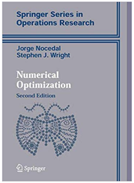
The Power of Nonconvex Optimization in Solving Random Quadratic Systems of Equations

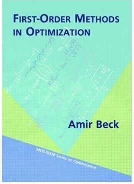
Yuxin Chen (Princeton) Emmanuel Candès (Stanford)

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

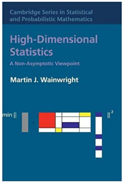
Agenda

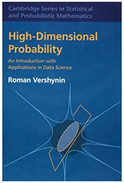
1. The power of nonconvex optimization in solving random quadratic systems of equations (Aug. 28)
2. Random initialization and implicit regularization in nonconvex statistical estimation (Aug. 29)
3. The projected power method: an efficient nonconvex algorithm for joint discrete assignment from pairwise data (Sep. 3)
4. Spectral methods meets asymmetry: two recent stories (Sep. 4)
5. Inference and uncertainty quantification for noisy matrix completion (Sep. 5)





nonconvex optimization



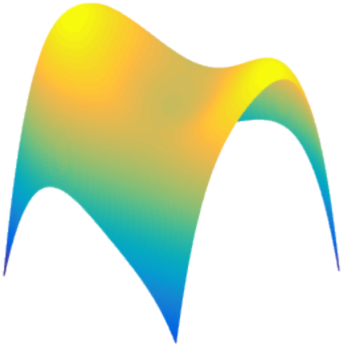


(high-dimensional) statistics

Nonconvex problems are everywhere

Maximum likelihood estimation is usually nonconvex

$$\begin{array}{rrcl} \text{maximize}_{\boldsymbol{x}} & \ell(\boldsymbol{x}; \text{data}) & \rightarrow & \text{may be nonconcave} \\ \text{subj. to} & \boldsymbol{x} \in \mathcal{S} & \rightarrow & \text{may be nonconvex} \end{array}$$

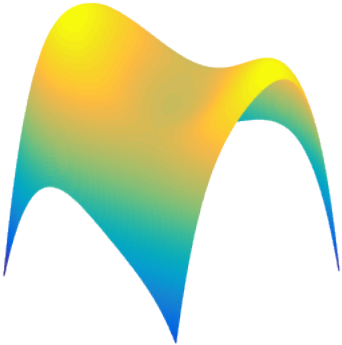


Nonconvex problems are everywhere

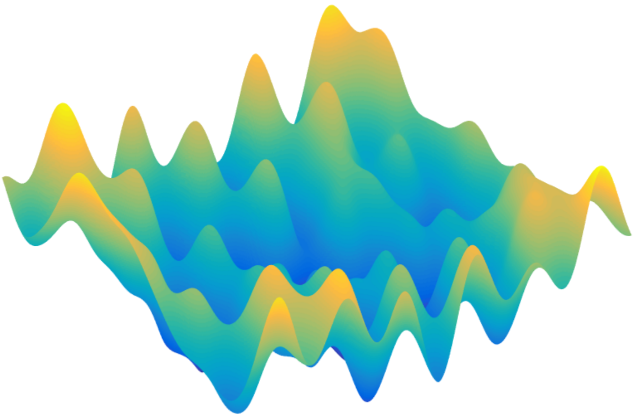
Maximum likelihood estimation is usually nonconvex

$$\begin{array}{rrcl} \text{maximize}_{\boldsymbol{x}} & \ell(\boldsymbol{x};\, \mathrm{data}) & \rightarrow & \text{may be nonconcave} \\ \text{subj. to} & \boldsymbol{x} \in \mathcal{S} & \rightarrow & \text{may be nonconvex} \end{array}$$

- low-rank matrix completion
- robust principal component analysis
- graph clustering
- dictionary learning
- blind deconvolution
- learning neural nets
- ...



Nonconvex optimization may be super scary



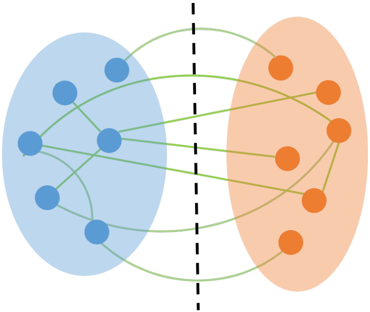
There may be bumps everywhere and exponentially many local optima

e.g. 1-layer neural net (Auer, Herbster, Warmuth '96; Vu '98)

Example: solving quadratic programs is hard

Finding maximum cut in a graph is

$$\begin{array}{ll} \text{maximize}_{\boldsymbol{x}} & \boldsymbol{x}^\top \boldsymbol{W} \boldsymbol{x} \\ \text{subj. to} & x_i^2 = 1, \quad i = 1, \cdots, n \end{array}$$



Example: solving quadratic programs is hard

“I can’t find an efficient algorithm, but neither can all these people.”

Fig credit: coding horror

$1,000,000 question

One strategy: convex relaxation

Can relax into convex problems by

- finding convex surrogates (e.g. compressed sensing, matrix completion)
- lifting into higher dimensions (e.g. Max-Cut)

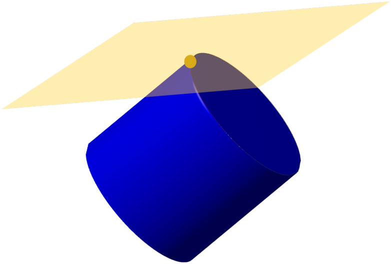
Example of convex surrogate: low-rank matrix completion

— Fazel '02, Recht, Parrilo, Fazel '10, Candès, Recht '09

$$\text{minimize}_{M} \quad \text{rank}(M) \qquad \text{subj. to data constraints}$$

$\Downarrow$ cvx surrogate

$$\text{minimize}_{M} \quad \text{nuc-norm}(M) \qquad \text{subj. to data constraints}$$



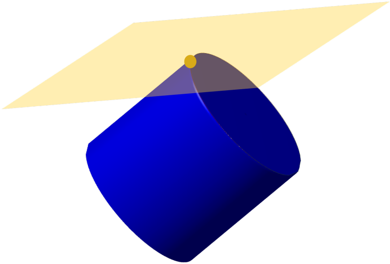
Example of convex surrogate: low-rank matrix completion

— Fazel '02, Recht, Parrilo, Fazel '10, Candès, Recht '09

$$\text{minimize}_{\boldsymbol{M}} \quad \text{rank}(\boldsymbol{M}) \qquad \text{subj. to data constraints}$$

