td>
<td>90.0</td>
</tr>
<tr>
<td></td>
<td>LADM</td>
<td>0.0603</td>
<td>194</td>
<td>0.5480</td>
<td>0.5024</td>
<td>90.0</td>
</tr>
<tr>
<td></td>
<td>LADMAP</td>
<td>0.0145</td>
<td>46</td>
<td>0.5480</td>
<td>0.5024</td>
<td>90.0</td>
</tr>
<tr>
<td>((15, 20, 300, 5))</td>
<td>APG</td>
<td>0.0869</td>
<td>106</td>
<td>2.4824</td>
<td>1.0341</td>
<td>80.0</td>
</tr>
<tr>
<td></td>
<td>ADM</td>
<td>0.1526</td>
<td>185</td>
<td>0.6519</td>
<td>0.4078</td>
<td>83.7</td>
</tr>
<tr>
<td></td>
<td>LADM</td>
<td>0.2943</td>
<td>363</td>
<td>0.6518</td>
<td>0.4076</td>
<td>86.7</td>
</tr>
<tr>
<td></td>
<td>LADMAP</td>
<td>0.0336</td>
<td>41</td>
<td>0.6518</td>
<td>0.4076</td>
<td>86.7</td>
</tr>
<tr>
<td>((20, 25, 500, 5))</td>
<td>APG</td>
<td>1.8837</td>
<td>117</td>
<td>2.8905</td>
<td>2.4017</td>
<td>72.4</td>
</tr>
<tr>
<td></td>
<td>ADM</td>
<td>3.7139</td>
<td>225</td>
<td>1.1191</td>
<td>1.0170</td>
<td>80.0</td>
</tr>
<tr>
<td></td>
<td>LADM</td>
<td>8.1574</td>
<td>508</td>
<td>0.6379</td>
<td>0.4268</td>
<td>80.0</td>
</tr>
<tr>
<td></td>
<td>LADMAP</td>
<td>0.7762</td>
<td>40</td>
<td>0.6379</td>
<td>0.4268</td>
<td>84.6</td>
</tr>
<tr>
<td>((30, 30, 900, 5))</td>
<td>APG</td>
<td>6.1252</td>
<td>116</td>
<td>3.0667</td>
<td>0.9199</td>
<td>69.4</td>
</tr>
<tr>
<td></td>
<td>ADM</td>
<td>11.7185</td>
<td>220</td>
<td>0.6865</td>
<td>0.4866</td>
<td>76.0</td>
</tr>
<tr>
<td></td>
<td>LADM</td>
<td>N.A.</td>
<td>N.A.</td>
<td>N.A.</td>
<td>N.A.</td>
<td>N.A.</td>
</tr>
<tr>
<td></td>
<td>LADMAP</td>
<td>2.3891</td>
<td>44</td>
<td>0.6864</td>
<td>0.4294</td>
<td>80.1</td>
</tr>
</tbody>
</table>
LADM with Parallel Splitting and Adaptive Penalty (LADMPSAP)

- Model problem:

\[
\begin{align*}
\min_{x_1, \ldots, x_n} & \sum_{i=1}^{n} f_i(x_i), \\
\text{s.t.} & \sum_{i=1}^{n} A_i(x_i) = b.
\end{align*}
\]

\[
\begin{align*}
\min_{X} & \|X\|_* + \frac{1}{2\mu} \|b - P(X)\|^2, \\
\text{s.t.} & X \geq 0,
\end{align*}
\]

\[
\begin{align*}
\min_{X, e} & \|X\|_* + \frac{1}{2\mu} \|e\|^2, \\
\text{s.t.} & b = P(X) + e, \quad X \geq 0,
\end{align*}
\]

\[
\begin{align*}
\min_{X, Y, e} & \|X\|_* + \frac{1}{2\mu} \|e\|^2, \\
\text{s.t.} & b = P(Y) + e, \quad X = Y, \quad Y \geq 0,
\end{align*}
\]

\[
\begin{align*}
\min_{X, Y, e} & \|X\|_* + \frac{1}{2\mu} \|e\|^2 + \chi_{Y \geq 0}(Y), \\
\text{s.t.} & b = P(Y) + e, \quad X = Y.
\end{align*}
\]

LADM with Parallel Splitting and Adaptive Penalty (LADMPSAP)

- Can we naively generalize two-block LADMAP for multi-block problems?

No!

