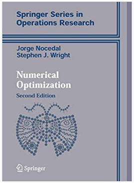
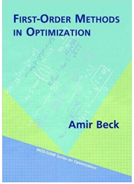
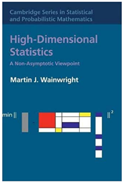
Solving Random Quadratic Systems of Equations Is Nearly as Easy as Solving Linear Systems

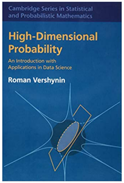
Yuxin Chen (Princeton)   Emmanuel Candès (Stanford)

Y. Chen, E. J. Candès, Communications on Pure and Applied Mathematics
vol. 70, no. 5, pp. 822-883, May 2017





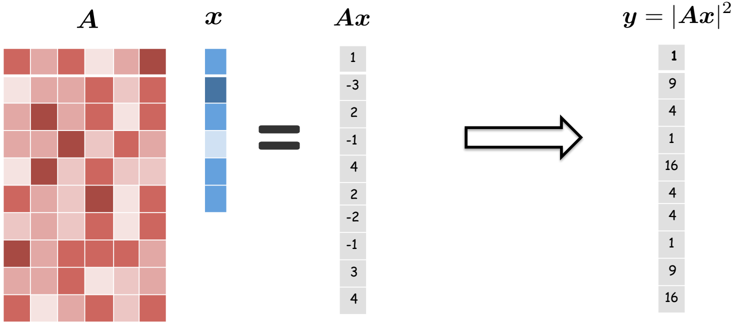




nonconvex optimization

(high-dimensional) statistics

Solving quadratic systems of equations



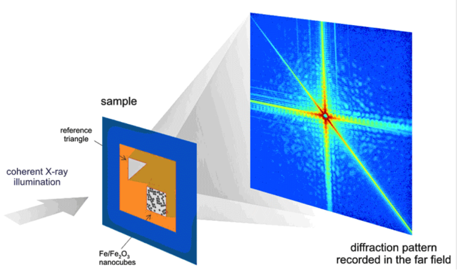
Solve for $\boldsymbol{x} \in \mathbb{C}^n$ in m quadratic equations

$$y_k \quad \approx \quad |\langle \boldsymbol{a}_k, \boldsymbol{x} \rangle|^2, \qquad k = 1, \ldots, m$$

Motivation: a missing phase problem in imaging science

Detectors record intensities of diffracted rays

- $x(t_1, t_2) \longrightarrow$ Fourier transform $\hat{x}(f_1, f_2)$

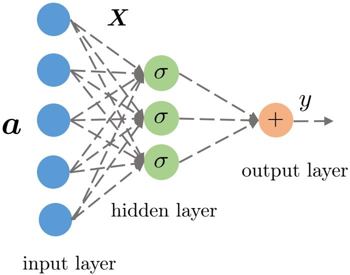


intensity of electrical field: $\left|\hat{x}(f_1, f_2)\right|^2 = \left|\int x(t_1, t_2) e^{-i2\pi(f_1 t_1 + f_2 t_2)} \mathrm{d}t_1 \mathrm{d}t_2\right|^2$

Phase retrieval: recover true signal $x(t_1, t_2)$ from intensity measurements

Motivation: learning neural nets with quadratic activation

— *Soltanolkotabi, Javanmard, Lee '17, Li, Ma, Zhang '17*



input features: $\boldsymbol{a}$; weights: $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_r]$

$$\text{output:} \quad y = \sum_{i=1}^{r} \sigma(\boldsymbol{a}^\top \boldsymbol{x}_i) \overset{\sigma(z)=z^2}{:=} \sum_{i=1}^{r} (\boldsymbol{a}^\top \boldsymbol{x}_i)^2$$

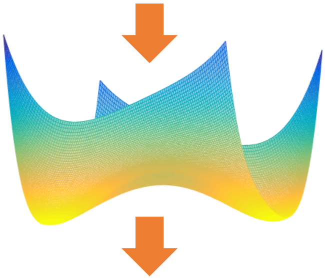
Solving quadratic systems is NP-complete *in general* ...

"I can't find an efficient algorithm, but neither can all these people."

Fig credit: coding horror

Statistical models come to rescue

statistical models



benign landscape

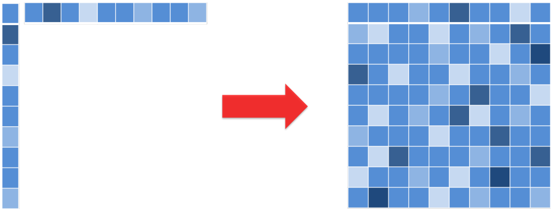
tractable algorithms

When data are generated by $\underbrace{\text{certain statistical / randomized models}}_{\text{e.g. } \boldsymbol{a}_k \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{I}_n)}$, problems are often much nicer than worst-case instances

Convex relaxation

Lifting: introduce $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^*$ to linearize constraints

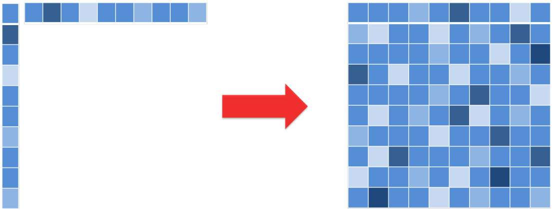
$$y_k \;=\; |\boldsymbol{a}_k^* \boldsymbol{x}|^2 = \boldsymbol{a}_k^*(\boldsymbol{x}\boldsymbol{x}^*)\boldsymbol{a}_k \qquad \Longrightarrow \qquad y_k = \boldsymbol{a}_k^* \boldsymbol{X} \boldsymbol{a}_k$$



Convex relaxation

Lifting: introduce $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^*$ to linearize constraints

$$y_k \;=\; |\boldsymbol{a}_k^* \boldsymbol{x}|^2 = \boldsymbol{a}_k^* (\boldsymbol{x}\boldsymbol{x}^*) \boldsymbol{a}_k \qquad \Longrightarrow \qquad y_k = \boldsymbol{a}_k^* \boldsymbol{X} \boldsymbol{a}_k$$

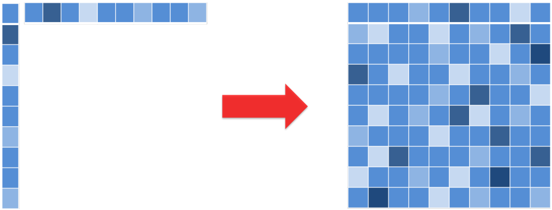


$$\begin{array}{lll} \text{find} & \boldsymbol{X} \succeq \boldsymbol{0} & \\ \text{s.t.} & y_k \;=\; \boldsymbol{a}_k^* \boldsymbol{X} \boldsymbol{a}_k, & k = 1, \cdots, m \\ & \color{red}{\mathrm{rank}(\boldsymbol{X}) = 1} & \end{array}$$

