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

Yuejie Chi

Carnegie Mellon University

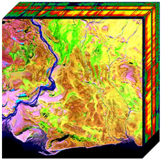
CAMDA Conference
May 2023

Sensing, computing, and imaging advances

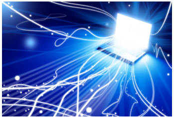
New imaging/sensing modalities allow us to probe the nature in unprecedented manners.

healthcare

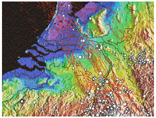
Radio astronomy



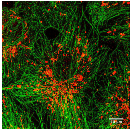
hyperspectral



Internet traffic



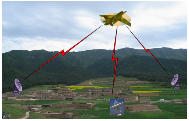
seismic imaging



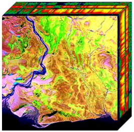
microscopy

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

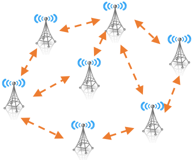


radar imaging

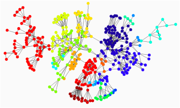


hyperspectral imaging

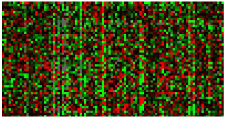
recommendation systems



localization



community detection



bioinformatics

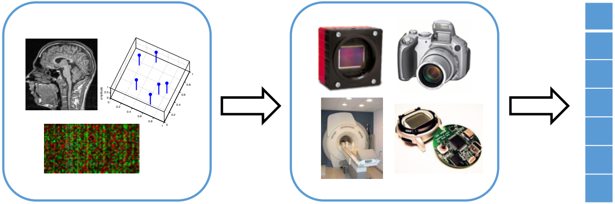
Low-rank representations encode latent structures

A canonical problem: low-rank matrix sensing

$\boldsymbol{M} \in \mathbb{R}^{n_1 \times n_2}$
$\mathrm{rank}(\boldsymbol{M}) = r$

$\mathcal{A}(\cdot)$
linear map

$\boldsymbol{y} \in \mathbb{R}^m$



$$\boldsymbol{y} = \mathcal{A}(\boldsymbol{M}) + \text{noise}$$

Recover $\boldsymbol{M}$ in the sample-starved regime:

$$\underbrace{(n_1 + n_2)r}_{\text{degree of freedom}} \lesssim \underbrace{m}_{\text{sensing budget}} \ll \underbrace{n_1 n_2}_{\text{ambient dimension}}$$

Convex relaxation via nuclear norm minimization

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

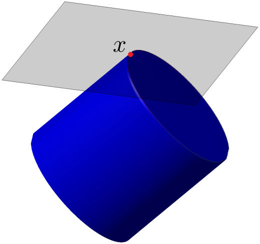
Convex relaxation via nuclear norm minimization

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

$\Downarrow$ cvx surrogate

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

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



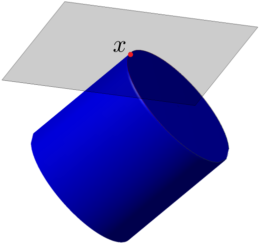
Convex relaxation via nuclear norm minimization

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

$\Downarrow$ cvx surrogate

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

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, ...

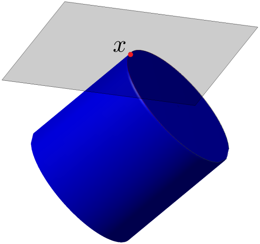
Convex relaxation via nuclear norm minimization

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

$$\Big\Downarrow \text{ cvx surrogate}$$

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

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

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

Low-rank matrix factorization

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

$$\Downarrow$$

$$\min_{\operatorname{rank}(\boldsymbol{Z})=r} \quad \frac{1}{2}\left\|\boldsymbol{y} - \mathcal{A}(\boldsymbol{Z})\right\|_2^2$$

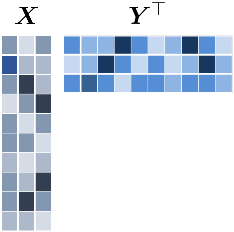
Low-rank matrix factorization

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

$$\Downarrow$$

$$\min_{\color{blue}{\mathrm{rank}(\boldsymbol{Z})=r}} \quad \frac{1}{2}\left\|\boldsymbol{y}-\mathcal{A}(\boldsymbol{Z})\right\|_2^2$$

$$\Downarrow \qquad \boldsymbol{Z} =$$



$$\min_{\boldsymbol{X}\in\mathbb{R}^{n_1\times r},\boldsymbol{Y}\in\mathbb{R}^{n_2\times r}} \quad f(\boldsymbol{X},\boldsymbol{Y})=\frac{1}{2}\left\|\boldsymbol{y}-\mathcal{A}(\boldsymbol{X}\boldsymbol{Y}^\top)\right\|_2^2$$

Low-rank matrix factorization

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

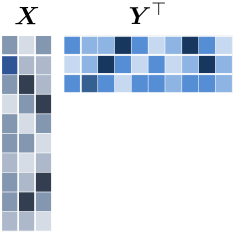
$$\Downarrow$$

$$\min_{\text{rank}(\boldsymbol{Z})=r} \quad \frac{1}{2}\left\|\boldsymbol{y}-\mathcal{A}(\boldsymbol{Z})\right\|_2^2$$

more scalable,
but nonconvex!

