Scaling and Scalability: Accelerating Ill-conditioned Low-rank Estimation

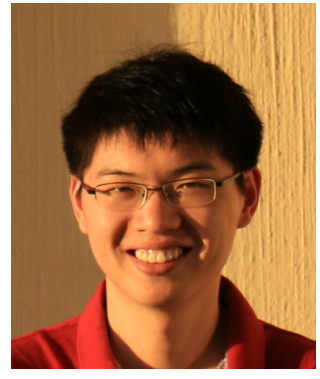
Yuejie Chi

Carnegie Mellon University

IEEE SAM TC Webinar, December 2021

Tian Tong

Carnegie Mellon



Cong Ma

University of Chicago

Sensing and imaging advances

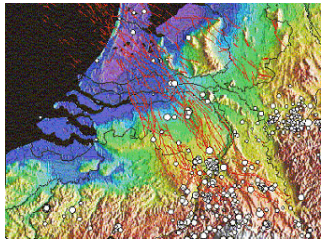
New imaging/sensing modalities allow us to probe the nature in unprecedented manners.

healthcare

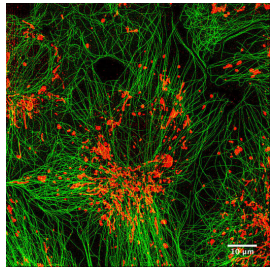
Radio astronomy

hyperspectral

Internet traffic



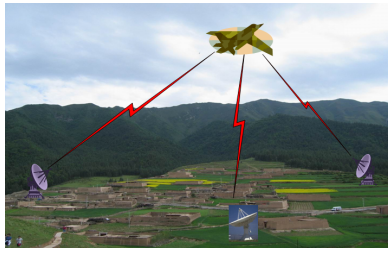
seismic imaging



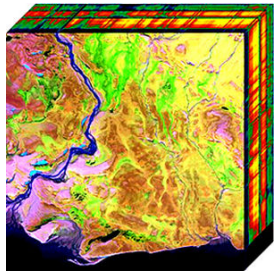
microscopy

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

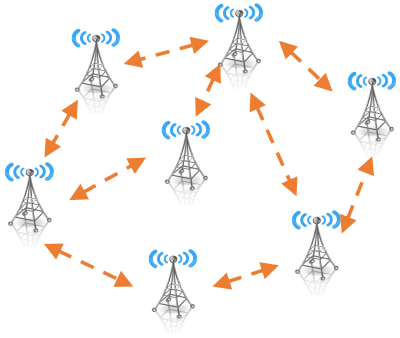


radar imaging

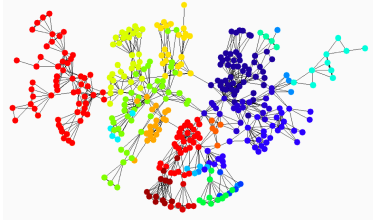


hyperspectral imaging

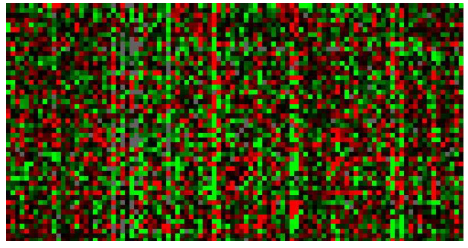
recommendation systems



localization



community detection



bioinformatics

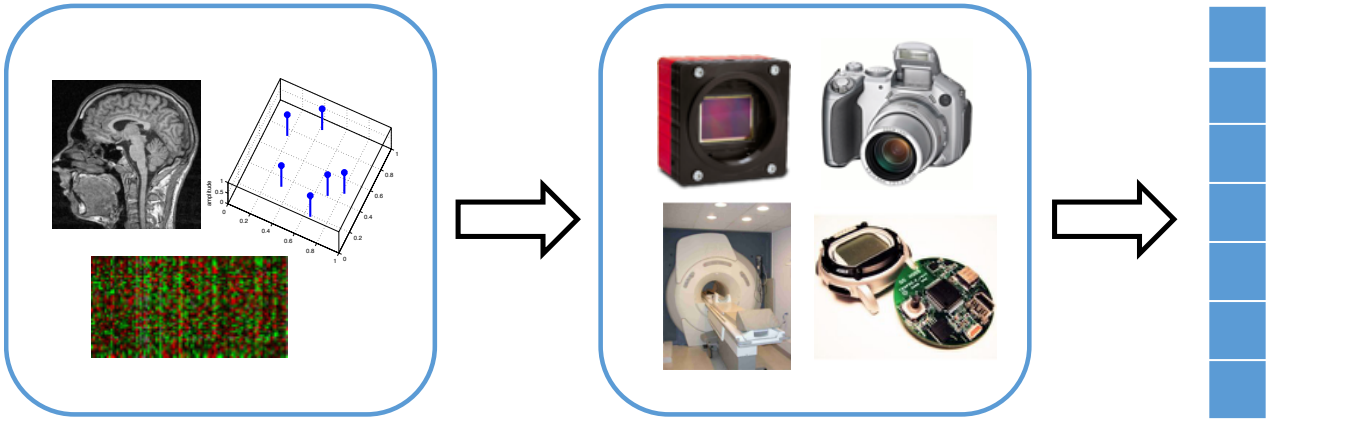
Low-rank representations encode latent structures

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 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

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

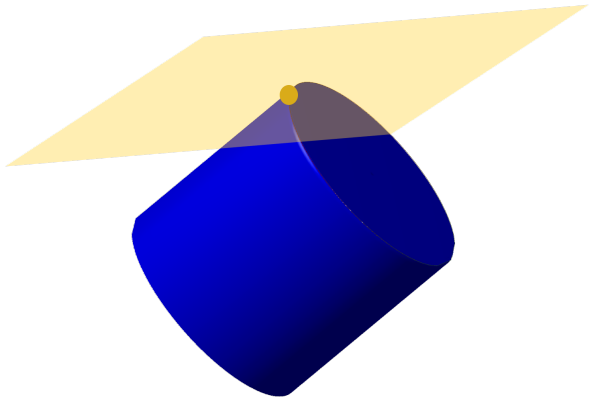
Convex relaxation via nuclear norm minimization

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

⇩ cvx surrogate

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

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



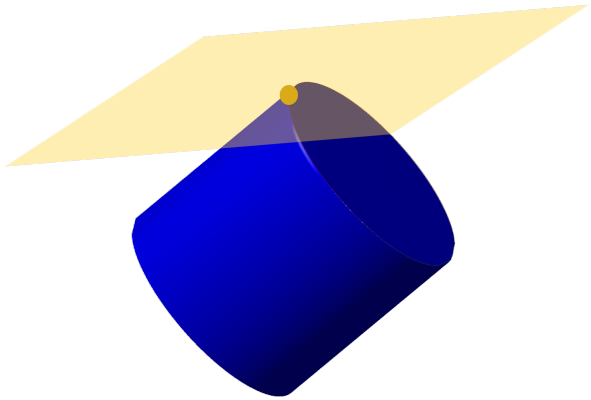
Convex relaxation via nuclear norm minimization

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

$\Downarrow$ cvx surrogate

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

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, ...

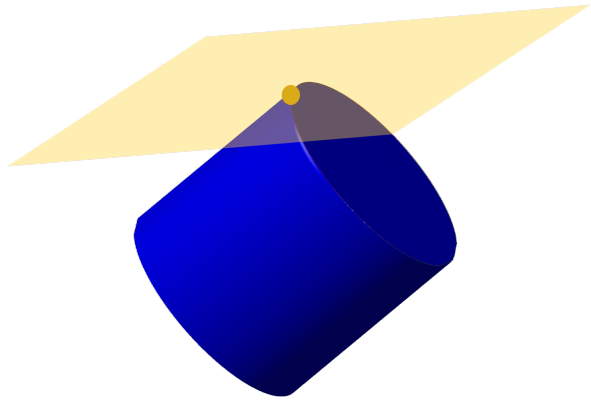
Convex relaxation via nuclear norm minimization

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

$\Downarrow$ cvx surrogate

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

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, ...

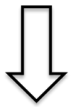
Poor scalability: operate in the *ambient* matrix space

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

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



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

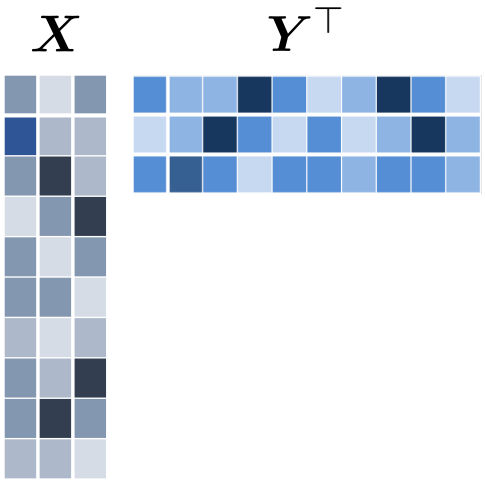
Low-rank matrix factorization

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

$$\Downarrow$$

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

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



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

Low-rank matrix factorization

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

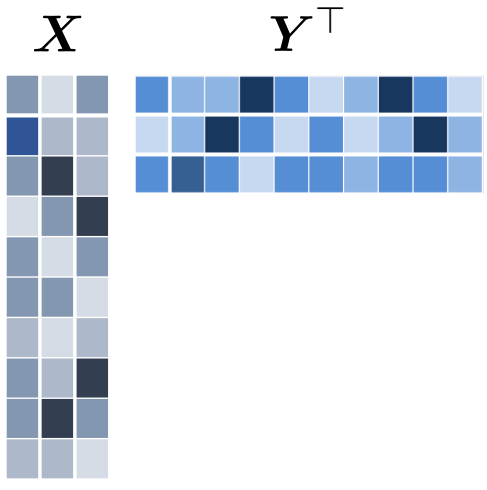
$$\Downarrow$$

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

more scalable,
but nonconvex!

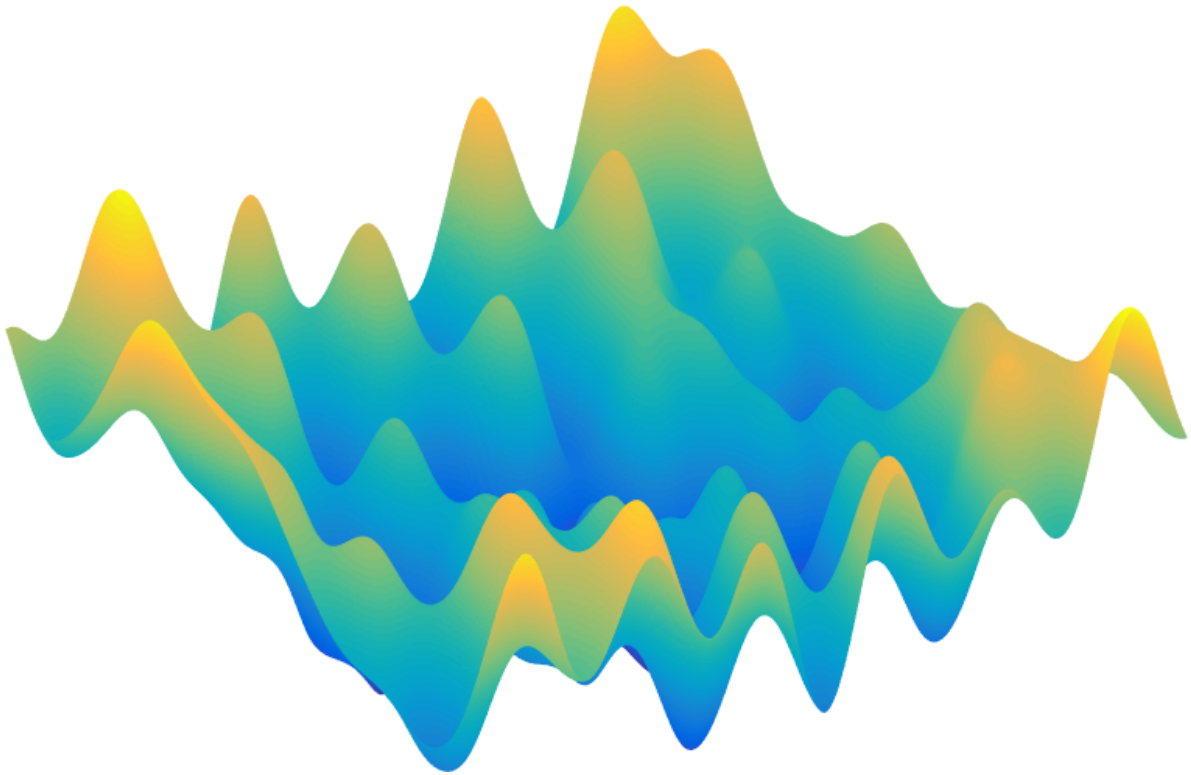
$$\Downarrow$$

$$\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{Y}^\top$$



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

Nonconvex problems are hard (in theory)!



"...in fact, the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity.