Convex relaxation

Lifting: introduce $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^*$ to linearize constraints

$$y_k = |\boldsymbol{a}_k^* \boldsymbol{x}|^2 = \boldsymbol{a}_k^*(\boldsymbol{x}\boldsymbol{x}^*)\boldsymbol{a}_k \quad \Longrightarrow \quad y_k = \boldsymbol{a}_k^* \boldsymbol{X} \boldsymbol{a}_k$$

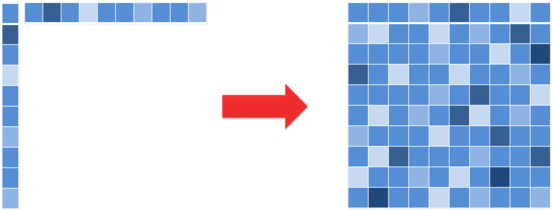


$$\begin{array}{ll} \text{find} & \boldsymbol{X} \succeq \boldsymbol{0} \\ \text{s.t.} & y_k = \boldsymbol{a}_k^* \boldsymbol{X} \boldsymbol{a}_k, \qquad k = 1, \cdots, m \\ & \cancel{\operatorname{rank}(\boldsymbol{X}) = 1} \end{array}$$

Convex relaxation

Lifting: introduce $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^*$ to linearize constraints

$$y_k \;=\; |\boldsymbol{a}_k^*\boldsymbol{x}|^2 = \boldsymbol{a}_k^*(\boldsymbol{x}\boldsymbol{x}^*)\boldsymbol{a}_k \qquad \Longrightarrow \qquad y_k = \boldsymbol{a}_k^*\boldsymbol{X}\boldsymbol{a}_k$$



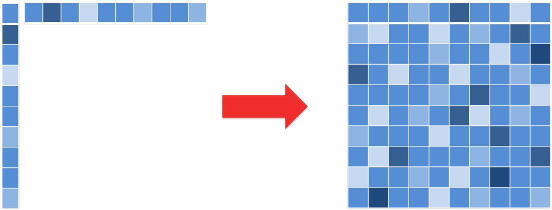
$$\begin{array}{lll} \text{find} & \boldsymbol{X} \succeq \boldsymbol{0} & \\ \text{s.t.} & y_k \;=\; \boldsymbol{a}_k^*\boldsymbol{X}\boldsymbol{a}_k, & k = 1, \cdots, m \\ & \cancel{\text{rank}(\boldsymbol{X}) = 1} & \end{array}$$

Works well if $\{\boldsymbol{a}_k\}$ are random

Convex relaxation

Lifting: introduce $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^*$ to linearize constraints

$$y_k = |\boldsymbol{a}_k^* \boldsymbol{x}|^2 = \boldsymbol{a}_k^* (\boldsymbol{x}\boldsymbol{x}^*) \boldsymbol{a}_k \quad \Longrightarrow \quad y_k = \boldsymbol{a}_k^* \boldsymbol{X} \boldsymbol{a}_k$$

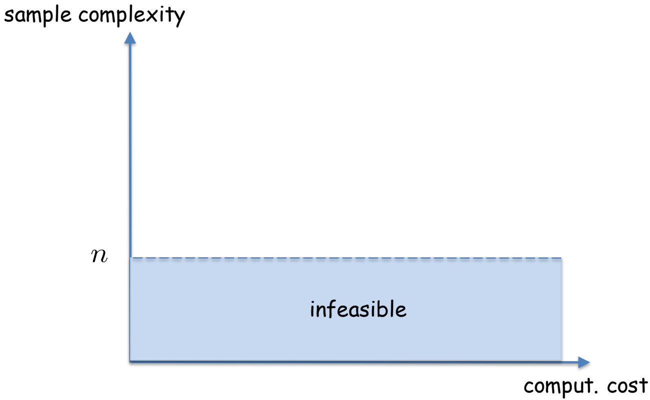


$$\begin{array}{ll} \text{find} & \boldsymbol{X} \succeq \boldsymbol{0} \\ \text{s.t.} & y_k = \boldsymbol{a}_k^* \boldsymbol{X} \boldsymbol{a}_k, \qquad k = 1, \cdots, m \\ & \cancel{\mathrm{rank}(\boldsymbol{X}) = 1} \end{array}$$

Works well if $\{\boldsymbol{a}_k\}$ are random, but huge increase in dimensions

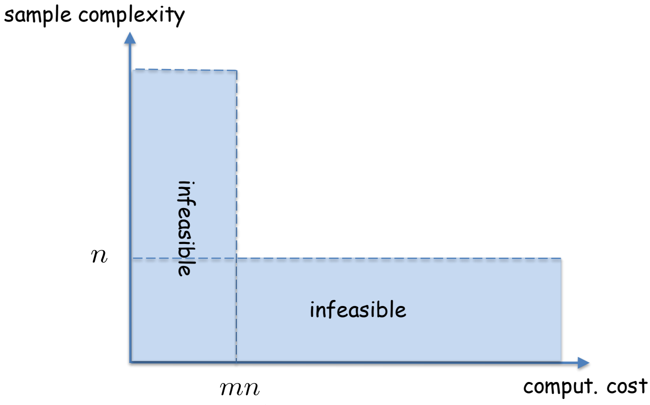
Prior art (before our work)

n: # unknowns; m: sample size (# eqns); $\boldsymbol{y} = |\boldsymbol{A}\boldsymbol{x}|^2,\ \boldsymbol{A} \in \mathbb{R}^{m\times n}$



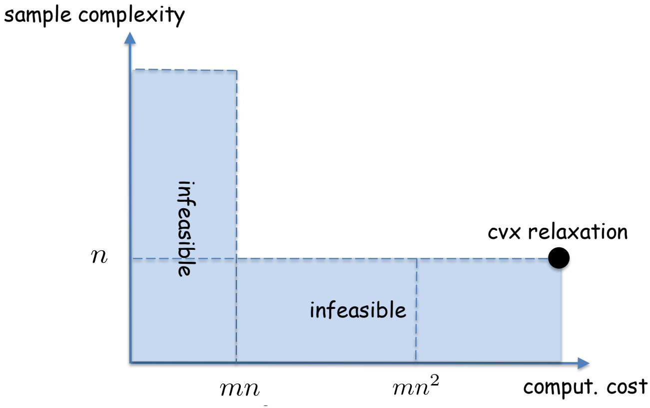
Prior art (before our work)

n: # unknowns; m: sample size (# eqns); $\boldsymbol{y} = |\boldsymbol{A}\boldsymbol{x}|^2,\ \boldsymbol{A} \in \mathbb{R}^{m \times n}$