Actually, the naive generalization of LADMAP may be divergent, e.g., when applied to the following problem with $n \geq 5$:

$$
\min_{x_1, \ldots, x_n} \sum_{i=1}^{n} \|x_i\|_1, \quad \text{s.t.} \quad \sum_{i=1}^{n} A_i x_i = b.
$$


LADM with Parallel Splitting and Adaptive Penalty (LADMPSAP)

\[
x_{i}^{k+1} = \arg\min_{x_{i}} f_{i}(x_{i}) + \frac{\eta_{i}\beta_{k}}{2} \left\| x_{i} - x_{i}^{k} + A_{i}^{*} \left( \lambda^{k} + \beta_{k} \left( \sum_{j=1}^{n} A_{i}(x_{j}^{k}) - b \right) \right) \right\|^2 / (\eta_{i}\beta_{k}),
\]

\[i = 1, \ldots, n,
\]

\[
\lambda^{k+1} = \lambda^{k} + \beta_{k} \left( \sum_{i=1}^{n} A_{i}(x_{i}^{k+1}) - b \right),
\]

\[
\beta_{k+1} = \min(\beta_{\text{max}}, \rho_{\beta_{k}}),
\]

where

\[
\rho = \begin{cases} 
\rho_{0}, & \text{if } \beta_{k} \max \left( \{ \sqrt{\eta_{i}} \left\| x_{i}^{k+1} - x_{i}^{k} \right\|, i = 1, \ldots, n \} \right) / \| b \| < \varepsilon_{2}, \\
1, & \text{otherwise},
\end{cases}
\]

with \( \rho_{0} > 1 \) being a constant and \( 0 < \varepsilon_{2} \ll 1 \) being a threshold.

LADM with Parallel Splitting and Adaptive Penalty (LADMPSAP)

**Theorem:** If \( \{\beta_k\} \) is non-decreasing and upper bounded, \( \eta_i > n\|A_i\|^2, i = 1, \cdots, n \), then \( \{(x_i^k, \lambda^k)\} \) generated by LADMPSAP converges to a KKT point of the problem.

**Remark:** When \( n = 2 \), LADMPSAP is weaker than LADMAP:

\[
\eta_i > 2\|A_i\|^2 \text{ vs. } \eta_i > \|A_i\|^2.
\]


LADM with Parallel Splitting and Adaptive Penalty (LADMPSAP)

• Model problem:

\[
\min_{x_1, \ldots, x_n} \sum_{i=1}^{n} f_i(x_i), \quad \text{s.t.} \quad \sum_{i=1}^{n} A_i(x_i) = b, \quad x_i \in X_i, \quad i = 1, \ldots, n,
\]

where \( X_i \subseteq \mathbb{R}^{d_i} \) is a closed convex set.

\[\Downarrow\]

\[
\min_{x_1, \ldots, x_{2n}} \sum_{i=1}^{n} f_i(x_i) + \sum_{i=n+1}^{2n} \chi_{x_i \in X_{i-n}}(x_i), \quad \text{s.t.} \quad \sum_{i=1}^{n} A_i(x_i) = b, \quad x_i = x_{n+i}, \quad i = 1, \ldots, n.
\]

**Theorem:** If \( \{\beta_k\} \) is non-decreasing and upper bounded, \( x_{n+1}, \ldots, x_{2n} \) are auxiliary variables, \( \eta_i > n\|A_i\|^2 + 2, \eta_{n+i} > 2, \quad i = 1, \ldots, n \), then \( \{(x_i^k), \lambda^k\} \) generated by LADMPSAP converges to a KKT point of the problem.

\[
\eta_i > 2n(\|A_i\|^2 + 1), \quad \eta_{n+i} > 2n, \quad i = 1, \ldots, n
\]

Experiment

\[
\min_X \|X\|_* + \frac{1}{2\mu} \|b - P(X)\|^2, \quad s.t. \quad X \geq 0,
\]

\[\downarrow\]

\[
\min_{X,Y,e} \|X\|_* + \frac{1}{2\mu} \|e\|^2 + \chi_{Y \geq 0}(Y), \quad s.t. \quad b = P(Y) + e, \quad X = Y.
\]

(a) Original  (b) Corrupted  (c) FPCA  (d) LADM  (e) LADMPSPAP

Experiment

Table 1: Numerical comparison on the NMC problem with synthetic data, average of 10 runs. $q$, $t$ and $d_r$ denote, respectively, sample ratio, the number of measurements $t=q(mn)$ and the “degree of freedom” defined by $d_r=r(m+n-r)$ for a matrix with rank $r$ and $q$. Here we set $m=n$ and fix $r=10$ in all the tests.

