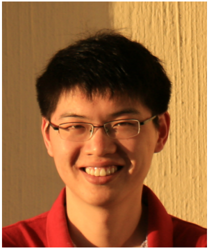
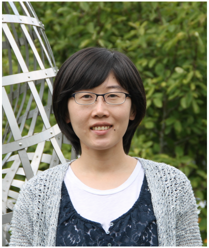
Random Initialization in Nonconvex Statistical Estimation

Yuxin Chen

Electrical Engineering, Princeton University



Cong Ma
Princeton ORFE



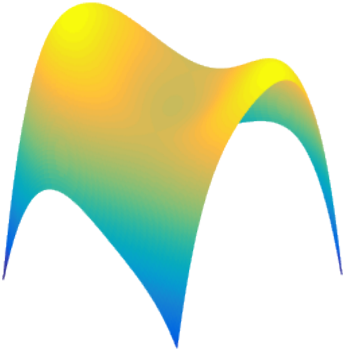
Yuejie Chi
CMU ECE

Jianqing Fan
Princeton ORFE

Nonconvex problems are everywhere

Empirical risk minimization is usually nonconvex

$$\text{minimize}_{\boldsymbol{x}} \quad f(\boldsymbol{x}; \text{data})$$

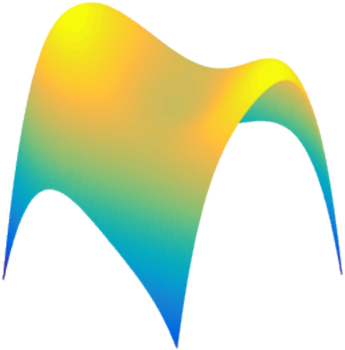


Nonconvex problems are everywhere

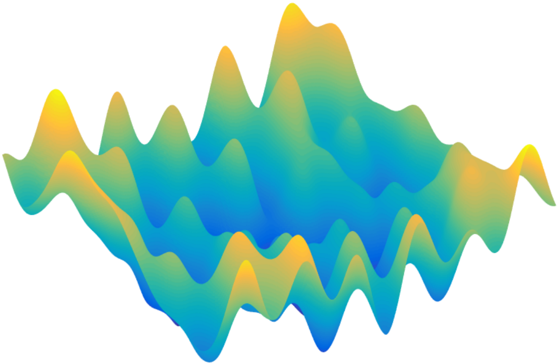
Empirical risk minimization is usually nonconvex

$$\text{minimize}_{\boldsymbol{x}} \quad f(\boldsymbol{x}; \text{data})$$

- low-rank matrix completion
- blind deconvolution
- dictionary learning
- mixture models
- deep neural nets
- ...



Nonconvex optimization may be super scary

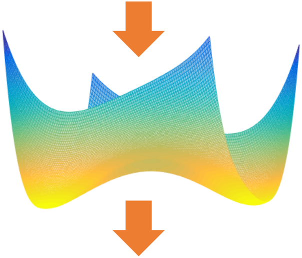


There may be bumps everywhere and exponentially many local optima

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

Statistical models come to rescue

statistical models



benign
landscape

tractable algorithms

When data are generated by certain statistical models, problems might be much nicer than worst-case instances

Example: low-rank matrix recovery

$$\underset{\boldsymbol{U}\in\mathbb{R}^{n\times r}}{\text{minimize}} \quad f(\boldsymbol{U}) := \sum_{i=1}^{m} \big(\langle \boldsymbol{A}_i, \boldsymbol{U}\boldsymbol{U}^\top\rangle - \langle \boldsymbol{A}_i, \boldsymbol{U}^\star\boldsymbol{U}^{\star\top}\rangle\big)^2$$

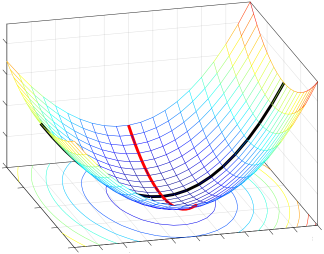
where entries of $\boldsymbol{A}_i$ are i.i.d. Gaussian

Example: low-rank matrix recovery

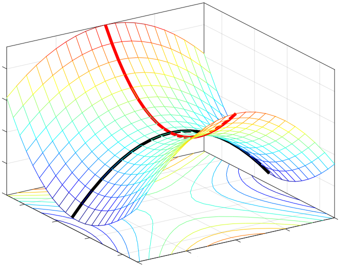
$$\underset{\boldsymbol{U}\in\mathbb{R}^{n\times r}}{\text{minimize}} \quad f(\boldsymbol{U}) := \sum_{i=1}^{m} \left(\langle \boldsymbol{A}_i, \boldsymbol{U}\boldsymbol{U}^\top \rangle - \langle \boldsymbol{A}_i, \boldsymbol{U}^\star \boldsymbol{U}^{\star\top} \rangle \right)^2$$

where entries of $\boldsymbol{A}_i$ are i.i.d. Gaussian

- *no spurious local minima* under large enough sample size (Bhojanapalli, Srebro '16)



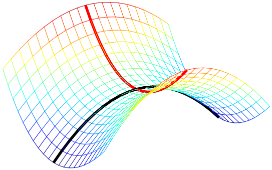
global minimum

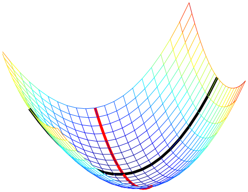


saddle point

Separation of landscape analysis and generic algorithm design

landscape analysis (statistics)

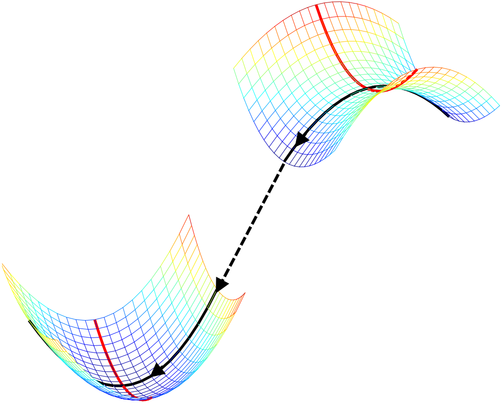




Separation of landscape analysis and generic algorithm design

landscape analysis
(statistics)

generic algorithms
(optimization)



Separation of landscape analysis and generic algorithm design

landscape analysis (statistics)

generic algorithms (optimization)

- 2-layer linear neural network (Baldi, Hornik '89)
- dictionary learning (Sun et al. '15)
- phase retrieval (Sun et al. '16, Davis et al. '17)
- matrix completion (Ge et al. '16, Chen et al. '17)
- matrix sensing (Bhojanapalli et al. '16, Li et al. '16)
- empirical risk mininimization (Mei et al. '16)
- synchronization (Bandeira et al. '16)
- robust PCA (Ge et al. '17)
- inverting deep neural nets (Hand et al. '17)
- 1-hidden-layer neural nets (Ge et al. '17)
- blind deconvolution (Zhang et al. '18, Li et al. '18)
- ...