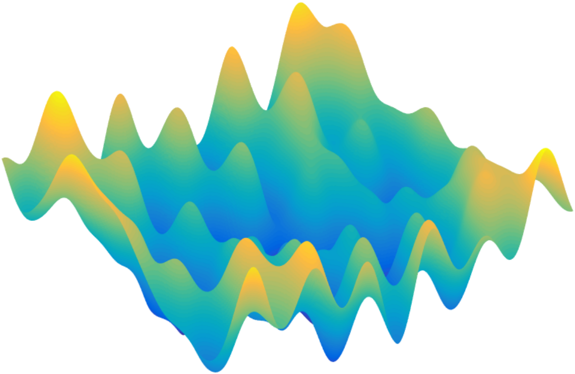
$$\Downarrow$$

$\boldsymbol{Z} =$



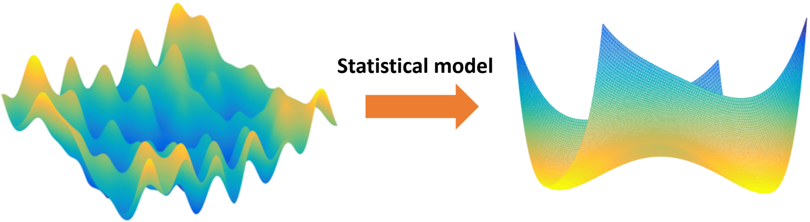
$$\min_{\boldsymbol{X}\in\mathbb{R}^{n_1\times r},\boldsymbol{Y}\in\mathbb{R}^{n_2\times r}} \quad f(\boldsymbol{X},\boldsymbol{Y})=\frac{1}{2}\left\|\boldsymbol{y}-\mathcal{A}(\boldsymbol{X}\boldsymbol{Y}^\top)\right\|_2^2$$

Nonconvex problems are hard (in theory)!



Nonconvex problems are hard (in theory)!

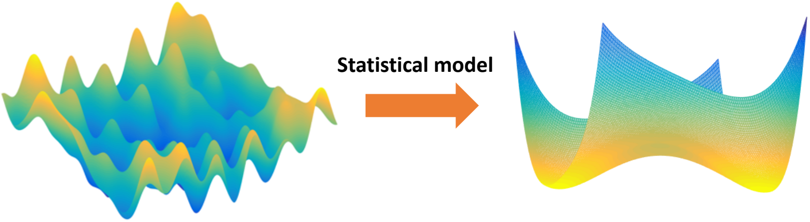
Statistics meets optimization



worst case

average case

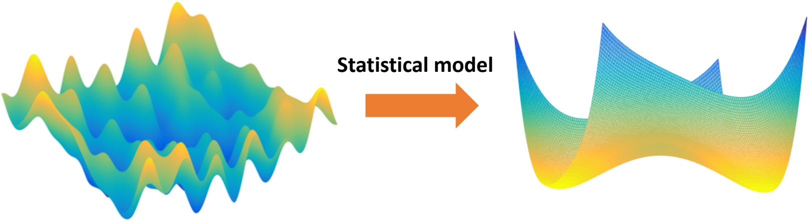
Statistics meets optimization



worst case average case

Simple algorithms can be efficient for nonconvex problems!

Statistics meets optimization



worst case average case

Simple algorithms can be efficient for nonconvex problems!

Vanilla gradient descent (GD):

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta \nabla f(\boldsymbol{x}_t)$$

for $t = 0, 1, \ldots$

Low-rank matrix sensing: GD with balancing regularization

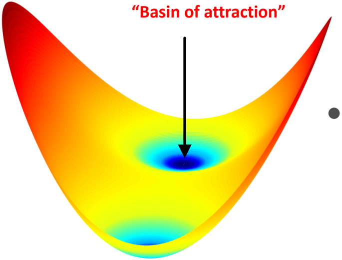
$$\min_{\boldsymbol{X},\boldsymbol{Y}} \quad f(\boldsymbol{X},\boldsymbol{Y}) = \frac{1}{2}\left\|\boldsymbol{y} - \mathcal{A}(\boldsymbol{X}\boldsymbol{Y}^\top)\right\|_2^2 = \frac{1}{2}\left\|\mathcal{A}(\boldsymbol{M} - \boldsymbol{X}\boldsymbol{Y}^\top)\right\|_2^2$$

Low-rank matrix sensing: GD with balancing regularization

$$\min_{\boldsymbol{X},\boldsymbol{Y}} f_{\mathrm{reg}}(\boldsymbol{X},\boldsymbol{Y}) = \frac{1}{2}\left\|\boldsymbol{y} - \mathcal{A}(\boldsymbol{X}\boldsymbol{Y}^\top)\right\|_2^2 + \frac{1}{8}\left\|\boldsymbol{X}^\top\boldsymbol{X} - \boldsymbol{Y}^\top\boldsymbol{Y}\right\|_{\mathrm{F}}^2$$

Low-rank matrix sensing: GD with balancing regularization

$$\min_{\boldsymbol{X},\boldsymbol{Y}} f_{\mathrm{reg}}(\boldsymbol{X},\boldsymbol{Y}) = \frac{1}{2}\left\|\boldsymbol{y} - \mathcal{A}(\boldsymbol{X}\boldsymbol{Y}^\top)\right\|_2^2 + \frac{1}{8}\left\|\boldsymbol{X}^\top\boldsymbol{X} - \boldsymbol{Y}^\top\boldsymbol{Y}\right\|_{\mathrm{F}}^2$$



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

$$(\boldsymbol{X}_0, \boldsymbol{Y}_0) \leftarrow \mathsf{SVD}_r(\mathcal{A}^*(\boldsymbol{y}))$$

- **Gradient iterations:**

$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t - \eta\,\nabla_{\boldsymbol{X}} f_{\mathrm{reg}}(\boldsymbol{X}_t, \boldsymbol{Y}_t)$$
$$\boldsymbol{Y}_{t+1} = \boldsymbol{Y}_t - \eta\,\nabla_{\boldsymbol{Y}} f_{\mathrm{reg}}(\boldsymbol{X}_t, \boldsymbol{Y}_t)$$

for $t = 0, 1, \ldots$

Prior art: GD for asymmetric low-rank matrix sensing

Theorem (Tu et al., ICML 2016)

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

$$\|\boldsymbol{X}_t \boldsymbol{Y}_t^\top - \boldsymbol{M}\|_{\mathrm{F}} \leq \varepsilon \cdot \sigma_{\min}(\boldsymbol{M})$$

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

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

Prior art: GD for asymmetric low-rank matrix sensing

Theorem (Tu et al., ICML 2016)

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

$$\|\boldsymbol{X}_t \boldsymbol{Y}_t^\top - \boldsymbol{M}\|_{\mathrm{F}} \leq \varepsilon \cdot \sigma_{\min}(\boldsymbol{M})$$

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

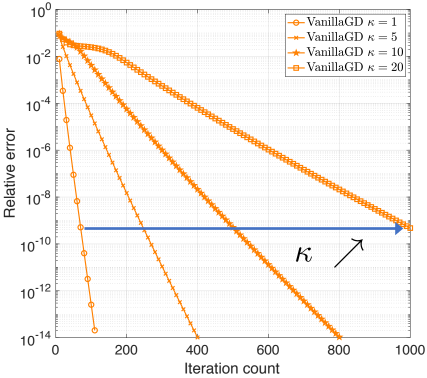
$$m \gtrsim (n_1 + n_2) r^2 \kappa^2.$$

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, ...)