<table>
<thead>
<tr>
<th>X</th>
<th>LADM</th>
<th>LADMPSAP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$n$</td>
<td>$q$</td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>1000</td>
<td>20%</td>
<td>10.05</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>5.03</td>
</tr>
<tr>
<td>5000</td>
<td>20%</td>
<td>50.05</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>25.03</td>
</tr>
<tr>
<td>10000</td>
<td>10%</td>
<td>50.03</td>
</tr>
</tbody>
</table>

Table 1: Numerical comparison on the image inpainting problem.

<table>
<thead>
<tr>
<th>Method</th>
<th>#Iter.</th>
<th>Time(s)</th>
<th>PSNR</th>
<th>FA</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPCA</td>
<td>179</td>
<td>228.99</td>
<td>27.77dB</td>
<td>9.41E-4</td>
</tr>
<tr>
<td>LADM</td>
<td>228</td>
<td>207.95</td>
<td>26.98dB</td>
<td>2.92E-3</td>
</tr>
<tr>
<td>LADMPASAP</td>
<td>143</td>
<td>134.89</td>
<td>31.39dB</td>
<td>0</td>
</tr>
</tbody>
</table>
LADM with Parallel Splitting and Adaptive Penalty (LADMPSAP)

Enhanced convergence results:

**Theorem 1:** If \( \{\beta_k\} \) is non-decreasing and \( \sum_{k=1}^{+\infty} \beta_k^{-1} = +\infty \), \( \eta_i > n\|A_i\|^2 \), \( \partial f_i(x) \) is bounded, \( i = 1, \cdots, n \), then the sequence \( \{x_i^k\} \) generated by LADMPSAP converges to an optimal solution to the model problem.

**Theorem 2:** If \( \{\beta_k\} \) is non-decreasing, \( \eta_i > n\|A_i\|^2 \), \( \partial f_i(x) \) is bounded, \( i = 1, \cdots, n \), then \( \sum_{k=1}^{+\infty} \beta_k^{-1} = +\infty \) is also the necessary condition for the global convergence of \( \{x_i^k\} \) generated by LADMPSAP to an optimal solution to the model problem.

With the above analysis, when all the subgradients of the component objective functions are bounded we can remove the upper bound \( \beta_{\text{max}} \).
LADM with Parallel Splitting and Adaptive Penalty (LADMPASAP)

Define $\mathbf{x} = (\mathbf{x}_1^T, \cdots, \mathbf{x}_n^T)^T$, $\mathbf{x}^* = ((\mathbf{x}_1^*)^T, \cdots, (\mathbf{x}_2^*)^T)^T$ and $f(\mathbf{x}) = \sum_{i=1}^n f_i(\mathbf{x}_i)$, where $(\mathbf{x}_1^*, \cdots, \mathbf{x}_2^*, \lambda^*)$ is a KKT point of the model problem.

**Proposition:** $\tilde{\mathbf{x}}$ is an optimal solution to the model problem iff there exists $\alpha > 0$, such that

$$f(\tilde{\mathbf{x}}) - f(\mathbf{x}^*) + \sum_{i=1}^n \langle \mathbf{A}_i^*(\lambda^*), \tilde{\mathbf{x}}_i - \mathbf{x}_i^* \rangle + \alpha \left\| \sum_{i=1}^n \mathbf{A}_i(\tilde{\mathbf{x}}_i) - \mathbf{b} \right\|^2 = 0.$$ 

Our criterion for checking the optimality of a solution is much simpler than that in He et al. 2011, which has to compare with all $(\mathbf{x}_1, \cdots, \mathbf{x}_n, \lambda) \in \mathbb{R}^{d_1} \times \cdots \times \mathbb{R}^{d_n} \times \mathbb{R}^m$.