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

$$\text{minimize}_{\boldsymbol{M}} \quad \text{nuc-norm}(\boldsymbol{M}) \qquad \text{subj. to data constraints}$$



Robust variation used everyday by Netflix

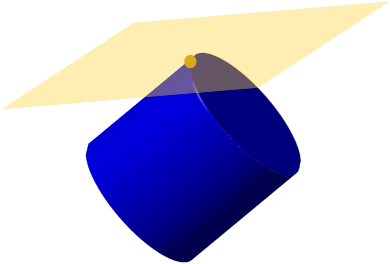
Example of convex surrogate: low-rank matrix completion

— Fazel '02, Recht, Parrilo, Fazel '10, Candès, Recht '09

$$\text{minimize}_{\boldsymbol{M}} \quad \text{rank}(\boldsymbol{M}) \qquad \text{subj. to data constraints}$$

$\Downarrow$ cvx surrogate

$$\text{minimize}_{\boldsymbol{M}} \quad \text{nuc-norm}(\boldsymbol{M}) \qquad \text{subj. to data constraints}$$

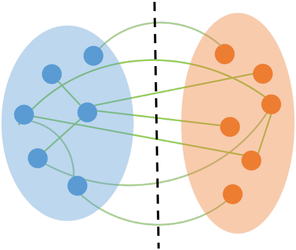


Robust variation used everyday by Netflix

Problem: operate in *full* matrix space even though $\boldsymbol{X}$ is low-rank

Example of lifting: Max-Cut

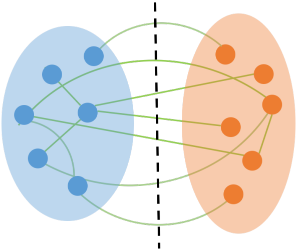
— Goemans, Williamson '95



$$\begin{array}{ll} \text{maximize}_{\boldsymbol{x}} & \boldsymbol{x}^\top \boldsymbol{W} \boldsymbol{x} \\ \text{subj. to} & x_i^2 = 1, \quad i = 1, \cdots, n \end{array}$$

Example of lifting: Max-Cut

— Goemans, Williamson '95



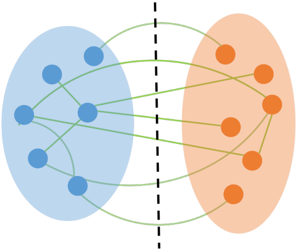
$$\begin{aligned} \text{maximize}_{\boldsymbol{x}} \quad & \boldsymbol{x}^\top \boldsymbol{W} \boldsymbol{x} \\ \text{subj. to} \quad & x_i^2 = 1, \quad i = 1, \cdots, n \end{aligned}$$

$\Downarrow$ let $\boldsymbol{X}$ be $\boldsymbol{x}\boldsymbol{x}^\top$

$$\begin{aligned} \text{maximize}_{\boldsymbol{X}} \quad & \langle \boldsymbol{X}, \boldsymbol{W} \rangle \\ \text{subj. to} \quad & \boldsymbol{X}_{i,i} = 1, \quad i = 1, \cdots, n \\ & \boldsymbol{X} \succeq \boldsymbol{0} \\ & \text{rank}(\boldsymbol{X}) = 1 \end{aligned}$$

Example of lifting: Max-Cut

— Goemans, Williamson '95



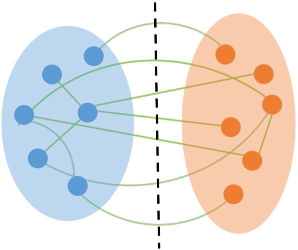
$$\begin{aligned} \text{maximize}_{\boldsymbol{x}} \quad & \boldsymbol{x}^\top \boldsymbol{W} \boldsymbol{x} \\ \text{subj. to} \quad & x_i^2 = 1, \quad i = 1, \cdots, n \end{aligned}$$

$\Downarrow$ let $\boldsymbol{X}$ be $\boldsymbol{x}\boldsymbol{x}^\top$

$$\begin{aligned} \text{maximize}_{\boldsymbol{X}} \quad & \langle \boldsymbol{X}, \boldsymbol{W} \rangle \\ \text{subj. to} \quad & \boldsymbol{X}_{i,i} = 1, \quad i = 1, \cdots, n \\ & \boldsymbol{X} \succeq \boldsymbol{0} \\ & \cancel{\text{rank}(\boldsymbol{X}) = 1} \end{aligned}$$

Example of lifting: Max-Cut

— Goemans, Williamson '95



$$\begin{array}{rl} \text{maximize}_{\boldsymbol{x}} & \boldsymbol{x}^\top \boldsymbol{W} \boldsymbol{x} \\ \text{subj. to} & x_i^2 = 1, \quad i = 1, \cdots, n \end{array}$$

$\Downarrow$ let $\boldsymbol{X}$ be $\boldsymbol{x}\boldsymbol{x}^\top$

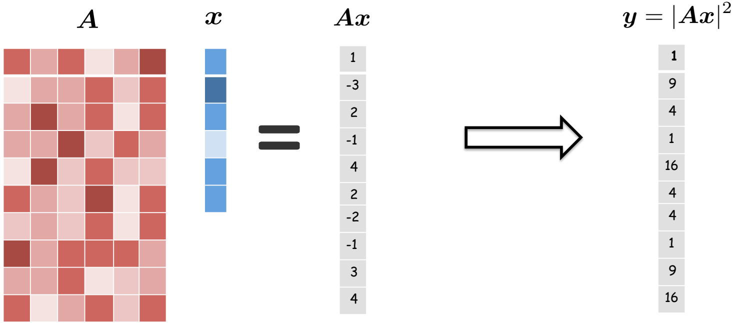
$$\begin{array}{rl} \text{maximize}_{\boldsymbol{X}} & \langle \boldsymbol{X}, \boldsymbol{W} \rangle \\ \text{subj. to} & \boldsymbol{X}_{i,i} = 1, \quad i = 1, \cdots, n \\ & \boldsymbol{X} \succeq \boldsymbol{0} \\ & \cancel{\text{rank}(\boldsymbol{X}) = 1} \end{array}$$

Problem: explosion in dimensions ($\mathbb{R}^n \to \mathbb{R}^{n \times n}$)

How about optimizing nonconvex problems directly without lifting?