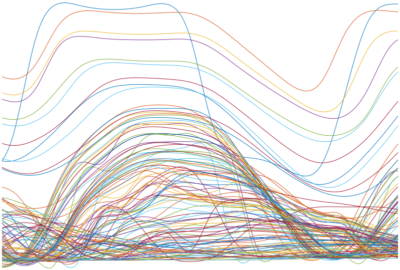
Convergence slows down for ill-conditioned matrices

$$\min_{\boldsymbol{X},\boldsymbol{Y}} \quad f(\boldsymbol{X},\boldsymbol{Y}) = \frac{1}{2}\left\|\mathcal{P}_\Omega(\boldsymbol{X}\boldsymbol{Y}^\top - \boldsymbol{M})\right\|_{\mathrm{F}}^2$$

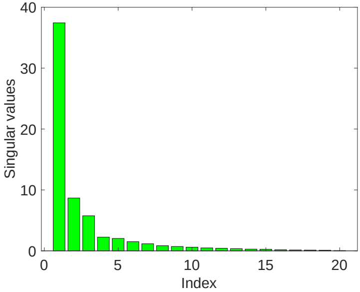


Vanilla GD converges in $O\left(\kappa \log \frac{1}{\varepsilon}\right)$ iterations.

Condition number can be large



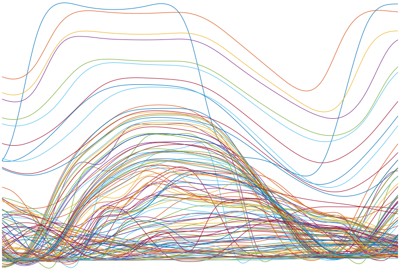
chlorine concentration levels
120 junctions, 180 time slots



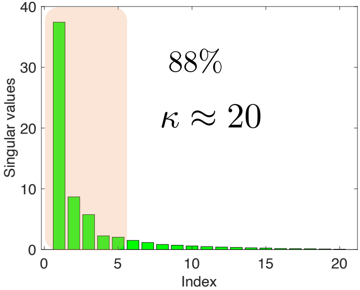
power-law spectrum

Data source: `www.epa.gov/water-research/epanet`

Condition number can be large



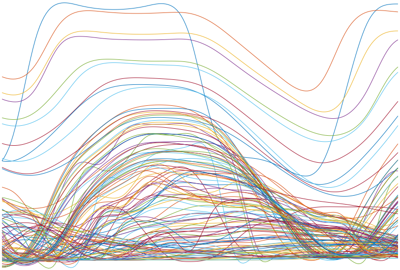
chlorine concentration levels
120 junctions, 180 time slots



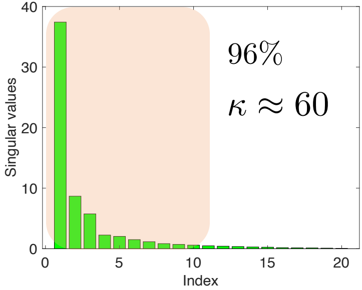
rank-5 approximation

Data source: `www.epa.gov/water-research/epanet`

Condition number can be large



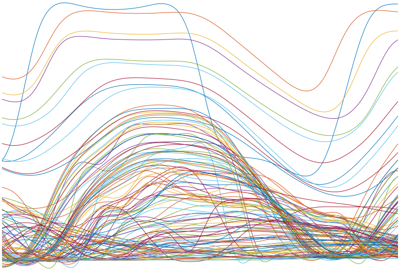
chlorine concentration levels
120 junctions, 180 time slots



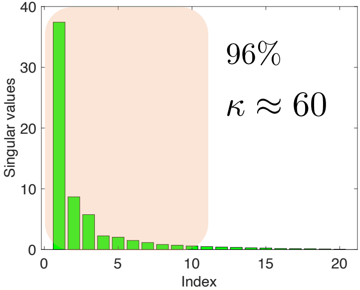
rank-10 approximation

Data source: `www.epa.gov/water-research/epanet`

Condition number can be large



chlorine concentration levels
120 junctions, 180 time slots

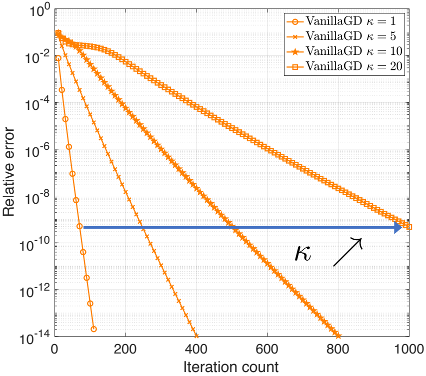


rank-10 approximation

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 $O(\log \frac{1}{\varepsilon})$*?*

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?

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?

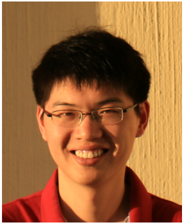
Generalization to tensors:

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

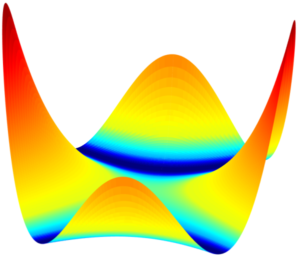


Tian Tong
CMU→Amazon

Cong Ma
UChicago

Our recipe: scaled gradient descent (ScaledGD)

$$f(\boldsymbol{X},\boldsymbol{Y}) = \frac{1}{2}\left\|\boldsymbol{y} - \mathcal{A}(\boldsymbol{X}\boldsymbol{Y}^\top)\right\|_2^2$$



- **Spectral initialization:** find an initial point in the "basin of attraction".
- **Scaled gradient iterations:**