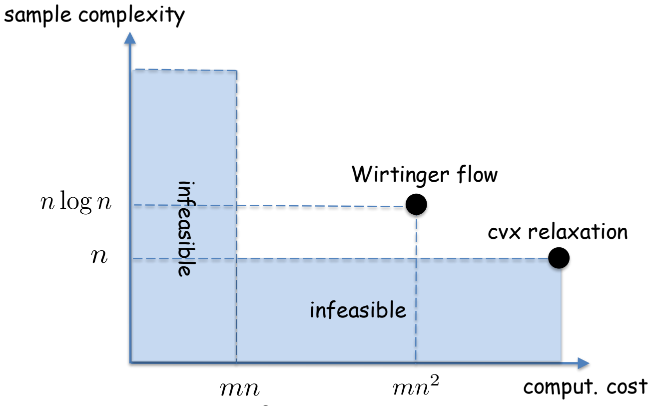
Prior art (before our work)

n: # unknowns; m: sample size (# eqns); $\boldsymbol{y} = |\boldsymbol{A}\boldsymbol{x}|^2, \boldsymbol{A} \in \mathbb{R}^{m \times n}$



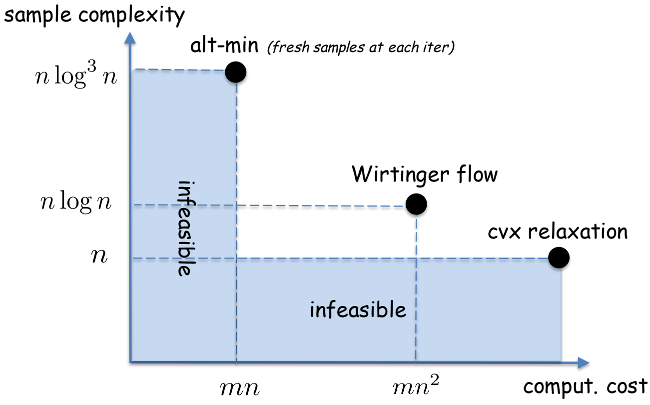
Prior art (before our work)

n: # unknowns; m: sample size (# eqns); $\boldsymbol{y} = |\boldsymbol{A}\boldsymbol{x}|^2, \boldsymbol{A} \in \mathbb{R}^{m \times n}$



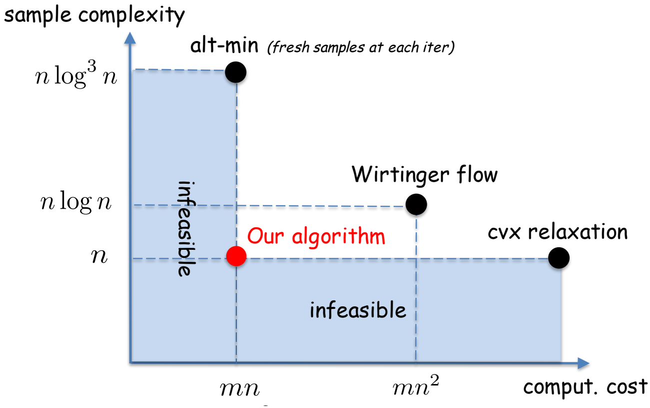
Prior art (before our work)

n: # unknowns; m: sample size (# eqns); $\boldsymbol{y} = |\boldsymbol{A}\boldsymbol{x}|^2,\ \boldsymbol{A} \in \mathbb{R}^{m \times n}$



A glimpse of our results

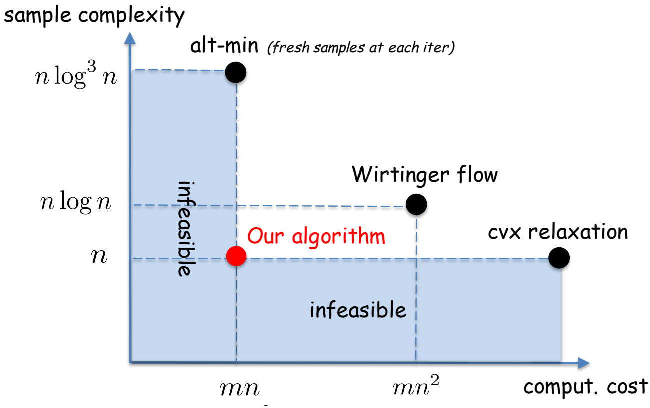
n: # unknowns; m: sample size (# eqns); $\boldsymbol{y} = |\boldsymbol{A}\boldsymbol{x}|^2, \boldsymbol{A} \in \mathbb{R}^{m \times n}$



This work: random quadratic systems are solvable in linear time!

A glimpse of our results

n: # unknowns; m: sample size (# eqns); $\boldsymbol{y} = |\boldsymbol{A}\boldsymbol{x}|^2, \boldsymbol{A} \in \mathbb{R}^{m \times n}$



This work: random quadratic systems are solvable in linear time!

✓ *minimal sample size*

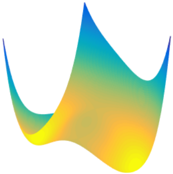
✓ *optimal statistical accuracy*

A first impulse: maximum likelihood estimate

$$\text{minimize}_{\boldsymbol{z}} \quad f(\boldsymbol{z}) = \quad \frac{1}{m}\sum\nolimits_{k=1}^{m} f_k(\boldsymbol{z})$$

A first impulse: maximum likelihood estimate

$$\text{minimize}_{\boldsymbol{z}} \quad f(\boldsymbol{z}) = \quad \frac{1}{m}\sum_{k=1}^{m} f_k(\boldsymbol{z})$$

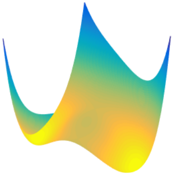


- Gaussian data: $\quad y_k \sim |\boldsymbol{a}_k^* \boldsymbol{x}|^2 + \mathcal{N}(0, \sigma^2)$

$$f_k(\boldsymbol{z}) = \left(y_k - |\boldsymbol{a}_k^* \boldsymbol{z}|^2\right)^2$$

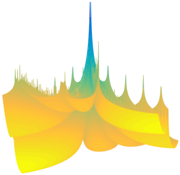
A first impulse: maximum likelihood estimate

$$\text{minimize}_{\boldsymbol{z}} \quad f(\boldsymbol{z}) = \frac{1}{m}\sum_{k=1}^{m} f_k(\boldsymbol{z})$$



- Gaussian data: $y_k \sim |\boldsymbol{a}_k^* \boldsymbol{x}|^2 + \mathcal{N}(0, \sigma^2)$

$$f_k(\boldsymbol{z}) = \left(y_k - |\boldsymbol{a}_k^* \boldsymbol{z}|^2\right)^2$$



