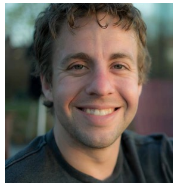
Short-Flat Decompositions and Faster Algorithms for Linear Inverse Problems

Kevin Tian (UT Austin)

Simons Institute Optimization and Algorithm Design Workshop

Based on joint work with:



Jonathan Kelner (MIT), Jerry Li (MSR), Allen Liu (MIT), Aaron Sidford (Stanford)

Roadmap

- Overview
  - Sparse recovery: SOTA and what's new
  - Matrix completion: SOTA and what's new
- Sparse recovery
  - Short-flat decompositions
  - Projected gradient descent
- Matrix completion

Sparse recovery

$$\mathbf{A}x^\star = b$$
$$\mathbf{A} \in \mathbb{R}^{n \times d}$$

$$\begin{pmatrix} — & a_1 & — \\ — & a_2 & — \\ & \vdots & \\ — & a_n & — \end{pmatrix} \begin{pmatrix} [x^\star]_1 \\ [x^\star]_2 \\ \vdots \\ [x^\star]_d \end{pmatrix} = \begin{pmatrix} [b]_1 \\ [b]_2 \\ \vdots \\ [b]_n \end{pmatrix}$$

Underconstrained regime: $n \ll d$

Clearly impossible in the worst case. Need to assume more!

Sparse recovery

$$\mathbf{A}x^\star = b$$
$$\mathbf{A} \in \mathbb{R}^{n\times d}$$

Standard assumption: $x^\star$ is s-sparse

$$\begin{pmatrix} — & a_1 & — \\ — & a_2 & — \\ & \vdots & \\ — & a_n & — \end{pmatrix} \begin{pmatrix} [x^\star]_1 \\ [x^\star]_2 \\ \vdots \\ [x^\star]_d \end{pmatrix} = \begin{pmatrix} [b]_1 \\ [b]_2 \\ \vdots \\ [b]_n \end{pmatrix}$$

Sparse recovery

$$\mathbf{A}x^\star = b$$
$$\mathbf{A} \in \mathbb{R}^{n \times d}$$
$$x^\star \text{ is } s\text{-sparse}$$

$$\begin{pmatrix} — & a_1 & — \\ — & a_2 & — \\ & \vdots & \\ — & a_n & — \end{pmatrix} \begin{pmatrix} [x^\star]_1 \\ [x^\star]_2 \\ \vdots \\ [x^\star]_d \end{pmatrix} = \begin{pmatrix} [b]_1 \\ [b]_2 \\ \vdots \\ [b]_n \end{pmatrix}$$

Assume: $\mathbf{A}$ satisfies RNP (no sparse vectors in kernel)

Algo: $$\boxed{\min_{\mathbf{A}x=b} \|x\|_1}$$

Polynomial time algorithms
(convex programming)

Sparse recovery

$$\mathbf{A}x^\star = b$$
$$\mathbf{A} \in \mathbb{R}^{n\times d}$$
$$x^\star \text{ is } s\text{-sparse}$$

$$\begin{pmatrix} — & a_1 & — \\ — & a_2 & — \\ & \vdots & \\ — & a_n & — \end{pmatrix} \begin{pmatrix} [x^\star]_1 \\ [x^\star]_2 \\ \vdots \\ [x^\star]_d \end{pmatrix} = \begin{pmatrix} [b]_1 \\ [b]_2 \\ \vdots \\ [b]_n \end{pmatrix}$$

Assume: $\mathbf{A}$ satisfies RNP (no sparse vectors in kernel)

Algo: $\boxed{\min_{\mathbf{A}x=b} \|x\|_1}$

Polynomial time algorithms
(convex programming)

Upshot: very flexible + general!

- Extends to noisy settings
- Essentially minimal assumptions

…potentially expensive in high-dim.

Sparse recovery

$$\mathbf{A}x^\star = b$$
$$\mathbf{A} \in \mathbb{R}^{n \times d}$$
$$x^\star \text{ is } s\text{-sparse}$$

$$\begin{pmatrix} — & a_1 & — \\ — & a_2 & — \\ & \vdots & \\ — & a_n & — \end{pmatrix} \begin{pmatrix} [x^\star]_1 \\ [x^\star]_2 \\ \vdots \\ [x^\star]_d \end{pmatrix} = \begin{pmatrix} [b]_1 \\ [b]_2 \\ \vdots \\ [b]_n \end{pmatrix}$$

Assume: $\mathbf{A}$ satisfies RNP

(**A** is near-isometry on sparse vectors) $\mathbf{A}$ satisfies RIP

$$\Omega(\|v\|_2) = \|\mathbf{A}v\|_2 = O(\|v\|_2)$$
$$\text{for all } \|v\|_0 = O(s)$$

Algo: $$\boxed{\min_{\mathbf{A}x=b} \|x\|_1}$$

Polynomial time algorithms
(convex programming)

Nearly-linear time algorithms

Sparse recovery

$$\mathbf{A}x^\star = b$$
$$\mathbf{A} \in \mathbb{R}^{n\times d}$$
$$x^\star \text{ is } s\text{-sparse}$$

$$\begin{pmatrix} — & a_1 & — \\ — & a_2 & — \\ & \vdots & \\ — & a_n & — \end{pmatrix} \begin{pmatrix} [x^\star]_1 \\ [x^\star]_2 \\ \vdots \\ [x^\star]_d \end{pmatrix} = \begin{pmatrix} [b]_1 \\ [b]_2 \\ \vdots \\ [b]_n \end{pmatrix}$$

Assume: $\mathbf{A}$ satisfies RNP

($\mathbf{A}$ is near-isometry on sparse vectors) $\mathbf{A}$ satisfies RIP

Algo: $\boxed{\min_{\mathbf{A}x=b} \|x\|_1}$

Polynomial time algorithms (convex programming)

Greedy: Pursuit, OMP
Non-convex: IHT, CoSaMP
Convex: Projected GD

Nearly-linear time algorithms

Sparse recovery

- Theory: both work under standard generative models
- Practice: fast methods much more brittle [Davenport, Needell, Wakin '13], [Jain, Tewari, Kar '14], [Polania, Carrillo, Blanco-Velasco, Barner '14], [Zhang, Wei, Wei, Li, Liu, Liu '16], …

What's going on?

Assume: $\mathbf{A}$ satisfies RNP

$\mathbf{A}$ satisfies RIP

Algo: $\min_{\mathbf{A}x=b} \|x\|_1$

Greedy: Pursuit, OMP
Non-convex: IHT, CoSaMP
Convex: Projected GD

Polynomial time algorithms
(convex programming)

Nearly-linear time algorithms

Sparse recovery

Theory vs. practice: *what's going on?*

Not broken by semi-random adversary!

