A tale of preconditioning and overparameterization in ill-conditioned low-rank estimation

Yuejie Chi

Carnegie Mellon University

CAMDA Conference
May 2023
New imaging/sensing modalities allow us to probe the nature in unprecedented manners.

The large amount of data brings exciting opportunities that call for new tools that are scalable in computation and memory.
Low-rank matrices in data science

- radar imaging
- hyperspectral imaging
- recommendation systems
- localization
- community detection
- bioinformatics

Low-rank representations encode latent structures
A canonical problem: low-rank matrix sensing

\[ \mathbf{M} \in \mathbb{R}^{n_1 \times n_2} \]
\[ \text{rank}(\mathbf{M}) = r \]

\[ \mathbf{y} = \mathcal{A}(\mathbf{M}) + \text{noise} \]

Recover \( \mathbf{M} \) in the sample-starved regime:

\[ (n_1 + n_2)r \lesssim m \ll n_1 n_2 \]

degree of freedom
sensing budget
ambient dimension
Convex relaxation via nuclear norm minimization

\[ \min_{Z \in \mathbb{R}^{n_1 \times n_2}} \text{rank}(Z) \quad \text{s.t.} \quad y \approx A(Z) \]

Significant developments in the last decade:
Fazel '02, Recht, Parrilo, Fazel '10, Candès, Recht '09, Candès, Tao '10, Cai et al. '10, Gross '10, Negahban, Wainwright '11, Sanghavi et al. '13, Chen, Chi '14, ...
Convex relaxation via nuclear norm minimization

\[
\begin{align*}
\min_{Z \in \mathbb{R}^{n_1 \times n_2}} & \quad \text{rank}(Z) \\
\text{s.t.} & \quad y \approx A(Z)
\end{align*}
\]

\[\downarrow\text{cvx surrogate}\]

\[
\begin{align*}
\min_{Z \in \mathbb{R}^{n_1 \times n_2}} & \quad \|Z\|_* \\
\text{s.t.} & \quad y \approx A(Z)
\end{align*}
\]

where \(\| \cdot \|_*\) is the nuclear norm.
Convex relaxation via nuclear norm minimization

\[
\min_{Z \in \mathbb{R}^{n_1 \times n_2}} \text{rank}(Z) \quad \text{s.t.} \quad y \approx A(Z)
\]

\[
\min_{Z \in \mathbb{R}^{n_1 \times n_2}} \| Z \|_* \quad \text{s.t.} \quad y \approx A(Z)
\]

where \( \| \cdot \|_* \) is the nuclear norm.

Significant developments in the last decade:

Fazel ’02, Recht, Parrilo, Fazel ’10, Candès, Recht ’09, Candès, Tao ’10, Cai et al. ’10, Gross ’10,
Negahban, Wainwright ’11, Sanghavi et al. ’13, Chen, Chi ’14, ...
Convex relaxation via nuclear norm minimization

$$\min_{Z \in \mathbb{R}^{n_1 \times n_2}} \text{rank}(Z) \quad \text{s.t.} \quad y \approx A(Z)$$

$\downarrow$ cvx surrogate

$$\min_{Z \in \mathbb{R}^{n_1 \times n_2}} \|Z\|_* \quad \text{s.t.} \quad y \approx A(Z)$$

where $\| \cdot \|_*$ is the nuclear norm.

**Significant developments in the last decade:**

Fazel ’02, Recht, Parrilo, Fazel ’10, Candès, Recht ’09, Candès, Tao ’10, Cai et al. ’10, Gross ’10, Negahban, Wainwright ’11, Sanghavi et al. ’13, Chen, Chi ’14, ...