- cubic regularization (Nesterov, Polyak '06)
- gradient descent (Lee et al. '16)
- trust region method (Sun et al. '16)
- Carmon et al. '16
- perturbed GD (Jin et al. '17)
- perturbed accelerated GD (Jin et al. '17)
- Agarwal et al. '17
- Natasha (Allen-Zhu '17)
- ...

Separation of landscape analysis and generic algorithm design

landscape analysis
(statistics)

generic algorithms
(optimization)

- 2-layer linear neural network (Baldi, Hornik '89)
- dictionary learning (Sun et al. '15)
- phase retrieval (Sun et al. '16, Davis et al. '17)
- matrix completion (Ge et al. '16, Chen et al. '17)
- matrix sensing (Bhojanapalli et al. '16, Li et al. '16)
- empirical risk mininimization (Mei et al. '16)
- synchronization (Bandeira et al. '16)
- robust PCA (Ge et al. '17)
- inverting deep neural nets (Hand et al. '17)
- 1-hidden-layer neural nets (Ge et al. '17)
- blind deconvolution (Zhang et al. '18, Li et al. '18)
- ...

- cubic regularization (Nesterov, Polyak '06)
- gradient descent (Lee et al. '16)
- trust region method (Sun et al. '16)
- Carmon et al. '16
- perturbed GD (Jin et al. '17)
- perturbed accelerated GD (Jin et al. '17)
- Agarwal et al. '17
- Natasha (Allen-Zhu '17)
- ...

Issue: conservative computational guarantees for specific problems
(e.g. solving quadratic systems, matrix completion)

This talk: blending landscape and convergence analysis

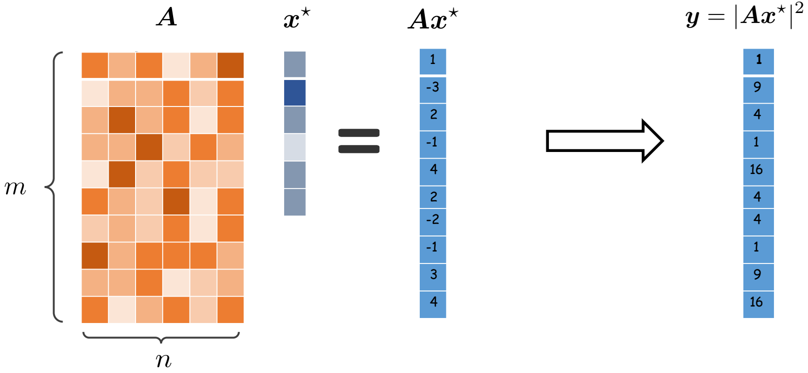
This talk: blending landscape and convergence analysis

$\Downarrow$

Even simplest possible nonconvex methods
can be remarkably efficient under suitable statistical models

A case study: solving random quadratic systems of equations

Solving quadratic systems of equations



Estimate $\boldsymbol{x}^\star \in \mathbb{R}^n$ from m random quadratic measurements

$$y_k = \left(\boldsymbol{a}_k^\top \boldsymbol{x}^\star\right)^2 + \text{noise}, \qquad k = 1, \ldots, m$$

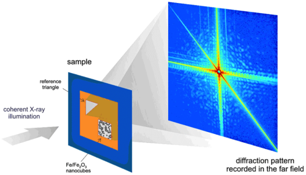
assume w.l.o.g. $\|\boldsymbol{x}^\star\|_2 = 1$

Motivation: phase retrieval

Detectors record intensities of diffracted rays

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

Fig credit: Stanford SLAC



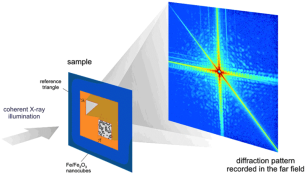
intensity of electrical field: $\left|\widehat{x}(f_1, f_2)\right|^2 = \left|\int x(t_1, t_2) e^{-i2\pi(f_1 t_1 + f_2 t_2)} \mathrm{d}t_1 \mathrm{d}t_2\right|^2$

Motivation: phase retrieval

Detectors record intensities of diffracted rays

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

Fig credit: Stanford SLAC

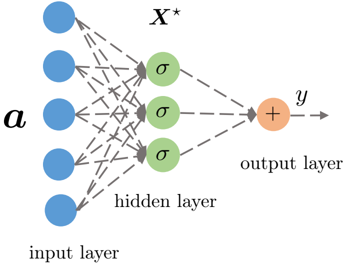


intensity of electrical field: $\left|\widehat{x}(f_1, f_2)\right|^2 = \left|\int x(t_1, t_2) e^{-i2\pi(f_1 t_1 + f_2 t_2)} \mathrm{d}t_1 \mathrm{d}t_2\right|^2$

Phase retrieval: recover signal $x(t_1, t_2)$ from intensity $\left|\widehat{x}(f_1, f_2)\right|^2$

Motivation: learning neural nets with quadratic activation

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

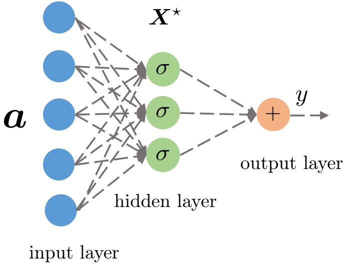


input features: $\boldsymbol{a}$; weights: $\boldsymbol{X}^\star = [\boldsymbol{x}_1^\star, \cdots, \boldsymbol{x}_r^\star]$