Assume: $\mathbf{A}$ satisfies RNP

Algo: $$\min_{\mathbf{A}x=b} ||x||_1$$

Polynomial time algorithms
(convex programming)

Easily broken by semi-random adversary!

$\mathbf{A}$ satisfies RIP

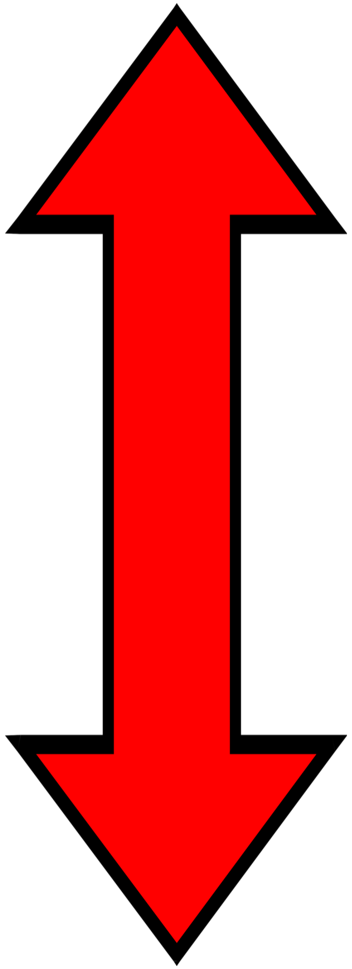
Greedy: Pursuit, OMP
Non-convex: IHT, CoSaMP
Convex: Projected GD

Nearly-linear time algorithms

Semi-random models

Easy! Polynomial-time (very fast?)

"Fully random"



"Worst-case"

Hard! (NP-hard, info-impossible?)

Semi-random models

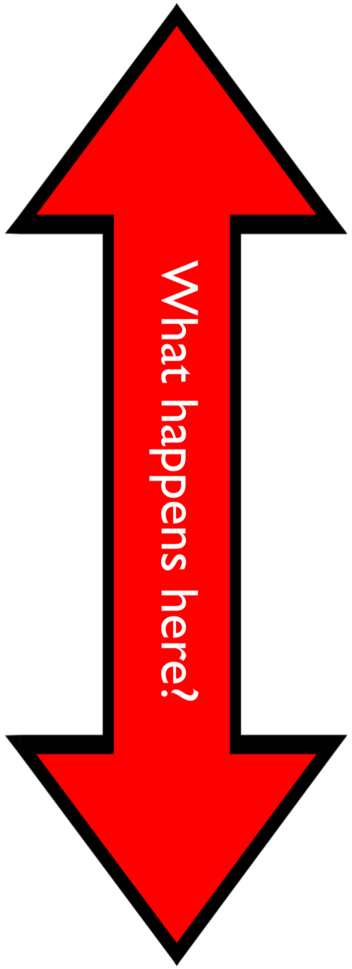
Philosophy

- "Beyond best-case analysis"
- Main q: design algorithms which are *robust* to input assumption violations?

If everything works when life is easy, choose the algorithm that is most robust to assumptions.

Easy! Polynomial-time (very fast?)

"Fully random"

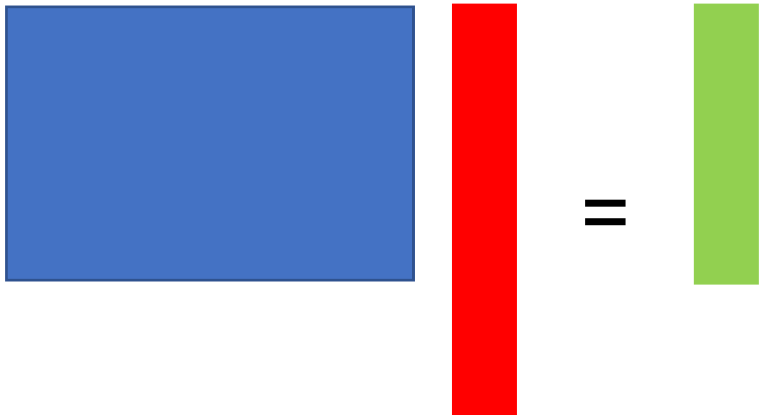


"Worst-case"

Hard! (NP-hard, info-impossible?)

Semi-random sparse recovery

$$\mathbf{A}x^\star = b$$
$$\mathbf{A} \in \mathbb{R}^{n \times d}$$



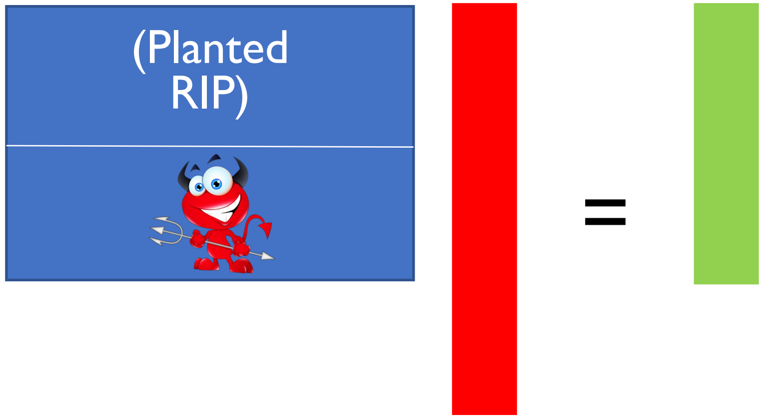
Nearly-linear time algos:
assume restricted
isometry property (RIP)

$$\Omega(\|v\|_2) = \|\mathbf{A}v\|_2 = O(\|v\|_2)$$

$$\text{for all } \|v\|_0 = O(s)$$

Semi-random sparse recovery

$$\mathbf{A}x^\star = b$$
$$\mathbf{A} \in \mathbb{R}^{n\times d}$$



Basic semi-random adversary:

1. Take RIP matrix **G**
2. Augment with additional "consistent" measurements
3. Shuffle matrix, present **A**

RIP: for all $\|v\|_0 = O(s)$

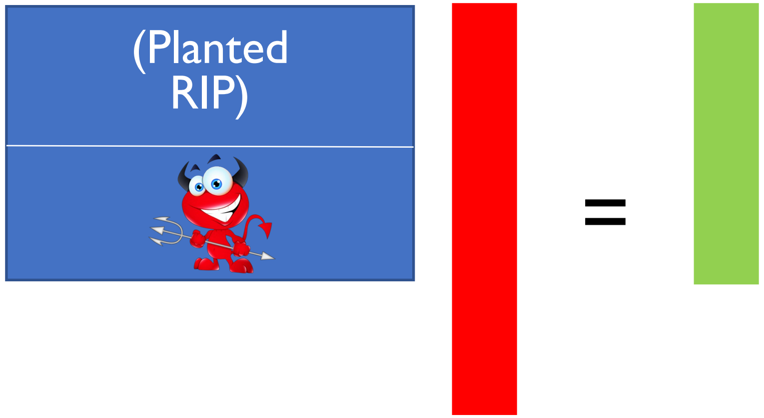
$$\Omega(\|v\|_2) = \|\mathbf{A}v\|_2 = O(\|v\|_2)$$