**Poor scalability:** operate in the ambient matrix space
Low-rank matrix factorization

\[
\min_{Z \in \mathbb{R}^{n_1 \times n_2}} \text{rank}(Z) \quad \text{s.t.} \quad y \approx A(Z)
\]
Low-rank matrix factorization

\[
\min_{\mathbf{Z} \in \mathbb{R}^{n_1 \times n_2}} \text{rank}(\mathbf{Z}) \quad \text{s.t.} \quad \mathbf{y} \approx \mathbf{A}(\mathbf{Z})
\]

\[
\min_{\text{rank}(\mathbf{Z}) = r} \frac{1}{2} \| \mathbf{y} - \mathbf{A}(\mathbf{Z}) \|_2^2
\]
Low-rank matrix factorization

\[
\begin{align*}
\min_{Z \in \mathbb{R}^{n_1 \times n_2}} & \quad \text{rank}(Z) \\
\text{s.t.} & \quad y \approx A(Z)
\end{align*}
\]

\[
\min_{\text{rank}(Z) = r} \frac{1}{2} \| y - A(Z) \|_2^2
\]

\[
f(X, Y) = \frac{1}{2} \left\| y - A(XY^\top) \right\|_2^2
\]
Low-rank matrix factorization

\[
\begin{align*}
\min_{Z \in \mathbb{R}^{n_1 \times n_2}} & \quad \text{rank}(Z) \\
\text{s.t.} & \quad y \approx A(Z)
\end{align*}
\]

\[
\begin{align*}
\min_{\text{rank}(Z)=r} & \quad \frac{1}{2} \|y - A(Z)\|_2^2
\end{align*}
\]

more scalable, but nonconvex!

\[
Z = \begin{bmatrix} X \\ Y^\top \end{bmatrix}
\]

\[
\begin{align*}
\min_{X \in \mathbb{R}^{n_1 \times r}, Y \in \mathbb{R}^{n_2 \times r}} & \quad f(X, Y) = \frac{1}{2} \|y - A(XY^\top)\|_2^2
\end{align*}
\]
Nonconvex problems are hard (in theory)!
Nonconvex problems are hard (in theory)!
Statistics meets optimization

Simple algorithms can be efficient for nonconvex problems!

Vanilla gradient descent (GD):

$$x_{t+1} = x_t - \eta \nabla f(x_t)$$

for $t = 0, 1, \ldots$
Simple algorithms can be efficient for nonconvex problems!
Vanilla gradient descent (GD):

\[ \mathbf{x}_{t+1} = \mathbf{x}_t - \eta \nabla f(\mathbf{x}_t) \]

for \( t = 0, 1, \ldots \)
Low-rank matrix sensing: GD with balancing regularization

\[
\min_{X,Y} \quad f(X, Y) = \frac{1}{2} \left\| y - A(XY^\top) \right\|^2_2 = \frac{1}{2} \left\| A(M - XY^\top) \right\|^2_2
\]
Low-rank matrix sensing: GD with balancing regularization

\[
\min_{X,Y} f_{\text{reg}}(X, Y) = \frac{1}{2} \left\| y - A(XY^\top) \right\|^2_2 + \frac{1}{8} \left\| X^\top X - Y^\top Y \right\|^2_F
\]
Low-rank matrix sensing: GD with balancing regularization

\[
\min_{X,Y} f_{\text{reg}}(X, Y) = \frac{1}{2} \left\| y - \mathcal{A}(XY^\top) \right\|_2^2 + \frac{1}{8} \left\| X^\top X - Y^\top Y \right\|_F^2
\]

- **Spectral initialization**: find an initial point in the “basin of attraction”.
  \[
  (X_0, Y_0) \leftarrow \text{SVD}_r(\mathcal{A}^*(y))
  \]

- **Gradient iterations**:
  \[
  X_{t+1} = X_t - \eta \nabla_X f_{\text{reg}}(X_t, Y_t)
  \\
  Y_{t+1} = Y_t - \eta \nabla_Y f_{\text{reg}}(X_t, Y_t)
  \]
  for \( t = 0, 1, \ldots \)
Prior art: GD for asymmetric low-rank matrix sensing

**Theorem (Tu et al., ICML 2016)**

Suppose $M = X_\ast Y_\ast^\top$ is rank-$r$ and has a condition number $\kappa = \sigma_{\max}(M)/\sigma_{\min}(M)$. For low-rank matrix sensing with i.i.d. Gaussian design, vanilla GD (with spectral initialization) achieves

$$\|X_t Y_t^\top - M\|_F \leq \varepsilon \cdot \sigma_{\min}(M)$$

- **Computational:** within $O(\kappa \log \frac{1}{\varepsilon})$ iterations;
- **Statistical:** as long as the sample complexity satisfies

$$m \gtrsim (n_1 + n_2)r^2\kappa^2.$$
Theorem (Tu et al., ICML 2016)