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

Motivation: learning neural nets with quadratic activation

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

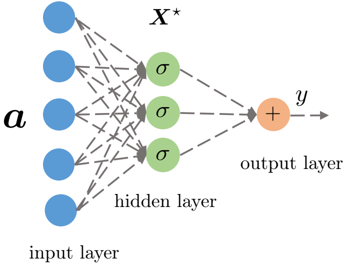


input features: $\boldsymbol{a}$; weights: $\boldsymbol{X}^\star = [\boldsymbol{x}_1^\star, \cdots, \boldsymbol{x}_r^\star]$

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

Motivation: learning neural nets with quadratic activation

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



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

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

We consider simplest model when $r = 1$

A natural least squares formulation

$$\text{minimize}_{\boldsymbol{x}\in\mathbb{R}^n} \quad f(\boldsymbol{x}) = \frac{1}{4m}\sum_{k=1}^{m}\left[\left(\boldsymbol{a}_k^\top \boldsymbol{x}\right)^2 - y_k\right]^2$$

A natural least squares formulation

$$\text{minimize}_{\boldsymbol{x}\in\mathbb{R}^n} \quad f(\boldsymbol{x}) = \frac{1}{4m}\sum_{k=1}^{m}\left[\left(\boldsymbol{a}_k^\top \boldsymbol{x}\right)^2 - y_k\right]^2$$

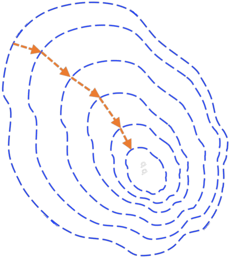
- **issue:** $f(\cdot)$ is highly nonconvex
  $\longrightarrow$ *computationally challenging!*

Wirtinger flow (Candès, Li, Soltanolkotabi '14)

$$\text{minimize}_{\boldsymbol{x}} \quad f(\boldsymbol{x}) = \frac{1}{4m} \sum_{k=1}^{m} \Big[\big(\boldsymbol{a}_k^\top \boldsymbol{x} \big)^2 - y_k \Big]^2$$

Wirtinger flow (Candès, Li, Soltanolkotabi '14)

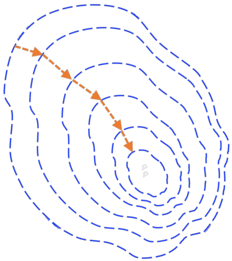
$$\text{minimize}_{\boldsymbol{x}} \quad f(\boldsymbol{x}) = \frac{1}{4m} \sum_{k=1}^{m} \left[\left(\boldsymbol{a}_k^\top \boldsymbol{x} \right)^2 - y_k \right]^2$$



- **spectral initialization:** $\boldsymbol{x}^0 \leftarrow$ leading eigenvector of certain data matrix

Wirtinger flow (Candès, Li, Soltanolkotabi '14)

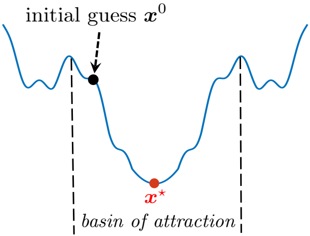
$$\text{minimize}_{\boldsymbol{x}} \quad f(\boldsymbol{x}) = \frac{1}{4m} \sum_{k=1}^{m} \left[\left(\boldsymbol{a}_k^\top \boldsymbol{x} \right)^2 - y_k \right]^2$$



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

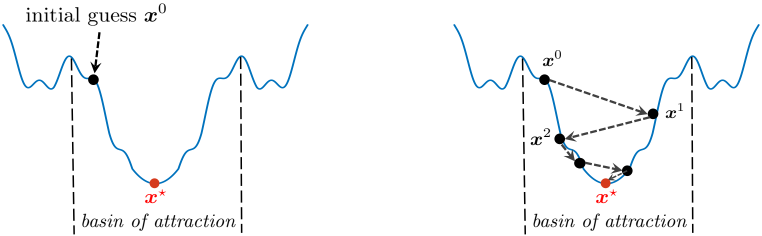
$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta_t \nabla f(\boldsymbol{x}^t), \qquad t = 0, 1, \cdots$$

Rationale of two-stage approach



1. initialize within $\underbrace{\text{local basin sufficiently close to } \boldsymbol{x}^\star}_{\text{(restricted) strongly convex; no saddles / spurious local mins}}$

Rationale of two-stage approach



1. initialize within $\underbrace{\text{local basin sufficiently close to } \boldsymbol{x}^\star}_{\text{(restricted) strongly convex; no saddles / spurious local mins}}$

2. iterative refinement

A highly incomplete list of two-stage methods

phase retrieval:

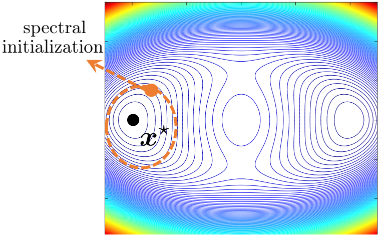
- Netrapalli, Jain, Sanghavi '13
- Candès, Li, Soltanolkotabi '14
- Chen, Candès '15
- Cai, Li, Ma '15
- Wang, Giannakis, Eldar '16
- Zhang, Zhou, Liang, Chi '16
- Kolte, Ozgur '16
- Zhang, Chi, Liang '16
- Soltanolkotabi '17
- Vaswani, Nayer, Eldar '16
- Chi, Lu '16
- Wang, Zhang, Giannakis, Akcakaya, Chen '16
- Tan, Vershynin '17
- Ma, Wang, Chi, Chen '17
- Duchi, Ruan '17
- Jeong, Gunturk '17
- Yang, Yang, Fang, Zhao, Wang, Neykov '17
- Qu, Zhang, Wright '17
- Goldstein, Studer '16
- Bahmani, Romberg '16
- Hand, Voroninski '16
- Wang, Giannakis, Saad, Chen '17
- Barmherzig, Sun '17
- ...

other problems:

- Keshavan, Montanari, Oh '09
- Sun, Luo '14
- Chen, Wainwright '15
- Tu, Boczar, Simchowitz, Soltanolkotabi, Recht '15
- Zheng, Lafferty '15
- Balakrishnan, Wainwright, Yu '14
- Chen, Suh '15
- Chen, Candès '16
- Li, Ling, Strohmer, Wei '16
- Yi, Park, Chen, Caramanis '16
- Jin, Kakade, Netrapalli '16
- Huang, Kakade, Kong, Valiant '16
- Ling, Strohmer '17
- Li, Ma, Chen, Chi '18
- Aghasi, Ahmed, Hand '17
- Lee, Tian, Romberg '17
- Li, Chi, Zhang, Liang '17
- Cai, Wang, Wei '17
- Abbe, Bandeira, Hall '14
- Chen, Kamath, Suh, Tse '16
- Zhang, Zhou '17
- Boumal '16
- Zhong, Boumal '17
- ...