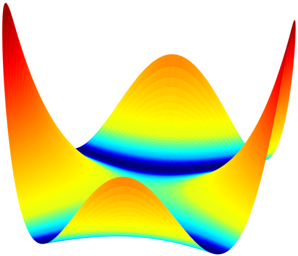
$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t - \eta \nabla_{\boldsymbol{X}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t) \underbrace{(\boldsymbol{Y}_t^\top \boldsymbol{Y}_t)^{-1}}_{\texttt{preconditioner}}$$

$$\boldsymbol{Y}_{t+1} = \boldsymbol{Y}_t - \eta \nabla_{\boldsymbol{Y}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t) \underbrace{(\boldsymbol{X}_t^\top \boldsymbol{X}_t)^{-1}}_{\texttt{preconditioner}}$$

for $t = 0, 1, \ldots$

Our recipe: scaled gradient descent (ScaledGD)

$$f(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{2} \left\| \boldsymbol{y} - \mathcal{A}(\boldsymbol{X}\boldsymbol{Y}^\top) \right\|_2^2$$



- **Spectral initialization:** find an initial point in the "basin of attraction".
- **Scaled gradient iterations:**

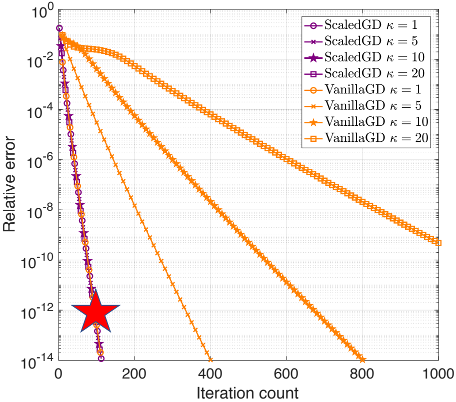
$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t - \eta \nabla_{\boldsymbol{X}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t) \underbrace{(\boldsymbol{Y}_t^\top \boldsymbol{Y}_t)^{-1}}_{\texttt{preconditioner}}$$

$$\boldsymbol{Y}_{t+1} = \boldsymbol{Y}_t - \eta \nabla_{\boldsymbol{Y}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t) \underbrace{(\boldsymbol{X}_t^\top \boldsymbol{X}_t)^{-1}}_{\texttt{preconditioner}}$$

for $t = 0, 1, \ldots$

ScaledGD is a *preconditioned* gradient method *without* balancing regularization!

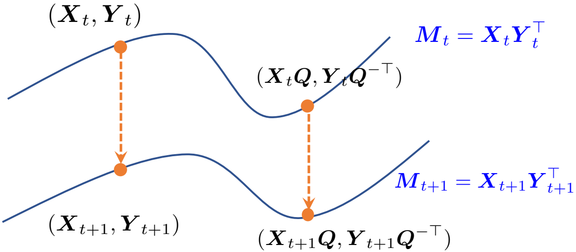
ScaledGD for low-rank matrix completion



Huge computational saving: ScaledGD converges in an κ-independent manner with a minimal overhead!

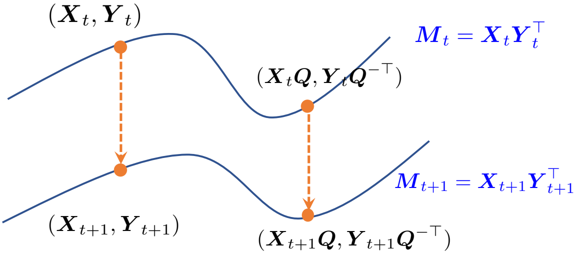
A closer look at ScaledGD

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



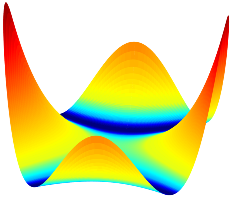
A closer look at ScaledGD

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



New distance metric as Lyapunov function:

$$\mathrm{dist}^2\left(\begin{bmatrix}\boldsymbol{X}\\ \boldsymbol{Y}\end{bmatrix}, \begin{bmatrix}\boldsymbol{X}_\star\\ \boldsymbol{Y}_\star\end{bmatrix}\right) = \inf_{\boldsymbol{Q}\in \mathrm{GL}(r)} \left\|(\boldsymbol{X}\boldsymbol{Q} - \boldsymbol{X}_\star)\boldsymbol{\Sigma}_\star^{1/2}\right\|_{\mathrm{F}}^2 + \left\|(\boldsymbol{Y}\boldsymbol{Q}^{-\top} - \boldsymbol{Y}_\star)\boldsymbol{\Sigma}_\star^{1/2}\right\|_{\mathrm{F}}^2$$



$+$ 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

$$\|\boldsymbol{X}_t \boldsymbol{Y}_t^\top - \boldsymbol{M}\|_{\mathrm{F}} \lesssim \varepsilon \cdot \sigma_{\min}(\boldsymbol{M})$$

- **Computational:** *within* $O\big(\log \frac{1}{\varepsilon}\big)$ *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

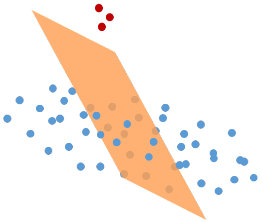
$$\|\boldsymbol{X}_t \boldsymbol{Y}_t^\top - \boldsymbol{M}\|_{\mathrm{F}} \lesssim \varepsilon \cdot \sigma_{\min}(\boldsymbol{M})$$

- **Computational:** *within* $O\big(\log \frac{1}{\varepsilon}\big)$ *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



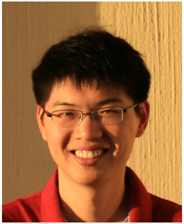
$$\begin{bmatrix} \checkmark & ? & ? & ? & \checkmark \\ ? & ? & \checkmark & \checkmark & ? \\ \checkmark & ? & ? & \checkmark & ? \\ ? & ? & \checkmark & ? & ? \\ \checkmark & ? & ? & ? & ? \\ ? & \checkmark & ? & ? & \checkmark \end{bmatrix}$$

	Robust PCA		**Matrix completion**	
Algorithms	corruption fraction	iteration complexity	sample complexity	iteration complexity
GD	$\frac{1}{\mu r^{3/2}\kappa^{3/2} \vee \mu r \kappa^2}$	$\kappa \log \frac{1}{\varepsilon}$	$(\mu \vee \log n)\mu n r^2 \kappa^2$	$\kappa \log \frac{1}{\varepsilon}$
ScaledGD	$\frac{1}{\mu r^{3/2}\kappa}$	$\log \frac{1}{\varepsilon}$	$(\mu\kappa^2 \vee \log n)\mu n r^2 \kappa^2$	$\log \frac{1}{\varepsilon}$

Huge computation savings at comparable sample complexities!