A case study: solving random quadratic systems of equations

Solving quadratic systems of equations



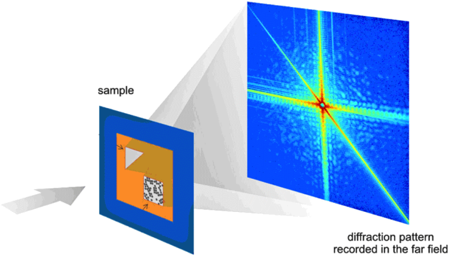
Solve for $\boldsymbol{x} \in \mathbb{C}^n$ in m quadratic equations

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

Motivation: a missing phase problem in imaging science

Detectors record intensities of diffracted rays

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

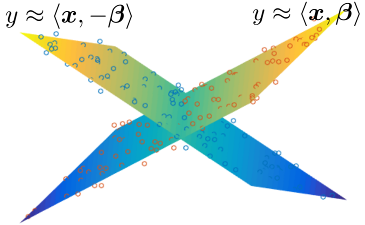


intensity of electrical field: $\left|\hat{x}(f_1, f_2)\right|^2 = \left|\int x(t_1, t_2) e^{-i2\pi(f_1 t_1 + f_2 t_2)} \mathrm{d}t_1 \mathrm{d}t_2\right|^2$

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

Motivation: latent variable models

Example: mixture of regression



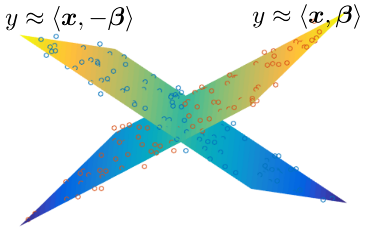
- Samples $\{(y_k, \boldsymbol{x}_k)\}$: drawn from one of two unknown regressors $\boldsymbol{\beta}$ and $-\boldsymbol{\beta}$

$$y_k \approx \begin{cases} \langle \boldsymbol{x}_k, \boldsymbol{\beta} \rangle, & \text{with prob. } 0.5 \\ \langle \boldsymbol{x}_k, -\boldsymbol{\beta} \rangle, & \text{else} \end{cases}$$

(labels: latent variables)

Motivation: latent variable models

Example: mixture of regression



- Samples $\{(y_k, \boldsymbol{x}_k)\}$: drawn from one of two unknown regressors $\boldsymbol{\beta}$ and $-\boldsymbol{\beta}$

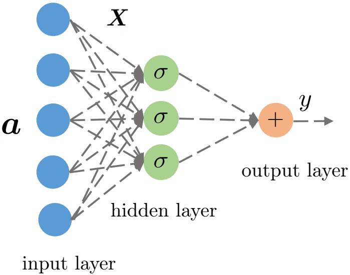
$$y_k \approx \begin{cases} \langle \boldsymbol{x}_k, \boldsymbol{\beta} \rangle, & \text{with prob. } 0.5 \\ \langle \boldsymbol{x}_k, -\boldsymbol{\beta} \rangle, & \text{else} \end{cases} \qquad \text{(labels: latent variables)}$$

— equivalent to observing $|y_k|^2 \approx |\langle \boldsymbol{x}_k, \boldsymbol{\beta} \rangle|^2$

- Goal: estimate $\boldsymbol{\beta}$

Motivation: learning neural nets with quadratic activation

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



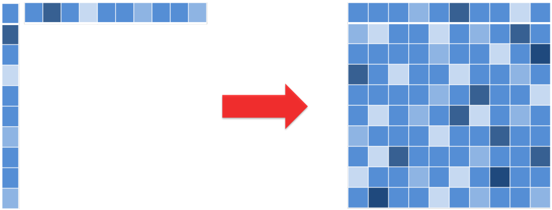
input features: $\boldsymbol{a}$; weights: $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_r]$

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

An equivalent view: low-rank factorization

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

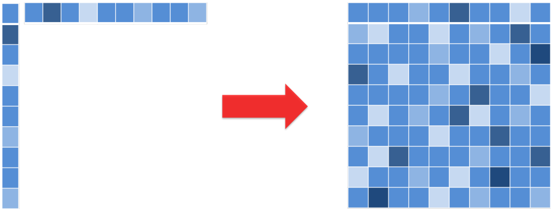
$$y_k = |\boldsymbol{a}_k^* \boldsymbol{x}|^2 = \boldsymbol{a}_k^* (\boldsymbol{x}\boldsymbol{x}^*) \boldsymbol{a}_k \qquad \Longrightarrow \qquad y_k = \boldsymbol{a}_k^* \boldsymbol{X} \boldsymbol{a}_k$$



An equivalent view: low-rank factorization

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

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

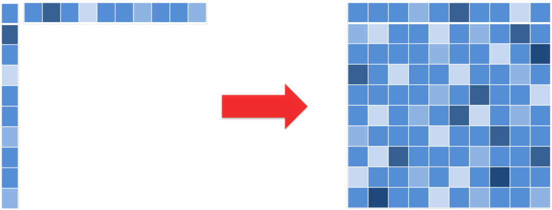


$$\begin{aligned} \text{find} \quad & \boldsymbol{X} \succeq \boldsymbol{0} \\ \text{s.t.} \quad & y_k = \boldsymbol{a}_k^* \boldsymbol{X} \boldsymbol{a}_k, \qquad k = 1, \cdots, m \\ & \operatorname{rank}(\boldsymbol{X}) = 1 \end{aligned}$$

An equivalent view: low-rank factorization

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

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

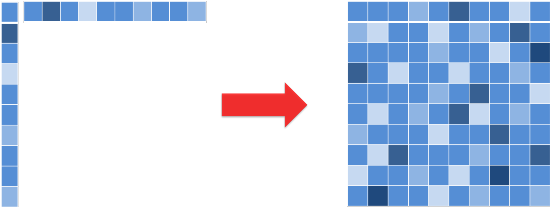


$$\begin{aligned} \text{find} \quad & \boldsymbol{X} \succeq \boldsymbol{0} \\ \text{s.t.} \quad & y_k = \boldsymbol{a}_k^* \boldsymbol{X} \boldsymbol{a}_k, \qquad k = 1, \cdots, m \\ & \cancel{\operatorname{rank}(\boldsymbol{X}) = 1} \end{aligned}$$

