On the Convergence and Complexity of Nonconvex ADMM

Shiqian Ma

Department of Mathematics, UC Davis

Based on Joint Works with Bo Jiang, Tianyi Lin, Junyu Zhang, Shuzhong Zhang

ADMM and Proximal Splitting Methods in Optimization
DIMACS, Rutgers University
June 11-13, 2018
Nonconvex objective, convex constraints

\[
\min \ f(x_1, x_2, \ldots, x_N) + \sum_{i=1}^{N-1} r_i(x_i)
\]
\[
\text{s.t. } \sum_{i=1}^{N} A_i x_i = b, \ x_i \in \mathcal{X}_i, \ i = 1, \ldots, N - 1,
\]

- \( f \) is differentiable, can be nonconvex
- \( r_i \) can be nonsmooth and nonconvex
- \( \mathcal{X}_i \subseteq \mathbb{R}^{n_i} \) is closed convex set
- Note that \( r_N \) and \( \mathcal{X}_N \) are missing
- Further, require \( A_N \) full row rank (can be relaxed by additional assumptions)
- These restrictions can be removed, with a worse complexity bound
Examples

- **Matrix robust PCA:**

  \[
  \begin{align*}
  & \min_{X,Y,Z,E,B} \quad \|Z - XY^\top\|_F^2 + \alpha R(E) + \gamma \|B\|_F^2 \\
  & \text{s.t.} \quad M = Z + E + B
  \end{align*}
  \]

  \(R(E)\) promotes sparsity; nonconvex
  - SCAD ([Fan-Li-2001])
  - MCP ([Zhang-2010])
  - log-sum penalty ([Candès-Wakin-Boyd-2007])
  - capped-\(\ell_1\) penalty ([Zhang-Zhang, 2010, 2013]).

- **Tensor robust PCA in Tucker-rank:**

  \[
  \begin{align*}
  & \min_{X_i,C,Z,E,B} \quad \|Z - C \times_1 X_1 \times_2 X_2 \ldots \times_d X_d\|_F^2 + \alpha R(E) + \gamma \|B\|_F^2 \\
  & \text{s.t.} \quad \mathcal{T} = Z + E + B
  \end{align*}
  \]
[Hong-Luo-Razaviyayn-2014]: Consensus and sharing problem

- **Consensus problem:** \( \min \sum_{k=1}^{K} g_k(x) + h(x), \) s.t., \( x \in \mathcal{X} \)
ADMM solves the following reformulation

\[
\min \sum_{k=1}^{K} g_k(x_k) + h(x) \\
\text{s.t.} \quad x_k = x, \forall k = 1, \ldots, K, x \in \mathcal{X}. \tag{1}
\]

Require \( g_k \) smooth (possibly nonconvex), \( h \) convex nonsmooth.

- **Sharing problem:**
\[
\min \sum_{k=1}^{K} g_k(x_k) + \ell(\sum_{k=1}^{K} A_k x_k), \text{ s.t., } x_k \in \mathcal{X}_k, k = 1, \ldots, K
\]
ADMM solves the following reformulation

\[
\min \sum_{k=1}^{K} g_k(x_k) + \ell(x) \\
\text{s.t.} \quad \sum_{k=1}^{K} A_k x_k = x, x_k \in \mathcal{X}_k, k = 1, \ldots, K \tag{2}
\]

Require \( g_k \) smooth (possibly nonconvex), \( \ell \) smooth.
Existing works on nonconvex ADMM

- (Hong-2016): Jacobi ADMM for solving

\[
\min f(x), \text{ s.t., } Ax = b, x \in \mathbb{R}^n
\]

where \( f \) is smooth and nonconvex.

- Analyzed iteration complexity for obtaining an \( \epsilon \)-stationary solution

- \( \epsilon \)-stationary solution is defined as \((x^*, \lambda^*)\) such that

\[
\|\nabla_x \mathcal{L}_\gamma(x^*, \lambda^*)\|^2 + \|Ax^* - b\|^2 \leq \epsilon
\]

- Wang, Yin, Zeng, 2015.