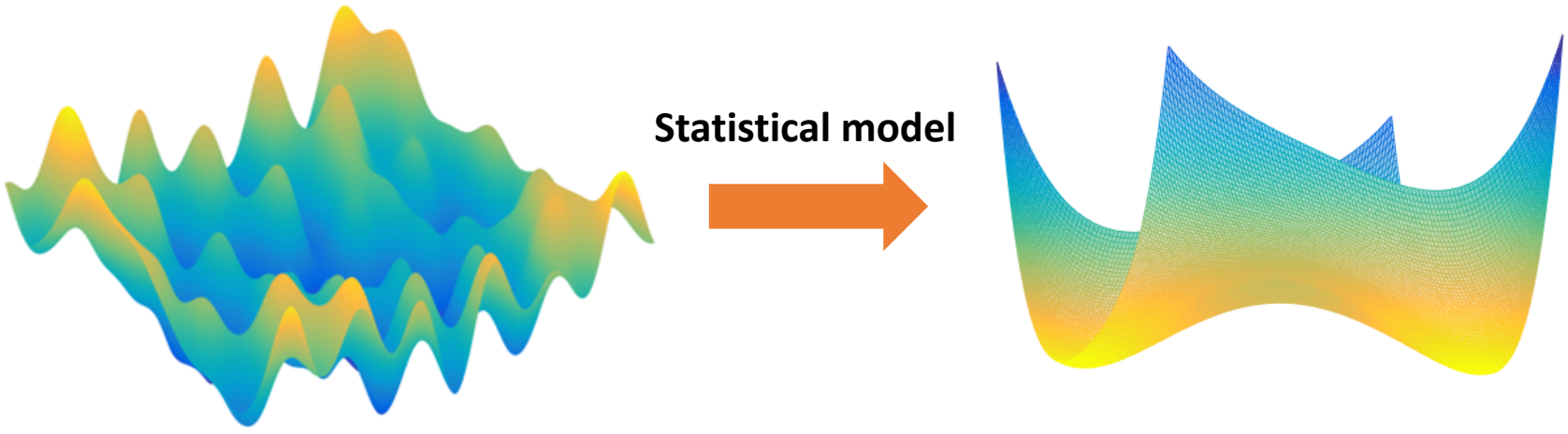
R. T. Rockafellar, in SIAM Review, 1993

Nonconvex problems are hard (in theory)!

"...in fact, the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity.

R. T. Rockafellar, in SIAM Review, 1993

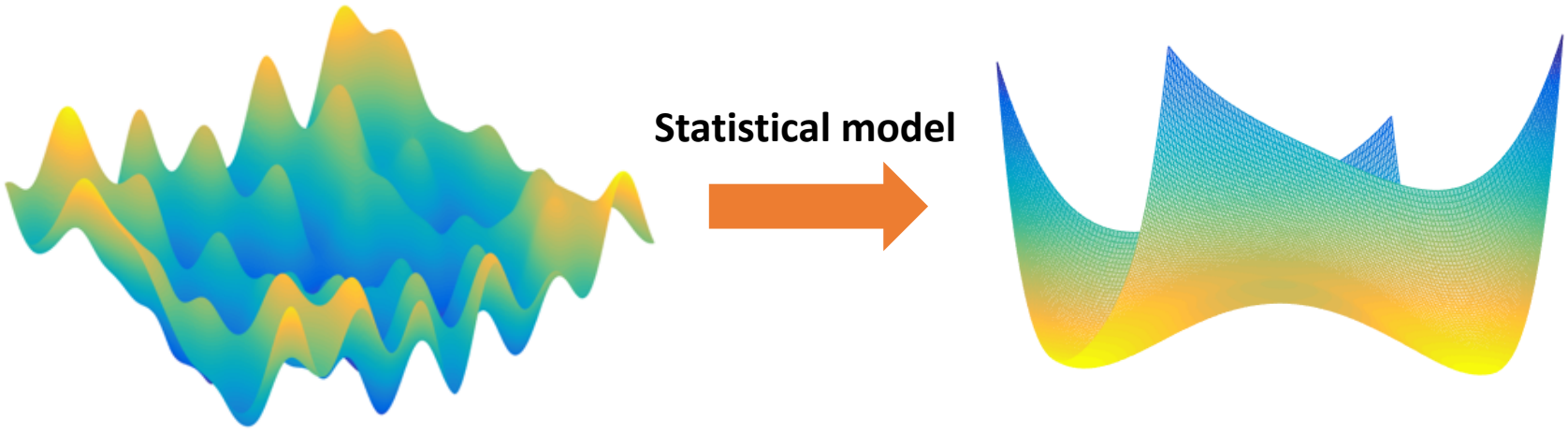
Statistics meets optimization



worst case

average case

Statistics meets optimization

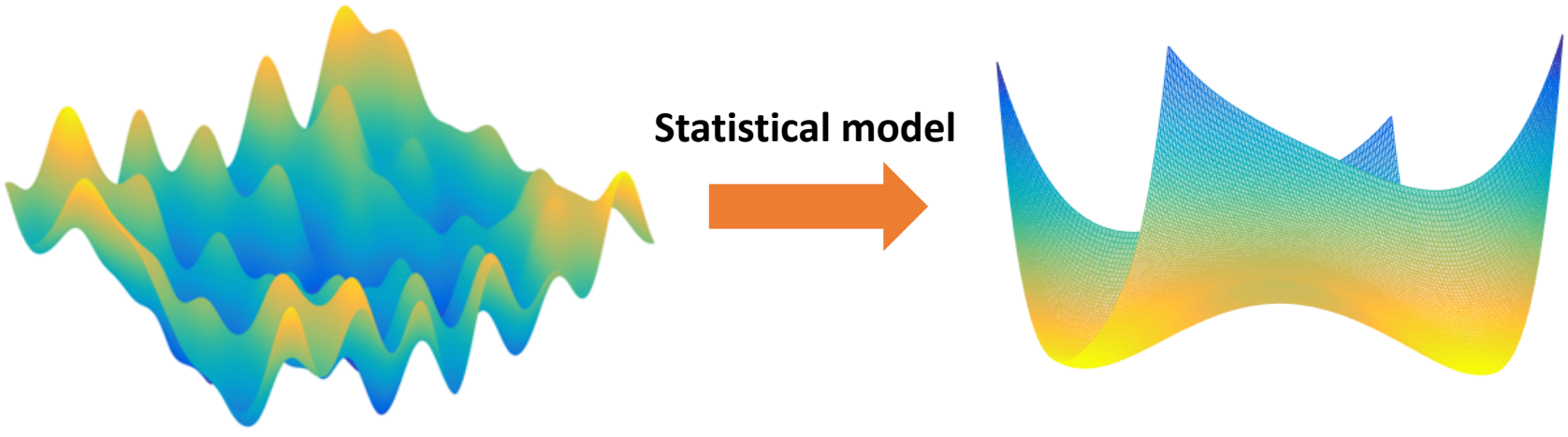


worst case

average case

Simple algorithms can be efficient for nonconvex problems!

Statistics meets optimization



worst case

average case

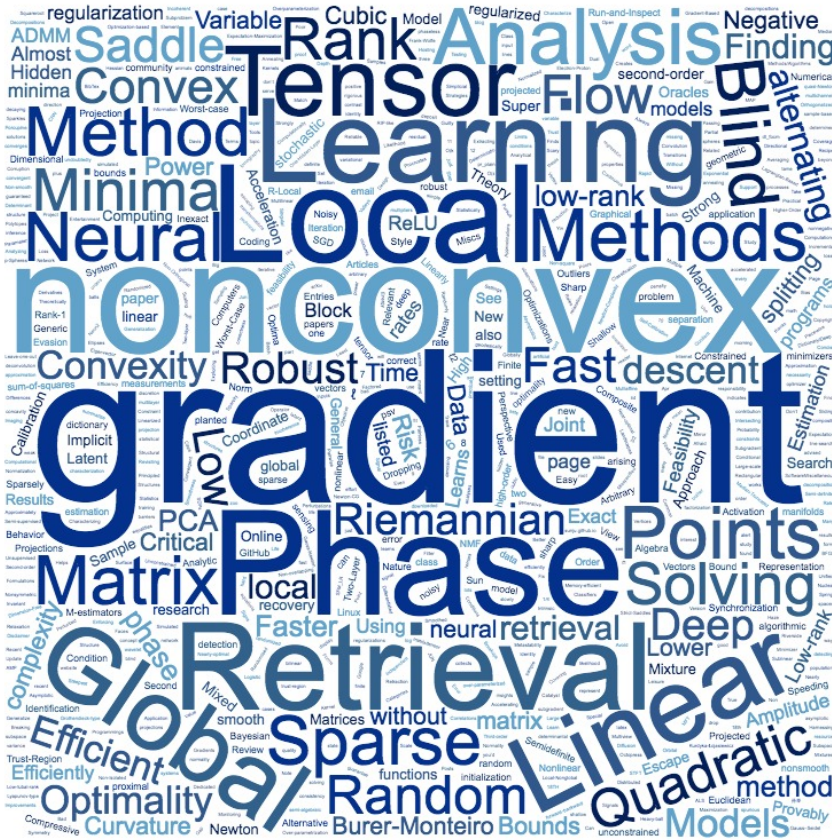
Simple algorithms can be efficient for nonconvex problems!

Vanilla gradient descent (GD):

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

for $t = 0, 1, \ldots$

Recent developments: provable nonconvex optimization



"Nonconvex Optimization Meets Low-Rank Matrix Factorization: An Overview," Chi, Lu, Chen, TSP 2019

Phase retrieval: Netrapalli et al. '13, Candès, Li, Soltanolkotabi '14, Chen, Candès '15, Cai, Li, Ma '15, Zhang et al. '16, Wang et al. '16, Sun, Qu, Wright '16, Ma et al. '17, Chen et al. '18, Soltani, Hegde '18, Ruan and Duchi, '18, ...

Matrix sensing/completion: Keshavan et al. '09, Jain et al. '09, Hardt '13, Jain et al. '13, Sun, Luo '15, Chen, Wainwright '15, Tu et al. '15, Zheng, Lafferty '15, Bhojanapalli et al. 16, Ge, Lee, Ma '16, Jin et al. '16, Ma et al. '17, Chen and Li '17, Cai et al. '18, Li, Zhu, Tang, Wakin '18, Charisopoulos et al. '19, ...

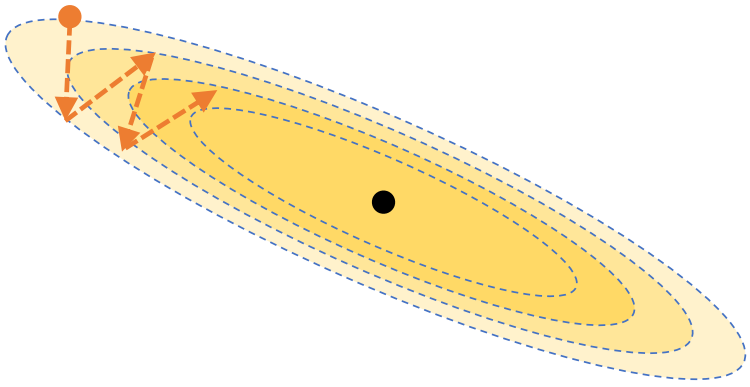
Blind deconvolution/demixing: Li et al. '16, Lee et al. '16, Cambareri, Jacques '16, Ling, Strohmer '16, Huang, Hand '16, Ma et al. '17, Zhang et al. '18, Li, Bresler '18, Dong, Shi '18, Shi, Chi '19, Qu et al. '19...

Dictionary learning: Arora et al. '14, Sun et al. '15, Chatterji, Bartlett '17, Bai et al. '18, Gilboa et al. '18, Rambhatla et al. '19, Qu et al. '19,...

Robust principal component analysis: Netrapalli et al. '14, Yi et al. '16, Gu et al. '16, Ge et al. '17, Cherapanamjeri et al. '17, Vaswani et al. '18, Maunu et al. '19, ...

Deep learning: Zhong et al. '17, Bai, Mei, Montanari '17, Du et al. '17, Ge, Lee, Ma '17, Du et al. '18, Soltanolkotabi and Oymak, '18...

Acceleration via preconditioning

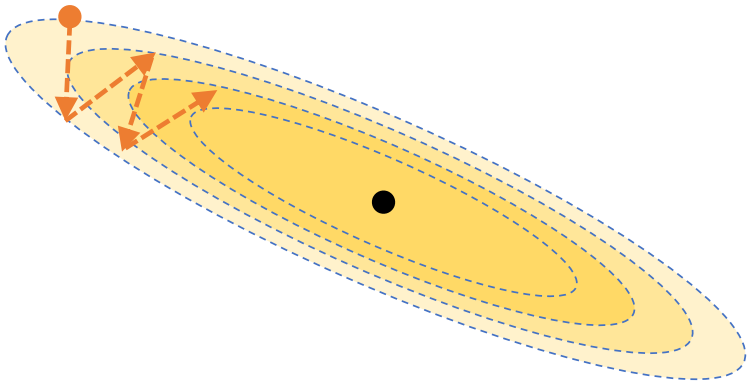


Vanilla GD:

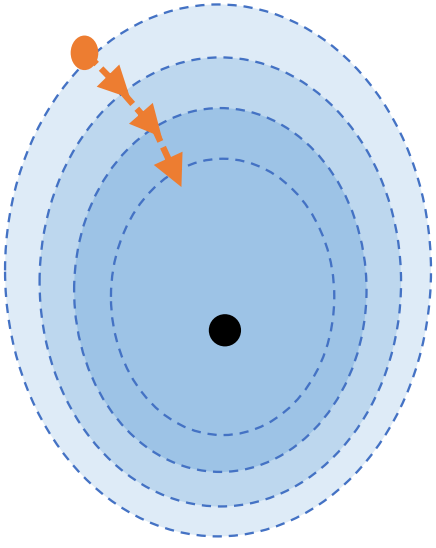
$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta \nabla f(\boldsymbol{x}_t)$$

☹ **Slows down with ill-conditioning.**

Acceleration via preconditioning



Preconditioning



Vanilla GD:

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

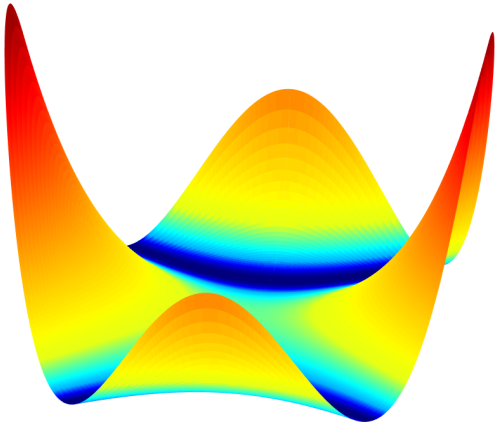
☹ **Slows down with ill-conditioning.**

Preconditioned GD:

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

☺ **Preconditioning helps!**

Robustness via nonsmooth optimization

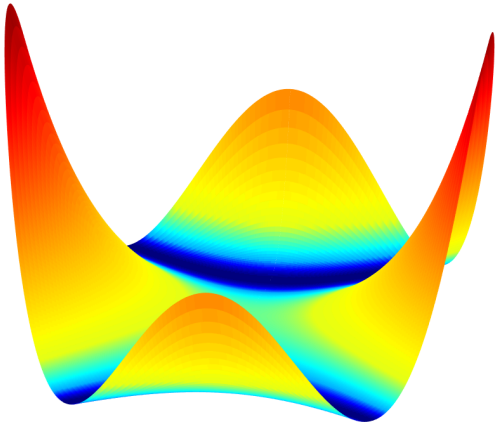


Least squares:

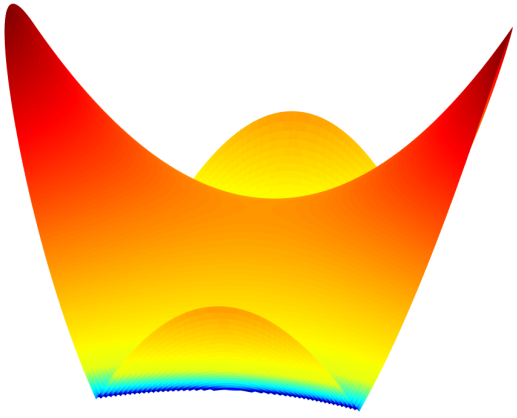
$$f(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{2} \left\| \boldsymbol{y} - \mathcal{A}(\boldsymbol{X}\boldsymbol{Y}^\top) \right\|_2^2$$

☹ **Sensitive to outliers.**

Robustness via nonsmooth optimization



Nonsmooth



Least squares:

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

☹ **Sensitive to outliers.**

Least absolute deviation:

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

☺ **Nonsmoothness helps!**

This talk

Optimization geometry:

When and why does simple gradient descent work well for low-rank matrix estimation?

This talk

Optimization geometry:

When and why does simple gradient descent work well for low-rank matrix estimation?

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

Optimization geometry:

When and why does simple gradient descent work well for low-rank matrix estimation?

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

Optimization geometry:

When and why does simple gradient descent work well for low-rank matrix estimation?

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:

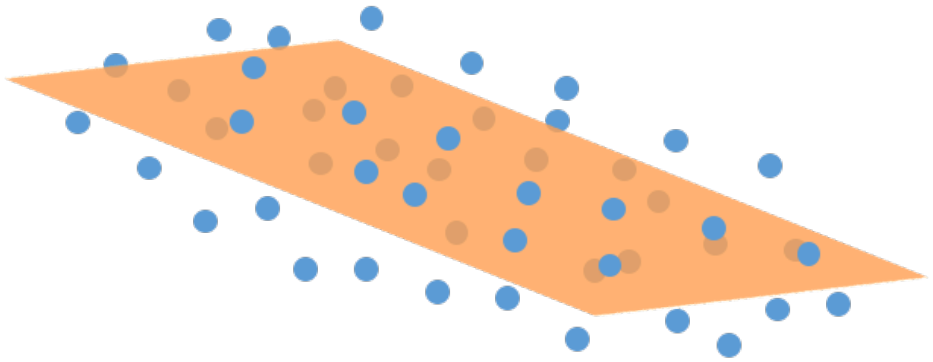
Can we design provably robust variants that are simultaneously oblivious to the presence of outliers?

Generalization to tensors:

Can we generalize to higher-dimensional objects?

Warm-up: understanding the geometry of low-rank matrix factorization

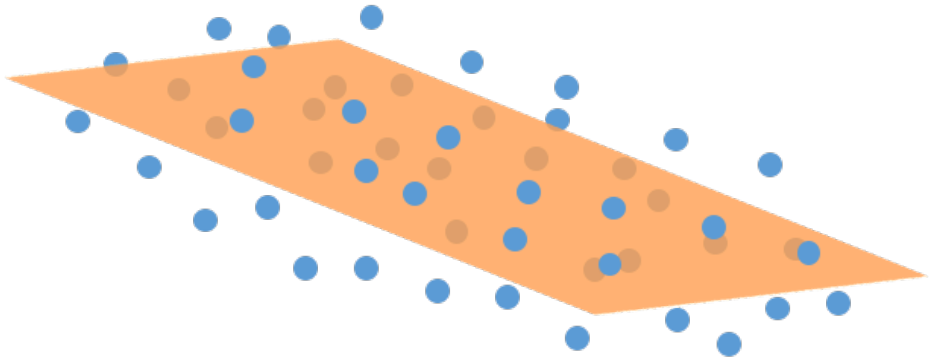
Revisiting PCA: in search of low-rank representation



Given $\boldsymbol{M} \succeq 0 \in \mathbb{R}^{n \times n}$ (e.g. sample covariance matrix), find its best rank-r approximation:

$$\underbrace{\widehat{\boldsymbol{M}} = \mathsf{argmin}_{\boldsymbol{Z}} \, \|\boldsymbol{Z} - \boldsymbol{M}\|_{\mathrm{F}}^2 \quad \text{s.t.} \quad \mathsf{rank}(\boldsymbol{Z}) \le r}_{\text{nonconvex optimization!}}$$

Revisiting PCA: in search of low-rank representation



This problem admits a closed-form solution:

- let $\boldsymbol{M} = \sum_{i=1}^{n} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^\top$ be eigen-decomposition of $\boldsymbol{M}$ ($\lambda_1 \geq \cdots \lambda_r > \lambda_{r+1} \cdots \geq \lambda_n$), then

$$\widehat{\boldsymbol{M}} = \sum_{i=1}^{r} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^\top$$

— *nonconvex, but tractable*

An optimization viewpoint

Low-rank factorization: if we factorize $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{X}^\top$ with $\boldsymbol{X} \in \mathbb{R}^{n\times r}$, then it leads to a nonconvex problem:

$$\text{minimize}_{\boldsymbol{X}\in\mathbb{R}^{n\times r}} \quad f(\boldsymbol{X}) = \|\boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{M}\|_{\mathrm{F}}^2$$

An optimization viewpoint

Low-rank factorization: if we factorize $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{X}^\top$ with $\boldsymbol{X} \in \mathbb{R}^{n\times r}$, then it leads to a nonconvex problem:

$$\text{minimize}_{\boldsymbol{X}\in\mathbb{R}^{n\times r}} \quad f(\boldsymbol{X}) = \|\boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{M}\|_{\mathrm{F}}^2$$

Theorem (Baldi and Hornik, 1989)

Suppose $\boldsymbol{M}$ has a strict eigen-gap between λ_r and λ_{r+1}, the critical points of $f(\boldsymbol{X})$ can be categorized into

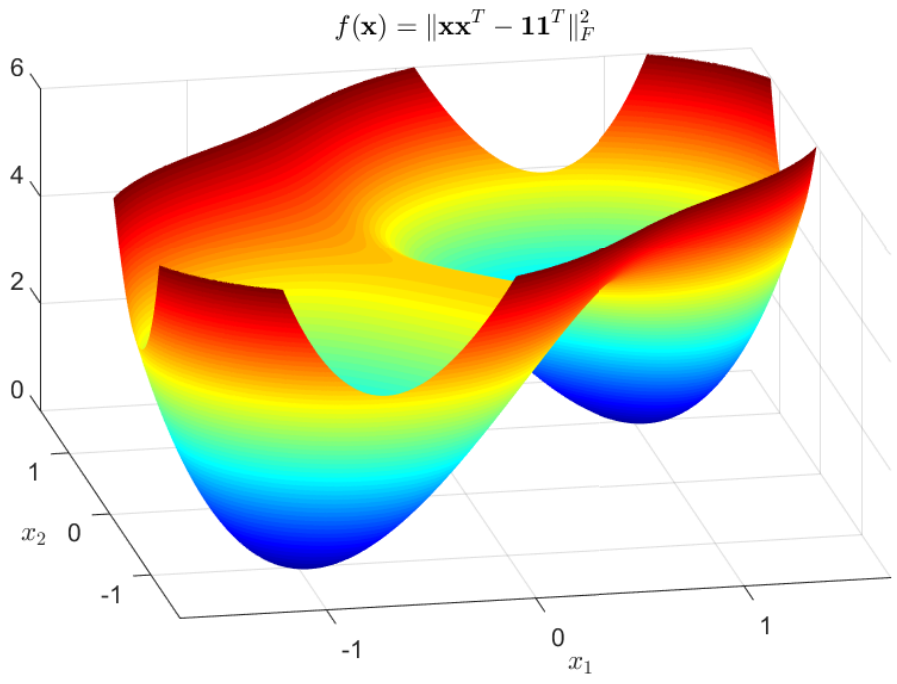
- *global minima;*
- *strict saddle points, from which there exist directions to strictly decrease $f(\boldsymbol{X})$.*

In other words, all local minima are global minima!

Baldi and Hornik. "Neural networks and principal component analysis: Learning from examples without local minima." Neural networks 2.1 (1989): 53-58.

Benign landscape of PCA

For example, for 2-dimensional case $f(\boldsymbol{x}) = \left\| \boldsymbol{x}\boldsymbol{x}^\top - \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \right\|_{\mathrm{F}}^2$



global minima: $\boldsymbol{x} = \pm \begin{bmatrix} 1 \\ 1 \end{bmatrix}$; strict saddles: $\boldsymbol{x} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, and $\pm \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

— No "spurious" local minima!

Local geometry: the hidden convexity

$$f(\boldsymbol{X}) := \left\| \boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{X}_\star \boldsymbol{X}_\star^\top \right\|_{\mathrm{F}}^2, \qquad \boldsymbol{X} \in \mathbb{R}^{n\times r}$$

A modified distance metric:

$$\mathrm{dist}^2(\boldsymbol{X}, \boldsymbol{X}_\star) = \|\boldsymbol{X}\boldsymbol{H}_{\boldsymbol{X}} - \boldsymbol{X}_\star\|_{\mathrm{F}}^2,$$

where $\boldsymbol{H}_{\boldsymbol{X}} := \mathrm{argmin}_{\boldsymbol{H}\in\mathcal{O}^{r\times r}} \|\boldsymbol{X}\boldsymbol{H} - \boldsymbol{X}_\star\|_{\mathrm{F}}^2$.

Local geometry: the hidden convexity

$$f(\boldsymbol{X}) := \left\| \boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{X}_\star \boldsymbol{X}_\star^\top \right\|_{\mathrm{F}}^2, \qquad \boldsymbol{X} \in \mathbb{R}^{n\times r}$$

A modified distance metric:

$$\mathrm{dist}^2(\boldsymbol{X}, \boldsymbol{X}_\star) = \|\boldsymbol{X}\boldsymbol{H}_{\boldsymbol{X}} - \boldsymbol{X}_\star\|_{\mathrm{F}}^2,$$

where $\boldsymbol{H}_{\boldsymbol{X}} := \mathrm{argmin}_{\boldsymbol{H}\in\mathcal{O}^{r\times r}} \|\boldsymbol{X}\boldsymbol{H} - \boldsymbol{X}_\star\|_{\mathrm{F}}^2$.

Restricted strong convexity:

$$\mathsf{vec}(\boldsymbol{V})^\top \nabla^2 f(\boldsymbol{X}) \mathsf{vec}(\boldsymbol{V}) \gtrsim \|\boldsymbol{V}\|_{\mathrm{F}}^2, \quad \boldsymbol{V} := \boldsymbol{X}\boldsymbol{H}_{\boldsymbol{X}} - \boldsymbol{X}_\star.$$

Local geometry: the hidden convexity

$$f(\boldsymbol{X}) := \left\| \boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{X}_\star \boldsymbol{X}_\star^\top \right\|_{\mathrm{F}}^2, \qquad \boldsymbol{X} \in \mathbb{R}^{n\times r}$$

A modified distance metric:

$$\mathrm{dist}^2(\boldsymbol{X}, \boldsymbol{X}_\star) = \|\boldsymbol{X}\boldsymbol{H}_{\boldsymbol{X}} - \boldsymbol{X}_\star\|_{\mathrm{F}}^2,$$

where $\boldsymbol{H}_{\boldsymbol{X}} := \mathrm{argmin}_{\boldsymbol{H}\in\mathcal{O}^{r\times r}} \|\boldsymbol{X}\boldsymbol{H} - \boldsymbol{X}_\star\|_{\mathrm{F}}^2$.

Restricted strong convexity:

$$\mathsf{vec}(\boldsymbol{V})^\top \nabla^2 f(\boldsymbol{X}) \mathsf{vec}(\boldsymbol{V}) \gtrsim \|\boldsymbol{V}\|_{\mathrm{F}}^2, \quad \boldsymbol{V} := \boldsymbol{X}\boldsymbol{H}_{\boldsymbol{X}} - \boldsymbol{X}_\star.$$

Linear convergence: The GD iterates obey

$$\mathrm{dist}^2(\boldsymbol{X}_t, \boldsymbol{X}_\star) \le \left(1 - \frac{c}{\kappa}\right)^t \mathrm{dist}^2(\boldsymbol{X}_0, \boldsymbol{X}_\star), \qquad t \ge 0,$$

as long as $\mathrm{dist}^2(\boldsymbol{X}_0, \boldsymbol{X}_\star) \lesssim \lambda_1$. Here, $\kappa := \lambda_1/\lambda_r$.