Suppose $M = X_* Y_*^\top$ is rank-$r$ and has a condition number $\kappa = \sigma_{\text{max}}(M)/\sigma_{\text{min}}(M)$. For low-rank matrix sensing with i.i.d. Gaussian design, vanilla GD (with spectral initialization) achieves

$$\|X_t Y_t^\top - M\|_F \leq \varepsilon \cdot \sigma_{\text{min}}(M)$$

- **Computational**: within $O(\kappa \log \frac{1}{\varepsilon})$ iterations;
- **Statistical**: as long as the sample complexity satisfies

$$m \gtrsim (n_1 + n_2) r^2 \kappa^2.$$

Similar results hold for many low-rank problems: matrix completion, robust PCA, etc...

(Netrapalli et al. '13, Candès, Li, Soltanolkotabi '14, Sun and Luo '15, Chen and Wainwright '15, Zheng and Lafferty '15, Ma et al. '17, ....)
Convergence slows down for ill-conditioned matrices

\[
\min_{X, Y} f(X, Y) = \frac{1}{2} \left\| \mathcal{P}_\Omega (XY^\top - M) \right\|_F^2
\]

Vanilla GD converges in \( O(\kappa \log \frac{1}{\varepsilon}) \) iterations.
Condition number can be large

chlorine concentration levels
120 junctions, 180 time slots

power-law spectrum

Data source: www.epa.gov/water-research/epanet
Condition number can be large

chlorine concentration levels
120 junctions, 180 time slots

rank-5 approximation

Data source: www.epa.gov/water-research/epanet
Condition number can be large

chlorine concentration levels
120 junctions, 180 time slots

rank-10 approximation

Data source: www.epa.gov/water-research/epanet
Condition number can be large

chlorine concentration levels
120 junctions, 180 time slots

rank-10 approximation

Must mind the condition number!

Data source: www.epa.gov/water-research/epanet
Getting rid of the condition number?

Can we accelerate the convergence rate of GD to $O(\log \frac{1}{\varepsilon})$?
Acceleration for ill-conditioned matrix estimation:

Can we design provably fast gradient algorithms that are insensitive to the condition number of low-rank matrices?
This talk: the power of preconditioning

Acceleration for ill-conditioned matrix estimation:
Can we design provably fast gradient algorithms that are insensitive to the condition number of low-rank matrices?

Robustness to adversarial outliers:
Can we design provably robust variants that are simultaneously oblivious to the presence of outliers?
This talk: the power of preconditioning

Acceleration for ill-conditioned matrix estimation:
Can we design provably fast gradient algorithms that are insensitive to the condition number of low-rank matrices?

Robustness to adversarial outliers:
Can we design provably robust variants that are simultaneously oblivious to the presence of outliers?

Generalization to tensors:
Can we generalize to higher-dimensional objects?
This talk: the power of preconditioning

**Acceleration for ill-conditioned matrix estimation:**
Can we design provably fast gradient algorithms that are insensitive to the condition number of low-rank matrices?

**Robustness to adversarial outliers:**
Can we design provably robust variants that are simultaneously oblivious to the presence of outliers?

**Generalization to tensors:**
Can we generalize to higher-dimensional objects?

**Going beyond spectral initialization and exact parameterization:**
Can we still succeed with a misspecified rank?
Accelerating gradient descent for ill-conditioned low-rank matrix estimation

Tian Tong
CMU→Amazon

Cong Ma
UChicago
Our recipe: scaled gradient descent (ScaledGD)

\[ f(X, Y) = \frac{1}{2} \left\| y - A(XY^\top) \right\|_2^2 \]

- **Spectral initialization**: find an initial point in the “basin of attraction”.

- **Scaled gradient iterations**:

  \[
  X_{t+1} = X_t - \eta \nabla_X f(X_t, Y_t) \left( Y_t^\top Y_t \right)^{-1}
  \]

  \[
  Y_{t+1} = Y_t - \eta \nabla_Y f(X_t, Y_t) \left( X_t^\top X_t \right)^{-1}
  \]

  for \( t = 0, 1, \ldots \)
Our recipe: scaled gradient descent (ScaledGD)

\[ f(X, Y) = \frac{1}{2} \| y - A(XY^\top) \|_2^2 \]

- **Spectral initialization**: find an initial point in the “basin of attraction”.