- Many others ......
Back to our problem

\[
\begin{align*}
\min & \quad f(x_1, x_2, \ldots, x_N) + \sum_{i=1}^{N-1} r_i(x_i) \\
\text{s.t.} & \quad \sum_{i=1}^{N} A_i x_i = b, \quad x_i \in \mathcal{X}_i, \quad i = 1, \ldots, N - 1,
\end{align*}
\]

Assumptions:

- $\nabla f$ is Lipschitz continuous
- $f$ and $r_i$ are lower bounded.
- We consider two settings under which we can show complexity of ADMM.

<table>
<thead>
<tr>
<th>Setting</th>
<th>$r_i$, $i = 1, \ldots, N - 1$</th>
<th>$\mathcal{X}_i$, $i = 1, \ldots, N - 1$</th>
<th>$\epsilon$-stationary solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting 1</td>
<td>Lipschitz continuous</td>
<td>$\mathcal{X}_i \subset \mathbb{R}^{n_i}$ compact</td>
<td>Definition 1</td>
</tr>
<tr>
<td>Setting 2</td>
<td>lower semi-continuous</td>
<td>$\mathcal{X}_i = \mathbb{R}^{n_i}$</td>
<td>Definition 2</td>
</tr>
</tbody>
</table>
Definition 1 of $\epsilon$-stationary solution: Setting 1

Under the conditions in **Setting 1**, for $\epsilon \geq 0$, we call $(x_1^*, \ldots, x_N^*)$ an $\epsilon$-stationary solution if there exists a Lagrange multiplier $\lambda^*$ such that the following holds for any $(x_1, \ldots, x_N) \in X_1 \times \cdots \times X_{N-1} \times \mathbb{R}^{nN}$:

\[
\begin{align*}
(x_i - x_i^*)^\top \left[ g_i^* + \nabla_i f(x_1^*, \ldots, x_N^*) - A_i^\top \lambda^* \right] & \geq -\epsilon, \quad i = 1, \ldots, N - 1, \\
\left\| \nabla_N f(x_1^*, \ldots, x_{N-1}^*, x_N^*) - A_N^\top \lambda^* \right\| & \leq \epsilon, \\
\left\| \sum_{i=1}^N A_i x_i^* - b \right\| & \leq \epsilon,
\end{align*}
\]

where $g_i^*$ is a general subgradient of $r_i$ at point $x_i^*$. If $\epsilon = 0$, we call $(x_1^*, \ldots, x_N^*)$ a stationary solution.
Definition 2 of $\epsilon$-stationary solution: Setting 2

Under the conditions in **Setting 2**, for $\epsilon \geq 0$, we call $(x_1^*, \ldots, x_N^*)$ an $\epsilon$-stationary solution if there exists a Lagrange multiplier $\lambda^*$ such that the last 2 inequalities in Definition 1 hold and the following holds for any $(x_1, \cdots, x_N) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_{N-1} \times \mathbb{R}^{nN}$:

$$\text{dist} \left( -\nabla_i f(x_1^*, \cdots, x_N^*) + A_i^\top \lambda^*, \partial r_i(x_i^*) \right) \leq \epsilon, \quad i = 1, \ldots, N - 1,$$

where $\partial r_i(x_i^*)$ is the general subgradient of $r_i$ at $x_i^*$, $i = 1, 2, \ldots, N - 1$. If $\epsilon = 0$, we call $(x_1^*, \cdots, x_N^*)$ a stationary solution.
ADMM-g: An ADMM Variant for $A_N = I$

**ADMM-g**

Given $(x_1^0, x_2^0, \ldots, x_N^0) \in \mathcal{X}_1 \times \ldots \times \mathcal{X}_{N-1} \times \mathbb{R}^{nN}$, $\lambda^0 \in \mathbb{R}^m$.

for $k = 1, 2, \ldots, N$, do

for $i = 1, \ldots, N-1,$

$x_i^{k+1} := \text{argmin}_{x_i \in \mathcal{X}_i} \tilde{L}_\beta(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i, x_{i+1}^{k+1}, \ldots, x_N^k, \lambda^k) + \frac{1}{2} \|x_i - x_i^k\|_{H_i}^2$

$x_N^{k+1} := x_N^k - \gamma \nabla_N \mathcal{L}_\beta(x_1^{k+1}, x_2^{k+1}, \ldots, x_N^k, \lambda^k)$.