Semi-random sparse recovery

$$\min_{\mathbf{A}x=b} \|x\|_1$$

(polynomial time)

$$\mathbf{A}x^\star = b$$
$$\mathbf{A} \in \mathbb{R}^{n\times d}$$



Basic semi-random adversary:

1. Take RIP matrix **G**
2. Augment with additional "consistent" measurements
3. Shuffle matrix, present **A**

RIP: for all $\|v\|_0 = O(s)$

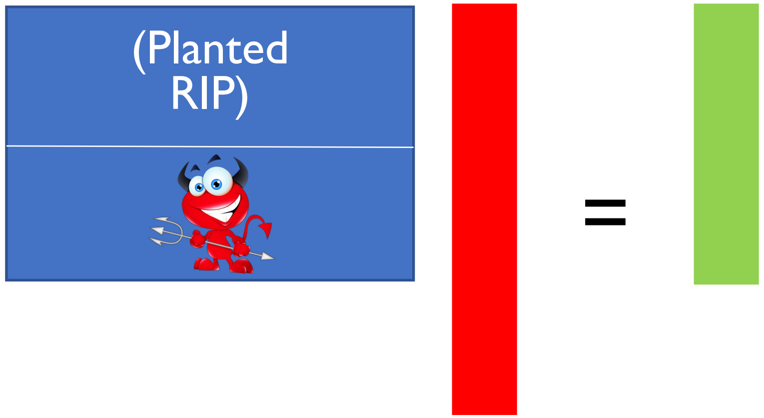
$$\Omega(\|v\|_2) = \|\mathbf{A}v\|_2 = O(\|v\|_2)$$

Semi-random sparse recovery

$$\min_{\mathbf{A}x=b} \|x\|_1$$

(polynomial time)

$$\mathbf{A}x^\star = b$$
$$\mathbf{A} \in \mathbb{R}^{n\times d}$$



Basic semi-random adversary:

1. Take RIP matrix **G**
2. Augment with additional "consistent" measurements
3. Shuffle matrix, present **A**

RIP: for all $\|v\|_0 = O(s)$

$$\Omega(\|v\|_2) = \|\mathbf{A}v\|_2 = O(\|v\|_2)$$

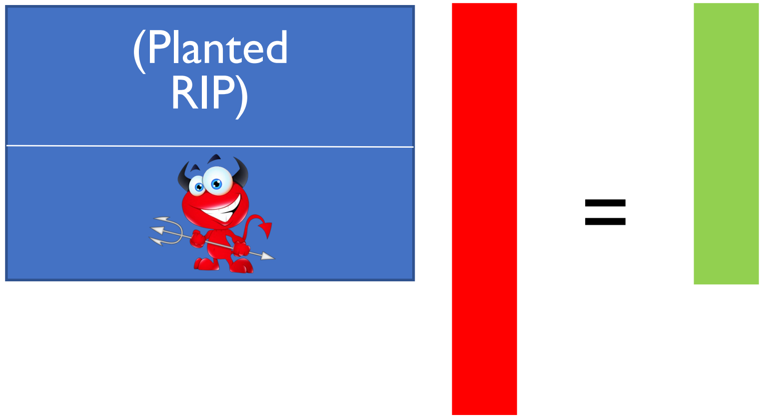
Fast algorithms?

1. Many greedy/non-convex iterative methods immediately fail (explicit counterexamples)

2. Convex iterative methods' analyses depend on *restricted conditioning*, easy to break

Our basic result

$$\mathbf{A}x^{\star} = b$$
$$\mathbf{A} \in \mathbb{R}^{n \times d}$$



pRIP adversary:

1. Take RIP matrix **G**
2. Augment with additional "consistent" measurements
3. Shuffle matrix, present **A**

Theorem [Kelner, Li, Liu, Sidford, Tian '23]:

Can solve linear systems in entrywise-bounded* pRIP **A** in time

$$\widetilde{O}(nd)$$

*satisfied by standard RIP constructions, e.g. Gaussian, subsampled Fourier/Hadamard matrices

Our general result

$$\|x - x^\star\|_2^2 = O\left(\frac{1}{m}\|\xi_{\text{top } m}\|_2^2\right)$$

$$\mathbf{A}x^\star + \xi = b$$
$$\mathbf{A} \in \mathbb{R}^{n \times d}$$

(Weighted RIP)

$\approx$

wRIP (>pRIP) adversary:

1. Exists diagonal reweighting **W** such that $\mathbf{A}^\top\mathbf{W}\mathbf{A}$ is RIP and **A** is entrywise bounded
2. We define $m := \frac{\|w\|_1}{\|w\|_\infty}$

Theorem [Kelner, Li, Liu, Sidford, Tian '23]:

Can solve noisy linear systems in entrywise-bounded wRIP **A** optimally in time

$$\widetilde{O}\left(d \cdot \frac{ns}{m}\right)$$

In pRIP model:

- When all of **A** is RIP and $n = m \gg s$, sublinear
- When **A** contains minimum $m \approx s$, linear

Matrix "sparse recovery"

$$\langle \mathbf{A}_i, \mathbf{X}^\star \rangle = b_i \quad \forall i \in [n]$$

$$\{\mathbf{A}_i\}_{i \in [n]}, \mathbf{X}^\star \in \mathbb{R}^{d \times d}$$

Matrix "sparse recovery"

$$\langle \mathbf{A}_i, \mathbf{X}^\star \rangle = b_i \quad \forall i \in [n]$$

$$\gtrsim dr$$

$$\{\mathbf{A}_i\}_{i \in [n]}, \mathbf{X}^\star \in \mathbb{R}^{d \times d}$$

Standard assumption: $\mathbf{X}^\star$ is rank-r

Matrix "sparse recovery"

$$\langle \mathbf{A}_i, \mathbf{X}^\star \rangle = b_i \quad \forall i \in [n]$$

$$\gtrsim dr$$

$$\{\mathbf{A}_i\}_{i \in [n]}, \mathbf{X}^\star \in \mathbb{R}^{d \times d}$$

Standard assumption: $\mathbf{X}^\star$ is rank-r

Poster

Robust Matrix Sensing in the Semi-Random Model

Xing Gao · Yu Cheng

Great Hall & Hall B1+B2 #1720

Theorem 1.1 (Semi-Random Matrix Sensing (Informal)). *Given a set of wRIP sensing matrices $\{A_i\}_{i=1}^n$ and observation vector $b = \mathcal{A}[X^*]$, we can compute X such that $\|X - X^*\|_F \le \epsilon$ with high probability in time $\widetilde{O}(nd^{\omega+1})$, where n is the number of sensing matrices and $O(d^\omega)$ represents the matrix multiplication time for $X \in \mathbb{R}^{d \times d}$.*

Ask Xing and Yu @ NeurIPS '23!