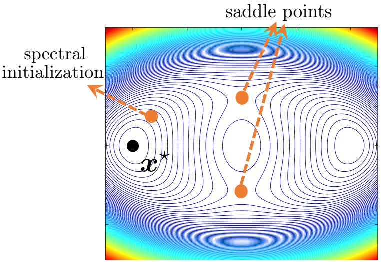
Is carefully-designed initialization necessary for fast convergence?

Initialization



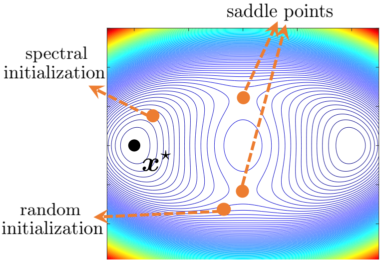
- spectral initialization gets us to (restricted) strongly cvx region

Initialization



- spectral initialization gets us to (restricted) strongly cvx region
- cannot initialize GD anywhere, e.g. might get stuck at saddles

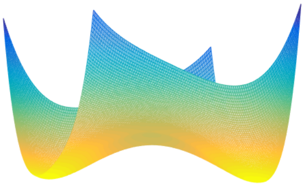
Initialization



- spectral initialization gets us to (restricted) strongly cvx region
- cannot initialize GD anywhere, e.g. might get stuck at saddles

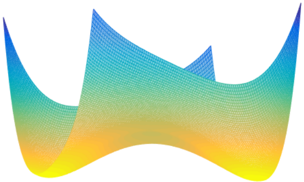
Can we initialize GD randomly, which is simpler and model-agnostic?

What does prior theory say?



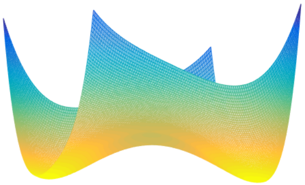
- **landscape:** no spurious local mins (Sun, Qu, Wright '16)

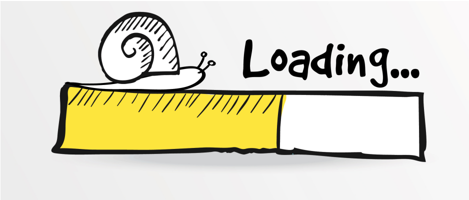
What does prior theory say?



- **landscape:** no spurious local mins (Sun, Qu, Wright '16)
- randomly initialized GD converges almost surely (Lee et al. '16)

What does prior theory say?



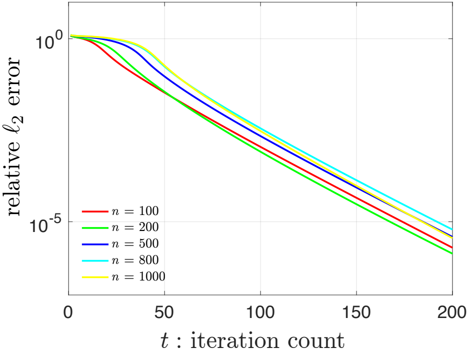


- **landscape:** no spurious local mins (Sun, Qu, Wright '16)
- randomly initialized GD converges almost surely (Lee et al. '16)

"almost surely" might mean "take forever"

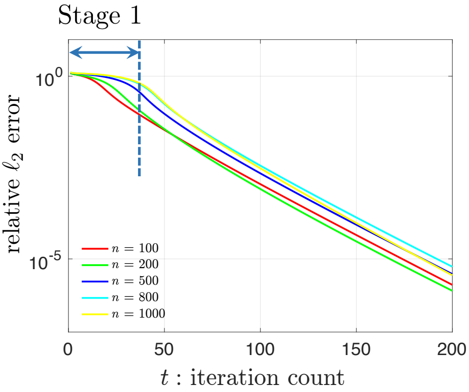
Numerical efficiency of randomly initialized GD

$$\eta = 0.1,\ \boldsymbol{a}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n),\ m = 10n,\ \boldsymbol{x}^0 \sim \mathcal{N}(\boldsymbol{0}, n^{-1}\boldsymbol{I}_n)$$



Numerical efficiency of randomly initialized GD

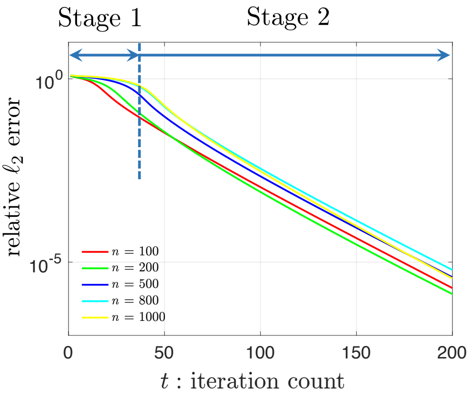
$$\eta = 0.1,\ \boldsymbol{a}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n),\ m = 10n,\ \boldsymbol{x}^0 \sim \mathcal{N}(\boldsymbol{0}, n^{-1}\boldsymbol{I}_n)$$