- Poisson data: $y_k \sim \text{Poisson}\left(|\boldsymbol{a}_k^* \boldsymbol{x}|^2\right)$

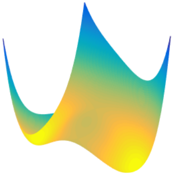
$$f_k(\boldsymbol{z}) = |\boldsymbol{a}_k^* \boldsymbol{z}|^2 - y_k \log |\boldsymbol{a}_k^* \boldsymbol{z}|^2$$

A first impulse: maximum likelihood estimate

$$\text{minimize}_{\boldsymbol{z}} \quad f(\boldsymbol{z}) = \frac{1}{m}\sum_{k=1}^{m} f_k(\boldsymbol{z})$$

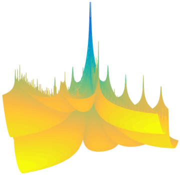
- Gaussian data: $y_k \sim |\boldsymbol{a}_k^* \boldsymbol{x}|^2 + \mathcal{N}(0, \sigma^2)$

$$f_k(\boldsymbol{z}) = \left(y_k - |\boldsymbol{a}_k^* \boldsymbol{z}|^2 \right)^2$$



- Poisson data: $y_k \sim \text{Poisson}\left(|\boldsymbol{a}_k^* \boldsymbol{x}|^2 \right)$

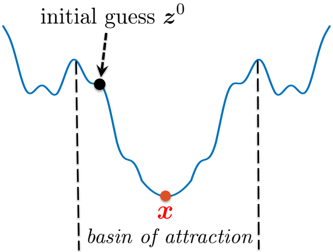
$$f_k(\boldsymbol{z}) = |\boldsymbol{a}_k^* \boldsymbol{z}|^2 - y_k \log |\boldsymbol{a}_k^* \boldsymbol{z}|^2$$



Problem: $f(\cdot)$ nonconvex, many local stationary points

A plausible nonconvex paradigm

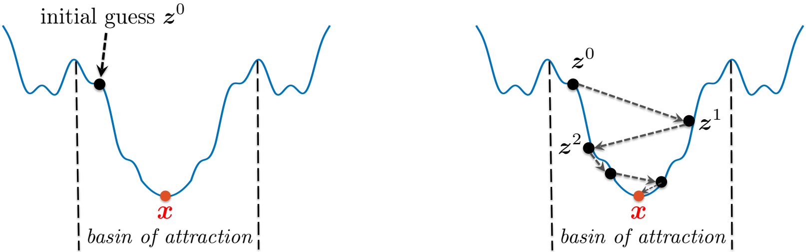
$$\text{minimize}_{\boldsymbol{z}} \quad f(\boldsymbol{z}) = \sum_{k=1}^{m} f_k(\boldsymbol{z})$$



1. initialize within $\underbrace{\text{local basin sufficiently close to } \boldsymbol{x}}_{\text{(hopefully) nicer landscape}}$

A plausible nonconvex paradigm

$$\text{minimize}_{\boldsymbol{z}} \quad f(\boldsymbol{z}) = \sum_{k=1}^{m} f_k(\boldsymbol{z})$$

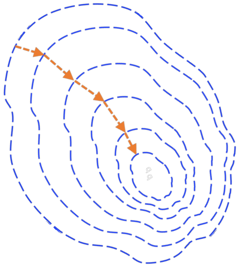


1. initialize within $\underbrace{\text{local basin sufficiently close to } \boldsymbol{x}}_{\text{(hopefully) nicer landscape}}$

2. iterative refinement

Wirtinger flow (Candès, Li, Soltanolkotabi '14)

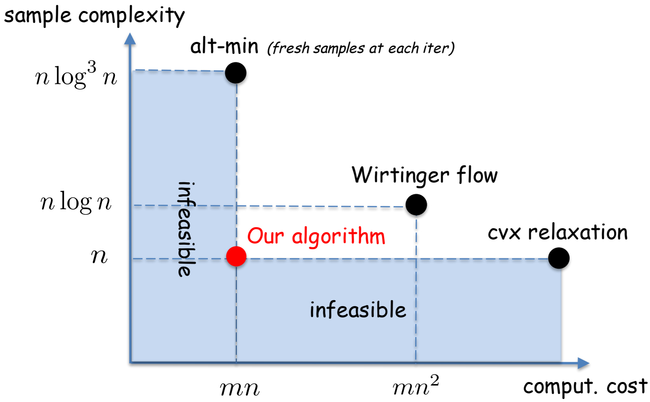
$$\text{minimize}_{\boldsymbol{z}} \quad f(\boldsymbol{z}) = \frac{1}{m}\sum_{k=1}^{m}\left[\left(\boldsymbol{a}_k^\top \boldsymbol{z}\right)^2 - y_k\right]^2$$



- **spectral initialization:** $\boldsymbol{z}^0 \leftarrow$ leading eigenvector of certain data matrix
- **(Wirtinger) gradient descent:**

$$\boldsymbol{z}^{t+1} = \boldsymbol{z}^t - \mu_t \nabla f(\boldsymbol{z}^t), \qquad t = 0, 1, \cdots$$

Performance guarantees for WF



- suboptimal computational cost?

  — n *times more expensive than linear-time algorithms*

- suboptimal sample complexity?

Iterative refinement stage: search directions

Wirtinger flow: $$\boldsymbol{z}^{t+1} = \boldsymbol{z}^t - \frac{\mu_t}{m}\sum_{k=1}^{m} \underbrace{\big(y_k - |\boldsymbol{a}_k^\top \boldsymbol{z}^t|^2\big)\boldsymbol{a}_k\boldsymbol{a}_k^\top \boldsymbol{z}^t}_{=\nabla f_k(\boldsymbol{z}^t)}$$

Iterative refinement stage: search directions

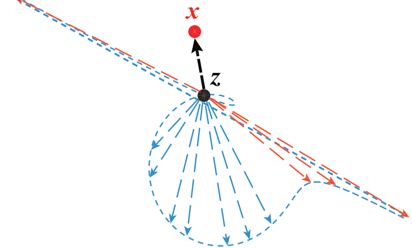
$$\text{Wirtinger flow:} \quad \boldsymbol{z}^{t+1} = \boldsymbol{z}^t - \frac{\mu_t}{m} \sum_{k=1}^{m} \underbrace{\left(y_k - |\boldsymbol{a}_k^\top \boldsymbol{z}^t|^2\right) \boldsymbol{a}_k \boldsymbol{a}_k^\top \boldsymbol{z}^t}_{= \nabla f_k(\boldsymbol{z}^t)}$$

Even in a local region around $\boldsymbol{x}$ (e.g. $\{\boldsymbol{z} \mid \|\boldsymbol{z} - \boldsymbol{x}\|_2 \leq 0.1 \|\boldsymbol{x}\|_2\}$):