Gradient descent for low-rank matrix sensing

Low-rank matrix sensing: GD with balancing regularization

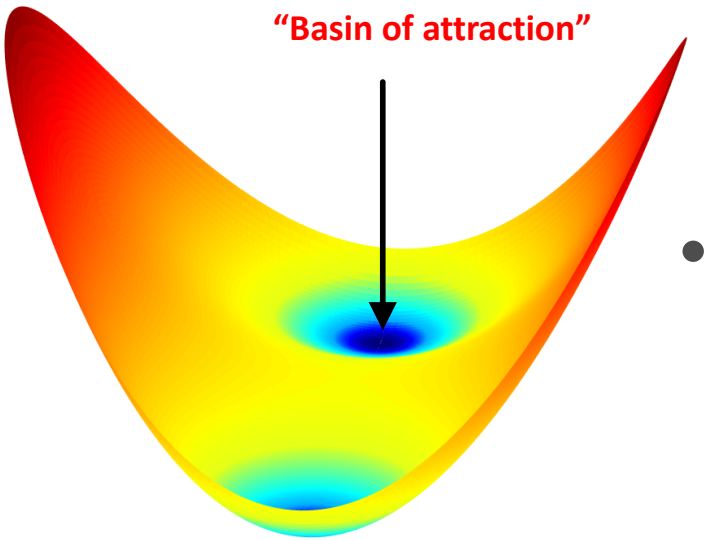
$$\min_{\boldsymbol{X},\boldsymbol{Y}} \quad f(\boldsymbol{X},\boldsymbol{Y}) = \frac{1}{2}\left\|\boldsymbol{y} - \mathcal{A}(\boldsymbol{X}\boldsymbol{Y}^\top)\right\|_2^2 = \frac{1}{2}\left\|\mathcal{A}(\boldsymbol{M} - \boldsymbol{X}\boldsymbol{Y}^\top)\right\|_2^2$$

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".

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

- **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*

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

- **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*

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

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

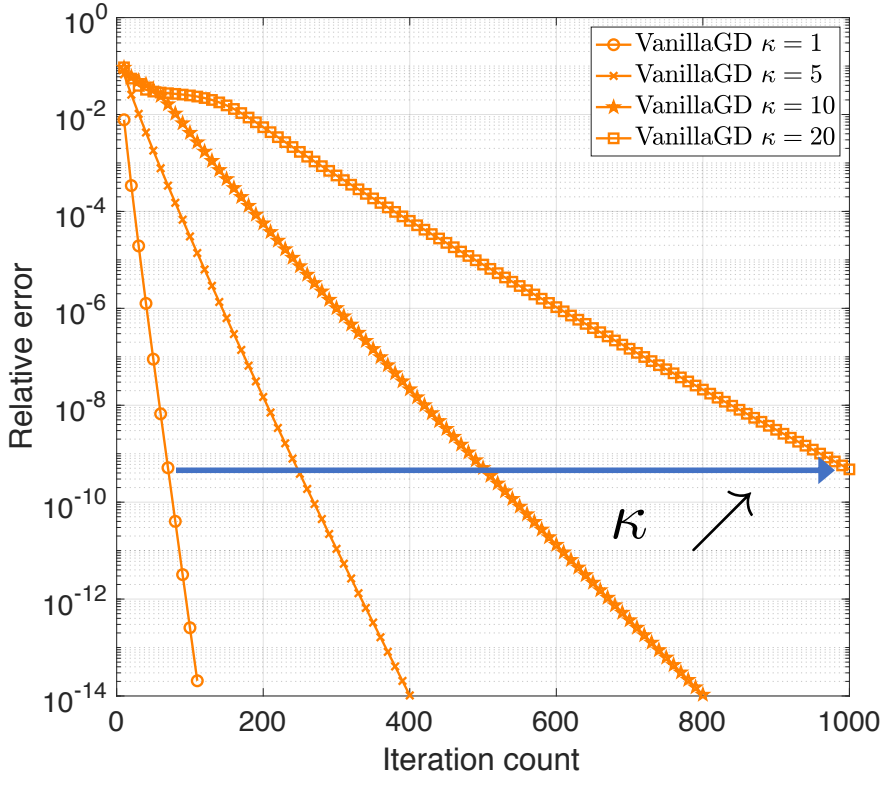
$$m \gtrsim (n_1 + n_2) r^2 \kappa^2.$$

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, ...)

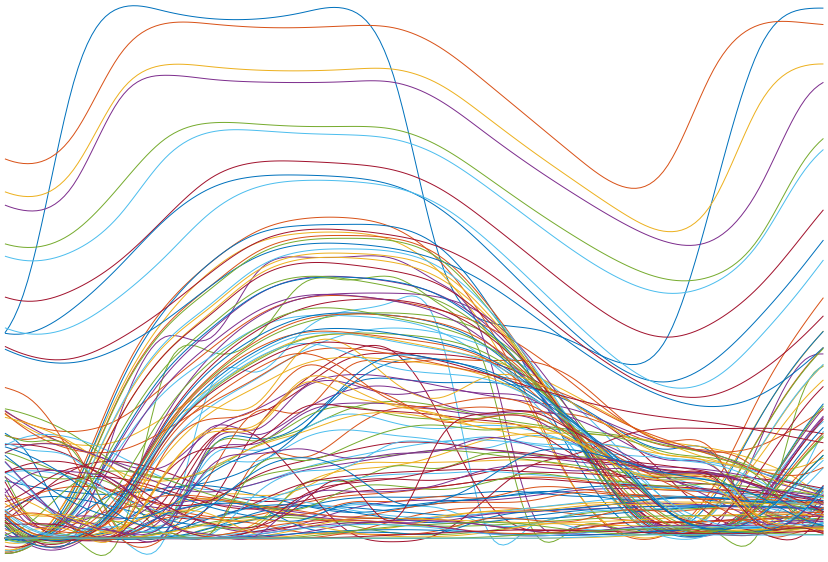
Convergence slows down for ill-conditioned matrices

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

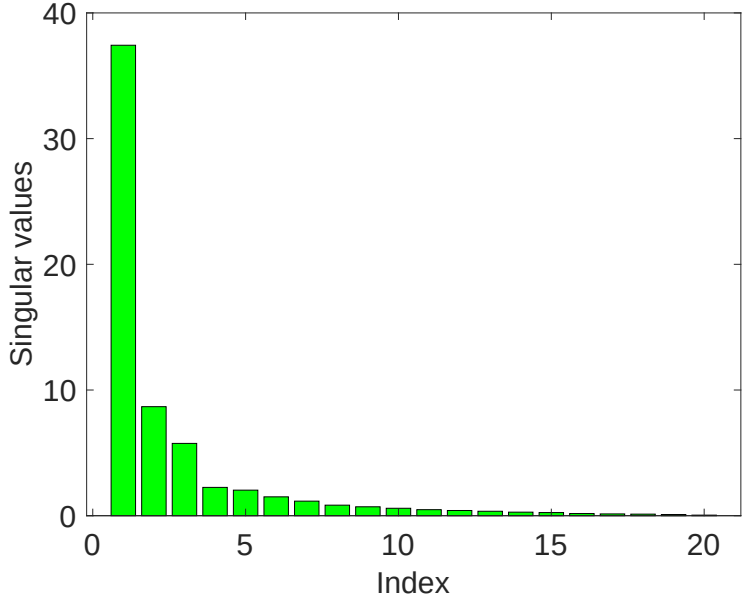


Vanilla GD converges in $O\left(\kappa \log \frac{1}{\varepsilon}\right)$ iterations.

Condition number can be large



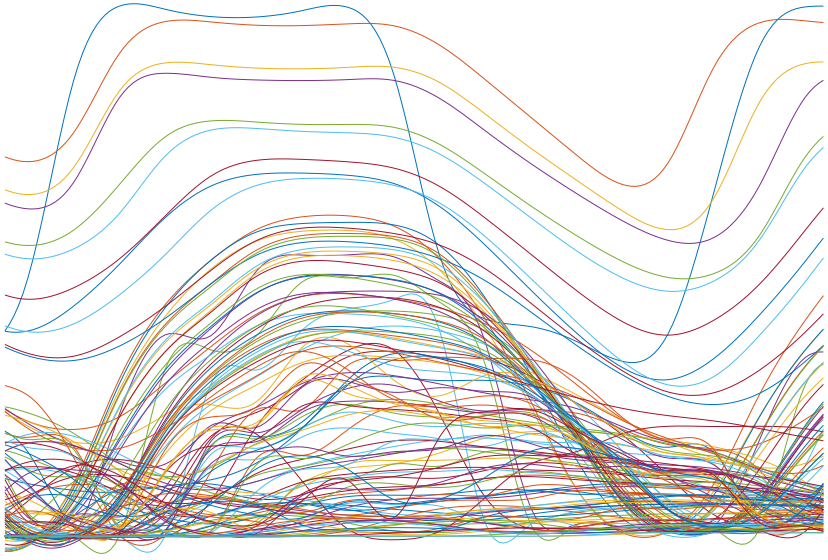
chlorine concentration levels
120 junctions, 180 time slots



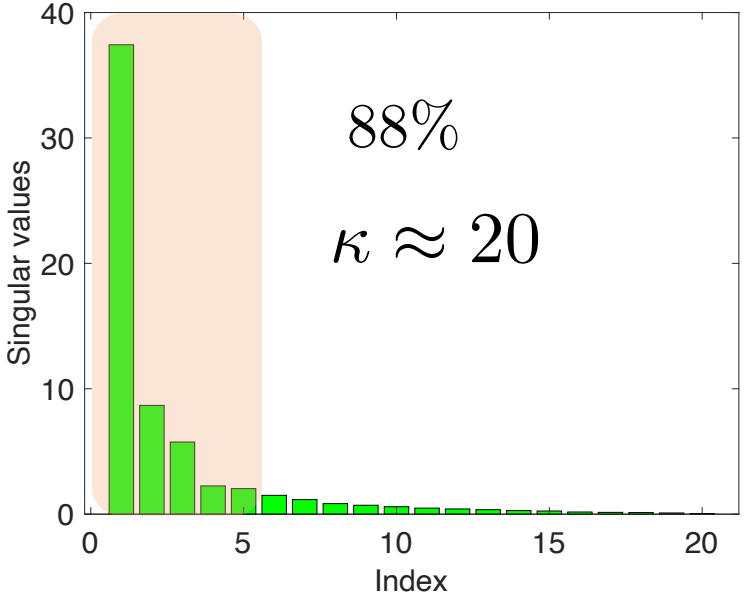
power-law spectrum

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

Condition number can be large



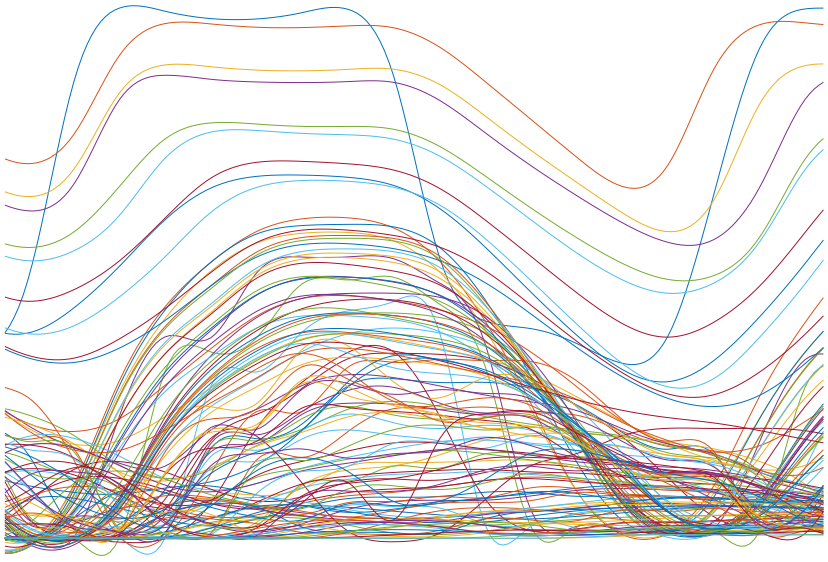
chlorine concentration levels
120 junctions, 180 time slots



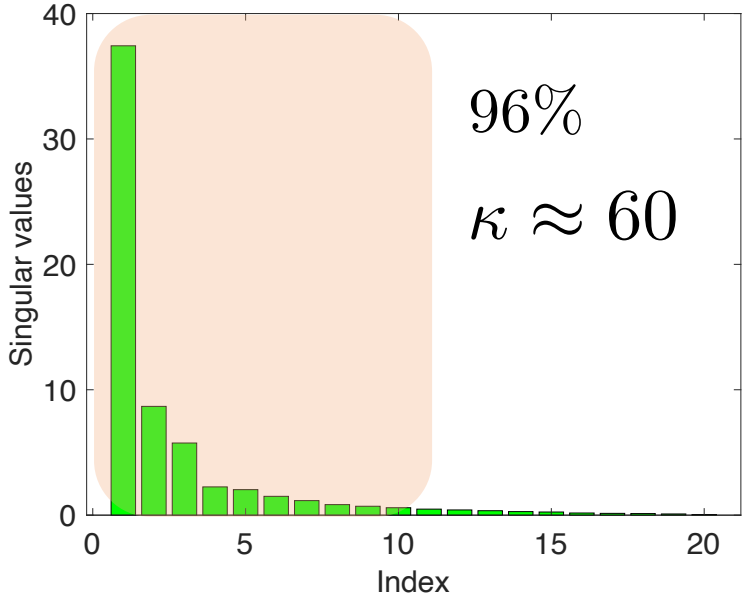
rank-5 approximation

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

Condition number can be large



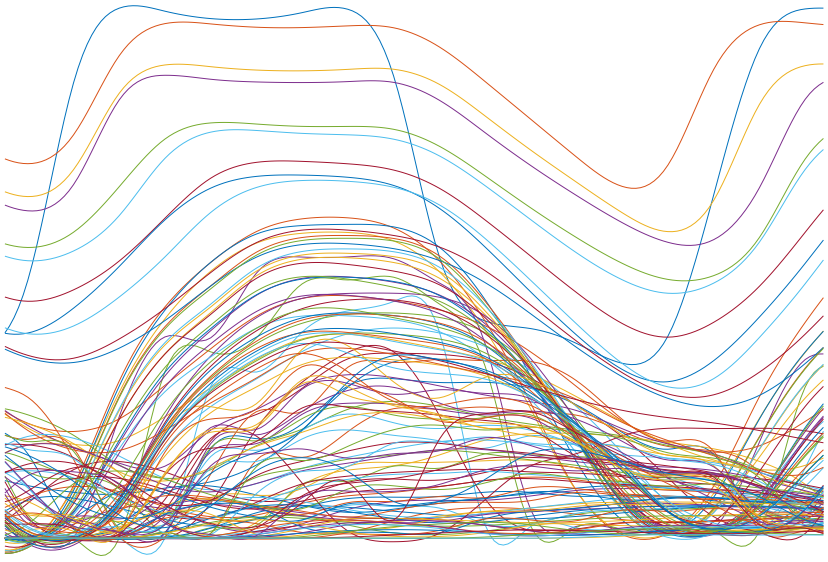
chlorine concentration levels
120 junctions, 180 time slots