Matrix completion

$$\langle \mathbf{A}_i, \mathbf{X}^\star \rangle = b_i \quad \forall i \in [n]$$

$$\gtrsim dr$$

$$\{\mathbf{A}_i\}_{i\in[n]}, \mathbf{X}^\star \in \mathbb{R}^{d\times d}$$

Standard assumption: $\mathbf{X}^\star$ is rank-r

$$\mathbf{A}_i = e_{j_i} e_{k_i}^\top \; \forall i \in [n]$$

2						
		3				
					10	
		-7				
				2		

Matrix completion

$$\langle \mathbf{A}_i, \mathbf{X}^\star \rangle = b_i \quad \forall i \in [n]$$

$$\gtrsim dr$$

$$\{\mathbf{A}_i\}_{i \in [n]}, \mathbf{X}^\star \in \mathbb{R}^{d \times d}$$

Standard assumption: $\mathbf{X}^\star$ is rank-r

"RIP"-type assumption impossible: dodge observations with single spike

$$\mathbf{A}_i = e_{j_i} e_{k_i}^\top \; \forall i \in [n]$$

2						
		3				
					10	
		-7				
				2		

Matrix completion

$$\langle \mathbf{A}_i, \mathbf{X}^\star \rangle = b_i \quad \forall i \in [n]$$

$$\gtrsim dr$$

$$\{\mathbf{A}_i\}_{i\in[n]}, \mathbf{X}^\star \in \mathbb{R}^{d\times d}$$

Standard assumptions:

$$\mathbf{X}^\star = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$$

$$\mathbf{U}, \mathbf{V} \in \mathbb{R}^{d\times r} \text{ are “spread”}$$

Matrix completion

$$\langle \mathbf{A}_i, \mathbf{X}^\star \rangle = b_i \quad \forall i \in [n]$$

$$\gtrsim dr$$

$$\{\mathbf{A}_i\}_{i \in [n]}, \mathbf{X}^\star \in \mathbb{R}^{d \times d}$$

Standard assumptions:

$$\mathbf{X}^\star = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$$

$$\mathbf{U}, \mathbf{V} \in \mathbb{R}^{d \times r} \text{ are “spread”}$$

$$\max_{i \in [d]} \|\mathbf{U}^\top e_i\|_2 \lesssim \sqrt{\frac{r}{d}}$$

“Incoherence”

Matrix completion

$$\langle \mathbf{A}_i, \mathbf{X}^\star \rangle = b_i \quad \forall i \in [n]$$

$$\gtrsim dr$$

$$\{\mathbf{A}_i\}_{i\in[n]}, \mathbf{X}^\star \in \mathbb{R}^{d\times d}$$

Standard assumptions:

$$\mathbf{X}^\star = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$$

$$\mathbf{U}, \mathbf{V} \in \mathbb{R}^{d\times r} \text{ are ``spread''}$$

Polynomial time:

[Recht '11]

$$\min_{\mathbf{X}^\star \text{ matches obs}} \|\mathbf{X}^\star\|_1$$

$$\approx dr \text{ samples}$$

Matrix completion

$$\langle \mathbf{A}_i, \mathbf{X}^\star \rangle = b_i \quad \forall i \in [n]$$

$$\gtrsim dr$$

$$\{\mathbf{A}_i\}_{i \in [n]}, \mathbf{X}^\star \in \mathbb{R}^{d \times d}$$

Standard assumptions:

$$\mathbf{X}^\star = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$$

$$\mathbf{U}, \mathbf{V} \in \mathbb{R}^{d \times r} \text{ are “spread”}$$

Polynomial time:

[Recht ‘11]

$$\min_{\mathbf{X}^\star \text{ matches obs}} \|\mathbf{X}^\star\|_1$$

$$\approx dr \text{ samples}$$

Near-linear time:

[Jain-Netrapalli ‘15]

PGD + clipping

$$\approx dr^7 \text{ time}$$

$$\approx dr^5 \text{ samples}$$

Matrix completion

Open questions:

1. Improved "fast" rates?
2. Beyond incoherence?

Polynomial time:

[Recht '11]

$$\min_{\mathbf{X}^\star \text{ matches obs}} \|\mathbf{X}^\star\|_1$$

$$\approx dr \text{ samples}$$

Near-linear time:

[Jain-Netrapalli '15]

PGD + clipping

$$\approx dr^7 \text{ time}$$

$$\approx dr^5 \text{ samples}$$

Matrix completion

Open questions:

1. Improved "fast" rates?
2. Beyond incoherence?
3. Noise-robustness?

$$\text{Observe: } \mathbf{M}^\star + \mathbf{N},\ \|\mathbf{N}\|_{\mathrm{F}} \le \Delta$$

$$\text{Recovery: } \|\mathbf{M} - \mathbf{M}^\star\|_{\mathrm{F}} \le \sqrt{d}\Delta$$

…SOTA even for polynomial time!
[Candes-Plan '10]

Polynomial time:
[Recht '11]

$$\min_{\mathbf{X}^\star \text{ matches obs}} \|\mathbf{X}^\star\|_1$$

$$\approx dr \text{ samples}$$

Near-linear time:
[Jain-Netrapalli '15]

PGD + clipping

$$\approx dr^7 \text{ time}$$

$$\approx dr^5 \text{ samples}$$

Matrix completion

Theorem, Part 1 [Kelner, Li, Liu, Sidford, Tian '23]:

From rank-r $\mathbf{M}^\star \in \mathbb{R}^{d \times d}$ + ($\|\mathbf{N}\|_{\mathsf{F}} \leq \Delta$),
can give $\mathbf{M} \in \mathbb{R}^{d \times d}, S \subseteq [d]$ with:

$$\|[\mathbf{M} - \mathbf{M}^\star]_{S \times S}\|_{\mathrm{F}} = O(\Delta), \; |S| \geq 0.99d$$

Polynomial time:

[Recht '11]

$$\min_{\mathbf{X}^\star \text{ matches obs}} \|\mathbf{X}^\star\|_1$$

$$\approx dr \text{ samples}$$

Near-linear time:

[Jain-Netrapalli '15]

PGD + clipping

$$\approx dr^7 \text{ time}$$

$$\approx dr^5 \text{ samples}$$

Matrix completion

Theorem, Part 1 [Kelner, Li, Liu, Sidford, Tian '23]:

From rank-r $\mathbf{M}^\star \in \mathbb{R}^{d\times d}$ + ($\|\mathbf{N}\|_\mathrm{F} \leq \Delta$),
can give $\mathbf{M} \in \mathbb{R}^{d\times d}, S \subseteq [d]$ with:

$$\|[\mathbf{M} - \mathbf{M}^\star]_{S\times S}\|_\mathrm{F} = O(\Delta),\ |S| \geq 0.99d$$

…using

$$\approx dr^{1+o(1)} \text{ samples}$$

$$\approx dr^{2+o(1)} \text{ time}$$

Polynomial time:

[Recht '11]

$$\min_{\mathbf{X}^\star \text{ matches obs}} \|\mathbf{X}^\star\|_1$$

$$\approx dr \text{ samples}$$

Near-linear time:

[Jain-Netrapalli '15]

PGD + clipping

$$\approx dr^7 \text{ time}$$

$$\approx dr^5 \text{ samples}$$

Matrix completion

Theorem, Part IIA [Kelner, Li, Liu, Sidford, Tian '23]:

From rank-r, "regular" $\mathbf{M}^\star \in \mathbb{R}^{d \times d}$ + ($\|\mathbf{N}\|_F \le \Delta$), can give $\mathbf{M} \in \mathbb{R}^{d \times d}$ with:

$$\|\mathbf{M} - \mathbf{M}^\star\|_{\mathrm{F}} = O(r^{1.5}\Delta)$$

…using

$$\approx dr^{1+o(1)} \text{ samples}$$

$$\approx dr^{2+o(1)} \text{ time}$$

Polynomial time:

[Recht '11]

$$\min_{\mathbf{X}^\star \text{ matches obs}} \|\mathbf{X}^\star\|_1$$

$$\approx dr \text{ samples}$$

Near-linear time:

[Jain-Netrapalli '15]

PGD + clipping

$$\approx dr^7 \text{ time}$$

$$\approx dr^5 \text{ samples}$$

Matrix completion

Theorem, Part 1B [Kelner, Li, Liu, Sidford, Tian '23]:

From rank-r, "incoherent" $\mathbf{M}^\star \in \mathbb{R}^{d\times d}$ + ($\|\mathbf{N}\|_F \leq \Delta$), can give $\mathbf{M} \in \mathbb{R}^{d\times d}$ with:

$$\|\mathbf{M} - \mathbf{M}^\star\|_F = O(r^{1.5}\Delta)$$

…using

$$\approx dr^{2+o(1)} \text{ samples}$$

$$\approx dr^{3+o(1)} \text{ time}$$

Polynomial time:

[Recht '11]

$$\min_{\mathbf{X}^\star \text{ matches obs}} \|\mathbf{X}^\star\|_1$$

$$\approx dr \text{ samples}$$

Near-linear time:

[Jain-Netrapalli '15]

PGD + clipping

$$\approx dr^7 \text{ time}$$

$$\approx dr^5 \text{ samples}$$

Roadmap

- Overview
  - Sparse recovery: SOTA and what's new
  - Matrix completion: SOTA and what's new
- Sparse recovery
  - Short-flat decompositions
  - Projected gradient descent
- Matrix completion

Optimization for sparse recovery?

RIP: $\text{for all } \|v\|_0 = O(s)$

$$\Omega(\|v\|_2) = \|\mathbf{A}v\|_2 = O(\|v\|_2)$$

"Restricted well-conditioning":
Well-conditioned restricted to some set

Optimization for sparse recovery?

RIP: for all $\|v\|_0 = O(s)$

$$\Omega(\|v\|_2) = \|\mathbf{A}v\|_2 = O(\|v\|_2)$$

$\Longrightarrow$

(folklore, proof via shelling)

RIP+: for all NS v

$$\Omega(\|v\|_2) = \|\mathbf{A}v\|_2 = O(\|v\|_2)$$

v is numerically sparse (NS) if

$$\frac{\|v\|_1^2}{\|v\|_2^2} = O(s)$$

Optimization for sparse recovery?

A first attempt
- Maintain $x - x^\star$ is NS
- ??????
- Profit

RIP+: for all NS v

$$\Omega(\|v\|_2) = \|\mathbf{A}v\|_2 = O(\|v\|_2)$$

v is numerically sparse (NS) if

$$\frac{\|v\|_1^2}{\|v\|_2^2} = O(s)$$

Optimization for sparse recovery?

A first attempt

- Maintain $x - x^\star$ is NS
- ??????
- Profit

Can maintain (?) via ℓ_1 projection

RIP+: for all NS v

$$\Omega(\|v\|_2) = \|\mathbf{A}v\|_2 = O(\|v\|_2)$$

Question:

How to reason about effect of projection?

v is numerically sparse (NS) if

$$\frac{\|v\|_1^2}{\|v\|_2^2} = O(s)$$

Key geometric insight

Lemma (informal): If you hit a unit *v* with a random Gaussian matrix, it is “flat” in all directions except *v*

Key geometric insight

$$\left(\frac{1}{n}\sum_{i\in[n]} a_i a_i^\top\right) v \qquad = \mathbf{A}^\top \mathbf{A}$$

Random Gaussian matrix

$$\{a_i\}_{i\in[n]} \sim_{\text{i.i.d.}} \mathcal{N}(0, \mathbf{I})$$

Lemma (informal): If you hit a unit *v* with a random Gaussian matrix, it is "flat" in all directions except *v*

Key geometric insight

$$\left(\frac{1}{n}\sum_{i\in[n]} a_i a_i^\top\right) v$$

Lemma (informal): If you hit a unit *v* with a random Gaussian matrix, it is "flat" in all directions except *v*

Write $a_i = \xi_i v + a_i^\perp$

$$\xi_i \sim \mathcal{N}(0,1),\ a_i^\perp \sim \mathcal{N}(0, \mathbf{I} - vv^\top)$$

Key geometric insight

$$\left(\frac{1}{n}\sum_{i\in[n]} a_i a_i^\top\right) v$$

Lemma (informal): If you hit a unit *v* with a random Gaussian matrix, it is "flat" in all directions except *v*

Write $a_i = \xi_i v + a_i^\perp$

$$a_i a_i^\top v = \xi_i^2 v + \xi_i a_i^\perp$$

Key geometric insight

$$\left(\frac{1}{n}\sum_{i\in[n]} a_i a_i^\top\right) v$$

Lemma (informal): If you hit a unit *v* with a random Gaussian matrix, it is "flat" in all directions except *v*