An equivalent view: low-rank factorization

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

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

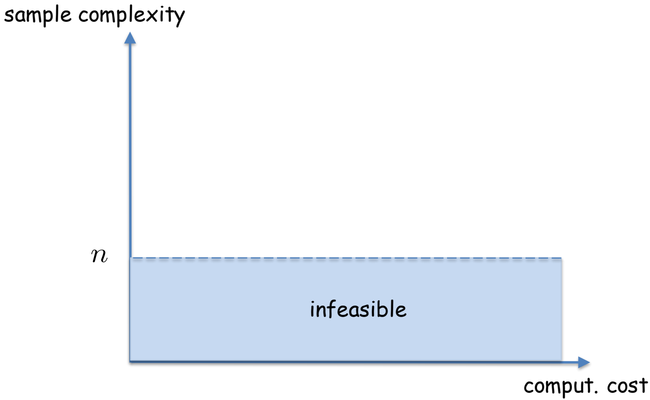


$$\begin{aligned} \text{find} \quad & \boldsymbol{X} \succeq \boldsymbol{0} \\ \text{s.t.} \quad & y_k \; = \; \boldsymbol{a}_k^* \boldsymbol{X} \boldsymbol{a}_k, \qquad k = 1, \cdots, m \\ & \cancel{\mathrm{rank}(\boldsymbol{X}) = 1} \end{aligned}$$

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

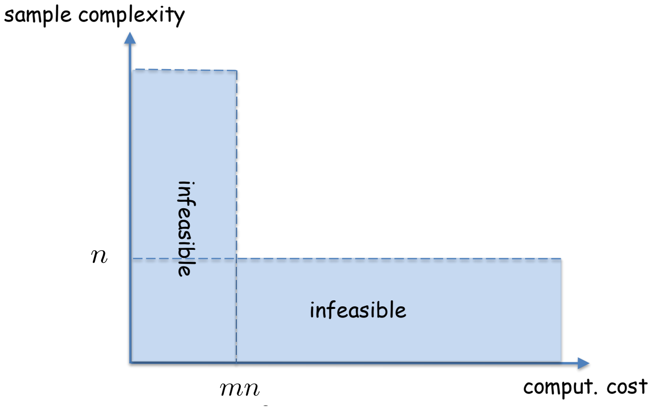
Prior art (before our work)

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



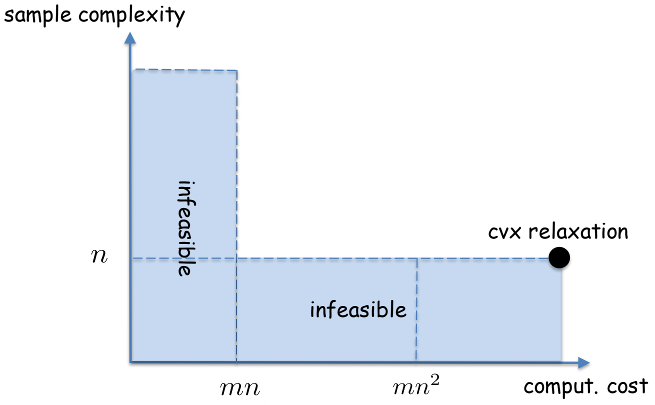
Prior art (before our work)

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



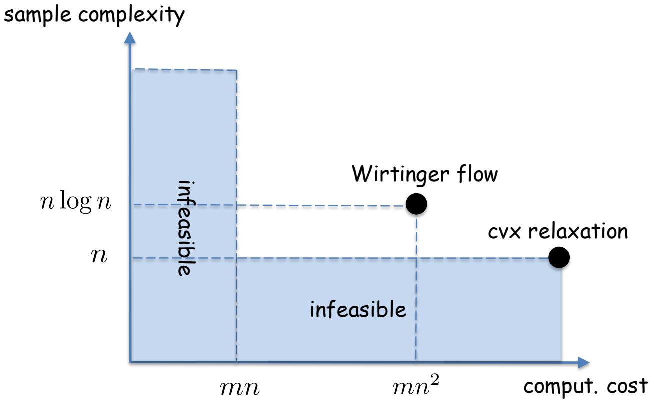
Prior art (before our work)

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



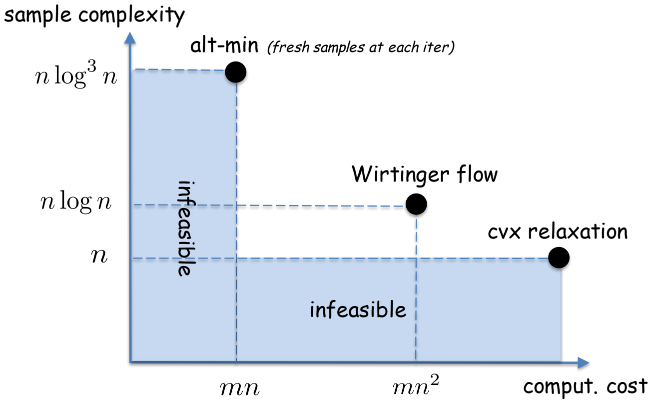
Prior art (before our work)

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



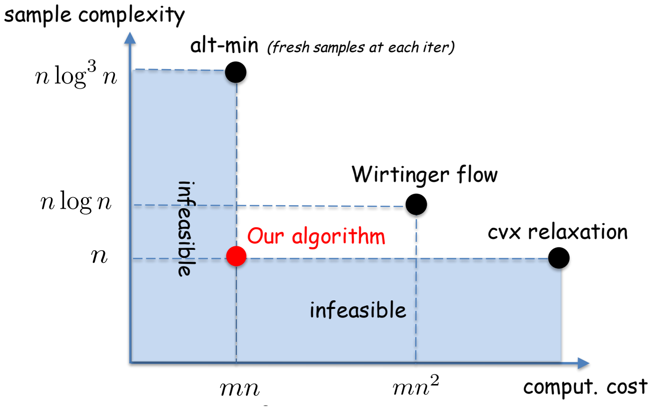
Prior art (before our work)

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

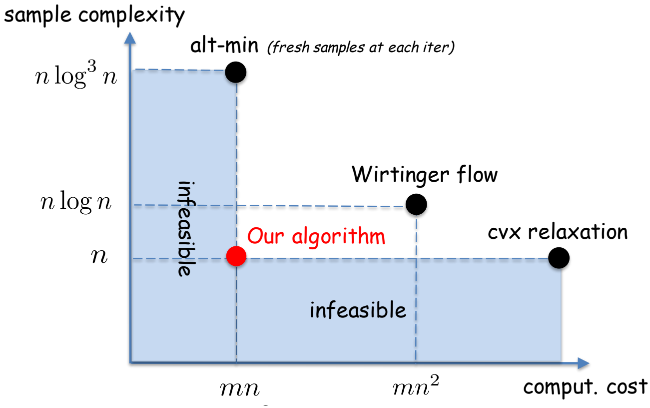
n: # unknowns; m: sample size (# eqns); $\boldsymbol{y} = |\boldsymbol{A}\boldsymbol{x}|^2,\ \boldsymbol{A} \in \mathbb{R}^{m \times n}$