$\lambda^{k+1} := \lambda^k - \beta \left( \sum_{i=1}^N A_i x_i^{k+1} - b \right)$.

end for

where $\tilde{L}_\beta$ denotes the linearization of $\mathcal{L}_\beta$ (only the smooth part is linearized)
Theorem (Jiang-Lin-M-Zhang-2016)

Under certain conditions (on $\beta, \gamma, H_i$), ADMM-g returns an $\epsilon$-stationary point in $O(1/\epsilon^2)$ iterations for both Setting 1 and Setting 2.

$$\beta > \max \left( \frac{18\sqrt{3} + 6}{13} L, \max_{i=1,2,...,N-1} \frac{6L^2}{\sigma_{\min}(H_i)} \right),$$

and

$$\gamma \in \left( \frac{13\beta - \sqrt{13\beta^2 - 12\beta L - 72L^2}}{6L^2 + \beta L + 13\beta^2}, \frac{13\beta + \sqrt{13\beta^2 - 12\beta L - 72L^2}}{6L^2 + \beta L + 13\beta^2} \right).$$
Proof for Complexity of ADMM-g

- Construct a potential function
  \[ \Psi_G(x_1, x_2, \ldots, x_N, \lambda, \bar{x}) = \mathcal{L}_\beta(x_1, x_2, \ldots, x_N, \lambda) + \frac{3}{\beta} \left[ \left( \beta - \frac{1}{\gamma} \right)^2 + L^2 \right] \|x_N - \bar{x}\|^2 \]

- **Lemma 1**: \( \Psi_G(x_1^{k+1}, \ldots, x_N^{k+1}, \lambda^{k+1}, x_N^k) \) monotonically decreases over \( k \geq 0 \).

- **Lemma 2**: \( \Psi_G(x_1^{k+1}, \ldots, x_N^{k+1}, \lambda^{k+1}, x_N^k) \) is uniformly lower-bounded.

- Combining Lemmas 1 and 2 yields \( \Psi_G(x_1^{k+1}, \ldots, x_N^{k+1}, \lambda^{k+1}, x_N^k) \) converges, from which it is easy to analyze iteration complexity.
ADMM-m: An ADMM Variant for $A_N$ of full row rank

ADMM-m

Given $(x_0^0, x_0^1, \cdots, x_0^N) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_{N-1} \times \mathbb{R}^{nN}, \lambda^0 \in \mathbb{R}^m$

for $k = 0, 1, \ldots,$ do

for $k = 1, \ldots, N-1,$

$x_{i}^{k+1} := \arg\min_{x_i \in \mathcal{X}_i} \mathcal{L}_\beta(x_1^{k+1}, \cdots, x_i^{k+1}, x_{i-1}, x_{i+1}, \cdots, x_N^k, \lambda^k) + \frac{1}{2} \| x_i - x_i^k \|^2_{H_i}

x_N^{k+1} := \arg\min_{x_N} U(x_1^{k+1}, \cdots, x_N^{k+1}, x_N, \lambda^k, x_N^k)

\lambda^{k+1} := \lambda^k - \beta \left( \sum_{i=1}^N A_i x_i^{k+1} - b \right)$

end for

\[
U(x_1, \cdots, x_{N-1}, x_N, \lambda, \bar{x}) = f(x_1, \cdots, x_{N-1}, \bar{x}) + (x_N - \bar{x})^\top \nabla_{x_N} f(x_1, \cdots, x_{N-1}, \bar{x})
\]

\[
+ \frac{L}{2} \| x_N - \bar{x} \|^2 - \left\langle \lambda, \sum_{i=1}^N A_i x_i - b \right\rangle + \frac{\beta}{2} \left\| \sum_{i=1}^N A_i x_i - b \right\|^2.
\]

Theorem (Jiang-Lin-M-Zhang-2016)

Under certain conditions, ADMM-m returns an $\epsilon$-stationary point in $O(1/\epsilon^2)$ iterations for both Setting 1 and Setting 2.
A Byproduct: Proximal BCD for Nonconvex Problems