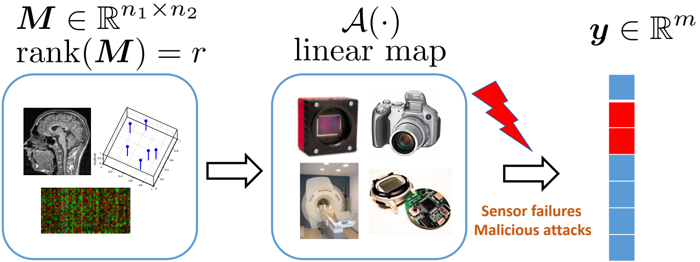
Robustness to outliers and corruptions?

Tian Tong
CMU→Amazon



Cong Ma
UChicago

Outlier-corrupted low-rank matrix sensing



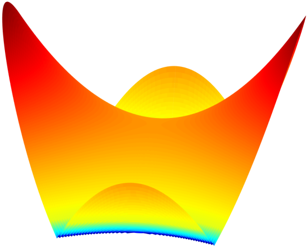
$$\boldsymbol{y} \;=\; \mathcal{A}(\boldsymbol{M}) + \underbrace{\boldsymbol{s}}_{\text{outliers}}, \qquad \mathcal{A}(\boldsymbol{M}) = \{\langle \boldsymbol{A}_i, \boldsymbol{M} \rangle\}_{i=1}^{m}$$

Arbitrary but sparse outliers: $\|\boldsymbol{s}\|_0 \le \alpha \cdot m$, where $0 \le \alpha < 1$ is fraction of outliers.

Dealing with outliers: subgradient methods

Least absolute deviation (LAD):

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



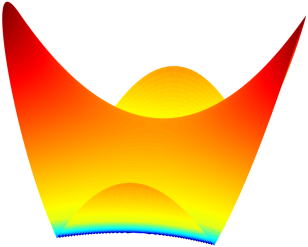
- **Median-truncated spectral initialization:** (Li et.al.'19).
- **Subgradient iterations:** (Charisopoulos et.al.'19; Li et al'18)

$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t - \eta_t \, \partial_{\boldsymbol{X}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t)$$
$$\boldsymbol{Y}_{t+1} = \boldsymbol{Y}_t - \eta_t \, \partial_{\boldsymbol{Y}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t)$$

Dealing with outliers: subgradient methods

Least absolute deviation (LAD):

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



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

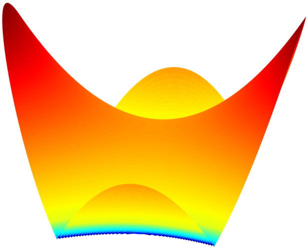
$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t - \eta_t \, \partial_{\boldsymbol{X}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t)$$
$$\boldsymbol{Y}_{t+1} = \boldsymbol{Y}_t - \eta_t \, \partial_{\boldsymbol{Y}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t)$$

Suffer from similar slow down due to ill-conditioning.

Dealing with outliers: scaled subgradient methods

Least absolute deviation (LAD):

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



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

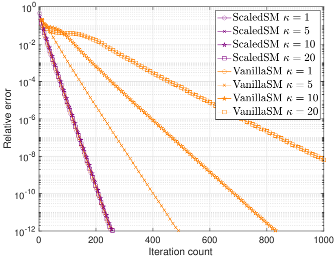
$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t - \eta_t \, \partial_{\boldsymbol{X}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t) \ \underbrace{(\boldsymbol{Y}_t^\top \boldsymbol{Y}_t)^{-1}}_{\texttt{preconditioner}}$$

$$\boldsymbol{Y}_{t+1} = \boldsymbol{Y}_t - \eta_t \, \partial_{\boldsymbol{Y}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t) \ \underbrace{(\boldsymbol{X}_t^\top \boldsymbol{X}_t)^{-1}}_{\texttt{preconditioner}}$$

where η_t is set as Polyak's or geometric decaying stepsize.

Performance guarantees

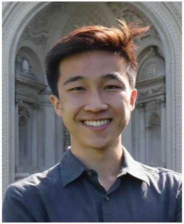
	matrix sensing	quadratic sensing
Subgradient Method (Charisopoulos et al, '19)	$\frac{\kappa}{(1-2\alpha)^2} \log \frac{1}{\epsilon}$	$\frac{r\kappa}{(1-2\alpha)^2} \log \frac{1}{\epsilon}$
ScaledSM (Tong, Ma, Chi, TSP '21)	$\frac{1}{(1-2\alpha)^2} \log \frac{1}{\epsilon}$	$\frac{r}{(1-2\alpha)^2} \log \frac{1}{\epsilon}$



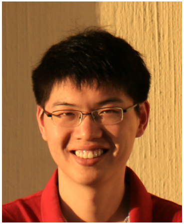
Robustness to both ill-conditioning and adversarial corruptions!