- $f(\cdot)$ is NOT strongly convex unless $m \gg n$
- $f(\cdot)$ has huge smoothness parameter

Iterative refinement stage: search directions

Wirtinger flow: $$\boldsymbol{z}^{t+1} = \boldsymbol{z}^{t} - \frac{\mu_t}{m}\sum_{k=1}^{m} \underbrace{\big(y_k - |\boldsymbol{a}_k^\top \boldsymbol{z}^t|^2\big)\boldsymbol{a}_k \boldsymbol{a}_k^\top \boldsymbol{z}^t}_{=\nabla f_k(\boldsymbol{z}^t)}$$



locus of $\{\nabla f_k(\boldsymbol{z})\}$

Problem: descent direction has large variability

Our solution: variance reduction via proper trimming

More adaptive rule:

$$\boldsymbol{z}^{t+1} = \boldsymbol{z}^{t} - \frac{\mu_t}{m} \sum_{i=1}^{m} \frac{y_i - |\boldsymbol{a}_i^\top \boldsymbol{z}^t|^2}{\boldsymbol{a}_i^\top \boldsymbol{z}^t} \boldsymbol{a}_i \mathbf{1}_{\mathcal{E}_1^i(\boldsymbol{z}^t) \cap \mathcal{E}_2^i(\boldsymbol{z}^t)}$$

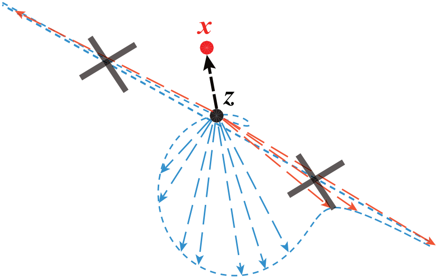
where $\mathcal{E}_1^i(\boldsymbol{z}) = \left\{ \alpha_z^{\mathsf{lb}} \leq \frac{|\boldsymbol{a}_i^\top \boldsymbol{z}|}{\|\boldsymbol{z}\|_2} \leq \alpha_z^{\mathsf{ub}} \right\}$; $\mathcal{E}_2^i(\boldsymbol{z}) = \left\{ |y_i - |\boldsymbol{a}_i^\top \boldsymbol{z}|^2| \leq \frac{\frac{\alpha_h}{m} \left\| \boldsymbol{y} - \mathcal{A}(\boldsymbol{z}\boldsymbol{z}^\top) \right\|_1 |\boldsymbol{a}_i^\top \boldsymbol{z}|}{\|\boldsymbol{z}\|_2} \right\}$

Our solution: variance reduction via proper trimming

More adaptive rule:

$$\boldsymbol{z}^{t+1} = \boldsymbol{z}^t - \frac{\mu_t}{m} \sum_{i=1}^{m} \frac{y_i - |\boldsymbol{a}_i^\top \boldsymbol{z}^t|^2}{\boldsymbol{a}_i^\top \boldsymbol{z}^t} \boldsymbol{a}_i \mathbf{1}_{\mathcal{E}_1^i(\boldsymbol{z}^t) \cap \mathcal{E}_2^i(\boldsymbol{z}^t)}$$

where $\mathcal{E}_1^i(\boldsymbol{z}) = \left\{ \alpha_z^{\mathsf{lb}} \leq \frac{|\boldsymbol{a}_i^\top \boldsymbol{z}|}{\|\boldsymbol{z}\|_2} \leq \alpha_z^{\mathsf{ub}} \right\};\ \mathcal{E}_2^i(\boldsymbol{z}) = \left\{ |y_i - |\boldsymbol{a}_i^\top \boldsymbol{z}|^2| \leq \frac{\frac{\alpha_h}{m} \left\| \boldsymbol{y} - \mathcal{A}(\boldsymbol{z}\boldsymbol{z}^\top) \right\|_1 |\boldsymbol{a}_i^\top \boldsymbol{z}|}{\|\boldsymbol{z}\|_2} \right\}$

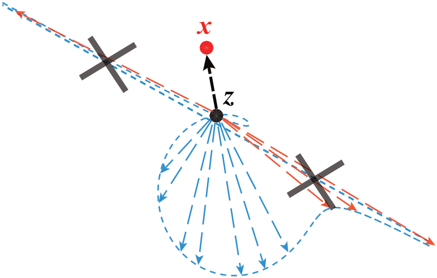


Our solution: variance reduction via proper trimming

More adaptive rule:

$$\boldsymbol{z}^{t+1} = \boldsymbol{z}^{t} - \frac{\mu_t}{m} \sum_{i=1}^{m} \frac{y_i - |\boldsymbol{a}_i^\top \boldsymbol{z}^t|^2}{\boldsymbol{a}_i^\top \boldsymbol{z}^t} \boldsymbol{a}_i \mathbf{1}_{\mathcal{E}_1^i(\boldsymbol{z}^t) \cap \mathcal{E}_2^i(\boldsymbol{z}^t)}$$

where $\mathcal{E}_1^i(\boldsymbol{z}) = \left\{ \alpha_z^{\mathsf{lb}} \leq \frac{|\boldsymbol{a}_i^\top \boldsymbol{z}|}{\|\boldsymbol{z}\|_2} \leq \alpha_z^{\mathsf{ub}} \right\}$; $\mathcal{E}_2^i(\boldsymbol{z}) = \left\{ |y_i - |\boldsymbol{a}_i^\top \boldsymbol{z}|^2| \leq \frac{\frac{\alpha_h}{m} \left\| \boldsymbol{y} - \mathcal{A}(\boldsymbol{z}\boldsymbol{z}^\top) \right\|_1 |\boldsymbol{a}_i^\top \boldsymbol{z}|}{\|\boldsymbol{z}\|_2} \right\}$



informally, $\boldsymbol{z}^{t+1} = \boldsymbol{z}^{t} - \frac{\mu}{m} \sum_{k \in \mathcal{T}} \nabla f_k(\boldsymbol{z}^t)$

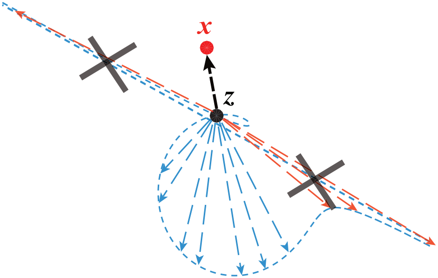
- $\mathcal{T}$ trims away excessively large grad components