- Multi-block problem

\[
\begin{align*}
\min \quad & F(x_1, x_2, \ldots, x_N) := f(x_1, x_2, \ldots, x_N) + \sum_{i=1}^{N} r_i(x_i) \\
\text{s.t.} & \quad x_i \in \mathcal{X}_i, \quad i = 1, \ldots, N,
\end{align*}
\]

\(f\) differentiable, \(r_i\) nonsmooth, and \(\mathcal{X}_i\) is closed convex set

**Proximal BCD Algorithm (P-BCD)**

Given \((x_1^0, x_2^0, \ldots, x_N^0) \in \mathcal{X}_1 \times \ldots \times \mathcal{X}_N\).

**for** \(k = 1, 2, \ldots, N,\) **do**

Update block \(x_i\) in a cyclic order:

\[
x_i^{k+1} := \arg\min_{x_i \in \mathcal{X}_i} F(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i, x_{i+1}^k \ldots, x_N^k) + \frac{1}{2} \| x_i - x_i^k \|_{H_i}^2
\]

**end for**
Proximal BCD and ADMM-g

- Introduce auxiliary variable $x_{N+1}$ and an arbitrary vector $b$
- Rewrite the multi-block problem as

\[
\min \ f(x_1, x_2, \ldots, x_N) + \sum_{i=1}^{N} r_i(x_i)
\]

\[
\text{s.t.} \quad x_{N+1} = b, \ x_i \in \mathcal{X}_i, \ i = 1, \ldots, N.
\]

- Satisfies Setting 1. Can be solved by ADMM-g.
- Easy to show: ADMM-g is equivalent to P-BCD.

**Theorem (Jiang-Lin-M-Zhang-2016)**

Under certain conditions, P-BCD returns an $\epsilon$-stationary point in $O(1/\epsilon^2)$ iterations.
Numerical results on robust tensor PCA

- Robust tensor PCA for 3rd-order tensor:

$$\min_{A,B,C,Z,E,B} \| Z - [A, B, C] \|^2 + \alpha \| E \|_1 + \| B \|^2$$

s.t. $$Z + E + B = T,$$

where $$T \in \mathbb{R}^{I_1 \times I_2 \times I_3}, A \in \mathbb{R}^{I_1 \times R}, B \in \mathbb{R}^{I_2 \times R}, C \in \mathbb{R}^{I_3 \times R}$$ and $$[A, B, C] := \sum_{r=1}^{R} a^r \otimes b^r \otimes c^r$$, $$R$$ denotes an estimate to the CP-rank, and $$\otimes$$ denotes outer product of vectors.
### Numerical Results

<table>
<thead>
<tr>
<th>R</th>
<th>ADMM-g</th>
<th></th>
<th>ADMM-m</th>
<th></th>
<th>BCD</th>
<th></th>
<th>proximal BCD</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tensor Size 10 × 20 × 30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>371.80</td>
<td>0.0362</td>
<td>19</td>
<td>395.25</td>
<td>0.0362</td>
<td>19</td>
<td>678.15</td>
<td>0.7093</td>
</tr>
<tr>
<td>10</td>
<td>632.10</td>
<td>0.0320</td>
<td>17</td>
<td>566.15</td>
<td>0.0320</td>
<td>17</td>
<td>1292.10</td>
<td>0.9133</td>
</tr>
<tr>
<td>15</td>
<td>529.25</td>
<td>0.0165</td>
<td>18</td>
<td>545.05</td>
<td>0.0165</td>
<td>18</td>
<td>1458.65</td>
<td>0.9224</td>
</tr>
<tr>
<td>Tensor Size 15 × 25 × 40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>516.30</td>
<td>0.0163</td>
<td>19</td>
<td>636.85</td>
<td>0.0437</td>
<td>17</td>
<td>611.25</td>
<td>0.8597</td>
</tr>
<tr>
<td>10</td>
<td>671.80</td>
<td>0.0345</td>
<td>17</td>
<td>723.20</td>
<td>0.0385</td>
<td>17</td>
<td>1223.60</td>
<td>0.9072</td>
</tr>
<tr>
<td>20</td>
<td>776.70</td>
<td>0.0341</td>
<td>16</td>
<td>922.25</td>
<td>0.0412</td>
<td>15</td>
<td>1716.05</td>
<td>0.9544</td>
</tr>
<tr>
<td>Tensor Size 30 × 50 × 70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>909.05</td>
<td>0.1021</td>
<td>13</td>
<td>1004.30</td>
<td>0.1006</td>
<td>13</td>
<td>1094.05</td>
<td>0.9271</td>
</tr>
<tr>
<td>20</td>
<td>1304.65</td>
<td>0.1233</td>
<td>7</td>
<td>1386.75</td>
<td>0.1387</td>
<td>6</td>
<td>1635.80</td>
<td>0.9668</td>
</tr>
<tr>
<td>40</td>
<td>1261.25</td>
<td>0.0623</td>
<td>10</td>
<td>1387.40</td>
<td>0.0779</td>
<td>7</td>
<td>2000.00</td>
<td>0.9798</td>
</tr>
</tbody>
</table>