Generalization to tensors

Tian Tong
CMU→Amazon

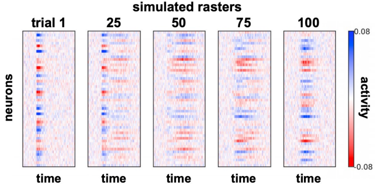


Harry Dong
CMU



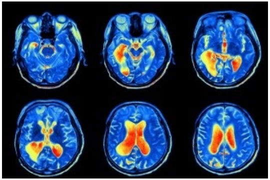
Cong Ma
UChicago

Capturing multi-way interactions by tensors

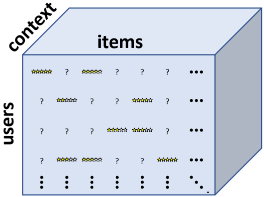


neural recordings

video surveillance



neuroimaging



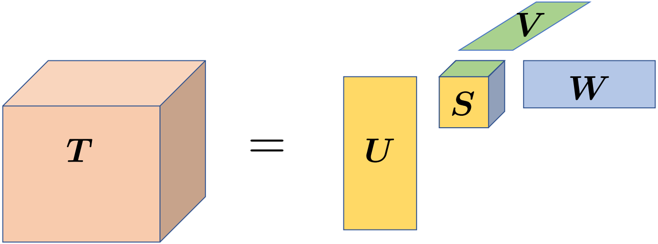
recommendation system

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

Low-rank tensor under Tucker decomposition

Low-rank Tucker decomposition of a tensor:

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



$$\boldsymbol{T} = (\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}) \cdot \boldsymbol{S},$$

where $\boldsymbol{U} \in \mathbb{R}^{n_1 \times r_1}$, $\boldsymbol{V} \in \mathbb{R}^{n_2 \times r_2}$, $\boldsymbol{W} \in \mathbb{R}^{n_3 \times r_3}$ and $\boldsymbol{S} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$.

Evidence that tensor problems are more challenging

Low-rank tensor recovery

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

Evidence that tensor problems are more challenging

Low-rank tensor recovery

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

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

Evidence that tensor problems are more challenging

Low-rank tensor recovery

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

- **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);

Evidence that tensor problems are more challenging

Low-rank tensor recovery

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

- **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?

$$\min_{\boldsymbol{F}=(\boldsymbol{U},\boldsymbol{V},\boldsymbol{W},\boldsymbol{S})} \; f(\boldsymbol{F}) = \frac{1}{2}\left\|\mathcal{A}((\boldsymbol{U},\boldsymbol{V},\boldsymbol{W})\cdot\boldsymbol{S}) - \boldsymbol{y}\right\|_2^2$$

How to construct scaled gradients for tensors?

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

Step 1: unfolding the tensor along mode-1:

$$\mathcal{M}_1(\boldsymbol{T}) = \boldsymbol{U}\ \underbrace{\mathcal{M}_1(\boldsymbol{S})(\boldsymbol{V}\otimes\boldsymbol{W})^\top}_{\breve{\boldsymbol{U}}^\top}$$

How to construct scaled gradients for tensors?

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

Step 1: unfolding the tensor along mode-1:

$$\boxed{\mathcal{M}_1(\boldsymbol{T})} = \boxed{\boldsymbol{U}}\ \underbrace{\boxed{\mathcal{M}_1(\boldsymbol{S})(\boldsymbol{V}\otimes\boldsymbol{W})^\top}}_{\breve{\boldsymbol{U}}^\top}$$

Step 2: Treat this as a matrix problem for updating factor $\boldsymbol{U}$:

$$\boldsymbol{U}_{t+1} = \boldsymbol{U}_t - \eta\nabla_{\boldsymbol{U}} f(\boldsymbol{F}_t)\left(\breve{\boldsymbol{U}}_t^\top\breve{\boldsymbol{U}}_t\right)^{-1}$$

How to construct scaled gradients for tensors?

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

Step 1: unfolding the tensor along mode-1:

$$\mathcal{M}_1(\boldsymbol{T}) = \boldsymbol{U} \underbrace{\mathcal{M}_1(\boldsymbol{S})(\boldsymbol{V}\otimes\boldsymbol{W})^\top}_{\breve{\boldsymbol{U}}^\top}$$

Step 2: Treat this as a matrix problem for updating factor $\boldsymbol{U}$:

$$\boldsymbol{U}_{t+1} = \boldsymbol{U}_t - \eta\nabla_{\boldsymbol{U}} f(\boldsymbol{F}_t)\left(\breve{\boldsymbol{U}}_t^\top\breve{\boldsymbol{U}}_t\right)^{-1}$$

Step 3: update the core tensor $\boldsymbol{S}$:

$$\boldsymbol{S}_{t+1} = \boldsymbol{S}_t - \eta\Big((\boldsymbol{U}_t^\top\boldsymbol{U}_t)^{-1}, (\boldsymbol{V}_t^\top\boldsymbol{V}_t)^{-1}, (\boldsymbol{W}_t^\top\boldsymbol{W}_t)^{-1}\Big)\cdot\nabla_{\boldsymbol{S}} f(\boldsymbol{F}_t)$$

ScaledGD for ill-conditioned low-rank tensor estimation

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

Scaled gradient iterations:

$$\begin{aligned}
\boldsymbol{U}_{t+1} &= \boldsymbol{U}_t - \eta\nabla_{\boldsymbol{U}} f(\boldsymbol{F}_t)\left(\breve{\boldsymbol{U}}_t^\top\breve{\boldsymbol{U}}_t\right)^{-1},\\
\boldsymbol{V}_{t+1} &= \boldsymbol{V}_t - \eta\nabla_{\boldsymbol{V}} f(\boldsymbol{F}_t)\left(\breve{\boldsymbol{V}}_t^\top\breve{\boldsymbol{V}}_t\right)^{-1},\\
\boldsymbol{W}_{t+1} &= \boldsymbol{W}_t - \eta\nabla_{\boldsymbol{W}} f(\boldsymbol{F}_t)\left(\breve{\boldsymbol{W}}_t^\top\breve{\boldsymbol{W}}_t\right)^{-1},\\
\boldsymbol{S}_{t+1} &= \boldsymbol{S}_t - \eta\left((\boldsymbol{U}_t^\top\boldsymbol{U}_t)^{-1}, (\boldsymbol{V}_t^\top\boldsymbol{V}_t)^{-1}, (\boldsymbol{W}_t^\top\boldsymbol{W}_t)^{-1}\right)\cdot\nabla_{\boldsymbol{S}} f(\boldsymbol{F}_t),
\end{aligned}$$

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

Key property: invariance to parameterization.