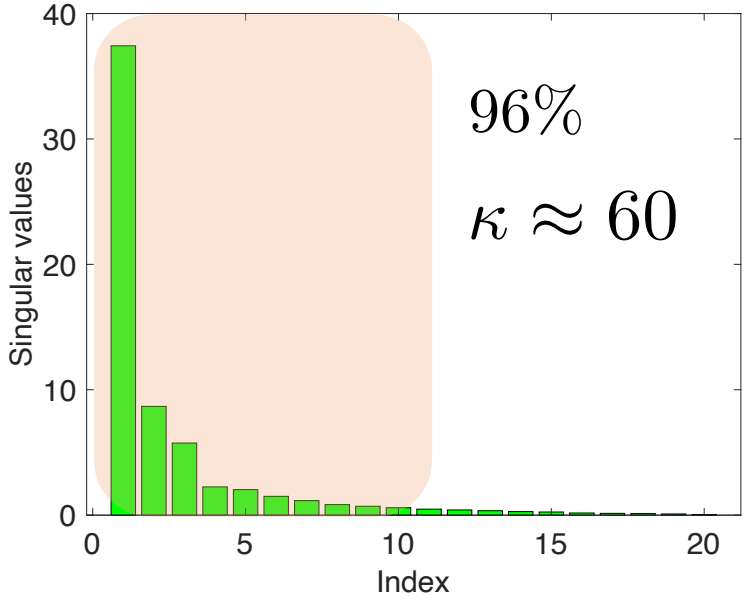


rank-10 approximation

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

Condition number can be large





chlorine concentration levels
120 junctions, 180 time slots

rank-10 approximation

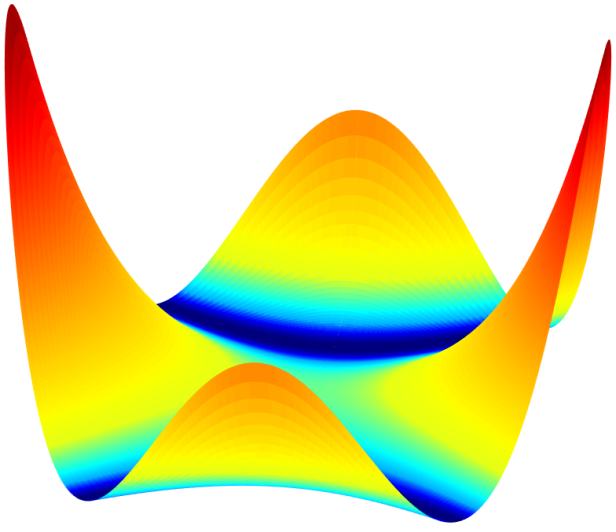
Can we accelerate the convergence rate of GD to $O(\log \frac{1}{\varepsilon})$?

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

Accelerating ill-conditioned matrix estimation

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:**

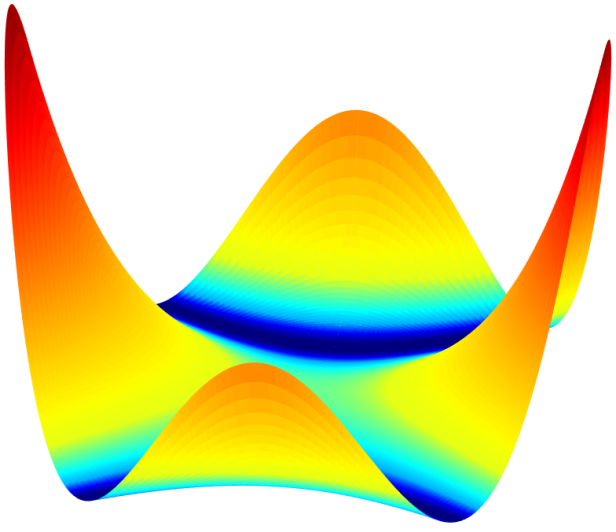
$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t - \eta \nabla_{\boldsymbol{X}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t) \underbrace{(\boldsymbol{Y}_t^\top \boldsymbol{Y}_t)^{-1}}_{\texttt{preconditioner}}$$

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

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:**

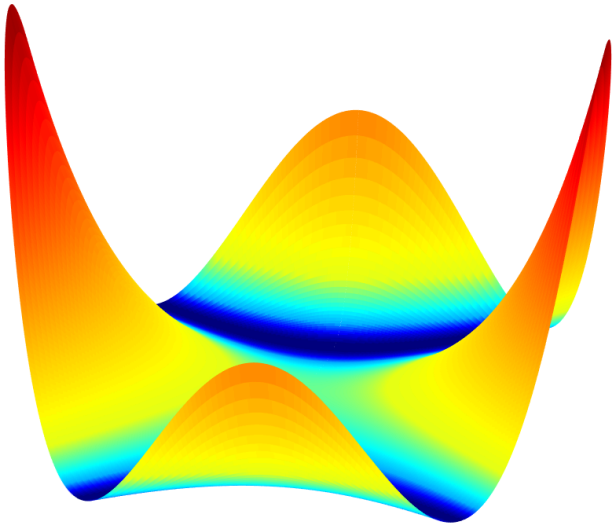
$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t - \eta \nabla_{\boldsymbol{X}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t) \underbrace{(\boldsymbol{Y}_t^\top \boldsymbol{Y}_t)^{-1}}_{\texttt{preconditioner}}$$

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

for $t = 0, 1, \ldots$

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:**

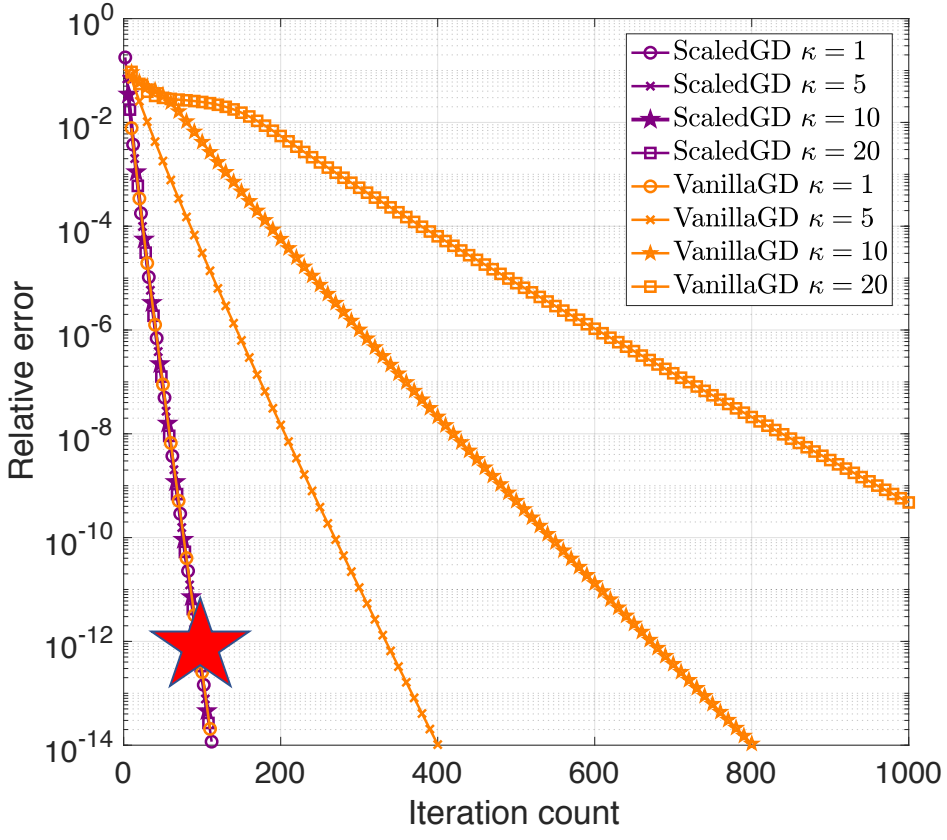
$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t - \eta \nabla_{\boldsymbol{X}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t) \underbrace{(\boldsymbol{Y}_t^\top \boldsymbol{Y}_t)^{-1}}_{\texttt{preconditioner}}$$

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

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!

ScaledGD as a quasi-Newton method

$$f(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{2} \left\| \boldsymbol{X}\boldsymbol{Y}^\top - \boldsymbol{X}_\star \boldsymbol{Y}_\star^\top \right\|_{\mathrm{F}}^2$$

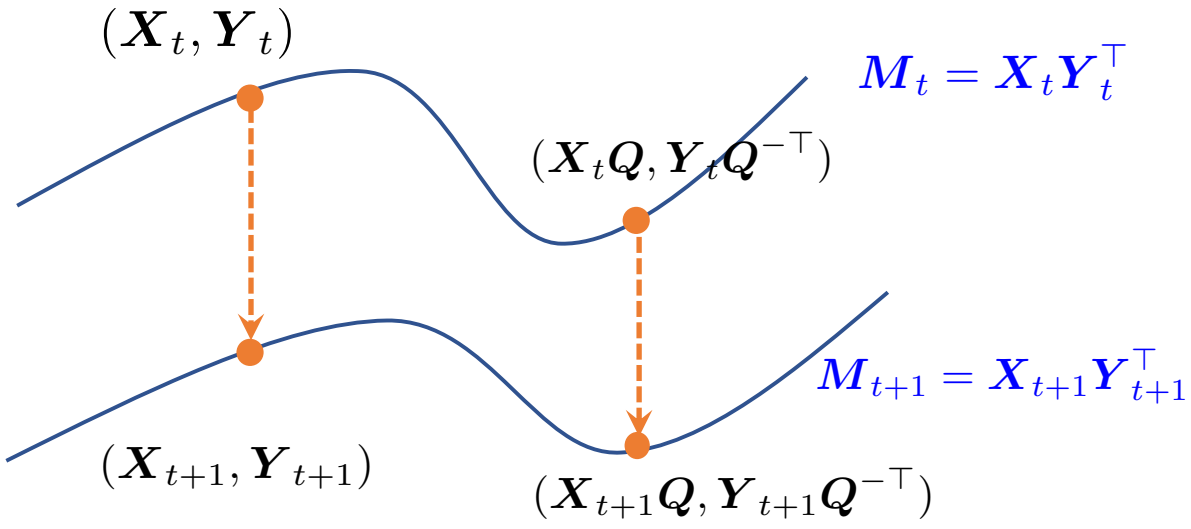
For low-rank matrix factorization, ScaledGD is equivalent to:

$$\begin{bmatrix} \mathsf{vec}(\boldsymbol{X}) \\ \mathsf{vec}(\boldsymbol{Y}) \end{bmatrix} \Longleftarrow \begin{bmatrix} \mathsf{vec}(\boldsymbol{X}) \\ \mathsf{vec}(\boldsymbol{Y}) \end{bmatrix} - \eta \begin{bmatrix} \nabla^2_{\boldsymbol{X},\boldsymbol{X}} f & \boldsymbol{0} \\ \boldsymbol{0} & \nabla^2_{\boldsymbol{Y},\boldsymbol{Y}} f \end{bmatrix}^{-1} \begin{bmatrix} \mathsf{vec}(\nabla_{\boldsymbol{X}} f) \\ \mathsf{vec}(\nabla_{\boldsymbol{Y}} f) \end{bmatrix}.$$

The preconditioners are chosen as the inverse of the block diagonal approximation of the Hessian to low-rank matrix factorization.