**Table:** Numerical results for tensor robust PCA with initial guess $R = R_{CP}$, averaged over 20 instances. Claim success if $Err < 0.01$
### Numerical Results (cont.)

<table>
<thead>
<tr>
<th>R</th>
<th>ADMM-g</th>
<th></th>
<th></th>
<th>ADMM-m</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>BCD</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>proximal BCD</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1830.65</td>
<td>0.0032</td>
<td>20</td>
<td>1758.90</td>
<td>0.0032</td>
<td>20</td>
<td>462.90</td>
<td>0.7763</td>
<td>0</td>
<td>1734.85</td>
<td>0.0032</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1493.20</td>
<td>0.0029</td>
<td>20</td>
<td>1586.00</td>
<td>0.0029</td>
<td>20</td>
<td>1277.15</td>
<td>0.9133</td>
<td>0</td>
<td>1137.15</td>
<td>0.0029</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1336.65</td>
<td>0.0078</td>
<td>19</td>
<td>1486.40</td>
<td>0.0031</td>
<td>20</td>
<td>1453.30</td>
<td>0.9224</td>
<td>0</td>
<td>945.05</td>
<td>0.0106</td>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tensor Size 10 × 20 × 30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1267.10</td>
<td>0.0019</td>
<td>20</td>
<td>1291.95</td>
<td>0.0019</td>
<td>20</td>
<td>609.45</td>
<td>0.8597</td>
<td>0</td>
<td>1471.10</td>
<td>0.0019</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1015.25</td>
<td>0.0019</td>
<td>20</td>
<td>1121.00</td>
<td>0.0164</td>
<td>19</td>
<td>1220.50</td>
<td>0.9072</td>
<td>0</td>
<td>1121.40</td>
<td>0.0019</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>814.95</td>
<td>0.0019</td>
<td>20</td>
<td>888.40</td>
<td>0.0019</td>
<td>20</td>
<td>1716.30</td>
<td>0.9544</td>
<td>0</td>
<td>736.70</td>
<td>0.0020</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>719.45</td>
<td>0.0009</td>
<td>20</td>
<td>608.25</td>
<td>0.0009</td>
<td>20</td>
<td>1094.10</td>
<td>0.9271</td>
<td>0</td>
<td>508.05</td>
<td>0.0327</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>726.95</td>
<td>0.0088</td>
<td>19</td>
<td>817.20</td>
<td>0.0220</td>
<td>17</td>
<td>1635.10</td>
<td>0.9668</td>
<td>0</td>
<td>539.25</td>
<td>0.0254</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>1063.55</td>
<td>0.0270</td>
<td>16</td>
<td>1122.75</td>
<td>0.0322</td>
<td>15</td>
<td>1998.05</td>
<td>0.9798</td>
<td>0</td>
<td>649.10</td>
<td>0.0246</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table:** Numerical results for tensor robust PCA with initial guess $R = R_{CP} + 1$, averaged over 20 instances. Claim success if $Err < 0.01$
Manifold Optimization: constraint set is nonconvex

\[
\begin{align*}
\min & \quad f(x_1, \cdots, x_N) + \sum_{i=1}^{N-1} r_i(x_i) \\
\text{s.t.} & \quad \sum_{i=1}^{N} A_i x_i = b \\
& \quad x_i \in \mathcal{M}_i, \quad i = 1, \ldots, N - 1, \\
& \quad x_i \in X_i, \quad i = 1, \ldots, N - 1
\end{align*}
\]