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*

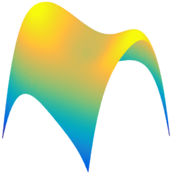
✓ *optimal statistical accuracy*

A first impulse: maximum likelihood estimate

$$\text{maximize}_{\boldsymbol{z}} \quad \ell(\boldsymbol{z}) = \quad \frac{1}{m} \sum\nolimits_{k=1}^{m} \ell_k(\boldsymbol{z})$$

A first impulse: maximum likelihood estimate

$$\text{maximize}_{\boldsymbol{z}} \quad \ell(\boldsymbol{z}) = \quad \frac{1}{m}\sum_{k=1}^{m} \ell_k(\boldsymbol{z})$$

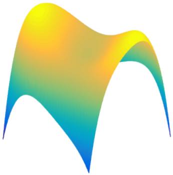


- Gaussian data: $\quad y_k \sim |\boldsymbol{a}_k^* \boldsymbol{x}|^2 + \mathcal{N}(0, \sigma^2)$

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

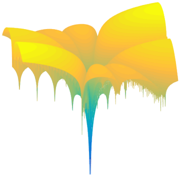
A first impulse: maximum likelihood estimate

$$\text{maximize}_{\boldsymbol{z}} \quad \ell(\boldsymbol{z}) = \quad \frac{1}{m}\sum_{k=1}^{m} \ell_k(\boldsymbol{z})$$



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

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



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

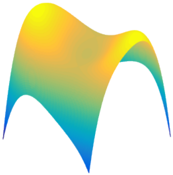
$$\ell_k(\boldsymbol{z}) = -|\boldsymbol{a}_k^* \boldsymbol{z}|^2 + y_k \log |\boldsymbol{a}_k^* \boldsymbol{z}|^2$$

A first impulse: maximum likelihood estimate

$$\text{maximize}_{\boldsymbol{z}} \quad \ell(\boldsymbol{z}) = \frac{1}{m}\sum_{k=1}^{m} \ell_k(\boldsymbol{z})$$

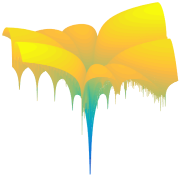
- Gaussian data: $y_k \sim |\boldsymbol{a}_k^* \boldsymbol{x}|^2 + \mathcal{N}(0, \sigma^2)$

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



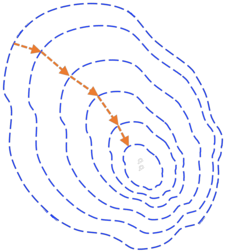
- Poisson data: $y_k \sim \text{Poisson}\big(|\boldsymbol{a}_k^* \boldsymbol{x}|^2\big)$

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



Problem: $-\ell$ nonconvex, many local stationary points

Wirtinger flow: Candès, Li, Soltanolkotabi '14



- **Spectral initialization:** $\boldsymbol{z}^0 \leftarrow$ leading eigenvector of

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

- **Iterative refinement:** for $t = 0, 1, \ldots$

$$\boldsymbol{z}^{t+1} = \boldsymbol{z}^t + \mu_t \nabla \ell(\boldsymbol{z}^t)$$

Already rich theory (see also Soltanolkotabi '14, Ma, Wang, Chi, Chen '17)

Interpretation of spectral initialization

Spectral initialization: $\boldsymbol{z}^0 \leftarrow$ leading eigenvector of

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

Interpretation of spectral initialization

Spectral initialization: $\boldsymbol{z}^0 \leftarrow$ leading eigenvector of

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

- Rationale: $\mathbb{E}[\boldsymbol{Y}] = \boldsymbol{I} + 2\boldsymbol{x}\boldsymbol{x}^*$ ($\|\boldsymbol{x}\|_2 = 1$) under Gaussian design

Interpretation of spectral initialization

Spectral initialization: $\boldsymbol{z}^0 \leftarrow$ leading eigenvector of

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

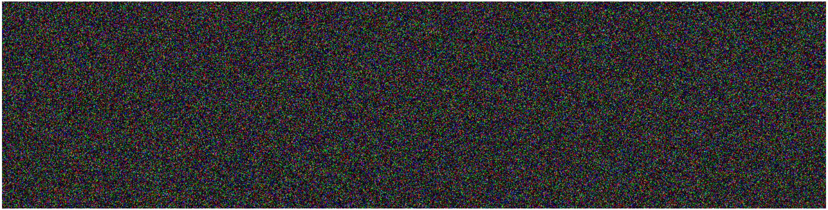
- Rationale: $\mathbb{E}[\boldsymbol{Y}] = \boldsymbol{I} + 2\boldsymbol{x}\boldsymbol{x}^*$ ($\|\boldsymbol{x}\|_2 = 1$) under Gaussian design
- Would succeed if $\boldsymbol{Y} \to \mathbb{E}[\boldsymbol{Y}]$

Empirical performance of initialization ($m = 12n$)

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

Empirical performance of initialization ($m = 12n$)

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



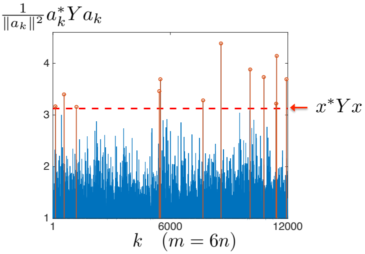
Spectral initialization

Improving initialization

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

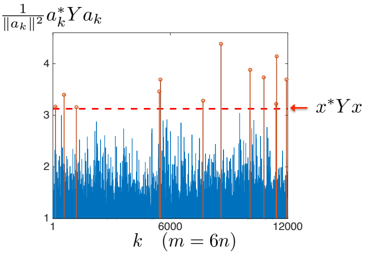
Improving initialization

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



Improving initialization

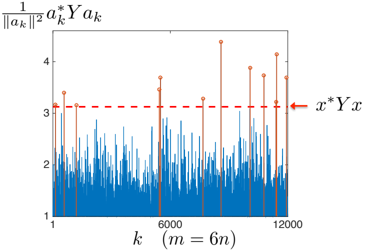
$$\boldsymbol{Y} = \frac{1}{m}\sum_k \underbrace{y_k \boldsymbol{a}_k \boldsymbol{a}_k^*}_{\text{heavy-tailed}} \quad \not\to \quad \mathbb{E}[\boldsymbol{Y}] \quad \text{unless } m \gg n$$