Write $a_i = \xi_i v + a_i^\perp$

$$a_i a_i^\top v = \xi_i^2 v + \xi_i a_i^\perp$$

$$\frac{1}{n}\sum_{i\in[n]} a_i a_i^\top v \approx v + \frac{1}{n}\sum_{i\in[n]} \xi_i a_i^\perp$$

Key geometric insight

$$\left(\frac{1}{n}\sum_{i\in[n]} a_i a_i^\top\right) v$$

Lemma (informal): If you hit a unit *v* with a random Gaussian matrix, it is "flat" in all directions except *v*

Write $a_i = \xi_i v + a_i^\perp$

$$a_i a_i^\top v = \xi_i^2 v + \xi_i a_i^\perp$$

$$\frac{1}{n}\sum_{i\in[n]} a_i a_i^\top v \approx v + \frac{1}{n}\sum_{i\in[n]} \xi_i a_i^\perp$$

(essentially random)

"flat" := ℓ_∞ bounded

Short-flat decompositions

Lemma (formal): let **A** be RIP with parameter *s*. For all NS unit *v*,

$$\mathbf{A}^\top \mathbf{A} v = p_v + e_v$$

$$\|p_v\|_2 = O(1), \|e_v\|_\infty = O\left(\frac{1}{\sqrt{s}}\right)$$

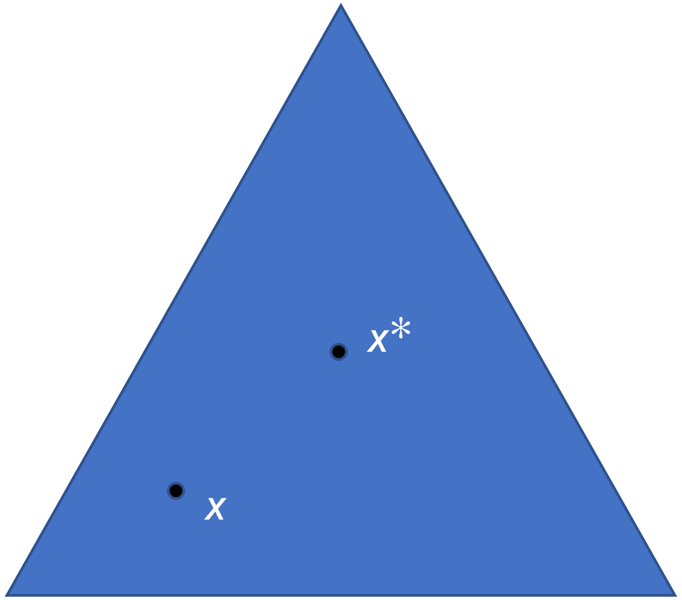
Why does PGD work?

Lemma: If you hit an NS unit *v* with $\mathbf{A}^{\mathsf{T}}\mathbf{A}$ where $\mathbf{A}$ is RIP, the result has a short-flat decomposition.

Let $v := x - x^*$

Suppose:

- $\| v \|_2 \leq 1$
- $\| v \|_1 \leq \sqrt{s}$



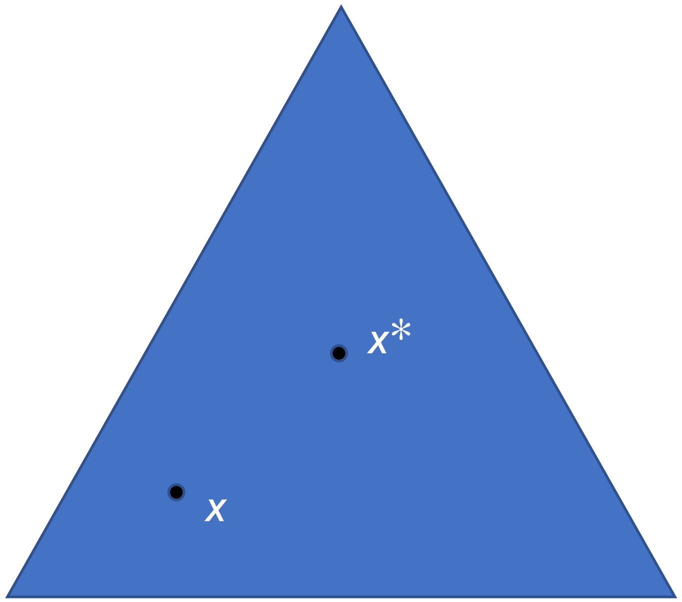
Why does PGD work?

Lemma: If you hit an NS unit *v* with $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ where **A** is RIP, the result has a short-flat decomposition.

Let $v := x - x^*$

Suppose:

- $\| v \|_2 \leq 1$
- $\| v \|_1 \leq \sqrt{s}$



Case 1: $\| v \|_2 \leq \frac{1}{2}$

Halve our radius ☺

Case 2: $\| v \|_2 \geq \frac{1}{2}$

Use $\mathbf{A}^{\mathrm{T}}(\mathbf{A}x - b) = \mathbf{A}^{\mathrm{T}}\mathbf{A}v$ as descent direction

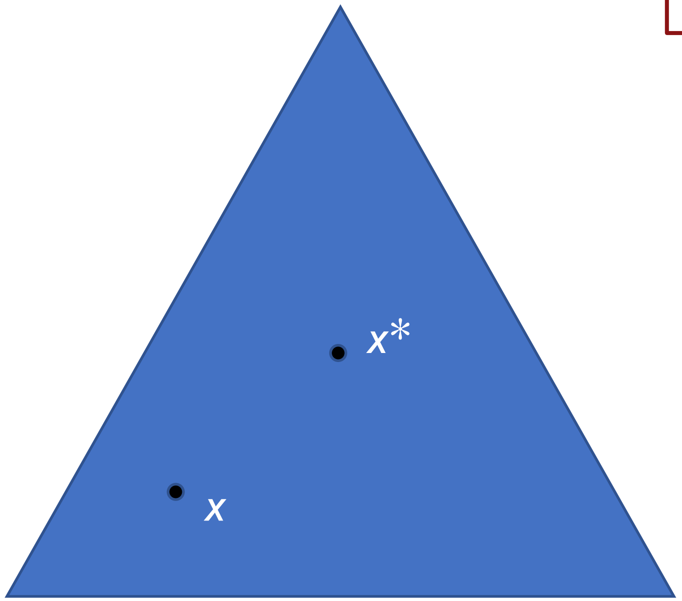
Why does PGD work?