Randomly initialized GD enters local basin within **tens of iterations**

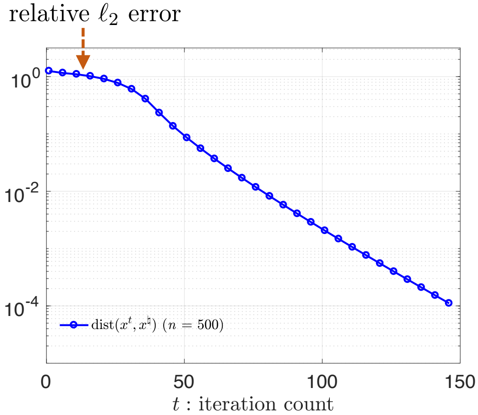
Numerical efficiency of randomly initialized GD

$$\eta = 0.1,\ \boldsymbol{a}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n),\ m = 10n,\ \boldsymbol{x}^0 \sim \mathcal{N}(\boldsymbol{0}, n^{-1}\boldsymbol{I}_n)$$

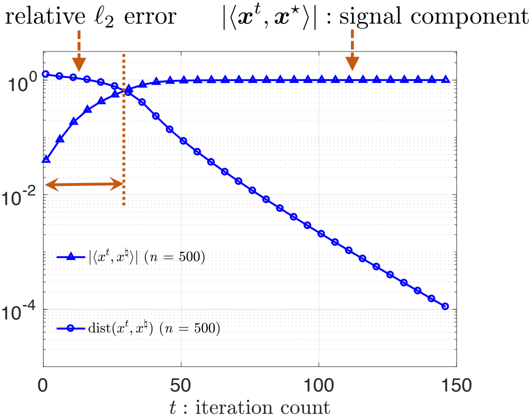


Randomly initialized GD enters local basin within **tens of iterations**

Exponential growth of signal strength in Stage 1



Exponential growth of signal strength in Stage 1



Numerically, a few iterations suffice for entering local region

Our theory: noiseless case

These numerical findings can be formalized when $\boldsymbol{a}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)$:

Our theory: noiseless case

These numerical findings can be formalized when $\boldsymbol{a}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)$:

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

Theorem 1 (Chen, Chi, Fan, Ma '18)

Under i.i.d. Gaussian design, GD with $\boldsymbol{x}^0 \sim \mathcal{N}(\mathbf{0}, n^{-1}\boldsymbol{I}_n)$ achieves

Our theory: noiseless case

These numerical findings can be formalized when $\boldsymbol{a}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n)$:

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

Theorem 1 (Chen, Chi, Fan, Ma '18)

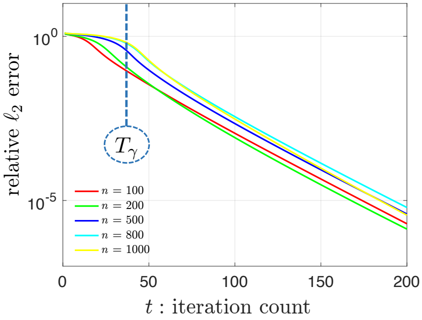
Under i.i.d. Gaussian design, GD with $\boldsymbol{x}^0 \sim \mathcal{N}(\boldsymbol{0}, n^{-1}\boldsymbol{I}_n)$ *achieves*

$$\mathrm{dist}(\boldsymbol{x}^t, \boldsymbol{x}^\star) \le \gamma(1-\rho)^{t-T_\gamma}\|\boldsymbol{x}^\star\|_2, \qquad t \ge T_\gamma$$

with high prob. for $T_\gamma \lesssim \log n$ *and some constants* $\gamma, \rho > 0$, *provided that step size* $\eta \asymp 1$ *and sample size* $m \gtrsim n\,\mathrm{poly}\log m$

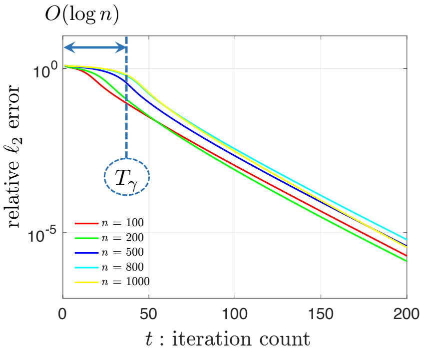
Our theory: noiseless case

$$\mathrm{dist}(\boldsymbol{x}^t, \boldsymbol{x}^\star) \le \gamma(1-\rho)^{t-T_\gamma}\|\boldsymbol{x}^\star\|_2, \quad t \ge T_\gamma \asymp \log n$$



Our theory: noiseless case

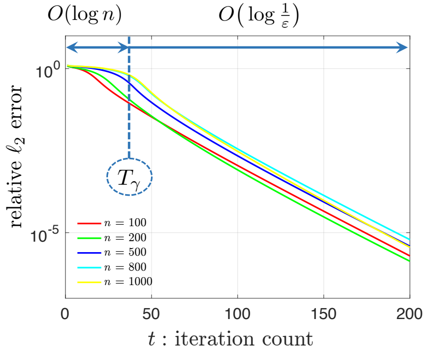
$$\mathrm{dist}(\boldsymbol{x}^t, \boldsymbol{x}^\star) \le \gamma(1-\rho)^{t-T_\gamma}\|\boldsymbol{x}^\star\|_2, \quad t \ge T_\gamma \asymp \log n$$



- *Stage 1:* takes $O(\log n)$ iterations to reach $\mathrm{dist}(\boldsymbol{x}^t, \boldsymbol{x}^\star) \le \gamma$ (e.g. $\gamma = 0.1$)

Our theory: noiseless case

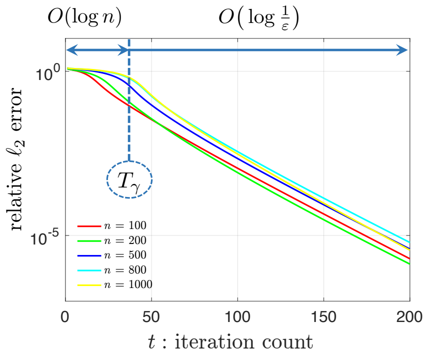
$$\mathrm{dist}(\boldsymbol{x}^t, \boldsymbol{x}^\star) \le \gamma(1-\rho)^{t-T_\gamma} \|\boldsymbol{x}^\star\|_2, \quad t \ge T_\gamma \asymp \log n$$



- *Stage 1:* takes $O(\log n)$ iterations to reach $\mathrm{dist}(\boldsymbol{x}^t, \boldsymbol{x}^\star) \le \gamma$ (e.g. $\gamma = 0.1$)
- *Stage 2:* linear (geometric) convergence