- \( f \) differentiable, nonconvex
- \( r_i \) nonconvex, nonsmooth
- \( \mathcal{M}_i \): Riemannian manifold
- \( X_i \): closed convex set

**Monograph:** (Absil-Mahony-Sepulchre-2009): “Optimization algorithms on matrix manifolds”
Tangent space and normal cone

**Definition: Tangent Space**

Consider a Riemannian manifold $\mathcal{M}$ embedded in a Euclidean space. For any $x \in \mathcal{M}$, the tangent space $T_x \mathcal{M}$ at $x$ is a linear subspace consists of the derivatives of all smooth curves on $\mathcal{M}$ passing $x$; that is

$$T_x \mathcal{M} = \{ \gamma'(0) : \gamma(0) = x, \gamma([-\delta, \delta]) \subset \mathcal{M}, \text{ for some } \delta > 0, \gamma \text{ is smooth} \} .$$

**Definition: Normal Cone**

Suppose $S$ is a closed subset on the Riemannian manifold $\mathcal{M}$. The Riemannian normal cone is defined as

$$\mathcal{N}_S(x) := \{ u \in T_x \mathcal{M} : \langle u, v \rangle_x \leq 0, \forall v \in T_S(x) \}$$
Consider: $f$ is smooth with Lipschitz gradient and $r_i$’s are convex and locally Lipschitz continuous. If there exists a Lagrange multiplier $\lambda^*$ such that

$$
\begin{align*}
\nabla_N f(x^*) - A_N^T \lambda^* &= 0, \\
\sum_{i=1}^N A_i x_i^* - b &= 0, \\
\text{Proj}_{\tau_{x_i^*} M_i} (\nabla_i f(x^*) - A_i^T \lambda^* + \partial r_i(x_i^*)) + \mathcal{N}_{x_i \cap M_i}(x_i^*) &\ni 0, \quad i = 1, \ldots, N - 1,
\end{align*}
$$

then $x^*$ is a stationary solution.
A natural definition of $\epsilon$-stationary solution

$x^*$ is said to be an $\epsilon$-stationary solution if there exists a multiplier $\lambda^*$ such that

\[
\begin{align*}
\|\nabla_N f(x^*) - A_N^T \lambda^*\| &\leq \epsilon, \\
\|\sum_{i=1}^{N} A_i x_i^* - b\| &\leq \epsilon, \\
\text{dist}\left(\text{Proj}_{\mathcal{T}_{x_i^*} \mathcal{M}_i} (-\nabla_i f(x^*) + A_i^T \lambda^* - \partial r_i(x_i^*)) , \mathcal{N}_{x_i \cap \mathcal{M}_i}(x_i^*)\right) &\leq \epsilon, \ i = 1, ..., N - 1.
\end{align*}
\]
An ADMM Variant

Nonconvex ADMM Variant on Riemannian Manifold

Given
\[(x_1^0, x_2^0, \ldots, x_N^0) \in (M_1 \cap X_1) \times (M_2 \cap X_2) \times \cdots \times (M_{N-1} \cap X_{N-1}) \times \mathbb{R}^{nN},\]
\[\lambda^0 \in \mathbb{R}^m, \beta > 0, \gamma > 0, H_i > 0, i = 1, \ldots, N - 1.\]

for \(k = 0, 1, \ldots\) do

for \(i = 1, 2, \ldots, N - 1,\)

\[x_i^{k+1} := \text{argmin}_{x_i \in M_i \cap X_i} L_\beta(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i, x_{i+1}^{k}, \ldots, x_N^k, \lambda^k) + \frac{1}{2} \|x_i - x_i^k\|_{H_i}^2;\]

\[x_N^{k+1} := x_N^k - \gamma \nabla_N L_\beta(x_1^{k+1}, \ldots, x_{N-1}^{k+1}, x_N^k, \lambda^k);\]

\[\lambda^{k+1} := \lambda^k - \beta(\sum_{i=1}^N A_i x_i^{k+1} - b).\]

- In Step 1, \(L_\beta\) can be replaced by its linearization.
- The subproblems are easy for many applications.