Lemma: If you hit an NS unit *v* with $\mathbf{A}^\mathsf{T}\mathbf{A}$ where $\mathbf{A}$ is RIP, the result has a short-flat decomposition.

Let $v := x - x^*$

Suppose:

- $\| v \|_2 \leq 1$
- $\| v \|_1 \leq \sqrt{s}$



Case 2 is good idea by Lemma:

$$\mathbf{A}^\top \mathbf{A} v \approx v + e$$

"flat" := ℓ_∞ bounded

Filtered by PGD against ℓ_1 ball + Hölder's inequality

Case 1: $\| v \|_2 \leq \frac{1}{2}$

Halve our radius ☺

Case 2: $\| v \|_2 \geq \frac{1}{2}$

Use $\mathbf{A}^\mathrm{T}(\mathbf{A}x - b) = \mathbf{A}^\mathrm{T}\mathbf{A}v$ as descent direction

Algorithm sketch

Input: s-sparse x_{in}, $\| x_{\text{in}} - x^* \|_2 \leq R$

Output: s-sparse x_{out}, $\| x_{\text{out}} - x^* \|_2 \leq \frac{R}{2}$

- $\mathcal{X} := \{x \mid \| x_{\text{in}} - x \|_1 = O(\sqrt{s})R\}$
- This set contains x^* by Cauchy-Schwarz

v is numerically sparse (NS) if

$$\frac{\|v\|_1^2}{\|v\|_2^2} = O(s)$$

Restricted W-C: for all NS v,

$$\frac{1}{n} \sum_{i \in [n]} \langle a_i, v \rangle^2 = [\Omega(1), O(1)]$$

Short-flat: for all NS unit v,

$$\mathbf{A}^\top \mathbf{A} v = p_v + e_v$$

$$\|p_v\|_2 = O(1), \|e_v\|_\infty = O\left(\frac{1}{\sqrt{s}}\right)$$

Algorithm sketch

Input: s-sparse x_{in}, $\| x_{\text{in}} - x^* \|_2 \le R$

Output: s-sparse x_{out}, $\| x_{\text{out}} - x^* \|_2 \le \frac{R}{2}$

- $\mathcal{X} := \{x \mid \| x_{\text{in}} - x \|_1 = O(\sqrt{s})R\}$
- $x \leftarrow x_{\text{in}}$
- For 10 iterations:
  - If v is not numerically sparse, we're done
  - If it is numerically sparse, we can PGD

v is numerically sparse (NS) if

$$\frac{\|v\|_1^2}{\|v\|_2^2} = O(s)$$

Restricted W-C: for all NS v,

$$\frac{1}{n} \sum_{i \in [n]} \langle a_i, v \rangle^2 = [\Omega(1), O(1)]$$

Short-flat: for all NS unit v,

$$\mathbf{A}^\top \mathbf{A} v = p_v + e_v$$

$$\|p_v\|_2 = O(1), \|e_v\|_\infty = O\left(\frac{1}{\sqrt{s}}\right)$$

Algorithm sketch

Input: s-sparse x_{in}, $\| x_{\text{in}} - x^* \|_2 \leq R$

Output: s-sparse x_{out}, $\| x_{\text{out}} - x^* \|_2 \leq \frac{R}{2}$

- $\mathcal{X} := \{x \mid \| x_{\text{in}} - x \|_1 = O(\sqrt{s})R\}$
- $x \leftarrow x_{\text{in}}$
- For 10 iterations:
  - $\Delta = \mathbf{A}x - b = \mathbf{A}v$ for $v = x - x^*$
  - If $\frac{1}{n}\sum_{1 \leq i \leq n} \Delta_i^2 \geq \Omega(1)$ and $\frac{1}{n}\mathbf{A}^{\mathrm{T}}\Delta$ has a short-flat decomposition:
    - $x \leftarrow \mathrm{argmin}_{x' \in \mathcal{X}} \| x' - (x - \eta \mathbf{A}^{\mathrm{T}}\Delta) \|_2$
    - Constant progress in distance to x^*

v is numerically sparse (NS) if

$$\frac{\|v\|_1^2}{\|v\|_2^2} = O(s)$$

Restricted W-C: for all NS *v*,

$$\frac{1}{n}\sum_{i \in [n]} \langle a_i, v \rangle^2 = [\Omega(1), O(1)]$$

Short-flat: for all NS unit *v*,

$$\mathbf{A}^\top \mathbf{A} v = p_v + e_v$$

$$\|p_v\|_2 = O(1), \|e_v\|_\infty = O\left(\frac{1}{\sqrt{s}}\right)$$

Algorithm sketch

Input: s-sparse x_{in}, $\| x_{\text{in}} - x^* \|_2 \leq R$

Output: s-sparse x_{out}, $\| x_{\text{out}} - x^* \|_2 \leq \frac{R}{2}$

- $\mathcal{X} := \{x \mid \| x_{\text{in}} - x \|_1 = O(\sqrt{s})R\}$
- $x \leftarrow x_{\text{in}}$
- For 10 iterations:
  - $\Delta = \mathbf{A}x - b = \mathbf{A}v$ for $v = x - x^*$
  - If $\frac{1}{n}\sum_{1 \leq i \leq n} \Delta_i^2 \geq \Omega(1)$ and $\frac{1}{n}\mathbf{A}^{\mathrm{T}}\Delta$ has a short-flat decomposition:
    - $x \leftarrow \text{argmin}_{x' \in \mathcal{X}} \| x' - (x - \eta \mathbf{A}^{\mathrm{T}}\Delta) \|_2$
    - Constant progress in distance to x^*
  - Else:
    - Break
    - Not numerically sparse, radius loose

v is numerically sparse (NS) if

$$\frac{\|v\|_1^2}{\|v\|_2^2} = O(s)$$

Restricted W-C: for all NS *v*,

$$\frac{1}{n}\sum_{i \in [n]} \langle a_i, v \rangle^2 = [\Omega(1), O(1)]$$

Short-flat: for all NS unit *v*,

$$\mathbf{A}^\top \mathbf{A} v = p_v + e_v$$

$$\|p_v\|_2 = O(1), \|e_v\|_\infty = O\left(\frac{1}{\sqrt{s}}\right)$$

Algorithm sketch

Input: s-sparse x_{in}, $\| x_{\mathrm{in}} - x^* \|_2 \leq R$

Output: s-sparse x_{out}, $\| x_{\mathrm{out}} - x^* \|_2 \leq \frac{R}{2}$

- $\mathcal{X} := \{x | \, \| x_{\mathrm{in}} - x \|_1 = O(\sqrt{s})R\}$
- $x \leftarrow x_{\mathrm{in}}$
- For 10 iterations:
  - $\Delta = \mathbf{A}x - b = \mathbf{A}v$ for $v = x - x^*$
  - If $\frac{1}{n}\sum_{1 \leq i \leq n} \Delta_i^2 \geq \Omega(1)$ and $\frac{1}{n}\mathbf{A}^{\mathrm{T}}\Delta$ has a short-flat decomposition:
    - $x \leftarrow \mathrm{argmin}_{x' \in \mathcal{X}} \| x' - (x - \eta \mathbf{A}^{\mathrm{T}}\Delta) \|_2$
    - Constant progress in distance to x^*
  - Else:
    - Break
- Return x truncated to s largest coordinates