- **Scaled gradient iterations**:
  
  \[
  X_{t+1} = X_t - \eta \nabla_X f(X_t, Y_t) \left( Y_t^\top Y_t \right)^{-1} \text{preconditioner}
  \]
  
  \[
  Y_{t+1} = Y_t - \eta \nabla_Y f(X_t, Y_t) \left( X_t^\top X_t \right)^{-1} \text{preconditioner}
  \]

  for \( t = 0, 1, \ldots \)

ScaledGD is a *preconditioned* gradient method *without* balancing regularization!
ScaledGD for low-rank matrix completion

Huge computational saving: ScaledGD converges in an $\kappa$-independent manner with a minimal overhead!
Invariance to invertible transforms: (Tanner and Wei, ’16; Mishra ’16)

\[(X_t, Y_t)\]

\[M_t = X_t Y_t^T\]

\[(X_t Q, Y_t Q^{-T})\]

\[M_{t+1} = X_{t+1} Y_{t+1}^T\]

\[(X_{t+1} Q, Y_{t+1} Q^{-T})\]
A closer look at ScaledGD

**Invariance to invertible transforms:** (Tanner and Wei, '16; Mishra '16)

\[
(X_t, Y_t) 
\]

\[
M_t = X_t Y_t^\top 
\]

\[
M_{t+1} = X_{t+1} Y_{t+1}^\top 
\]

**New distance metric as Lyapunov function:**

\[
\text{dist}^2 \left( \begin{bmatrix} X \\ Y \end{bmatrix}, \begin{bmatrix} X_* \\ Y_* \end{bmatrix} \right) = \inf_{Q \in \text{GL}(r)} \left\| (XQ - X_*) \Sigma_*^{1/2} \right\|_F^2 \\
+ \left\| (YQ^{-\top} - Y_*) \Sigma_*^{1/2} \right\|_F^2
\]

+ a careful trajectory-based analysis
Theoretical guarantees of ScaledGD

**Theorem (Tong, Ma and Chi, JMLR 2021)**

For low-rank matrix sensing with i.i.d. Gaussian design, ScaledGD with spectral initialization achieves

\[ \|X_t Y_t^\top - M\|_F \lesssim \varepsilon \cdot \sigma_{\min}(M) \]

- **Computational**: within \( O\left(\log \frac{1}{\varepsilon}\right) \) iterations;
- **Statistical**: the sample complexity satisfies

\[ m \gtrsim (n_1 + n_2) r^2 \kappa^2. \]
Theoretical guarantees of ScaledGD

Theorem (Tong, Ma and Chi, JMLR 2021)

For low-rank matrix sensing with i.i.d. Gaussian design, ScaledGD with spectral initialization achieves

\[ \|X_t Y_t^\top - M\|_F \lesssim \varepsilon \cdot \sigma_{\min}(M) \]

- **Computational:** within \( O(\log \frac{1}{\varepsilon}) \) iterations;
- **Statistical:** the sample complexity satisfies

\[ m \gtrsim (n_1 + n_2)r^2\kappa^2. \]

Strict improvement over Tu et al.: ScaledGD provably accelerates vanilla GD at the same sample complexity!
ScaledGD works more broadly

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Robust PCA</th>
<th>Matrix completion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>corruption fraction</td>
<td>iteration complexity</td>
</tr>
<tr>
<td>GD</td>
<td>$\frac{1}{\mu r^{3/2} \kappa^{3/2} \vee \mu r \kappa^2}$</td>
<td>$\kappa \log \frac{1}{\varepsilon}$</td>
</tr>
<tr>
<td>ScaledGD</td>
<td>$\frac{1}{\mu r^{3/2} \kappa}$</td>
<td>$\log \frac{1}{\varepsilon}$</td>
</tr>
</tbody>
</table>

Huge computation savings at comparable sample complexities!
Robustness to outliers and corruptions?