Our solution: variance reduction via proper trimming

More adaptive rule:

$$\boldsymbol{z}^{t+1} = \boldsymbol{z}^t - \frac{\mu_t}{m}\sum_{i=1}^{m} \frac{y_i - |\boldsymbol{a}_i^\top \boldsymbol{z}^t|^2}{\boldsymbol{a}_i^\top \boldsymbol{z}^t} \boldsymbol{a}_i \mathbf{1}_{\mathcal{E}_1^i(\boldsymbol{z}^t)\cap\mathcal{E}_2^i(\boldsymbol{z}^t)}$$

where $\mathcal{E}_1^i(\boldsymbol{z}) = \left\{\alpha_z^{\mathsf{lb}} \le \frac{|\boldsymbol{a}_i^\top \boldsymbol{z}|}{\|\boldsymbol{z}\|_2} \le \alpha_z^{\mathsf{ub}}\right\}$; $\mathcal{E}_2^i(\boldsymbol{z}) = \left\{|y_i - |\boldsymbol{a}_i^\top \boldsymbol{z}|^2| \le \frac{\frac{\alpha_h}{m}\left\|\boldsymbol{y}-\mathcal{A}(\boldsymbol{z}\boldsymbol{z}^\top)\right\|_1 |\boldsymbol{a}_i^\top \boldsymbol{z}|}{\|\boldsymbol{z}\|_2}\right\}$

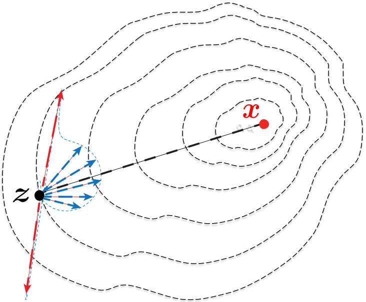


informally, $\boldsymbol{z}^{t+1} = \boldsymbol{z}^t - \frac{\mu}{m}\sum_{k\in\mathcal{T}} \nabla f_k(\boldsymbol{z}^t)$

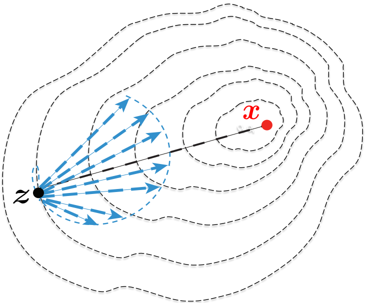
- $\mathcal{T}$ trims away excessively large grad components

Slight bias $\quad + \quad$ much reduced variance

Larger step size μ_t is feasible



without trimming: $\mu_t = O(1/n)$



with trimming: $\mu_t = O(1)$

With better-controlled descent directions, one proceeds far more aggressively

Initialization stage

Spectral initialization (e.g. alt-min, WF): $\boldsymbol{z}^0 \leftarrow$ leading eigenvector of

$$\boldsymbol{Y} := \frac{1}{m}\sum_{k=1}^{m} y_k \boldsymbol{a}_k \boldsymbol{a}_k^*$$

Initialization stage

Spectral initialization (e.g. alt-min, WF): $\boldsymbol{z}^0 \leftarrow$ leading eigenvector of

$$\boldsymbol{Y} := \frac{1}{m}\sum_{k=1}^{m} y_k \boldsymbol{a}_k \boldsymbol{a}_k^*$$

- Rationale: $\mathbb{E}[\boldsymbol{Y}] = \|\boldsymbol{x}\|_2^2\,\boldsymbol{I} + 2\boldsymbol{x}\boldsymbol{x}^*$ under i.i.d. Gaussian design

Initialization stage

Spectral initialization (e.g. alt-min, WF): $\boldsymbol{z}^0 \leftarrow$ leading eigenvector of

$$\boldsymbol{Y} := \frac{1}{m}\sum_{k=1}^{m} y_k \boldsymbol{a}_k \boldsymbol{a}_k^*$$

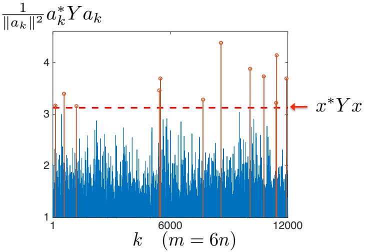
- Rationale: $\mathbb{E}[\boldsymbol{Y}] = \|\boldsymbol{x}\|_2^2\,\boldsymbol{I} + 2\boldsymbol{x}\boldsymbol{x}^*$ under i.i.d. Gaussian design
- Would succeed if $\boldsymbol{Y} \rightarrow \mathbb{E}[\boldsymbol{Y}]$

Improving initialization

$$\boldsymbol{Y} \;=\; \frac{1}{m}\sum_k \underbrace{y_k \boldsymbol{a}_k \boldsymbol{a}_k^*}_{\text{heavy-tailed}} \quad \not\rightarrow \quad \mathbb{E}[\boldsymbol{Y}] \quad \text{unless } m \gg n$$

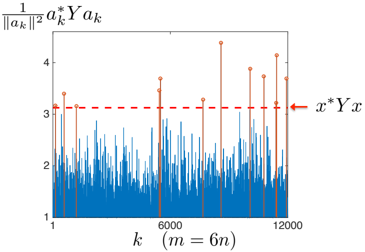
Improving initialization

$$\boldsymbol{Y} \;=\; \frac{1}{m}\sum_k \underbrace{y_k \boldsymbol{a}_k \boldsymbol{a}_k^*}_{\text{heavy-tailed}} \quad \not\to \quad \mathbb{E}[\boldsymbol{Y}] \quad \text{unless } m \gg n$$



Improving initialization

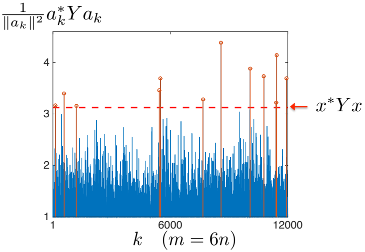
$$\boldsymbol{Y} \;=\; \frac{1}{m}\sum_k \underbrace{y_k \boldsymbol{a}_k \boldsymbol{a}_k^*}_{\text{heavy-tailed}} \quad \not\rightarrow \quad \mathbb{E}[\boldsymbol{Y}] \quad \text{unless } m \gg n$$



Problem large outliers $y_k = |\boldsymbol{a}_k^* \boldsymbol{x}|^2$ bear too much influence

Improving initialization

$$\boldsymbol{Y} = \frac{1}{m}\sum_k \underbrace{y_k \boldsymbol{a}_k \boldsymbol{a}_k^*}_{\text{heavy-tailed}} \quad \not\to \quad \mathbb{E}[\boldsymbol{Y}] \quad \text{unless } m \gg n$$



Problem large outliers $y_k = |\boldsymbol{a}_k^* \boldsymbol{x}|^2$ bear too much influence

Solution discard large samples and run PCA for $\frac{1}{m}\sum_k y_k \boldsymbol{a}_k \boldsymbol{a}_k^* \mathbf{1}_{\{|y_k| \lesssim \mathrm{Avg}\{|y_l|\}\}}$