v is numerically sparse (NS) if

$$\frac{\|v\|_1^2}{\|v\|_2^2} = O(s)$$

Restricted W-C: for all NS v,

$$\frac{1}{n}\sum_{i \in [n]} \langle a_i, v \rangle^2 = [\Omega(1), O(1)]$$

Short-flat: for all NS unit v,

$$\mathbf{A}^\top \mathbf{A} v = p_v + e_v$$

$$\|p_v\|_2 = O(1), \|e_v\|_\infty = O\left(\frac{1}{\sqrt{s}}\right)$$

Algorithm sketch

Input: s-sparse x_{in}, $\| x_{\text{in}} - x^* \|_2 \leq R$

Output: s-sparse x_{out}, $\| x_{\text{out}} - x^* \|_2 \leq \frac{R}{2}$

- $\mathcal{X} := \{x \mid \| x_{\text{in}} - x \|_1 = O(\sqrt{s})R\}$
- $x \leftarrow x_{\text{in}}$
- For 10 iterations:
  - $\Delta = \mathbf{A}x - b = \mathbf{A}v$ for $v = x - x^*$
  - If $\frac{1}{n}\sum_{1 \leq i \leq n} \Delta_i^2 \geq \Omega(1)$ and $\frac{1}{n}\mathbf{A}^{\mathrm{T}}\Delta$ has a short-flat decomposition:
    - $x \leftarrow \mathrm{argmin}_{x' \in \mathcal{X}} \| x' - (x - \eta \mathbf{A}^{\mathrm{T}}\Delta) \|_2$
    - Constant progress in distance to x^*
  - Else:
    - Break
- Return x truncated to s largest coordinates

Makes sense even in semi-random case!

We find planted solution in near-linear time.

Analysis sketch

$$\|x_t - x^\star\|_2^2 - \|x_{t+1} - x^\star\|_2^2 \geq 2\eta\langle \underbrace{\qquad g_t \qquad}_{:=\mathbf{A}^\top\mathbf{A}(x_t - x^\star)}, x_{t+1} - x^\star\rangle - \|x_t - x_{t+1}\|_2^2$$

Analysis sketch

$$\|x_t - x^\star\|_2^2 - \|x_{t+1} - x^\star\|_2^2 \geq 2\eta\langle \underbrace{g_t}_{:=\mathbf{A}^\top\mathbf{A}(x_t - x^\star)}, x_{t+1} - x^\star\rangle - \|x_t - x_{t+1}\|_2^2$$

$$\geq 2\eta\langle g_t, x_t - x^\star\rangle - 2\eta\langle e_t, x_t - x_{t+1}\rangle$$

$$-2\eta\langle p_t, x_t - x_{t+1}\rangle - \|x_t - x_{t+1}\|_2^2$$

Analysis sketch

$$\|x_t - x^\star\|_2^2 - \|x_{t+1} - x^\star\|_2^2 \geq 2\eta\langle \underbrace{g_t}_{:=\mathbf{A}^\top\mathbf{A}(x_t - x^\star)}, x_{t+1} - x^\star\rangle - \|x_t - x_{t+1}\|_2^2$$

$$\geq 2\eta\langle g_t, x_t - x^\star\rangle - 2\eta\langle e_t, x_t - x_{t+1}\rangle$$

big (restricted W-C) small (flatness + Hölder)

$$-2\eta\langle p_t, x_t - x_{t+1}\rangle - \|x_t - x_{t+1}\|_2^2$$

small (shortness + Young)

Roadmap

- Overview
  - Sparse recovery: SOTA and what's new
  - Matrix completion: SOTA and what's new
- Sparse recovery
  - Short-flat decompositions
  - Projected gradient descent
- Matrix completion

Matrix short-flat decomposition?

$$\frac{1}{p}\left[\mathbf{M} - \mathbf{M}^{\star}\right]_{\Omega}$$

"gradient" (i.e. scaled
residuals)

Matrix short-flat decomposition?

$$\frac{1}{p}\left[\mathbf{M}-\mathbf{M}^{\star}\right]_{\Omega} = \underbrace{\mathbf{M}-\mathbf{M}^{\star}}_{\mathbf{P}}$$

$$+ \underbrace{\frac{1}{p}\left[\mathbf{M}-\mathbf{M}^{\star}\right]_{\Omega} - \left(\mathbf{M}-\mathbf{M}^{\star}\right)}_{\mathbf{E}}$$

…hopefully flat
(opnorm bounded)?

Matrix short-flat decomposition?

$$\frac{1}{p}\left[\mathbf{M} - \mathbf{M}^{\star}\right]_{\Omega} - (\mathbf{M} - \mathbf{M}^{\star})$$

Matrix Bernstein controls opnorm via…

- "Prob. 1 bound": entrywise small
- "Variance bound": row-column norms small

Matrix short-flat decomposition?

$$\frac{1}{p}\left[\mathbf{M}-\mathbf{M}^{\star}\right]_{\Omega}-(\mathbf{M}-\mathbf{M}^{\star})$$

Matrix Bernstein controls opnorm via…

- "Prob. 1 bound": entrywise small
- "Variance bound": row-column norms small

Not true in general, but OK if we drop 1% of rows/cols.

Matrix short-flat decomposition?

$$\frac{1}{p}\left[\mathbf{M}-\mathbf{M}^{\star}\right]_{\Omega}-\left(\mathbf{M}-\mathbf{M}^{\star}\right)$$

Matrix Bernstein controls opnorm via…

- "Prob. 1 bound": entrywise small
- "Variance bound": row-column norms small

Not true in general, but OK if we drop 1% of rows/cols.

…recovering dropped rows/cols is most of the work…

…also need to maintain iterates are low-rank…

What else?

1. General framework for semi-random inverse problems?
   - Similar "fast algo/robust algo" gaps for other problems
   - Fine-grained guarantees?
2. Harder adversaries?
   - How far can we push definition of "bad" observations?
   - Weaker types of hidden structure?

Thank you!

Contact
kjtian.github.io
kjtian@cs.utexas.edu

Semi-Random Sparse Recovery in Nearly-Linear Time

Matrix Completion in Almost-Verification Time