Tian Tong
CMU→Amazon

Cong Ma
UChicago
Outlier-corrupted low-rank matrix sensing

\[ M \in \mathbb{R}^{n_1 \times n_2} \]
\[ \text{rank}(M) = r \]

\[ y = A(M) + s, \quad A(M) = \{\langle A_i, M \rangle\}_{i=1}^m \]

**Arbitrary but sparse outliers:** \[ \|s\|_0 \leq \alpha \cdot m, \text{ where } 0 \leq \alpha < 1 \text{ is fraction of outliers.} \]
Dealing with outliers: subgradient methods

Least absolute deviation (LAD):

$$\min_{X,Y} f(X,Y) = \| y - A(XY^\top) \|_1$$

- Median-truncated spectral initialization: (Li et.al.’19).
- Subgradient iterations: (Charisopoulos et.al.’19; Li et al.’18)

$$X_{t+1} = X_t - \eta_t \partial_X f(X_t, Y_t)$$
$$Y_{t+1} = Y_t - \eta_t \partial_Y f(X_t, Y_t)$$
Dealing with outliers: subgradient methods

Least absolute deviation (LAD):

$$\min_{X,Y} \ f(X,Y) = \left\| y - A(XY^\top) \right\|_1$$

- Median-truncated spectral initialization: (Li et.al.’19).
- Subgradient iterations: (Charisopoulos et.al.’19; Li et al’18)

$$X_{t+1} = X_t - \eta_t \partial_{X} f(X_t, Y_t)$$

$$Y_{t+1} = Y_t - \eta_t \partial_{Y} f(X_t, Y_t)$$

Suffer from similar slow down due to ill-conditioning.
Dealing with outliers: scaled subgradient methods

Least absolute deviation (LAD):

\[
\min_{X,Y} \quad f(X, Y) = \|y - A(XY^\top)\|_1
\]

- Median-truncated spectral initialization: (Li et.al.’19).
- Scaled subgradient iterations:

\[
\begin{align*}
X_{t+1} &= X_t - \eta_t \partial_X f(X_t, Y_t) \left( Y_t^\top Y_t \right)^{-1} \\
Y_{t+1} &= Y_t - \eta_t \partial_Y f(X_t, Y_t) \left( X_t^\top X_t \right)^{-1}
\end{align*}
\]

where \( \eta_t \) is set as Polyak’s or geometric decaying stepsize.
Performance guarantees

<table>
<thead>
<tr>
<th></th>
<th>matrix sensing</th>
<th>quadratic sensing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Subgradient Method</strong></td>
<td>$\frac{\kappa}{(1-2\alpha)^2} \log \frac{1}{\epsilon}$</td>
<td>$\frac{r\kappa}{(1-2\alpha)^2} \log \frac{1}{\epsilon}$</td>
</tr>
<tr>
<td>(Charisopoulos et al, '19)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ScaledSM</strong></td>
<td>$\frac{1}{(1-2\alpha)^2} \log \frac{1}{\epsilon}$</td>
<td>$\frac{r}{(1-2\alpha)^2} \log \frac{1}{\epsilon}$</td>
</tr>
<tr>
<td>(Tong, Ma, Chi, TSP '21)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Robustness to both ill-conditioning and adversarial corruptions!
Generalization to tensors

Tian Tong
CMU → Amazon

Harry Dong
CMU

Cong Ma
UChicago
Capturing multi-way interactions by tensors

High-order tensors capture multi-way interactions across modalities.

neural recordings

video surveillance

neuroimaging

recommendation system
Low-rank Tucker decomposition of a tensor:

\[
T(i_1, i_2, i_3) = \sum_{j_1, j_2, j_3} S(j_1, j_2, j_3) U(i_1, j_1) V(i_2, j_2) W(i_3, j_3)
\]

\[
T = (U, V, W) \cdot S,
\]

where \( U \in \mathbb{R}^{n_1 \times r_1} \), \( V \in \mathbb{R}^{n_2 \times r_2} \), \( W \in \mathbb{R}^{n_3 \times r_3} \) and \( S \in \mathbb{R}^{r_1 \times r_2 \times r_3} \).
Evidence that tensor problems are more challenging

**Low-rank tensor recovery**

Recover low-rank $T$ from $y = A(T)$. 