---


LADM with Parallel Splitting and Adaptive Penalty (LADMPsAP)

**Theorem 3:** Define $\bar{x}^K = \sum_{k=0}^{K} \gamma_k x^{k+1}$, where $\gamma_k = \beta_k^{-1} / \sum_{j=0}^{K} \beta_j^{-1}$. Then

$$f(\bar{x}^K) - f(x^*) + \sum_{i=1}^{n} A_i^*(\lambda^*), \bar{x}_i^K - x_i^* \rangle + \frac{\alpha \beta_0}{2} \left\| \sum_{i=1}^{n} A_i(\bar{x}_i^K) - b \right\|^2 \leq C_0 / \left( 2 \sum_{k=0}^{K} \beta_k^{-1} \right),$$

where $\alpha^{-1} = (n+1) \max \left( 1, \left\{ \frac{\| A_i \|^2}{\eta_i - n \| A_i \|^2}, i = 1, \ldots, n \right\} \right)$ and $C_0 = \sum_{i=1}^{n} \eta_i \| x_i^0 - x_i^* \|^2 + \beta_0^{-2} \| \lambda^0 - \lambda^* \|^2$.

*A much simpler proof of convergence rate (in ergodic sense)!*

Proximal LADMPSAP

- Even more general problem:

\[
\min_{x_1, \ldots, x_n} \sum_{i=1}^{n} f_i(x_i), \quad \text{s.t.} \quad \sum_{i=1}^{n} A_i(x_i) = b.
\]

\[
f_i(x_i) = g_i(x_i) + h_i(x_i),
\]

where both \( g_i \) and \( h_i \) are convex, \( g_i \) is \( C^{1,1} \):

\[
\|\nabla g_i(x) - \nabla g_i(y)\| \leq L_i \|x - y\|, \quad \forall x, y \in \mathbb{R}^{d_i},
\]

and \( h_i \) may not be differentiable but its proximal operation is easily solvable.
Proximal LADMPSPAP

- Linearize the augmented term to obtain:

\[
x_{i}^{k+1} = \arg\min_{x_i} h_i(x_i) + g_i(x_i) + \frac{\sigma_i^{(k)}}{2} \left\| x_i - x_i^k + A_i^\dagger(\hat{\lambda}^k) \sigma_i^{(k)} \right\|^2, \quad i = 1, \ldots, n,
\]

- Further linearize \( g_i \):

\[
x_{i}^{k+1} = \arg\min_{x_i} h_i(x_i) + g_i(x_i^k) + \frac{\sigma_i^{(k)}}{2} \left\| A_i^\dagger(\hat{\lambda}^k) \right\|^2
\]

\[
+ \langle \nabla g_i(x_i^k) + A_i^\dagger(\hat{\lambda}^k), x_i - x_i^k \rangle + \frac{\tau_i^{(k)}}{2} \left\| x_i - x_i^k \right\|^2
\]

\[
= \arg\min_{x_i} h_i(x_i) + \frac{\tau_i^{(k)}}{2} \left\| x_i - x_i^k + \frac{1}{\tau_i^{(k)}} [A_i^\dagger(\hat{\lambda}^k) + \nabla g_i(x_i^k)] \right\|^2.
\]

- Convergence condition:

\[
\tau_i^{(k)} = T_i + \beta_k \eta_i, \text{ where } T_i \geq L_i \text{ and } \eta_i > n \| A_i \|^2 \text{ are both positive constants.}
\]
Experiment

- Group Sparse Logistic Regression with Overlap

\[
\min_{\mathbf{w}, b} \frac{1}{s} \sum_{i=1}^{s} \log (1 + \exp (-y_i (\mathbf{w}^T \mathbf{x}_i + b))) + \mu \sum_{j=1}^{t} \| \mathbf{S}_j \mathbf{w} \|, \tag{1}
\]

where \( \mathbf{x}_i \) and \( y_i, i = 1, \cdots, s \), are the training data and labels, respectively, and \( \mathbf{w} \) and \( b \) parameterize the linear classifier. \( \mathbf{S}_j, j = 1, \cdots, t \), are the selection matrices, with only one 1 at each row and the rest entries are all zeros. The groups of entries, \( \mathbf{S}_j \mathbf{w}, j = 1, \cdots, t \), may overlap each other.