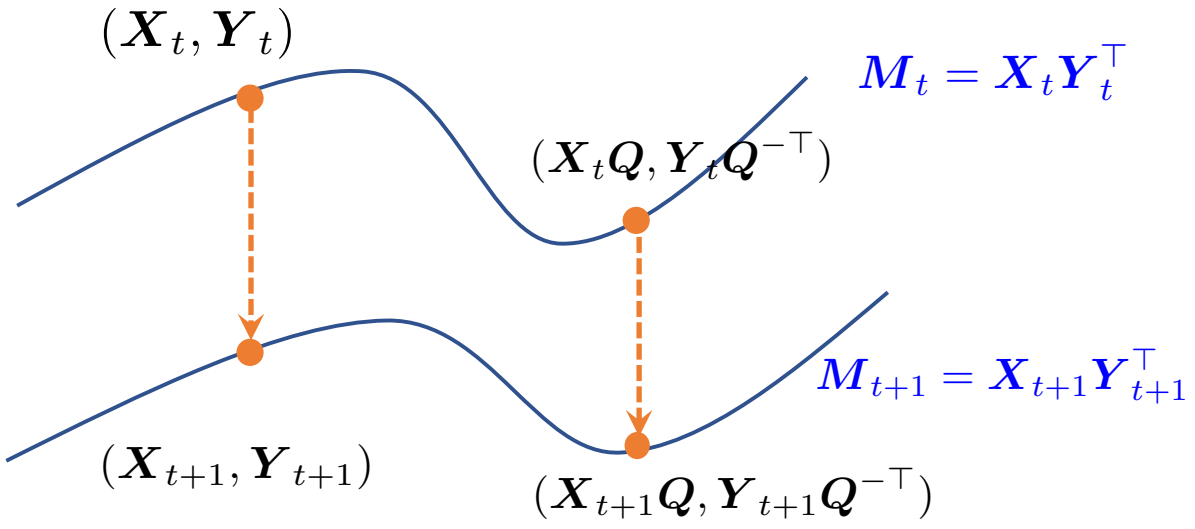
A closer look at ScaledGD

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



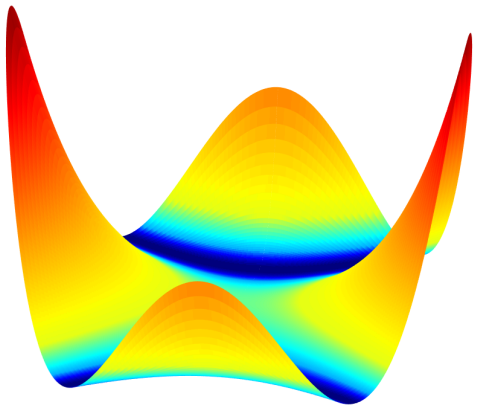
A closer look at ScaledGD

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



New distance metric as Lyapunov function:

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



+ a careful trajectory-based analysis

Theoretical guarantees of ScaledGD

Theorem (Tong, Ma and Chi, 2020)

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, 2020)

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

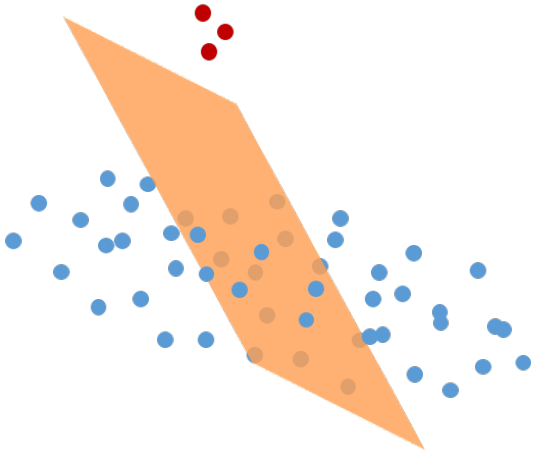
$$\|\boldsymbol{X}_t\boldsymbol{Y}_t^\top - \boldsymbol{M}\|_{\mathrm{F}} \lesssim \varepsilon \cdot \sigma_{\min}(\boldsymbol{M})$$

- **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!

Code available at `https://github.com/Titan-Tong/ScaledGD`

What about the run time?

The run time of ScaledGD is rather competitive, with additional suitability for parallel implementation.

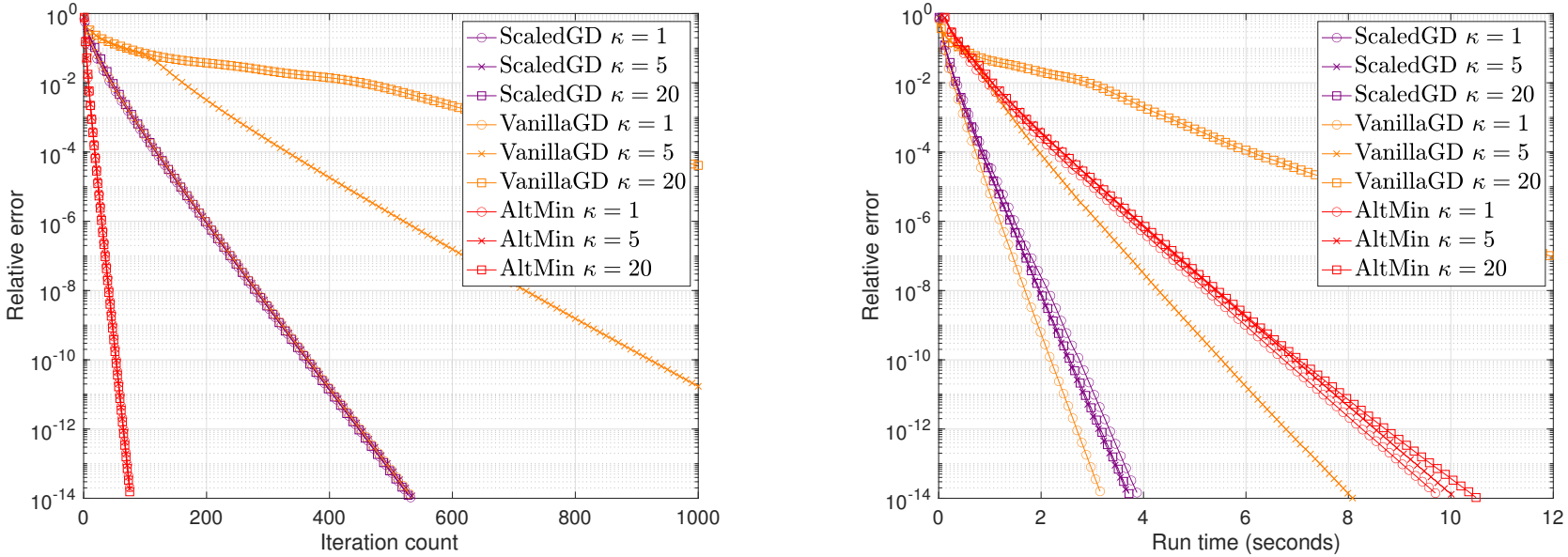
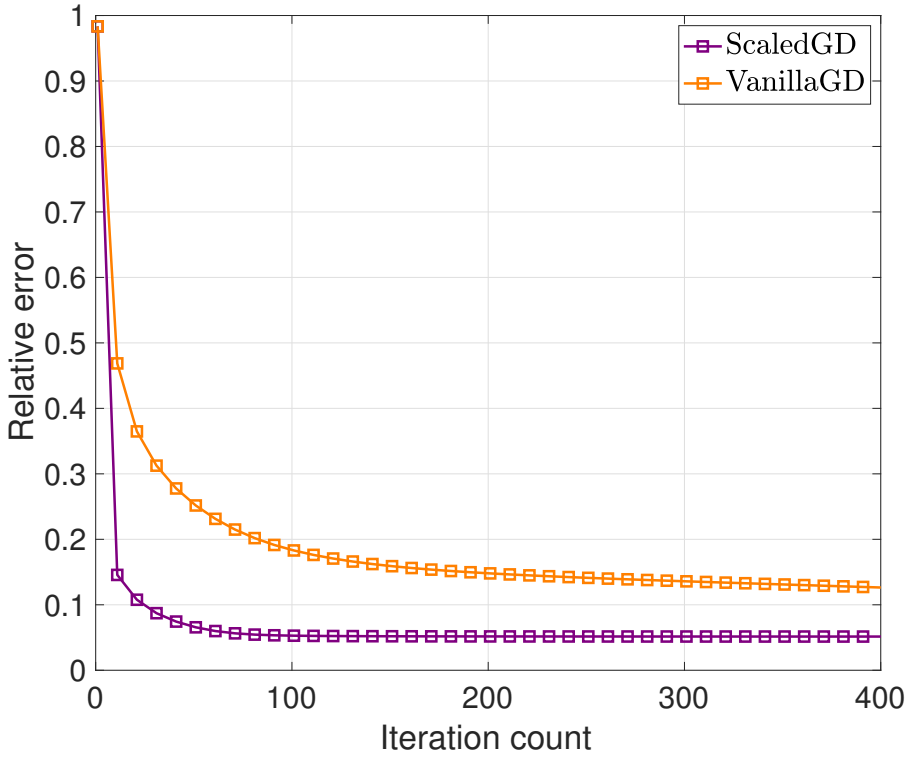


Figure: Run time for matrix completion with $n = 1000$, $p = 0.2$, $r = 50$.

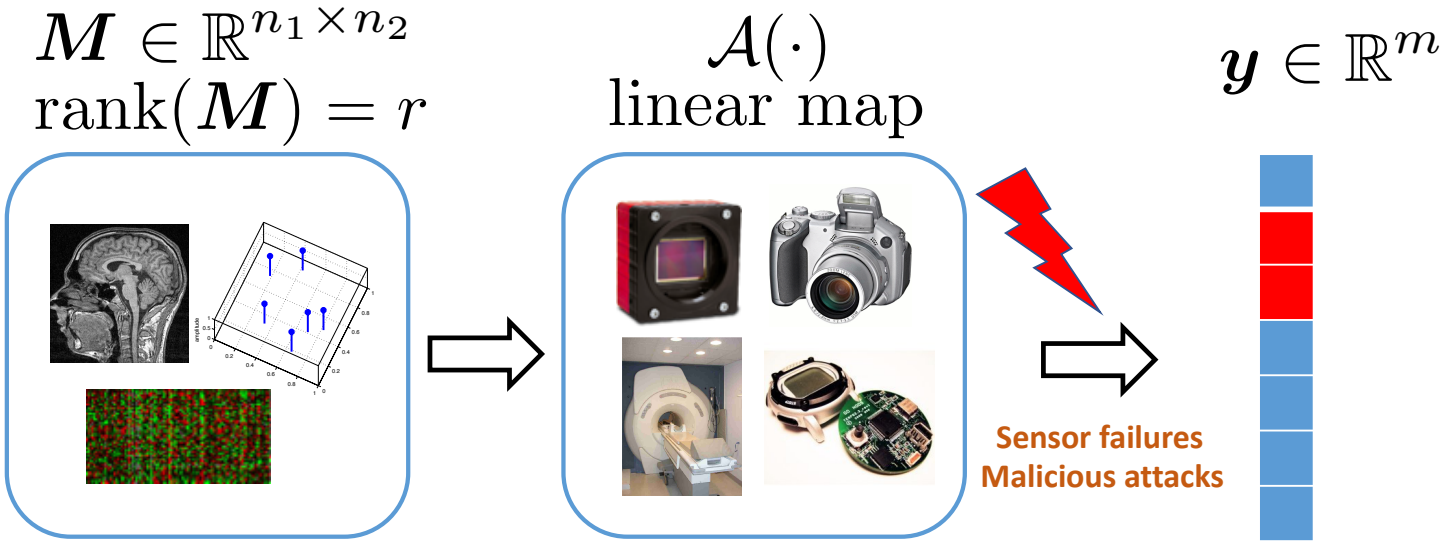
Numerical stability

ScaledGD converges faster than GD in a small number of iterations (they eventually reach the same accuracy).



Robustness to outliers and corruptions?

Outlier-corrupted low-rank matrix sensing



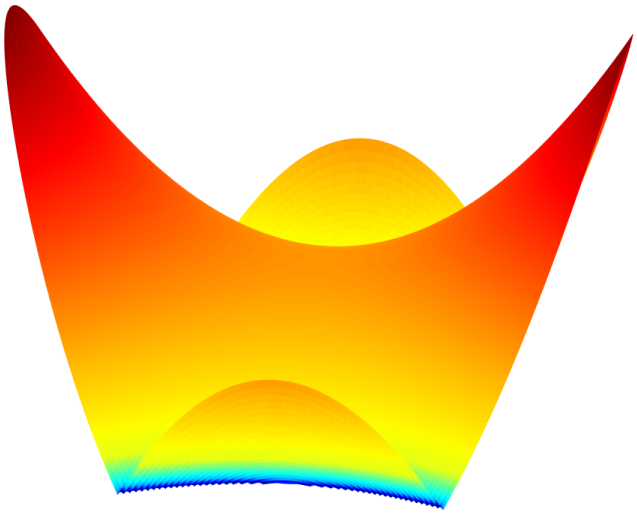
$$\boldsymbol{y} \;=\; \mathcal{A}(\boldsymbol{M}) + \underbrace{\boldsymbol{s}}_{\text{outliers}}, \quad \mathcal{A}(\boldsymbol{M}) = \{\langle \boldsymbol{A}_i, \boldsymbol{M} \rangle\}_{i=1}^m$$

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

Dealing with outliers: subgradient methods

Least absolute deviation (LAD):

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



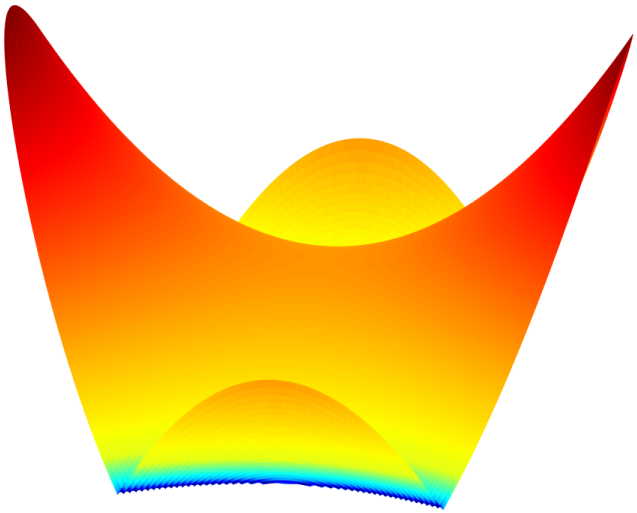
- **Median-truncated spectral initialization:** (Li et.al.'19).
- **Subgradient iterations:** (Charisopoulos et.al.'19; Li et al'18)

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

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)

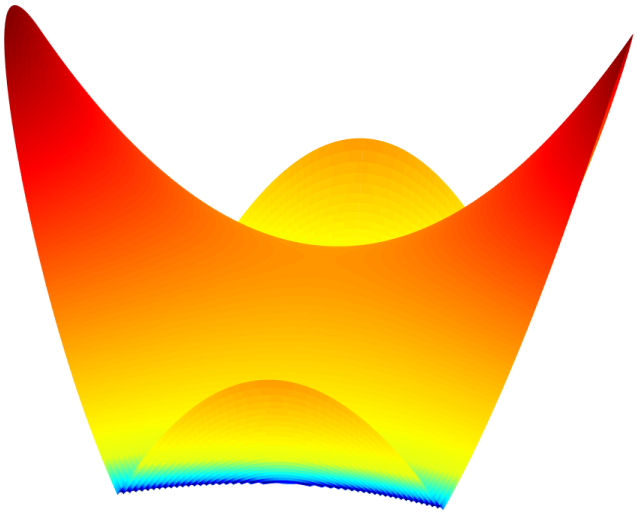
$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t - \eta_t \, \partial_{\boldsymbol{X}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t)$$
$$\boldsymbol{Y}_{t+1} = \boldsymbol{Y}_t - \eta_t \, \partial_{\boldsymbol{Y}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t)$$

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.