- Computation hardness: the nuclear norm of a tensor is NP-hard to compute (Hillar and Lim, '13);
- Computational barrier: polynomial-time algorithm exists when the sample size is above $\Omega(n^3/2)$ (Barak and Moitra, '16);
- Little existing results for the Tucker case: no provably efficient first-order algorithm for low-rank tensor completion (Han, Zhang, Willett, '20).
Evidence that tensor problems are more challenging

Low-rank tensor recovery

Recover low-rank $T$ from $y = \mathcal{A}(T)$.

- **Computation hardness:** the nuclear norm of a tensor is NP-hard to compute \cite{hillar2013nuclear};
Evidence that tensor problems are more challenging

Low-rank tensor recovery

\[ \text{Recover low-rank } T \text{ from } y = A(T). \]

- **Computation hardness:** the nuclear norm of a tensor is NP-hard to compute \((\text{Hillar and Lim, '13);}\)

- **Computational barrier:** polynomial-time algorithm exists when the sample size is above \(\Omega(n^{3/2})\) \((\text{Barak and Moitra, '16);}\)
Evidence that tensor problems are more challenging

Low-rank tensor recovery

Recover low-rank \( T \) from \( y = A(T) \).

- **Computation hardness**: the nuclear norm of a tensor is NP-hard to compute (Hillar and Lim, '13);

- **Computational barrier**: polynomial-time algorithm exists when the sample size is above \( \Omega(n^{3/2}) \) (Barak and Moitra, '16);

- **Little existing results for the Tucker case**: no provably efficient first-order algorithm for low-rank tensor completion (Han, Zhang, Willett, '20).
How to construct scaled gradients for tensors?

\[
\min_{F=(U,V,W,S)} f(F) = \frac{1}{2} \| A((U, V, W) \cdot S) - y \|_2^2
\]
How to construct scaled gradients for tensors?

\[
\begin{align*}
\min_{F=(U,V,W,S)} & \quad f(F) = \frac{1}{2} \| A((U, V, W) \cdot S) - y \|_2^2 \\
\end{align*}
\]

**Step 1:** unfolding the tensor along mode-1:

\[
\mathcal{M}_1(T) = U
\]
How to construct scaled gradients for tensors?

\[
\min_{F=(U,V,W,S)} f(F) = \frac{1}{2} \| A((U, V, W) \cdot S) - y \|_2^2
\]

**Step 1:** unfolding the tensor along mode-1:

\[
\mathcal{M}_1(T) = U
\]

**Step 2:** Treat this as a matrix problem for updating factor \( U \):

\[
U_{t+1} = U_t - \eta \nabla_U f(F_t) (\tilde{U}_t \tilde{U}_t)^{-1}
\]
How to construct scaled gradients for tensors?

\[
\min_{F=(U, V, W, S)} f(F) = \frac{1}{2} \| A((U, V, W) \cdot S) - y \|_2^2
\]

**Step 1:** unfolding the tensor along mode-1:

\[
\mathcal{M}_1(T) = U
\]

**Step 2:** Treat this as a matrix problem for updating factor \( U \):

\[
U_{t+1} = U_t - \eta \nabla_U f(F_t) (\bar{U}_t^\top \bar{U}_t)^{-1}
\]

**Step 3:** update the core tensor \( S \):

\[
S_{t+1} = S_t - \eta \left( (U_t^\top U_t)^{-1}, (V_t^\top V_t)^{-1}, (W_t^\top W_t)^{-1} \right) \cdot \nabla_S f(F_t)
\]
ScaledGD for ill-conditioned low-rank tensor estimation

\[
\min_{F=(U, V, W, S)} f(F) = \frac{1}{2} \| \mathcal{A}((U, V, W) \cdot S) - y \|_2^2
\]

Scaled gradient iterations:

\[
\begin{align*}
U_{t+1} &= U_t - \eta \nabla_U f(F_t) (\tilde{U}_t^\top \tilde{U}_t)^{-1}, \\
V_{t+1} &= V_t - \eta \nabla_V f(F_t) (\tilde{V}_t^\top \tilde{V}_t)^{-1}, \\
W_{t+1} &= W_t - \eta \nabla_W f(F_t) (\tilde{W}_t^\top \tilde{W}_t)^{-1}, \\
S_{t+1} &= S_t - \eta \left( (U_t^\top U_t)^{-1}, (V_t^\top V_t)^{-1}, (W_t^\top W_t)^{-1} \right) \cdot \nabla_S f(F_t),
\end{align*}
\]

where \( \tilde{U}_t := (V_t \otimes W_t) \mathcal{M}_1(S_t)^\top \), \( \tilde{V}_t := (U_t \otimes W_t) \mathcal{M}_2(S_t)^\top \), and \( \tilde{W}_t := (U_t \otimes V_t) \mathcal{M}_3(S_t)^\top \). Here, \( \mathcal{M}_k(S) \) is the matricization of \( S \) along the \( k \)-th mode.