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*

$$\|(\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t) \cdot \boldsymbol{S}_t - \boldsymbol{T}\|_{\mathrm{F}} \lesssim \varepsilon \cdot \sigma_{\min}(\boldsymbol{T})$$

- **Computational:** *within* $O\big(\log \frac{1}{\varepsilon}\big)$ *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*

$$\|(\boldsymbol{U}_t, \boldsymbol{V}_t, \boldsymbol{W}_t) \cdot \boldsymbol{S}_t - \boldsymbol{T}\|_{\mathrm{F}} \lesssim \varepsilon \cdot \sigma_{\min}(\boldsymbol{T})$$

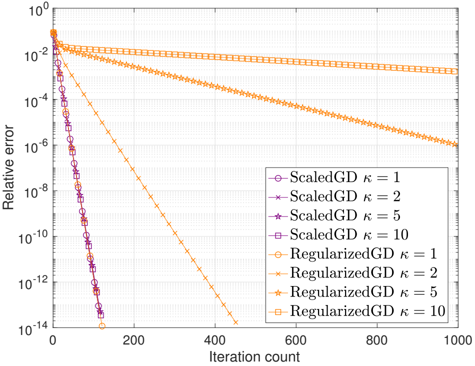
- **Computational:** *within* $O\left(\log \frac{1}{\varepsilon}\right)$ *iterations;*
- **Statistical:** *as long as the sample complexity satisfies*

$$n^3 p \gtrsim \mu^{3/2} r^{5/2} n^{3/2} \kappa^3 \log n.$$

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

Numerical evidence

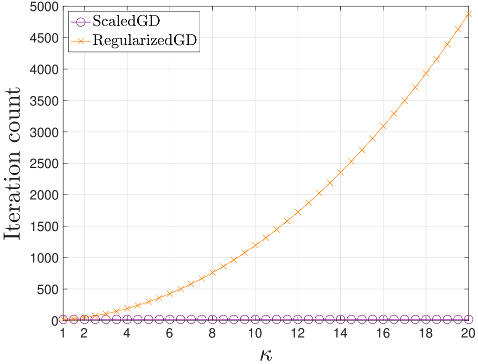
$$\min_{\boldsymbol{F}=(\boldsymbol{U},\boldsymbol{V},\boldsymbol{W},\boldsymbol{S})} \quad f(\boldsymbol{F}) = \frac{1}{2}\left\|\mathcal{P}_\Omega((\boldsymbol{U},\boldsymbol{V},\boldsymbol{W})\cdot\boldsymbol{S}) - \boldsymbol{T})\right\|_\mathrm{F}^2$$



The benefit of ScaledGD is even more evident for tensors!

Numerical evidence

$$\min_{\boldsymbol{F}=(\boldsymbol{U},\boldsymbol{V},\boldsymbol{W},\boldsymbol{S})} \; f(\boldsymbol{F}) = \frac{1}{2}\left\| \mathcal{P}_{\Omega}((\boldsymbol{U},\boldsymbol{V},\boldsymbol{W})\cdot\boldsymbol{S}) - \boldsymbol{T}) \right\|_{\mathrm{F}}^{2}$$



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

Data $=$ Sparse $+$ Low-rank

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

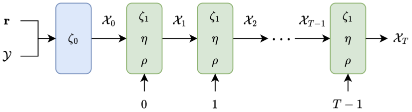
For a low-rank plus sparse tensor, ScaledGD with spectral initialization and iteration-varying thresholding converges at a constant rate, as long as the corruption level per fiber satisfies

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

Can use selective mode updates to accelerate computation!

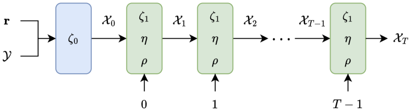
Hyperparameter tuning via self-supervised learning

unfolding + self-supervised learning



Hyperparameter tuning via self-supervised learning

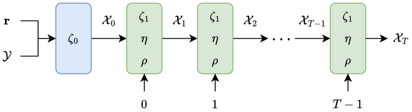
unfolding + self-supervised learning



some materials data

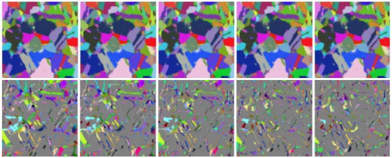
Hyperparameter tuning via self-supervised learning

unfolding + self-supervised learning



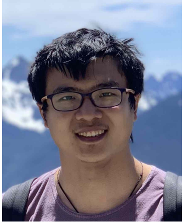
some materials data

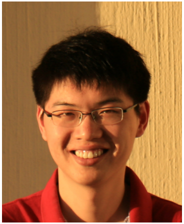
low-rank + sparse decomposition



"Deep Unfolded Tensor Robust PCA with Self-supervised Learning", Dong, Shah, Donegan, and Chi, ICASSP 2023.

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:

$$\boldsymbol{M} = \boldsymbol{X}\boldsymbol{X}^\top, \qquad \boldsymbol{X} \in \mathbb{R}^{n \times r'}, \qquad r' > r$$

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:

$$\boldsymbol{M} = \boldsymbol{X}\boldsymbol{X}^\top, \qquad \boldsymbol{X} \in \mathbb{R}^{n \times r'}, \qquad r' > r$$

ScaledGD:

$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t - \eta \nabla_{\boldsymbol{X}} f(\boldsymbol{X}_t) \underbrace{(\boldsymbol{X}_t^\top \boldsymbol{X}_t)^{-1}}_{\texttt{preconditioner}}$$

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:

$$\boldsymbol{M} = \boldsymbol{X}\boldsymbol{X}^\top, \qquad \boldsymbol{X} \in \mathbb{R}^{n\times r'}, \qquad r' > r$$

ScaledGD(λ):

$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t - \eta \nabla_{\boldsymbol{X}} f(\boldsymbol{X}_t) \underbrace{(\boldsymbol{X}_t^\top \boldsymbol{X}_t + \lambda \boldsymbol{I})^{-1}}_{\texttt{preconditioner}}$$

add regularization to stablize the preconditioner

Does preconditioning hurt generalization?