Our theory: noiseless case

$$\mathrm{dist}(\boldsymbol{x}^t, \boldsymbol{x}^\star) \le \gamma(1-\rho)^{t-T_\gamma}\|\boldsymbol{x}^\star\|_2, \quad t \ge T_\gamma \asymp \log n$$

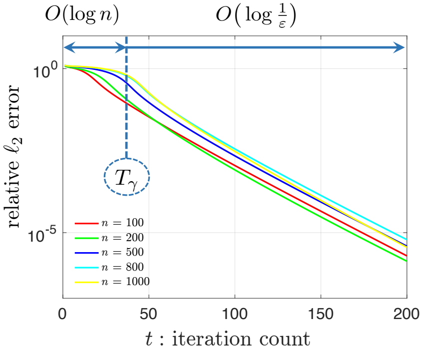


- *near-optimal computational cost*:

  — $O\left(\log n + \log \frac{1}{\varepsilon}\right)$ iterations to yield ε accuracy

Our theory: noiseless case

$$\mathrm{dist}(\boldsymbol{x}^t, \boldsymbol{x}^\star) \leq \gamma(1-\rho)^{t-T_\gamma}\|\boldsymbol{x}^\star\|_2, \quad t \geq T_\gamma \asymp \log n$$

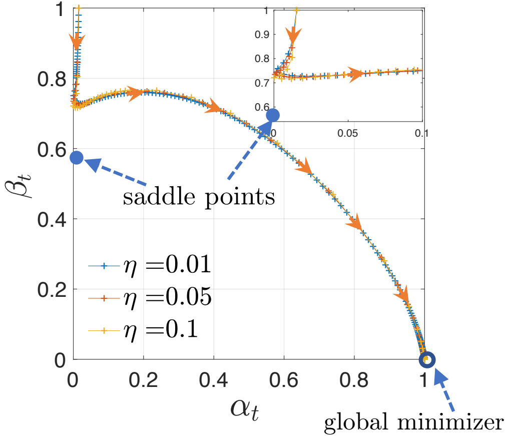


- *near-optimal computational cost*:

  — $O\left(\log n + \log \frac{1}{\varepsilon}\right)$ iterations to yield ε accuracy

- *near-optimal sample size*: $m \gtrsim n\mathsf{poly}\log m$

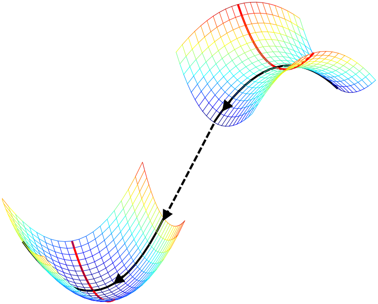
Automatic saddle avoidance



Randomly initialized GD never hits saddle points!

Other saddle-escaping schemes based on generic landscape analysis

	iteration complexity
trust-region (Sun et al. '16)	$n^7 + \log\log\frac{1}{\varepsilon}$
perturbed GD (Jin et al. '17)	$n^3 + n\log\frac{1}{\varepsilon}$
perturbed accelerated GD (Jin et al. '17)	$n^{2.5} + \sqrt{n}\log\frac{1}{\varepsilon}$
GD (ours) (Chen et al. '18)	$\log n + \log\frac{1}{\varepsilon}$



Generic optimization theory yields highly suboptimal convergence guarantees

A bit of analysis

What if we have infinite samples?

$$\textit{Gaussian designs: } \boldsymbol{a}_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{I}_n), \quad 1 \le k \le m$$

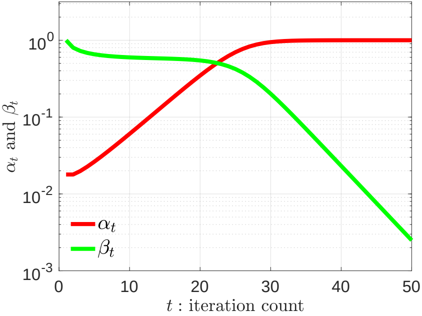
Population level (infinite samples)

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta \nabla F(\boldsymbol{x}^t),$$

where

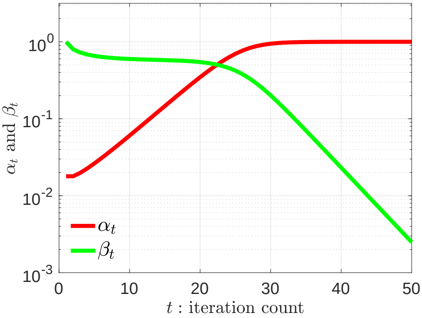
$$\nabla F(\boldsymbol{x}) := \mathbb{E}[\nabla f(\boldsymbol{x})] = (3\|\boldsymbol{x}\|_2^2 - 1)\boldsymbol{x} - 2(\boldsymbol{x}^{\star\top}\boldsymbol{x})\boldsymbol{x}^\star$$

Population-level state evolution



Let $\alpha_t := \underbrace{\left|\langle \boldsymbol{x}^t, \boldsymbol{x}^\star \rangle\right|}_{\text{signal strength}}$ and $\beta_t = \underbrace{\left\| \boldsymbol{x}^t - \langle \boldsymbol{x}^t, \boldsymbol{x}^\star \rangle \boldsymbol{x}^\star \right\|_2}_{\text{size of residual component}}$, then

Population-level state evolution



Let $\alpha_t := \underbrace{|\langle \boldsymbol{x}^t, \boldsymbol{x}^\star \rangle|}_{\text{signal strength}}$ and $\beta_t = \underbrace{\|\boldsymbol{x}^t - \langle \boldsymbol{x}^t, \boldsymbol{x}^\star \rangle \boldsymbol{x}^\star\|_2}_{\text{size of residual component}}$, then

$$\alpha_{t+1} = \{1 + 3\eta[1 - (\alpha_t^2 + \beta_t^2)]\}\alpha_t$$
$$\beta_{t+1} = \{1 + \eta[1 - 3(\alpha_t^2 + \beta_t^2)]\}\beta_t$$