Stepsize schedule

Polyak's stepsize:

$$\eta_t = \frac{f(\boldsymbol{X}_t\boldsymbol{Y}_t^\top) - f(\boldsymbol{M})}{\|\partial_{\boldsymbol{X}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t)(\boldsymbol{Y}_t^\top \boldsymbol{Y}_t)^{-1/2}\|_{\mathrm{F}}^2 + \|\partial_{\boldsymbol{Y}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t)(\boldsymbol{X}_t^\top \boldsymbol{X}_t)^{-1/2}\|_{\mathrm{F}}^2}.$$

- Use the distance concerted with preconditioners.
- Require the knowledge of the optimal value $f(\boldsymbol{M})$.

Stepsize schedule

Polyak's stepsize:

$$\eta_t = \frac{f(\boldsymbol{X}_t\boldsymbol{Y}_t^\top) - f(\boldsymbol{M})}{\|\partial_{\boldsymbol{X}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t)(\boldsymbol{Y}_t^\top \boldsymbol{Y}_t)^{-1/2}\|_{\mathrm{F}}^2 + \|\partial_{\boldsymbol{Y}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t)(\boldsymbol{X}_t^\top \boldsymbol{X}_t)^{-1/2}\|_{\mathrm{F}}^2}.$$

- Use the distance concerted with preconditioners.
- Require the knowledge of the optimal value $f(\boldsymbol{M})$.

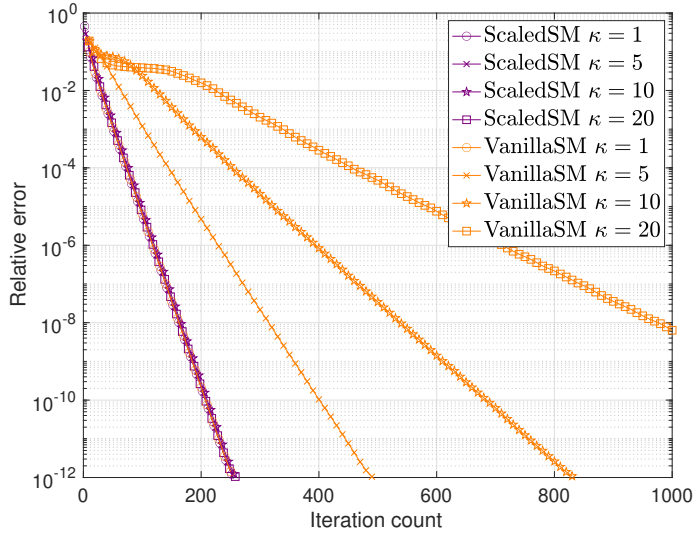
Geometrically decaying stepsize:

$$\eta_t = \frac{\lambda q^t}{\sqrt{\|\partial_{\boldsymbol{X}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t)(\boldsymbol{Y}_t^\top \boldsymbol{Y}_t)^{-1/2}\|_{\mathrm{F}}^2 + \|\partial_{\boldsymbol{Y}} f(\boldsymbol{X}_t, \boldsymbol{Y}_t)(\boldsymbol{X}_t^\top \boldsymbol{X}_t)^{-1/2}\|_{\mathrm{F}}^2}}$$

- Parameters λ, q need to be tuned.
- Perform similarly as Polyak's stepsize under well-tuned λ, q.

Performance guarantees

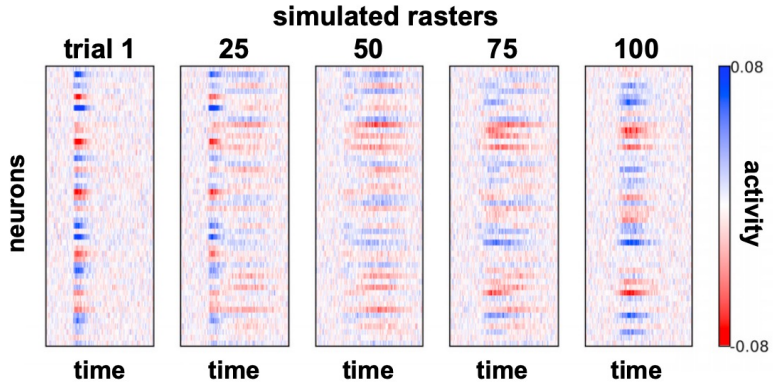
	matrix sensing	**quadratic sensing**
Subgradient Method (Charisopoulos et al, '19)	$\frac{\kappa}{(1-2\alpha)^2}\log\frac{1}{\epsilon}$	$\frac{r\kappa}{(1-2\alpha)^2}\log\frac{1}{\epsilon}$
ScaledSM (Tong, Ma, Chi, '20)	$\frac{1}{(1-2\alpha)^2}\log\frac{1}{\epsilon}$	$\frac{r}{(1-2\alpha)^2}\log\frac{1}{\epsilon}$



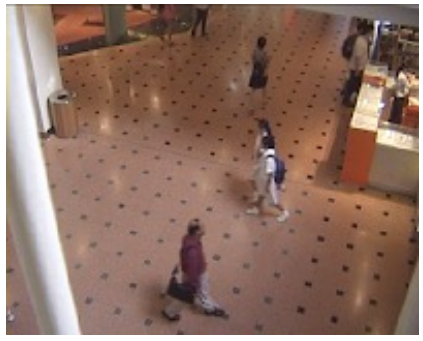
Robustness to both ill-conditioning and adversarial corruptions!

Generalization to tensors

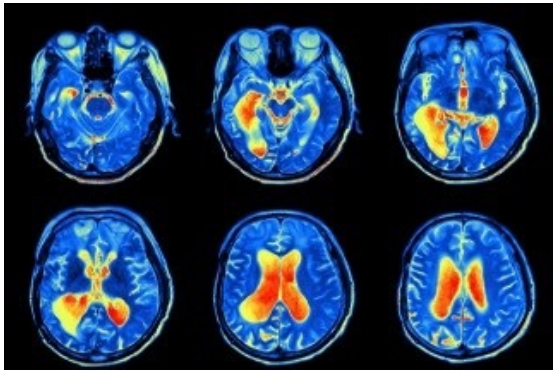
Capturing multi-way interactions by tensors



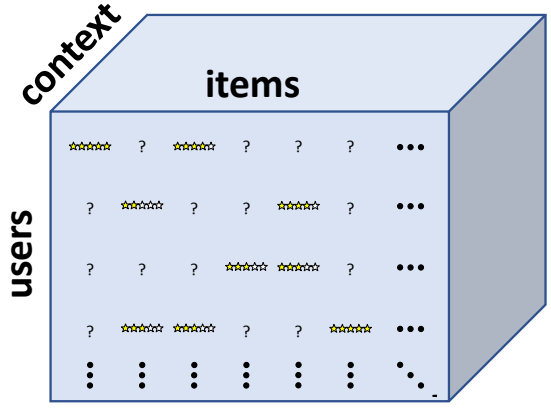
neural recordings



video surveillance



neuroimaging



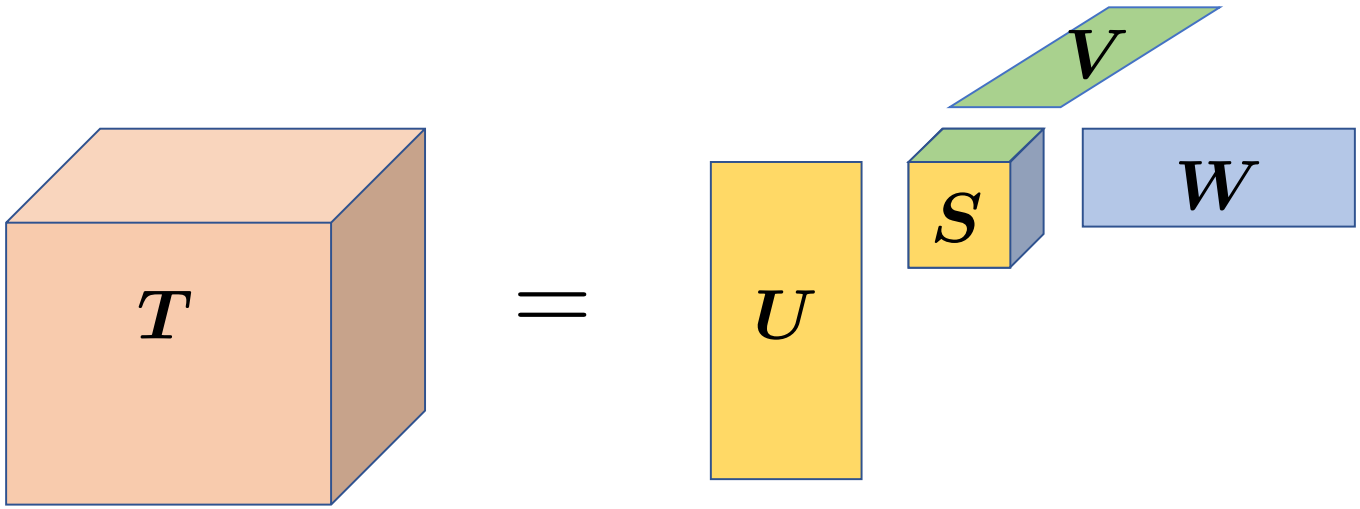
recommendation system

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?

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

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?

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

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 $\boldsymbol{U}$:

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

How to construct scaled gradients for tensors?

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

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 $\boldsymbol{S}$:

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

ScaledGD for ill-conditioned low-rank tensor estimation

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

Scaled gradient iterations:

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

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., 2021)

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., 2021)

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

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

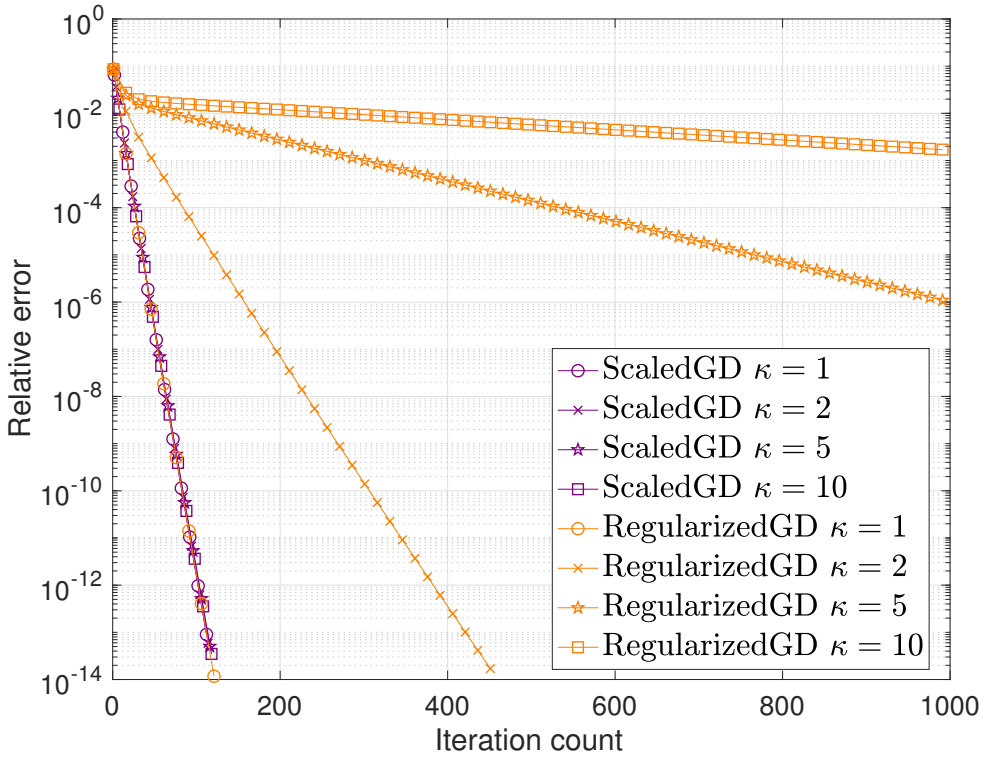
- **Computational:** *within* $O\big(\log \frac{1}{\varepsilon}\big)$ *iterations;*
- **Statistical:** *as long as the sample complexity satisfies*