Theorem: (Zhang-M-Zhang-2017)

Under certain conditions, this ADMM variant returns an \(\epsilon\)-stationary point in \(O(1/\epsilon^2)\) iterations.
Application 1: maximum bisection problem

- Given graph $G = (V, E)$. Find the best bisection, i.e., $(V_1, V_2)$ with $V_1 \cup V_2 = V$, $V_1 \cap V_2 = \emptyset$, $|V_1| = |V_2|$, that maximize the graph cut value:

$$\max_{V_1, V_2} \sum_{i \in V_1} \sum_{j \in V_2} W_{ij}$$

s.t. $V_1, V_2$ is a bisection of $V$.

- If we relax the constraint $|V_1| = |V_2|$ and only require \{\(V_1, V_2\)\} to be a partition of $V$, then it becomes the maximum cut problem.
Introduce binary assignment matrix $U \in \{0, 1\}^{n \times 2}$

Each node $i$ is represented by the $i$th row of $U$: $u_i^\top$

Note $u_i$ is a 2-dim binary vector with exactly one entry = 1

Nonconvex relaxation of max-bisection:

$$\min_U \langle W, UU^\top \rangle$$

s.t. $\|u_i\|^2 = 1, u_i \geq 0,$ for $i = 1, \ldots, n,$

$$\sum_{i=1}^n (u_i)_1 - \sum_{i=1}^n (u_i)_2 = 0.$$ 

After the relaxation is solved, round $u_i$ to an integer solution

$$u_i \leftarrow \begin{cases} (1, 0)^\top, & \text{if } (u_i)_1 \geq (u_i)_2, \\ (0, 1)^\top, & \text{otherwise}. \end{cases}$$

Then a greedy algorithm is applied to adjust current solution to a feasible bisection solution.
All subproblems have the following form:

$$\min_x b^T x$$

s.t. \( \|x\|_2 = 1, \ x \geq 0 \)

Can be solved analytically.
### Numerical Results

#### Graph Information

<table>
<thead>
<tr>
<th>Network</th>
<th>g05_60.0</th>
<th>g05_80.0</th>
<th>g05_100.0</th>
<th>pw01_100.0</th>
<th>pw09_100.0</th>
</tr>
</thead>
<tbody>
<tr>
<td># nodes</td>
<td>60</td>
<td>80</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td># edges</td>
<td>885</td>
<td>1580</td>
<td>2475</td>
<td>495</td>
<td>4455</td>
</tr>
</tbody>
</table>

**Table:** The test graphs from *Biq Mac Library*

<table>
<thead>
<tr>
<th>Network</th>
<th>our cut</th>
<th>SD</th>
<th>Ye</th>
<th>Frieze-Jerrum</th>
</tr>
</thead>
<tbody>
<tr>
<td>g05_60.0</td>
<td>1051.3</td>
<td>15.9773</td>
<td>1033.2</td>
<td>1045.4</td>
</tr>
<tr>
<td>g05_80.0</td>
<td>1822.7</td>
<td>15.3180</td>
<td>1778.5</td>
<td>1805.9</td>
</tr>
<tr>
<td>g05_100.0</td>
<td>2810.2</td>
<td>19.4413</td>
<td>2775.7</td>
<td>2799.8</td>
</tr>
<tr>
<td>pw01_100.0</td>
<td>3946.8</td>
<td>28.5032</td>
<td>3889.7</td>
<td>3944.3</td>
</tr>
<tr>
<td>pw09_100.0</td>
<td>26863.2</td>
<td>102.1318</td>
<td>26609</td>
<td>26764.1</td>
</tr>
</tbody>
</table>

**Table:** The column *SD* contains the standard deviations of our cut values in 20 rounds. All cut values are averaged over 20 runs.
ℓ_q-regularized sparse tensor PCA

Sparse tensor PCA

\[
\min_{i=1}^{N} \| T^{(i)} - C^{(i)} \times_1 U_1 \times \cdots \times_d U_d \|_F^2 + \alpha_1 \sum_{i=1}^{N} \| C^{(i)} \|_p^p + \alpha_2 \sum_{j=1}^{d} \| V_j \|_q^q + \frac{\mu}{2} \sum_{j=1}^{d} \| Y_j \|_2^2
\]

s.t. \( C^{(i)} \in \mathbb{R}^{m_1 \times \cdots \times m_d} \), \( i = 1, \ldots, N \)

\( U_j \in \mathbb{R}^{n_j \times m_j} \), \( U_j^T U_j = I \), \( j = 1, \ldots, d \)

\( V_j - U_j + Y_j = 0 \), \( j = 1, \ldots, d \).