2-parameter dynamics

Back to finite-sample analysis

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

Back to finite-sample analysis

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta \nabla f(\boldsymbol{x}^t) = \boldsymbol{x}^t - \eta \nabla F(\boldsymbol{x}^t) - \underbrace{\eta\big(\nabla f(\boldsymbol{x}^t) - \nabla F(\boldsymbol{x}^t)\big)}_{\text{residual}}$$

Back to finite-sample analysis

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta \nabla f(\boldsymbol{x}^t) = \boldsymbol{x}^t - \eta \nabla F(\boldsymbol{x}^t) - \underbrace{\eta\big(\nabla f(\boldsymbol{x}^t) - \nabla F(\boldsymbol{x}^t)\big)}_{\text{residual}}$$

— take one term in $\boldsymbol{x}^{\star\top}\big(\nabla f(\boldsymbol{x}^t) - \nabla F(\boldsymbol{x}^t)\big)$ for example:

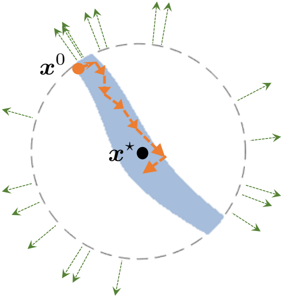
$$r_1 := \frac{1}{m} \sum_i \left(\boldsymbol{a}_{i,\perp}^\top \boldsymbol{x}_\perp^t\right)^3 a_{i,1}$$

Back to finite-sample analysis

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta \nabla f(\boldsymbol{x}^t) = \boldsymbol{x}^t - \eta \nabla F(\boldsymbol{x}^t) - \underbrace{\eta\big(\nabla f(\boldsymbol{x}^t) - \nabla F(\boldsymbol{x}^t)\big)}_{\text{residual}}$$

— take one term in $\boldsymbol{x}^{\star\top}\big(\nabla f(\boldsymbol{x}^t) - \nabla F(\boldsymbol{x}^t)\big)$ for example:

$$r_1 := \frac{1}{m}\sum_i \left(\boldsymbol{a}_{i,\perp}^\top \boldsymbol{x}_\perp^t\right)^3 a_{i,1}$$



a region with
well-controlled residual

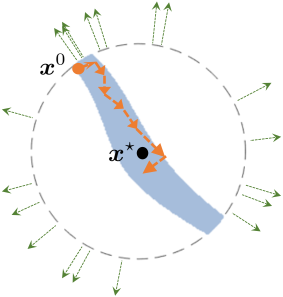
- $\underbrace{r_1 \asymp \frac{1}{\sqrt{m}}}_{\text{desired level}}$ if $\boldsymbol{x}^t$ is independent of $\{\boldsymbol{a}_l\}$

Back to finite-sample analysis

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta \nabla f(\boldsymbol{x}^t) = \boldsymbol{x}^t - \eta \nabla F(\boldsymbol{x}^t) - \underbrace{\eta\big(\nabla f(\boldsymbol{x}^t) - \nabla F(\boldsymbol{x}^t)\big)}_{\text{residual}}$$

— take one term in $\boldsymbol{x}^{\star\top}\big(\nabla f(\boldsymbol{x}^t) - \nabla F(\boldsymbol{x}^t)\big)$ for example:

$$r_1 := \frac{1}{m}\sum_i \big(\boldsymbol{a}_{i,\perp}^\top \boldsymbol{x}_\perp^t\big)^3 a_{i,1}$$



a region with
well-controlled residual

- $\underbrace{r_1 \asymp \frac{1}{\sqrt{m}}}_{\text{desired level}}$ if $\boldsymbol{x}^t$ is independent of $\{\boldsymbol{a}_l\}$
- **key analysis ingredient:** show $\boldsymbol{x}^t$ is "nearly-independent" of (some part of) $\{\boldsymbol{a}_l\}$

Key proof idea: leave-one-out analysis

Leave out a small amount of information from data and re-run GD

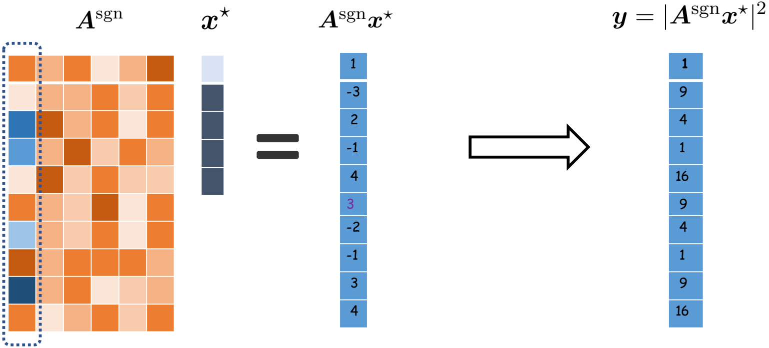
Key proof idea: leave-one-out analysis

Leave out a small amount of information from data and re-run GD

- Stein '72
- El Karoui, Bean, Bickel, Lim, Yu '13
- El Karoui '15
- Javanmard, Montanari '15
- Zhong, Boumal '17
- Lei, Bickel, El Karoui '17
- Sur, Chen, Candès '17
- Abbe, Fan, Wang, Zhong '17
- Chen, Fan, Ma, Wang '17

Key proof idea: leave-one-out analysis

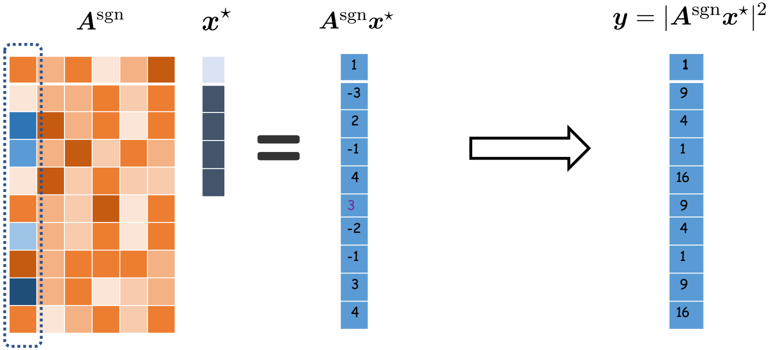
Leave out a small amount of information from data and re-run GD