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

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

Numerical evidence

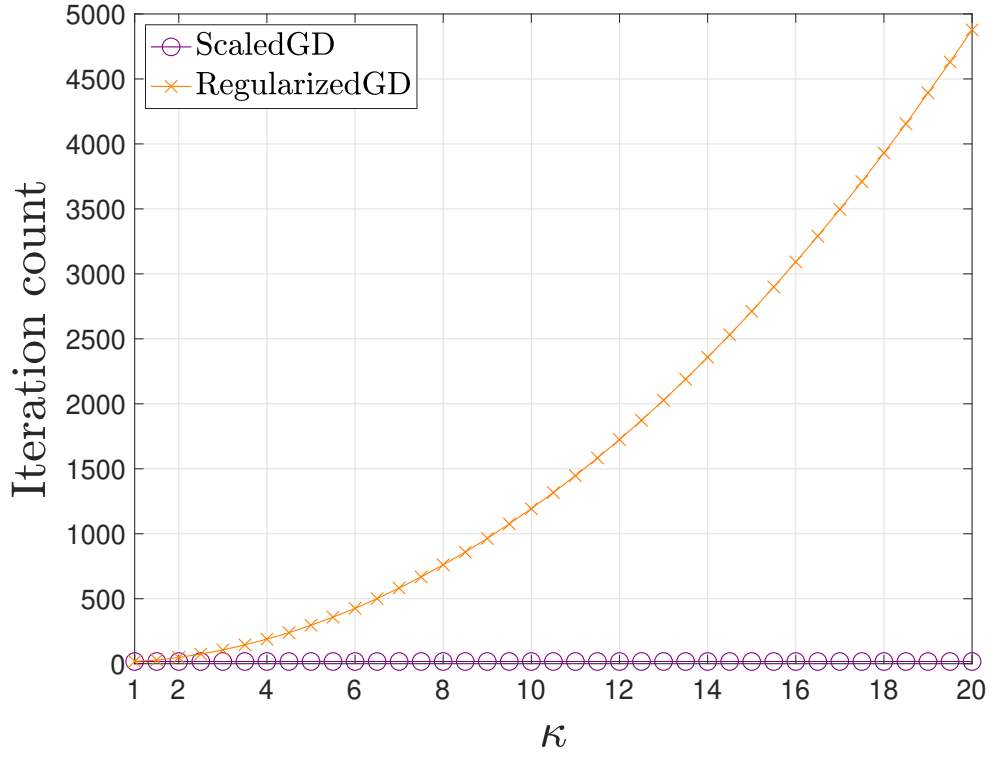
$$\min_{\boldsymbol{F}=(\boldsymbol{U},\boldsymbol{V},\boldsymbol{W},\boldsymbol{S})} \; f(\boldsymbol{F}) = \frac{1}{2}\left\|\mathcal{P}_{\Omega}((\boldsymbol{U},\boldsymbol{V},\boldsymbol{W})\cdot\boldsymbol{S}) - \boldsymbol{T})\right\|_{\mathrm{F}}^2$$



The benefit of ScaledGD is even more evident for tensors!

Numerical evidence

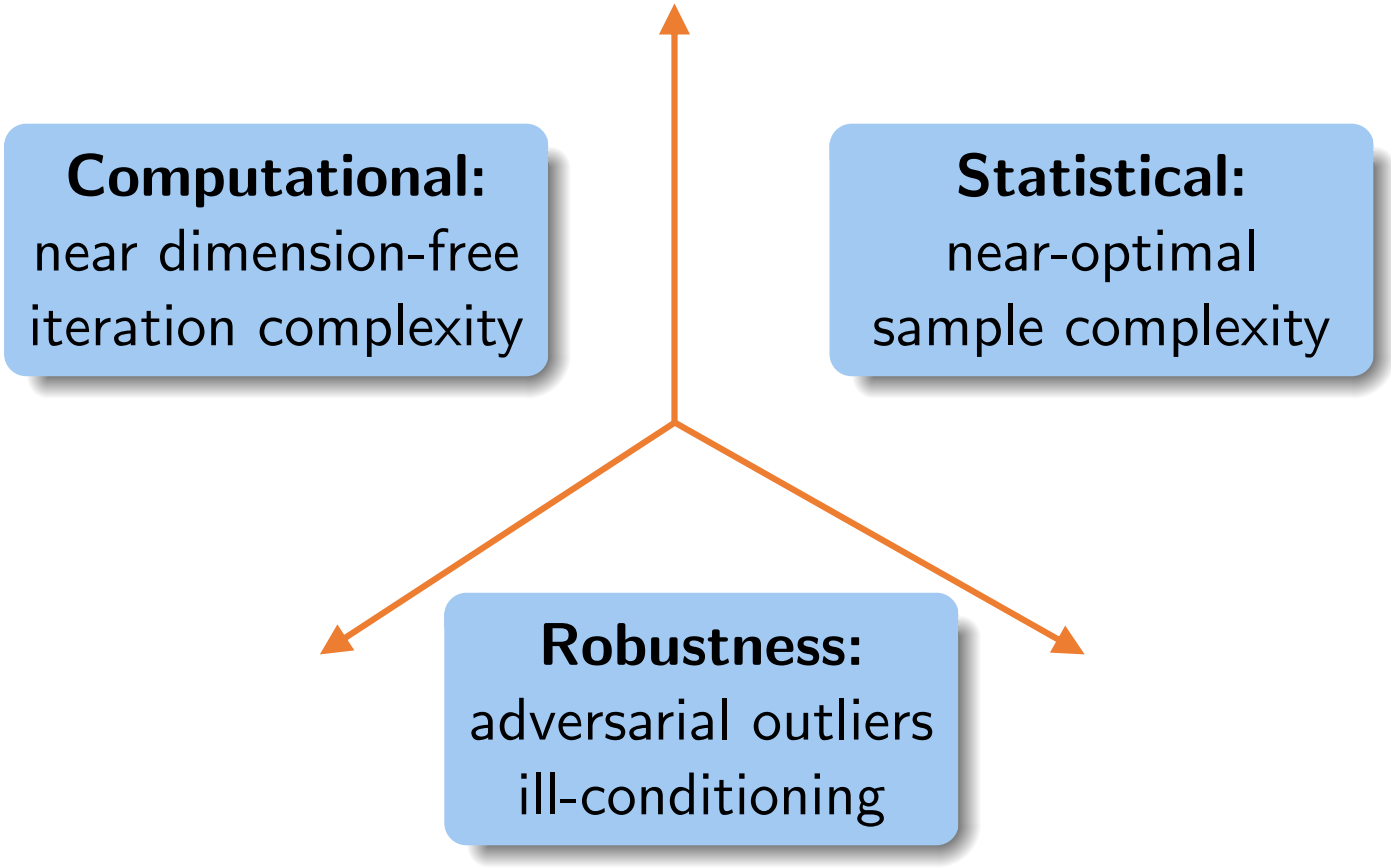
$$\min_{\boldsymbol{F}=(\boldsymbol{U},\boldsymbol{V},\boldsymbol{W},\boldsymbol{S})} \; f(\boldsymbol{F}) = \frac{1}{2}\left\|\mathcal{P}_\Omega((\boldsymbol{U},\boldsymbol{V},\boldsymbol{W})\cdot\boldsymbol{S}) - \boldsymbol{T}\right\|_{\mathrm{F}}^2$$



The benefit of ScaledGD is even more evident for tensors!

Concluding remarks

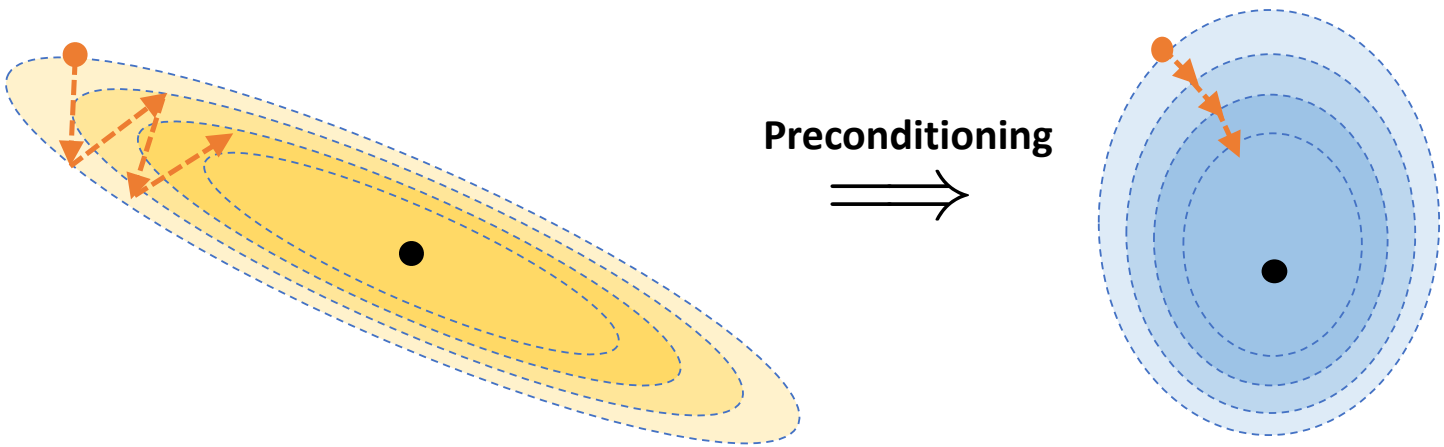
Bridging the theory-practice gap



Nonconvex low-rank matrix and tensor estimation:

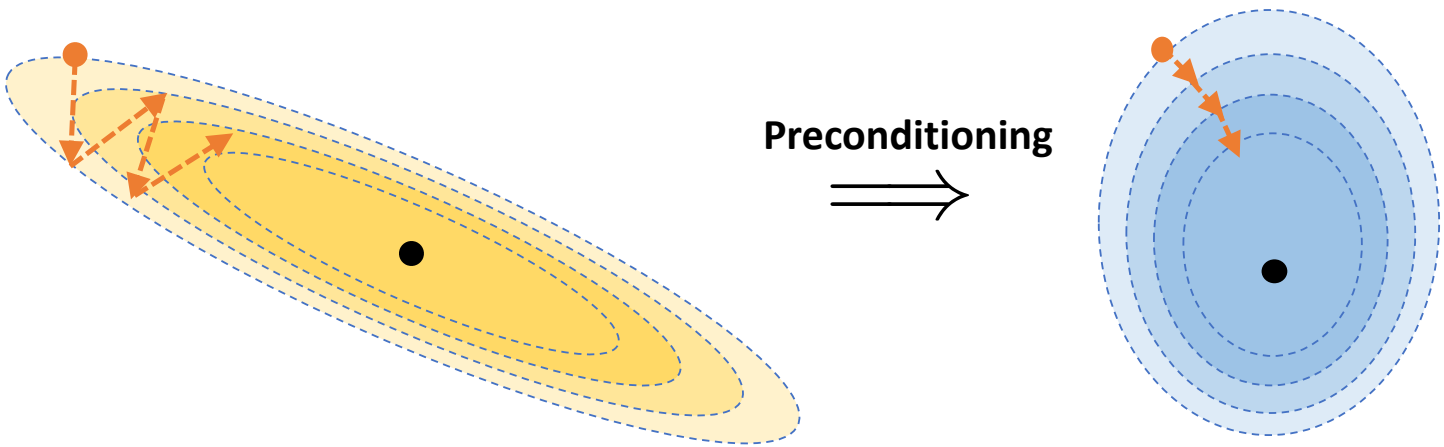
- identification and exploitation of benign geometric properties;
- analyzing iterate trajectories beyond black-box optimization;
- simple variants of GD lead to robust and accelerated convergence.

Preconditioning helps!



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

Preconditioning helps!



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

Promising directions:

- streaming/stochastic variants of ScaledGD;
- applications of ScaledGD to other problems.

Selected References

Overview:

1. Nonconvex Optimization Meets Low-Rank Matrix Factorization: An Overview, Y. Chi, Y. M. Lu and Y. Chen, *IEEE Trans. on Signal Processing*, 2019.
2. Spectral Methods for Data Science: A Statistical Perspective", Y. Chen, Y. Chi, J. Fan and C. Ma, *Foundations and Trends in Machine Learning*, 2021.
3. Harnessing Structures in Big Data via Guaranteed Low-Rank Matrix Estimation, Y. Chen and Y. Chi, *IEEE Signal Processing Magazine*, 2018.

Geometry of factored gradient descent:

1. Implicit Regularization for Nonconvex Statistical Estimation, C. Ma, K. Wang, Y. Chi and Y. Chen, *Foundations of Computational Mathematics*, 2020.
2. Beyond Procrustes: Balancing-free Gradient Descent for Asymmetric Low-Rank Matrix Sensing, C. Ma, Y. Li and Y. Chi, *IEEE Trans. on Signal Processing*, 2021.
3. Nonconvex Matrix Factorization from Rank-One Measurements, Y. Li, C. Ma, Y. Chen, and Y. Chi, *IEEE Trans. on Information Theory*, 2020.

Selected References

Robustness to ill-conditioning:

1. Accelerating ill-conditioned low-rank matrix estimation via scaled gradient descent, T. Tong, C. Ma, and Y. Chi, *Journal of Machine Learning Research*, 2021.
2. Scaling and scalability: Provable nonconvex low-rank tensor estimation from incomplete measurements, T. Tong, C. Ma, A. Prater-Bennette, E. Tripp, and Y. Chi, *arXiv preprint arXiv:2104.14526*, 2021.

Robustness to adversarial outliers:

1. Low-rank matrix recovery with scaled subgradient methods: Fast and robust convergence without the condition number, T. Tong, C. Ma, and Y. Chi, *IEEE Trans. on Signal Processing*, 2021.
2. Non-convex low-rank matrix recovery with arbitrary outliers via median-truncated gradient descent, Y. Li, Y. Chi, H. Zhang and Y. Liang, *Information and Inference: A Journal of the IMA*, 2020.

Thanks!

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

Advertisement: 2022 IEEE SPS Distinguished Lecturer

- nonconvex optimization for statistical estimation
- convex optimization for superresolution
- reinforcement learning
- communication-efficient distributed learning

Please consider inviting me to your local chapter!