Introducing \( \mathbf{\bar{w}} = (\mathbf{w}^T, b)^T, \mathbf{\bar{x}}_i = (\mathbf{x}_i^T, 1)^T, \mathbf{z} = (\mathbf{z}_1^T, \mathbf{z}_2^T, \cdots, \mathbf{z}_t^T)^T \), and \( \mathbf{\bar{S}} = (\mathbf{S}, \mathbf{0}) \), where \( \mathbf{S} = (\mathbf{S}_1^T, \cdots, \mathbf{S}_t^T)^T \), (1) can be rewritten as

\[
\min_{\mathbf{\bar{w}}, \mathbf{z}} \frac{1}{s} \sum_{i=1}^{s} \log (1 + \exp (-y_i (\mathbf{\bar{w}}^T \mathbf{\bar{x}}_i))) + \mu \sum_{j=1}^{t} \| \mathbf{z}_j \|, \quad s.t. \quad \mathbf{z} = \mathbf{\bar{S}} \mathbf{\bar{w}}, \tag{2}
\]

The Lipschitz constant of the gradient of logistic function with respect to \( \mathbf{\bar{w}} \) can be proven to be \( L_{\mathbf{\bar{w}}} \cdot \frac{1}{4s} \| \mathbf{\bar{X}} \|_2^2 \), where \( \mathbf{\bar{X}} = (\mathbf{\bar{x}}_1, \mathbf{\bar{x}}_2, \cdots, \mathbf{\bar{x}}_s) \).
## Experiment

<table>
<thead>
<tr>
<th>(s, p, t, q)</th>
<th>Method</th>
<th>Time</th>
<th>#Iter.</th>
<th>‖(\hat{\mathbf{w}} - \hat{\mathbf{w}}^*\‖_2</th>
<th>‖(\hat{\mathbf{z}} - \hat{\mathbf{z}}^*\‖_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(300, 901, 100, 10)</td>
<td>ADM</td>
<td>294.15</td>
<td>43</td>
<td>0.4800</td>
<td>0.4790</td>
</tr>
<tr>
<td></td>
<td>LADM</td>
<td>229.03</td>
<td>43</td>
<td>0.5331</td>
<td>0.5320</td>
</tr>
<tr>
<td></td>
<td>LADMPS</td>
<td>105.50</td>
<td>47</td>
<td>0.2088</td>
<td>0.2094</td>
</tr>
<tr>
<td></td>
<td>LADMPSAP</td>
<td>57.46</td>
<td>39</td>
<td>0.0371</td>
<td>0.0368</td>
</tr>
<tr>
<td></td>
<td>pLADMPSAP</td>
<td><strong>1.97</strong></td>
<td>141</td>
<td><strong>0.0112</strong></td>
<td><strong>0.0112</strong></td>
</tr>
<tr>
<td>(450, 1351, 150, 15)</td>
<td>ADM</td>
<td>450.96</td>
<td>33</td>
<td>0.4337</td>
<td>0.4343</td>
</tr>
<tr>
<td></td>
<td>LADM</td>
<td>437.12</td>
<td>36</td>
<td>0.5126</td>
<td>0.5133</td>
</tr>
<tr>
<td></td>
<td>LADMPS</td>
<td>201.30</td>
<td>39</td>
<td>0.1938</td>
<td>0.1937</td>
</tr>
<tr>
<td></td>
<td>LADMPSAP</td>
<td>136.64</td>
<td>37</td>
<td>0.0321</td>
<td>0.0306</td>
</tr>
<tr>
<td></td>
<td>pLADMPSAP</td>
<td><strong>4.16</strong></td>
<td>150</td>
<td><strong>0.0131</strong></td>
<td><strong>0.0131</strong></td>
</tr>
<tr>
<td>(600, 1801, 200, 20)</td>
<td>ADM</td>
<td>1617.09</td>
<td>62</td>
<td>1.4299</td>
<td>1.4365</td>
</tr>
<tr>
<td></td>
<td>LADM</td>
<td>1486.23</td>
<td>63</td>
<td>1.5200</td>
<td>1.5279</td>
</tr>
<tr>
<td></td>
<td>LADMPS</td>
<td>494.52</td>
<td>46</td>
<td>0.4915</td>
<td>0.4936</td>
</tr>
<tr>
<td></td>
<td>LADMPSAP</td>
<td>216.45</td>
<td>32</td>
<td>0.0787</td>
<td>0.0783</td>
</tr>
<tr>
<td></td>
<td>pLADMPSAP</td>
<td><strong>5.77</strong></td>
<td>127</td>
<td><strong>0.0276</strong></td>
<td><strong>0.0277</strong></td>
</tr>
</tbody>
</table>
Conclusions

• LADMAP, LADMPSAP, and P-LADMPSAP are very general methods for solving various convex programs.
• Adaptive penalty is important for fast convergence.
Thanks!

- zlin@pku.edu.cn
- http://www.cis.pku.edu.cn/faculty/vision/zlin/zlin.htm