- generate $\boldsymbol{A}^{\mathrm{sgn}}$ by randomly flipping $\mathrm{sgn}(a_{i,1})$, $\forall i$

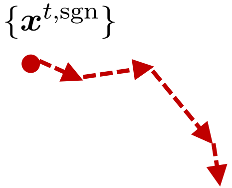
Key proof idea: leave-one-out analysis

Leave out a small amount of information from data and re-run GD



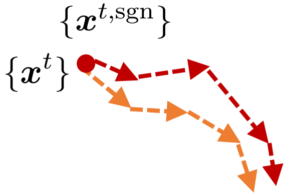
- generate $\boldsymbol{A}^{\mathrm{sgn}}$ by randomly flipping $\mathrm{sgn}(a_{i,1})$, $\forall i$
- generate auxiliary iterates $\{\boldsymbol{x}^{t,\mathrm{sgn}}\}$ by re-running GD w.r.t. $\boldsymbol{A}^{\mathrm{sgn}}$

Key proof idea: leave-one-out analysis



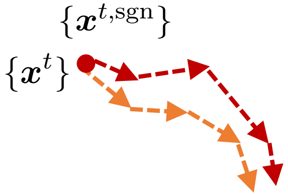
- Auxiliary iterate $\boldsymbol{x}^{t,\mathrm{sgn}}$ is independent of $\{\mathsf{sgn}(a_{i,1})\}$

Key proof idea: leave-one-out analysis



- Auxiliary iterate $\boldsymbol{x}^{t,\mathrm{sgn}}$ is independent of $\{\mathsf{sgn}(a_{i,1})\}$
- Auxiliary iterate $\boldsymbol{x}^{t,\mathrm{sgn}} \ \approx\ $ true iterate $\boldsymbol{x}^{t}$

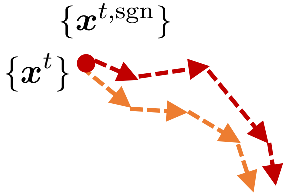
Key proof idea: leave-one-out analysis



- Auxiliary iterate $\boldsymbol{x}^{t,\mathrm{sgn}}$ is independent of $\{\mathsf{sgn}(a_{i,1})\}$
- Auxiliary iterate $\boldsymbol{x}^{t,\mathrm{sgn}} \approx$ true iterate $\boldsymbol{x}^{t}$

$$\Longrightarrow \quad \boldsymbol{x}^{t} \text{ is nearly independent of } \{\mathsf{sgn}(a_{i,1})\}$$

Key proof idea: leave-one-out analysis



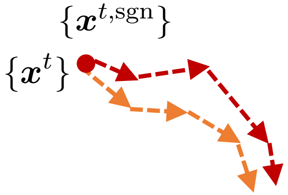
- Auxiliary iterate $\boldsymbol{x}^{t,\mathrm{sgn}}$ is independent of $\{\mathsf{sgn}(a_{i,1})\}$
- Auxiliary iterate $\boldsymbol{x}^{t,\mathrm{sgn}} \;\approx\;$ true iterate $\boldsymbol{x}^{t}$

$$\Longrightarrow \quad \boldsymbol{x}^{t} \text{ is nearly independent of } \{\mathsf{sgn}(a_{i,1})\}$$

- This makes it easy to control

$$r_1 = \tfrac{1}{m}\sum_i \left(\boldsymbol{a}_{i,\perp}^{\top}\boldsymbol{x}_{\perp}^{t}\right)^3 a_{i,1}$$

Key proof idea: leave-one-out analysis



- Auxiliary iterate $\boldsymbol{x}^{t,\mathrm{sgn}}$ is independent of $\{\mathsf{sgn}(a_{i,1})\}$
- Auxiliary iterate $\boldsymbol{x}^{t,\mathrm{sgn}} \;\approx\;$ true iterate $\boldsymbol{x}^{t}$

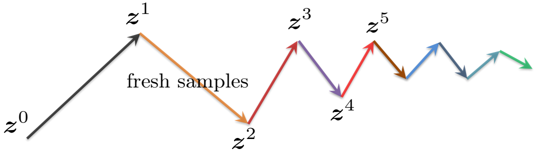
$$\Longrightarrow \quad \boldsymbol{x}^{t} \text{ is nearly independent of } \{\mathsf{sgn}(a_{i,1})\}$$

- This makes it easy to control

$$r_1 = \frac{1}{m}\sum_i \left(\boldsymbol{a}_{i,\perp}^{\top}\boldsymbol{x}_{\perp}^{t}\right)^3 |a_{i,1}|\, \mathsf{sgn}(a_{i,1})$$

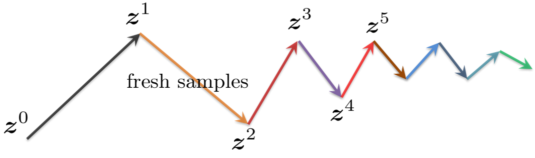
No need of sample splitting

- Several prior works use sample-splitting: require fresh samples at each iteration; not practical but helps analysis

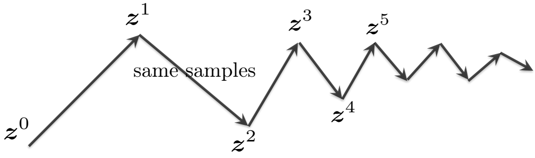


No need of sample splitting

- Several prior works use sample-splitting: require fresh samples at each iteration; not practical but helps analysis



- **This work:** reuses all samples in all iterations



Concluding remarks

Even simplest nonconvex methods
are remarkably efficient under suitable statistical models

smart initialization	sample splitting	saddle escaping
		

1. “Gradient descent with random initialization: ...”, Y. Chen, Y. Chi, J. Fan, C. Ma, accepted to Mathematical Programming

2. “Implicit regularization in nonconvex statistical estimation: ...”, C. Ma, K. Wang, Y. Chi, Y. Chen, arXiv:1711.10467

3. “Nonconvex optimization meets low-rank matrix factorization: An overview”, Y. Chi, Y. Lu, Y. Chen, arXiv:1809.09573