**Key property:** invariance to parameterization.
Theorem (Tong et. al., JMLR 2022)

For low-rank tensor completion under Bernoulli sampling, assume $n = n_1 = n_2 = n_3$, ScaledGD with spectral initialization and projection achieves

$$\| (U_t, V_t, W_t) \cdot S_t - T \|_F \lesssim \varepsilon \cdot \sigma_{\min}(T)$$

- **Computational:** within $O(\log \frac{1}{\varepsilon})$ iterations;
- **Statistical:** as long as the sample complexity satisfies

$$n^3 p \gtrsim \mu^{3/2} r^{5/2} n^{3/2} \kappa^3 \log n.$$
ScaledGD for low-rank tensor completion

**Theorem (Tong et. al., JMLR 2022)**

For low-rank tensor completion under Bernoulli sampling, assume \( n = n_1 = n_2 = n_3 \), ScaledGD with spectral initialization and projection achieves

\[
\| (U_t, V_t, W_t) \cdot S_t - T \|_F \lesssim \varepsilon \cdot \sigma_{\min}(T)
\]

- **Computational:** within \( O(\log \frac{1}{\varepsilon}) \) iterations;
- **Statistical:** as long as the sample complexity satisfies

\[
n^3 p \gtrsim \mu^{3/2} \kappa^{5/2} \mu^{3/2} \kappa^3 \log n.
\]

First provable linear convergence at a near-optimal sample complexity for low-Tucker-rank tensor completion!
Numerical evidence

\[
\min_{F=(U, V, W, S)} f(F) = \frac{1}{2} \| \mathcal{P}_\Omega((U, V, W) \cdot S) - T \|_F^2
\]

The benefit of ScaledGD is even more evident for tensors!
The benefit of ScaledGD is even more evident for tensors!
Tensor robust principal component analysis

Data = Sparse + Low-rank

Theorem (Dong, Tong, Ma, Chi, 2022)

For a low-rank plus sparse tensor, ScaledGD with spectral initialization and iteration-varying thresholding converges at a constant rate, as long as the corruption level per fiber satisfies

$$\alpha \lesssim \frac{1}{\mu^2 r^3 \kappa}.$$
Tensor robust principal component analysis

\[
\text{Data} = \text{Sparse} + \text{Low-rank}
\]

**Theorem (Dong, Tong, Ma, Chi, 2022)**

For a low-rank plus sparse tensor, ScaledGD with spectral initialization and iteration-varying thresholding converges at a constant rate, as long as the corruption level per fiber satisfies

\[
\alpha \lesssim \frac{1}{\mu^2 r^3 \kappa}.
\]

*Can use selective mode updates to accelerate computation!*
Hyperparameter tuning via self-supervised learning

unfolding + self-supervised learning

\[ r \rightarrow \xi_0 \rightarrow X_0 \rightarrow \xi_1 \rightarrow X_1 \rightarrow \xi_2 \rightarrow X_2 \rightarrow \cdots \rightarrow \xi_{T-1} \rightarrow X_{T-1} \rightarrow \eta \rightarrow \rho \rightarrow X_T \]

Hyperparameter tuning via self-supervised learning

unfolding + self-supervised learning

some materials data
Hyperparameter tuning via self-supervised learning

unfolding + self-supervised learning

low-rank + sparse decomposition

some materials data

Overparameterizing (Misspecified) ScaledGD?

Xingyu Xu
CMU

Yandi Shen
UChicago

Cong Ma
UChicago
What if we do not know the exact rank?

So far we have assumed the exact rank is given.... what if we do not know the exact rank?
What if we do not know the exact rank?

So far we have assumed the exact rank is given.... what if we do not know the exact rank?

Misspecification by overparameterization:

\[ M = XX^\top, \quad X \in \mathbb{R}^{n \times r'}, \quad r' > r \]
What if we do not know the exact rank?