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

Improving initialization

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

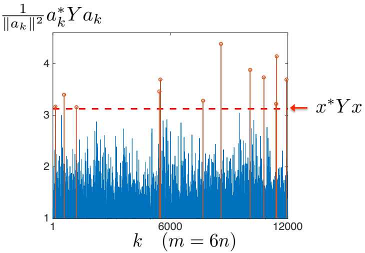


Problem large outliers $y_k = |\boldsymbol{a}_k^* \boldsymbol{x}|^2$ bear too much influence

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

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 $\frac{1}{m}\sum_k y_k \boldsymbol{a}_k \boldsymbol{a}_k^* \mathbf{1}_{\{|y_k| \lesssim \mathrm{Avg}\{|y_l|\}\}}$

— *improvable via more refined pre-processing*

(Wang, Giannakis, Eldar '16, Lu, Li '17, Mondelli, Montanari '17)

$$\frac{1}{m}\sum_k \rho(y_k) \boldsymbol{a}_k \boldsymbol{a}_k^* \qquad \text{e.g. } \rho(y_k) = \max\{y_k, a\}$$

Empirical performance of initialization ($m = 12n$)

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

Regularized spectral initialization

Iterative refinement stage: search directions

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

Iterative refinement stage: search directions

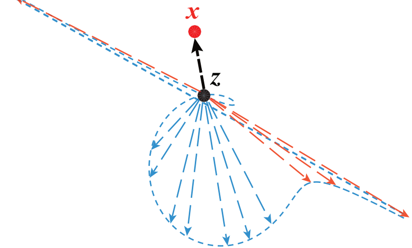
$$\text{Wirtinger flow:} \quad \boldsymbol{z}^{t+1} = \boldsymbol{z}^t - \frac{\mu_t}{m} \sum_{k=1}^{m} \underbrace{\left(y_k - |\boldsymbol{a}_k^\top \boldsymbol{z}^t|^2\right) \boldsymbol{a}_k \boldsymbol{a}_k^\top \boldsymbol{z}^t}_{= -\nabla \ell_k(\boldsymbol{z}^t)}$$

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



locus of $\{-\nabla \ell_k(\boldsymbol{z})\}$

Problem: descent direction has large variability

Our solution: variance reduction via proper trimming

More adaptive rule:

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

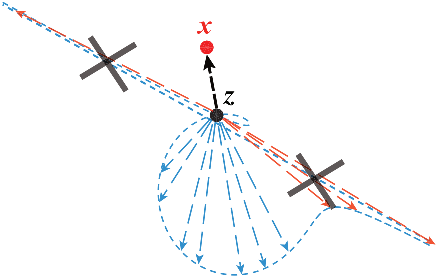
where $\mathcal{E}_1^i(\boldsymbol{z}) = \left\{ \alpha_z^{\mathsf{lb}} \leq \frac{|\boldsymbol{a}_i^\top \boldsymbol{z}|}{\|\boldsymbol{z}\|_2} \leq \alpha_z^{\mathsf{ub}} \right\};\ \mathcal{E}_2^i(\boldsymbol{z}) = \left\{ |y_i - |\boldsymbol{a}_i^\top \boldsymbol{z}|^2| \leq \frac{\frac{\alpha_h}{m}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

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

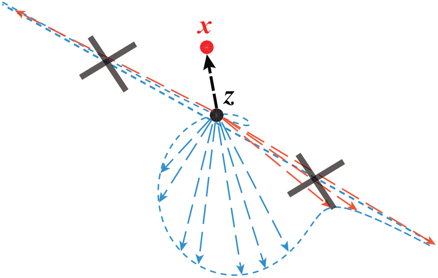


Our solution: variance reduction via proper trimming

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



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

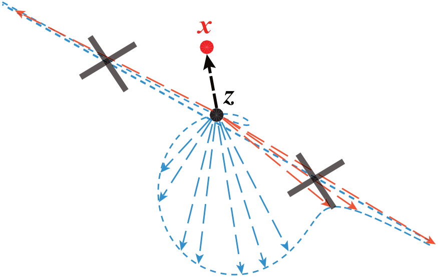
- $\mathcal{T}_t$ trims away excessively large grad components

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

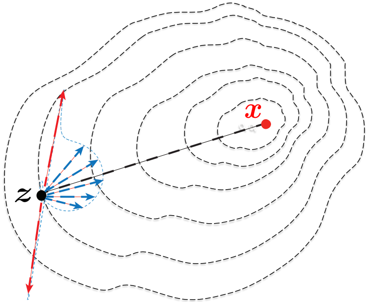


informally, $\boldsymbol{z}^{t+1} = \boldsymbol{z}^t + \frac{\mu}{m} \sum_{k \in \mathcal{T}_t} \nabla \ell_k(\boldsymbol{z}^t)$

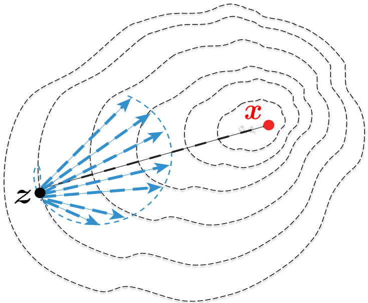
- $\mathcal{T}_t$ trims away excessively large grad components

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

Summary: truncated Wirtinger flows (TWF)

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

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

2. **Regularized gradient descent**

$$\boldsymbol{z}^{t+1} = \boldsymbol{z}^t + \mu_t \underbrace{\frac{1}{m}\sum\nolimits_{k\in\mathcal{T}_t}\nabla\ell_k(\boldsymbol{z}^t)}_{:=\nabla\ell^{\mathrm{tr}}(\boldsymbol{z}^t)}$$

Key idea: adaptively discard high-leverage data

Performance guarantees of TWF (noiseless data)

$$\text{dist}(\boldsymbol{z}, \boldsymbol{x}) := \min\{\|\boldsymbol{z} \pm \boldsymbol{x}\|_2\}$$

Theorem (Chen & Candès '15). Under i.i.d. Gaussian design, TWF achieves

$$\text{dist}(\boldsymbol{z}^t, \boldsymbol{x}) \lesssim (1-\rho)^t \, \|\boldsymbol{x}\|_2, \qquad t = 0, 1, \cdots$$

with high prob., provided that sample size $m \gtrsim n$. Here, $0 < \rho < 1$ is const.