Summary of proposed algorithm

1. **Regularized spectral initialization:** $\boldsymbol{z}^0 \leftarrow$ principal component of

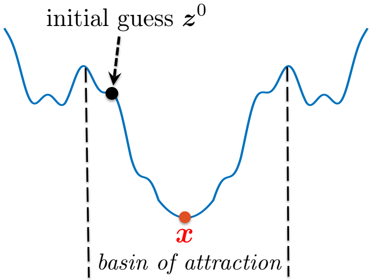
$$\frac{1}{m}\sum\nolimits_{k\in\mathcal{T}_0} y_k\,\boldsymbol{a}_k\boldsymbol{a}_k^*$$

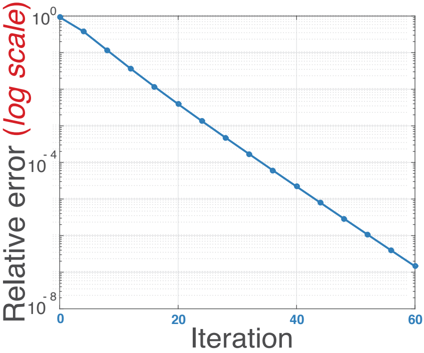
2. **Regularized gradient descent**

$$\boldsymbol{z}^{t+1} = \boldsymbol{z}^t - \frac{\mu_t}{m}\sum\nolimits_{k\in\mathcal{T}_t} \nabla f_k(\boldsymbol{z})$$

Adaptive and iteration-varying rules: discard high-leverage data $\{y_k : k \notin \mathcal{T}_t\}$

Theoretical guarantees (noiseless data)





Theorem (Chen & Candès) When $a_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)$ and $m \gtrsim n$, with high probability our algorithm attains ε accuracy in $\underbrace{O\big(\log \frac{1}{\varepsilon}\big) \text{ iterations}}_{\text{dimension-free linear convergence}}$

Computational complexity

$$\boldsymbol{A} := \{\boldsymbol{a}_k^*\}_{1 \le k \le m}$$

- **Initialization:** leading eigenvector $\rightarrow$ a few applications of $\boldsymbol{A}$ and $\boldsymbol{A}^*$

$$\sum_{k \in \mathcal{T}_0} y_k \, \boldsymbol{a}_k \boldsymbol{a}_k^* = \boldsymbol{A}^* \, \mathrm{diag}\{y_k \cdot 1_{k \in \mathcal{T}_0}\} \, \boldsymbol{A}$$

Computational complexity

$$\boldsymbol{A} := \{\boldsymbol{a}_k^*\}_{1\leq k\leq m}$$

- **Initialization:** leading eigenvector $\rightarrow$ a few applications of $\boldsymbol{A}$ and $\boldsymbol{A}^*$

$$\sum_{k\in\mathcal{T}_0} y_k\, \boldsymbol{a}_k \boldsymbol{a}_k^* = \boldsymbol{A}^*\ \mathrm{diag}\{y_k \cdot 1_{k\in\mathcal{T}_0}\}\, \boldsymbol{A}$$

- **Iterations:** one application of $\boldsymbol{A}$ and $\boldsymbol{A}^*$ per iteration

$$\boldsymbol{z}^{t+1} = \boldsymbol{z}^t - \frac{\mu_t}{m}\nabla f_{\mathsf{tr}}(\boldsymbol{z}^t)$$

$$\nabla f_{\mathsf{tr}}(\boldsymbol{z}^t) = \boldsymbol{A}^*\boldsymbol{\nu}$$

$$\boldsymbol{\nu} = 2\tfrac{|\boldsymbol{A}\boldsymbol{z}^t|^2 - \boldsymbol{y}}{\boldsymbol{A}\boldsymbol{z}^t} \cdot 1_{\mathcal{T}}$$

Computational complexity

$$\boldsymbol{A} := \{\boldsymbol{a}_k^*\}_{1 \leq k \leq m}$$

- **Initialization:** leading eigenvector $\rightarrow$ a few applications of $\boldsymbol{A}$ and $\boldsymbol{A}^*$

$$\sum_{k \in \mathcal{T}_0} y_k \, \boldsymbol{a}_k \boldsymbol{a}_k^* = \boldsymbol{A}^* \, \mathrm{diag}\{y_k \cdot 1_{k \in \mathcal{T}_0}\} \, \boldsymbol{A}$$

- **Iterations:** one application of $\boldsymbol{A}$ and $\boldsymbol{A}^*$ per iteration

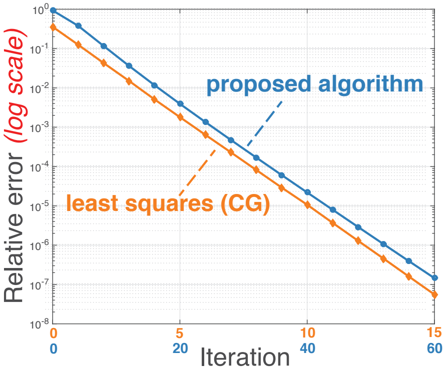
$$\boldsymbol{z}^{t+1} = \boldsymbol{z}^t - \frac{\mu_t}{m} \nabla f_{\mathsf{tr}}(\boldsymbol{z}^t)$$

$$\nabla f_{\mathsf{tr}}(\boldsymbol{z}^t) = \boldsymbol{A}^* \boldsymbol{\nu}$$
$$\boldsymbol{\nu} = 2 \frac{|\boldsymbol{A}\boldsymbol{z}^t|^2 - \boldsymbol{y}}{\boldsymbol{A}\boldsymbol{z}^t} \cdot 1_{\mathcal{T}}$$

Approximate runtime: several tens of applications of $\boldsymbol{A}$ and $\boldsymbol{A}^*$

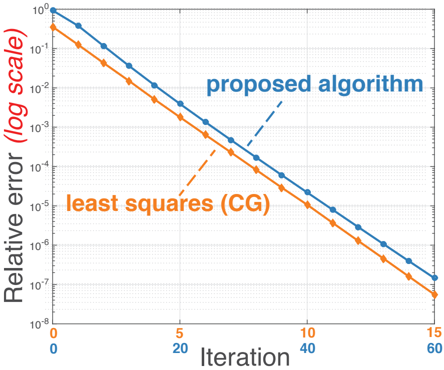
Numerical performance

- CG: solve $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$
- Our algorithm: solve $\boldsymbol{y} = |\boldsymbol{A}\boldsymbol{x}|^2$



Numerical performance

- CG: solve $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$
- Our algorithm: solve $\boldsymbol{y} = |\boldsymbol{A}\boldsymbol{x}|^2$