So far we have assumed the exact rank is given.... what if we do not know the exact rank?

Misspecification by overparameterization:

\[ M = XX^\top, \quad X \in \mathbb{R}^{n \times r'}, \quad r' > r \]

ScaledGD:

\[ X_{t+1} = X_t - \eta \nabla_X f(X_t) \left( X_t^\top X_t \right)^{-1} \]

_\text{preconditioner}

\textit{analysis break down and might be unstable}...
What if we do not know the exact rank?

So far we have assumed the exact rank is given…. what if we do not know the exact rank?

**Misspecification by overparameterization:**

\[ M = XX^\top, \quad X \in \mathbb{R}^{n \times r'}, \quad r' > r \]

**ScaledGD(\(\lambda\)):**

\[ X_{t+1} = X_t - \eta \nabla_X f(X_t) \left( X_t^\top X_t + \lambda I \right)^{-1} \]

*add regularization to stabilize the preconditioner*
Does preconditioning hurt generalization?

- Infinitely many global minima, not all generalize
- Can we still guarantee generalization?

**When Does Preconditioning Help or Hurt Generalization?**

*Shun-ichi Amari\(^1\), Jimmy Ba\(^2,3\), Roger Grosse\(^2,3\), Xuechen Li\(^4\), Atsushi Nitanda\(^5,6\), Taiji Suzuki\(^5,6\), Denny Wu\(^2,3\), Ji Xu\(^7\)*

\(^1\)RIKEN CBS,  \(^2\)University of Toronto,  \(^3\)Vector Institute, \(^4\)Google Research, Brain Team, \(^5\)University of Tokyo,  \(^6\)RIKEN AIP,  \(^7\)Columbia University

amari@brain.riken.jp, \{jba, rgrosse, lxuechen, dennywu\}@cs.toronto.edu, \{nitanda, taiji\}@mist.i.u-tokyo.ac.jp, jixu@cs.columbia.edu
Theoretical guarantees

**Theorem (Xu, Shen, Ma, Chi, ICML 2023)**

*For low-rank matrix sensing with i.i.d. Gaussian design, overparameterized ScaledGD(λ) with \( \lambda \asymp \sigma_{\min}(M) \), \( \eta \asymp 1 \), and \( X_0 \sim \alpha \mathcal{N}(0, 1/n) \) with sufficiently small \( \alpha \) achieves*

\[
\|X_t Y_t^\top - M\|_F \lesssim \varepsilon \cdot \sigma_{\min}(M)
\]

- **Computational:** *within \( O(\log \kappa \log(\kappa n) + \log \frac{1}{\varepsilon}) \) iterations;*
- **Statistical:** *the sample complexity satisfies*

\[
m \gtrsim nr^2 \text{poly}(\kappa).
\]

- *Our analysis also enables exact convergence under random initialization with correct rank specification.*
Comparison with overparameterized GD

ScaledGD picks up the signal component much faster than GD even from small random initialization!
Comparison with overparameterized GD

ScaledGD picks up the signal component much faster than GD even from small random initialization!

(Stöger and Soltanolkotabi, '21)
Comparison with overparameterized GD

ScaledGD picks up the signal component much faster than GD even from small random initialization!
Concluding remarks
Bridging the theory-practice gap

Nonconvex low-rank matrix and tensor estimation:

• identification and exploitation of benign geometric properties;
• analyzing iterate trajectories beyond black-box optimization;
• simple variants of GD lead to robust and accelerated convergence.

Computational: near dimension-free iteration complexity

Statistical: near-optimal sample complexity

Robustness: adversarial outliers ill-conditioning
Preconditioning helps!

Preconditioning dramatically increases the efficiency of vanilla gradient methods even for challenging nonconvex problems!

**Ongoing directions:**

- asymmetric ScaledGD with overparameterization.
- Generalizing the idea of ScaledGD to other learning and estimation problems.
Selected References

Overview:


ScaledGD for low-rank matrix estimation:


ScaledGD for low-rank tensor estimation:


• Fast and provable tensor robust principal component analysis via scaled gradient descent, *Information and Inference*, accepted.
Thanks!

https://users.ece.cmu.edu/~yuejiec/