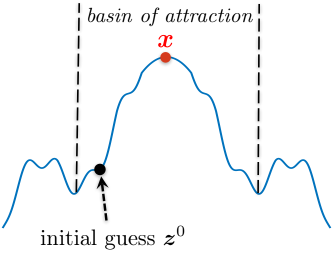
Performance guarantees of TWF (noiseless data)

$$\mathsf{dist}(\boldsymbol{z}, \boldsymbol{x}) := \min\{\|\boldsymbol{z} \pm \boldsymbol{x}\|_2\}$$

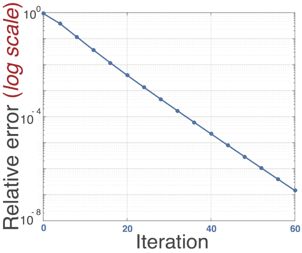
Theorem (Chen & Candès '15). Under i.i.d. Gaussian design, TWF achieves

$$\mathsf{dist}(\boldsymbol{z}^t, \boldsymbol{x}) \lesssim (1-\rho)^t \, \|\boldsymbol{x}\|_2, \qquad t = 0, 1, \cdots$$

with high prob., provided that sample size $m \gtrsim n$. Here, $0 < \rho < 1$ is const.



start within basin of attraction



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

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

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

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

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

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

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

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

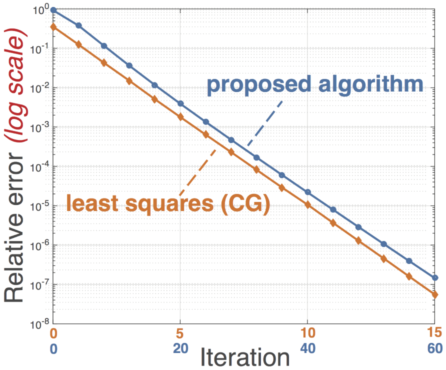
$$\boldsymbol{z}^{t+1} = \boldsymbol{z}^t + \frac{\mu_t}{m}\nabla\ell_{\mathsf{tr}}(\boldsymbol{z}^t)$$

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

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

Numerical surprise

- 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

After regularized spectral initialization

Empirical performance

After regularized spectral initialization

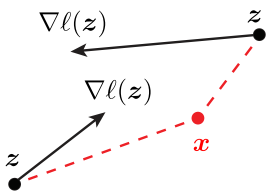
After 50 TWF iterations

Key convergence condition for gradient stage

If there are <u>many</u> samples:

$\forall \boldsymbol{z}$ s.t. $\mathrm{dist}(\boldsymbol{z}, \boldsymbol{x}) \leq \varepsilon \|\boldsymbol{x}\|_2$:

$$\langle \nabla \ell(\boldsymbol{z}),\ \boldsymbol{x} - \boldsymbol{z} \rangle \ \gtrsim\ \|\boldsymbol{z} - \boldsymbol{x}\|_2^2 + \|\nabla \ell(\boldsymbol{z})\|_2^2$$

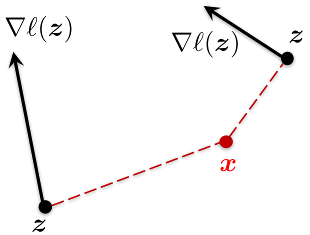


Key convergence condition for gradient stage

If there are **NOT** many samples, i.e. $m \asymp n$:

$\forall \boldsymbol{z}$ s.t. $\mathrm{dist}(\boldsymbol{z}, \boldsymbol{x}) \leq \varepsilon \|\boldsymbol{x}\|_2$:

$$\cancel{\langle \nabla \ell(\boldsymbol{z}),\ \boldsymbol{x} - \boldsymbol{z} \rangle \ \gtrsim\ \|\boldsymbol{z} - \boldsymbol{x}\|_2^2 + \|\nabla \ell(\boldsymbol{z})\|_2^2}$$

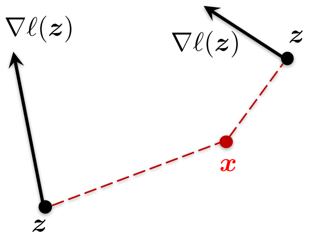


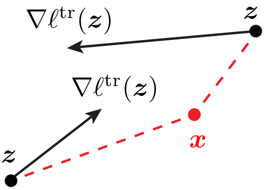
Key convergence condition for gradient stage

If there are **NOT** many samples, i.e. $m \asymp n$:

$\forall \boldsymbol{z}$ s.t. $\mathrm{dist}(\boldsymbol{z}, \boldsymbol{x}) \le \varepsilon \|\boldsymbol{x}\|_2$:

$$\langle \nabla \ell^{\mathrm{tr}}(\boldsymbol{z}),\ \boldsymbol{x} - \boldsymbol{z} \rangle \ \gtrsim\ \|\boldsymbol{z} - \boldsymbol{x}\|_2^2 + \|\nabla \ell^{\mathrm{tr}}(\boldsymbol{z})\|_2^2$$





Stability under noisy data

- Noisy data: $y_k = |\boldsymbol{a}_k^* \boldsymbol{x}|^2 + \eta_k$
- Signal-to-noise ratio:

$$\mathsf{SNR} := \frac{\sum_k |\boldsymbol{a}_k^* \boldsymbol{x}|^4}{\sum_k \eta_k^2} \approx \frac{3m\|\boldsymbol{x}\|^4}{\|\boldsymbol{\eta}\|^2}$$

- i.i.d. Gaussian design

Stability under noisy data

- Noisy data: $y_k = |\boldsymbol{a}_k^* \boldsymbol{x}|^2 + \eta_k$
- Signal-to-noise ratio:

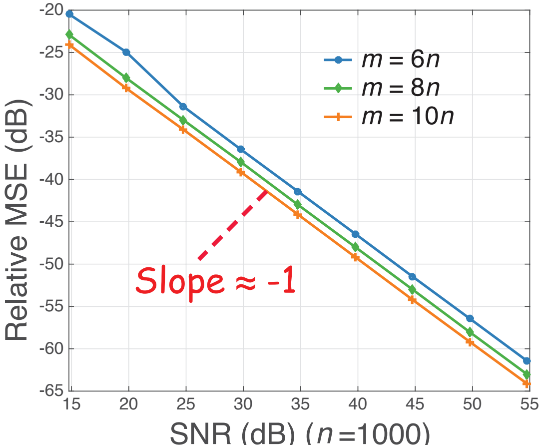
$$\mathsf{SNR} := \frac{\sum_k |\boldsymbol{a}_k^* \boldsymbol{x}|^4}{\sum_k \eta_k^2} \approx \frac{3m\|\boldsymbol{x}\|^4}{\|\boldsymbol{\eta}\|^2}$$