Subproblem for \( U_j \) is projection onto \( U_j^T U_j = I \), closed-form solution.

Subproblem for \( V_j \) amounts to solve a bunch of 1-dim problems

\[
\min_{x} ax^2 + bx + c|x|^q,
\]

where \( 0 < q < 1, a > 0, c > 0 \). Easily solvable for \( q = 1/2, 2/3 \).
Numerical Results

- Choose $p = 1$ and $q = 2/3$.
- The output $U_i$ is orthogonal but not sparse.
- Zero out $U_i$’s entries with magnitude smaller than $10^{-3}$.
- 10 instances are generated and average performance is reported.

<table>
<thead>
<tr>
<th></th>
<th>30 $\times$ 30 $\times$ 30, core 5 $\times$ 5 $\times$ 5</th>
<th>42 $\times$ 42 $\times$ 42, core 7 $\times$ 7 $\times$ 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$err_1$</td>
<td>0.0043</td>
<td>0.0803</td>
</tr>
<tr>
<td>SD</td>
<td>0.0028</td>
<td>0.0010</td>
</tr>
<tr>
<td>$err_2$</td>
<td>$2.7 \times 10^{-7}$</td>
<td>$1.2 \times 10^{-14}$</td>
</tr>
</tbody>
</table>

Table: Numerical performance for sparse tensor PCA: $err_1$ is averaged obj relative errors; $SD$ is standard deviation of $err_1$; $err_2$ is averaged orthogonality constraint violation.
Community detection (clustering)

- One possible way: symmetric orthogonal nonnegative matrix factorization.
- Suppose the adjacency matrix of the network is $A$, then the method aims to solve
  $$\min_{X \in \mathbb{R}^{n \times k}} \| A - XX^\top \|_F^2,$$
  s.t., $X^\top X = I_{k \times k}$, $X \geq 0$,

- $n$: number of nodes; $k$: number of communities (clusters).
- The orthogonality and nonnegativeness of the optimal solution $X^*$ indicate that there is exactly one positive entry in each row of $X^*$.
- Reconstruct the community structure by letting node $i$ belong to community $j$ if $X^*_{ij} > 0$.
- To use our algorithm, consider the following variant:
  $$\min_{X, Y, Z \in \mathbb{R}^{n \times k}} \| A - XX^\top \|_F^2 + \frac{\mu}{2} \| Z \|_F^2$$
  s.t. $X^\top X = I_{k \times k}$, $Y \geq 0$,
  $X - Y + Z = 0$. 

Shiqian Ma (UC Davis)
Numerical Results

- Compare with two existing spectral methods: SCORE (Jin-2015) and OCCAM (Zhang-Levina-Zhu-2014)
- Three real datasets:
  - American political blogs network: 1222 nodes and 2 communities specified by their political leaning
  - Caltech facebook network: with 597 nodes and 8 communities specified by their dorm number
  - Simmons College facebook network with 1168 nodes and 4 communities specified by their graduation years

<table>
<thead>
<tr>
<th>Network Name</th>
<th>ADMM</th>
<th>SCORE</th>
<th>OCCAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polblogs</td>
<td>5.07%</td>
<td>4.75%</td>
<td>4.91%</td>
</tr>
<tr>
<td>Caltech</td>
<td>23.68%</td>
<td>28.66%</td>
<td>34.21%</td>
</tr>
<tr>
<td>Simmons</td>
<td>20.61%</td>
<td>22.54%</td>
<td>23.92%</td>
</tr>
</tbody>
</table>

Table: Numerical performance: each algorithm is run for 20 times and averaged error rate is reported.
References


Thank you for your attention!