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

optimization

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* $\lambda \asymp \sigma_{\min}(\boldsymbol{M})$, $\eta \asymp 1$, *and* $\boldsymbol{X}_0 \sim \alpha \mathcal{N}(0, 1/n)$ *with sufficiently small* α *achieves*

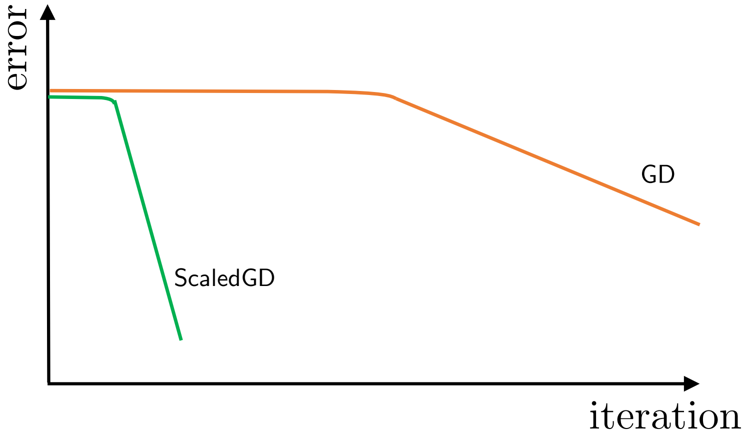
$$\|\boldsymbol{X}_t \boldsymbol{Y}_t^\top - \boldsymbol{M}\|_{\mathrm{F}} \lesssim \varepsilon \cdot \sigma_{\min}(\boldsymbol{M})$$

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

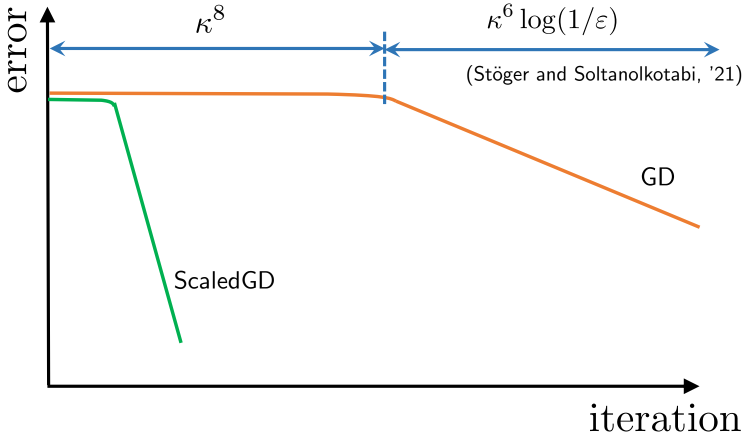
$$m \gtrsim nr^2 \textit{poly}(\kappa).$$

- Our analysis also enables exact convergence under random initialization with correct rank specification.

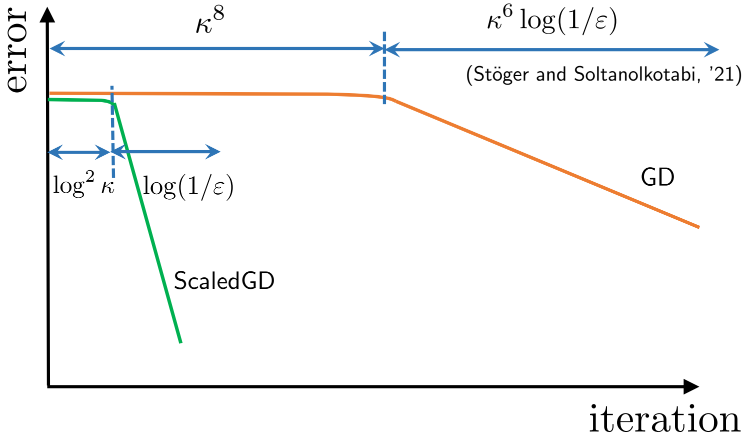
Comparison with overparameterized GD



Comparison with overparameterized GD



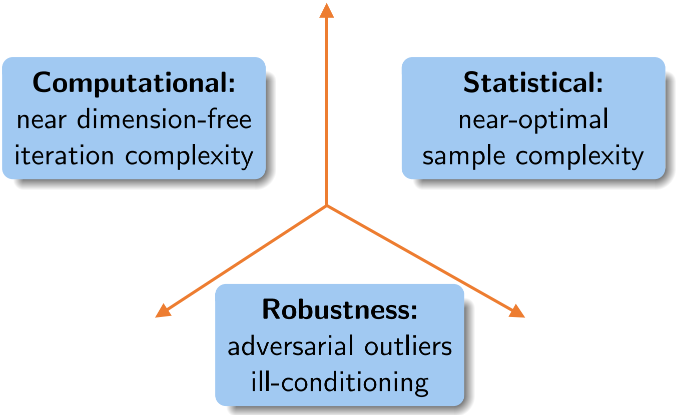
Comparison with overparameterized GD



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

Concluding remarks

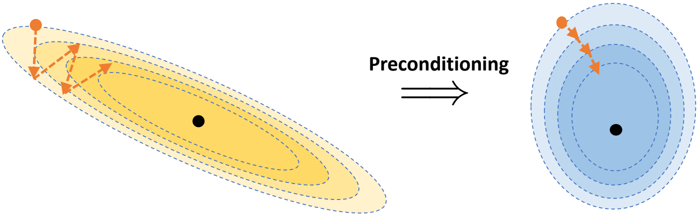
Bridging the theory-practice gap



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.

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:

- Nonconvex Optimization Meets Low-Rank Matrix Factorization: An Overview, *IEEE Trans. on Signal Processing*, 2019.

ScaledGD for low-rank matrix estimation:

- The Power of Preconditioning in Overparameterized Low-Rank Matrix Sensing, *arXiv preprint arXiv:2302.01186*, 2023. Short version at ICML 2023.
- Accelerating ill-conditioned low-rank matrix estimation via scaled gradient descent, *Journal of Machine Learning Research*, 2021.
- Low-rank matrix recovery with scaled subgradient methods: Fast and robust convergence without the condition number, *IEEE Trans. on Signal Processing*, 2021.

ScaledGD for low-rank tensor estimation:

- Scaling and scalability: Provable nonconvex low-rank tensor estimation from incomplete measurements, *Journal of Machine Learning Research*, 2022.
- Fast and provable tensor robust principal component analysis via scaled gradient descent, *Information and Inference*, accepted.

Thanks!

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