- i.i.d. Gaussian design

Theorem (Soltanolkotabi) WF converges to MLE

Theorem (Chen, Candès) Relative error of TWF converges to $O(\frac{1}{\sqrt{\mathsf{SNR}}})$

Relative MSE vs. SNR (Poisson data)



Empirical evidence: relative MSE scales inversely with SNR

This accuracy is nearly un-improvable (empirically)

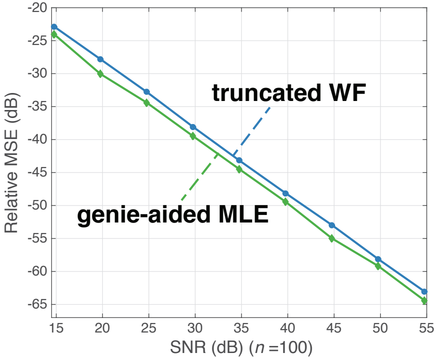
Comparison with genie-aided MLE (with sign info. revealed)

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

This accuracy is nearly un-improvable (empirically)

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

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



little empirical loss due to missing signs

This accuracy is nearly un-improvable (theoretically)

- Poisson data: $y_k \overset{\text{ind.}}{\sim} \text{Poisson}(\ |\boldsymbol{a}_k^* \boldsymbol{x}|^2\ )$
- Signal-to-noise ratio:

$$\text{SNR} \ \approx\ \frac{\sum_k |\boldsymbol{a}_k^* \boldsymbol{x}|^4}{\sum_k \text{Var}(y_k)} \ \approx\ 3\|\boldsymbol{x}\|^2$$

This accuracy is nearly un-improvable (theoretically)

- Poisson data: $y_k \overset{\text{ind.}}{\sim} \text{Poisson}(\ |\boldsymbol{a}_k^* \boldsymbol{x}|^2\ )$
- Signal-to-noise ratio:

$$\text{SNR} \ \approx\ \frac{\sum_k |\boldsymbol{a}_k^* \boldsymbol{x}|^4}{\sum_k \text{Var}(y_k)} \ \approx\ 3\|\boldsymbol{x}\|^2$$

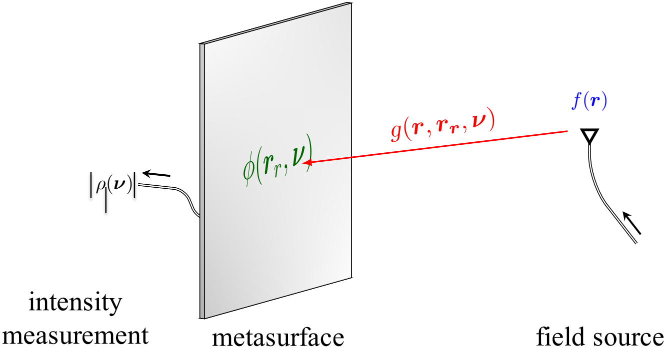
Theorem (Chen, Candès). Under i.i.d. Gaussian design, for any estimator $\hat{\boldsymbol{x}}$,

$$\inf_{\hat{\boldsymbol{x}}} \sup_{\boldsymbol{x}:\ \|\boldsymbol{x}\| \geq \log^{1.5} m} \frac{\mathbb{E}\left[\text{dist}\left(\hat{\boldsymbol{x}}, \boldsymbol{x}\right) \mid \{\boldsymbol{a}_k\}\right]}{\|\boldsymbol{x}\|} \ \gtrsim\ \frac{1}{\sqrt{\text{SNR}}},$$

provided that sample size $m \asymp n$.

Phaseless 3D computational imaging

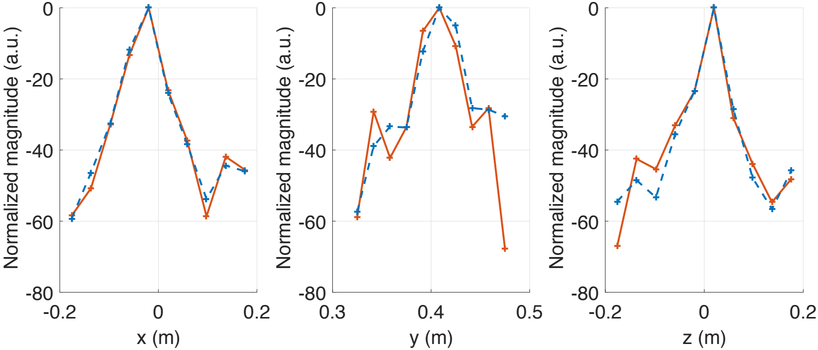
Fromenteze, Liu, Boyarsky, Gollub, & Smith '16



Measure intensities (with radiating metasurfaces) rather than complex signals for sub-centimeter wavelengths

Phaseless 3D computational imaging

Fromenteze, Liu, Boyarsky, Gollub, & Smith '16

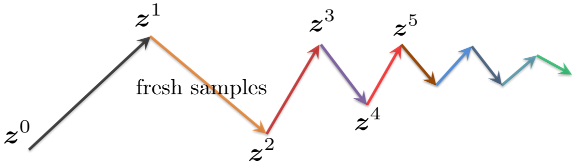


(red) phaseless reconstruction (blue) reconstruction w/ phase[1]

[1] This demonstration is proposed in microwave range as proof of concept

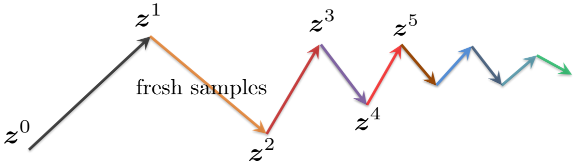
No need of sample splitting

- Several prior works use sample-splitting: require fresh samples at each iteration; not practical but much easier to analyze

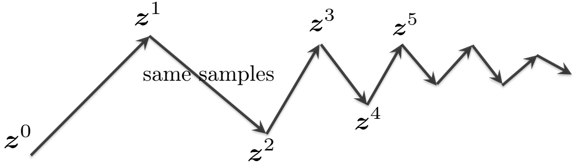


No need of sample splitting

- Several prior works use sample-splitting: require fresh samples at each iteration; not practical but much easier to analyze



- **Our works:** 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.)

Central message

- Simple nonconvex paradigms are surprisingly effective for computing MLE
- Importance of statistical thinking (initialization)

	statistical accuracy	comput. cost
convex relaxation		
nonconvex procedure		

- Y. Chen, E. Candès, "Solving random quadratic systems of equations is nearly as easy as solving linear systems," *Comm. Pure and Applied Math.*, 2017