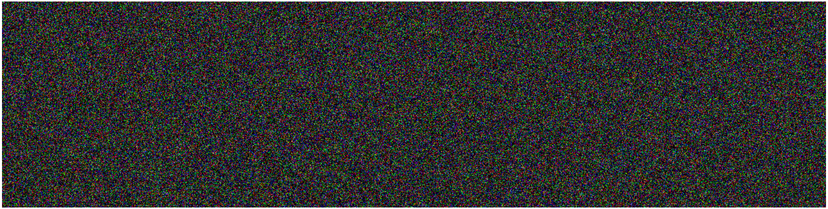
For random quadratic systems $(m = 8n)$

comput. cost of our algo. $\approx$ $\mathbf{4} \times$ comput. cost of least squares

Empirical performance ($m = 12n$)

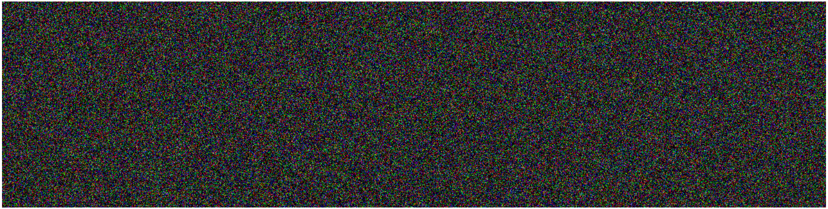
Ground truth $\boldsymbol{x} \in \mathbb{R}^{409600}$

Empirical performance ($m = 12n$)



Spectral initialization

Empirical performance ($m = 12n$)



Spectral initialization

Proposed: regularized spectral initialization

Empirical performance ($m = 12n$)

After regularized spectral initialization

Empirical performance ($m = 12n$)

After regularized spectral initialization

After 50 proposed iterations

Stability under noisy data

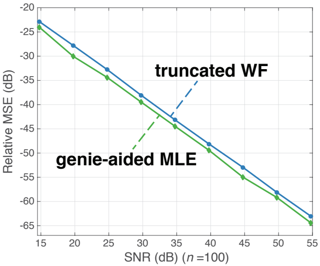
Comparison with genie-aided MLE (with phase info. revealed)

$$y_k \sim \text{Poisson}(\ |\boldsymbol{a}_k^*\boldsymbol{x}|^2\ ) \quad \text{and} \quad \varepsilon_k = \operatorname{sign}(\boldsymbol{a}_k^*\boldsymbol{x}) \qquad \text{(revealed by a genie)}$$

Stability under noisy data

Comparison with genie-aided MLE (with phase info. revealed)

$$y_k \sim \text{Poisson}(\ |\boldsymbol{a}_k^* \boldsymbol{x}|^2\ ) \quad \text{and} \quad \varepsilon_k = \mathrm{sign}\,(\boldsymbol{a}_k^* \boldsymbol{x}) \qquad \text{(revealed by a genie)}$$

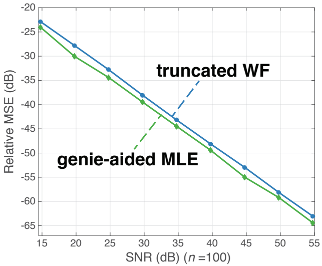


little empirical loss
due to missing signs

Stability under noisy data

Comparison with genie-aided MLE (with phase info. revealed)

$$y_k \sim \text{Poisson}(\ |\boldsymbol{a}_k^* \boldsymbol{x}|^2\ ) \quad \text{and} \quad \varepsilon_k = \text{sign}\left(\boldsymbol{a}_k^* \boldsymbol{x}\right) \qquad \text{(revealed by a genie)}$$

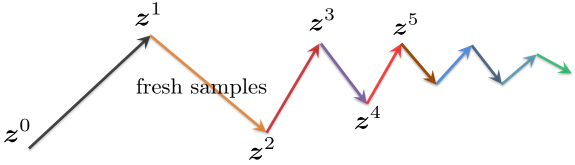


little empirical loss
due to missing signs

Theorem (Chen & Candès) Our algorithm achieves optimal statistical accuracy!

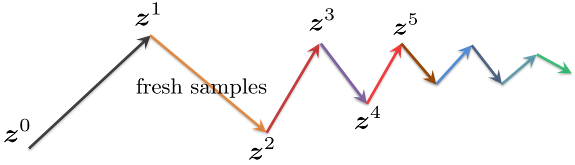
Deal with complicated dependencies across iterations

Several prior approaches: require fresh samples at each iteration

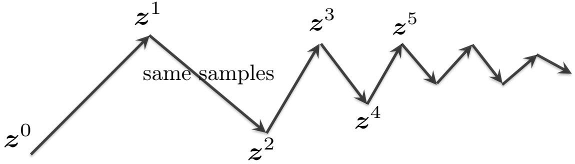


Deal with complicated dependencies across iterations

Several prior approaches: require fresh samples at each iteration



This approach: reuse all samples in all iterations



A small sample of more recent works

- other optimal algorithms
  - reshaped WF (Zhang et al.), truncated AF (Wang et al.), median-TWF (Zhang et al.)
  - alt-min w/o resampling (Waldspurger)
  - composite optimization (Duchi et al., Charisopoulos et al.)
  - approximate message passing (Ma et al.)
  - block coordinate descent (Barmherzig et al.)
  - PhaseMax (Goldstein et al., Bahmani et al., Salehi et al., Dhifallah et al., Hand et al.)
- stochastic algorithms (Kolte et al., Zhang et al., Lu et al., Tan et al., Jeong et al.)
- improved WF theory: iteration complexity $\rightarrow$ $O(\log n \log \frac{1}{\varepsilon})$ (Ma et al.)
- improved initialization (Lu et al., Wang et al., Mondelli et al.)
- random initialization (Chen et al.)
- structured quadratic systems (Cai et al., Soltanolkotabi, Wang et al., Yang et al., Qu et al.)
- geometric analysis (Sun et al., Davis et al.)
- low-rank generalization (White et al., Li et al., Vaswani et al.)

Concluding remarks

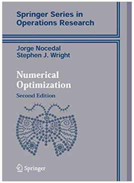
Achieves optimal bias-variance tradeoff by adaptively discarding high-leverage data

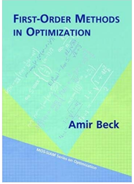
	comput. cost	sample size	statistical accuracy
cvx relaxation			
our non-cvx algo.			

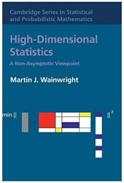
Concluding remarks

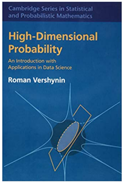
Achieves optimal bias-variance tradeoff by adaptively discarding high-leverage data

	comput. cost	sample size	statistical accuracy
cvx relaxation	👎	👍	👍
our non-cvx algo.	👍	👍	👍







nonconvex optimization

(high-dimensional) statistics