Random Initialization and Implicit Regularization in Nonconvex Statistical Estimation

Yuxin Chen

Electrical Engineering, Princeton University
Nonconvex problems are everywhere

Empirical risk minimization is usually nonconvex

\[
\text{minimize}_{x} \quad f(x; \text{data})
\]
Nonconvex problems are everywhere

Empirical risk minimization is usually nonconvex

\[
\minimize_x \quad f(x; \text{data})
\]

- 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)
Nonconvex optimization may be super scary

But they are solved on a daily basis via simple algorithms like (stochastic) gradient descent
When data are generated by certain statistical models, problems are often much nicer than worst-case instances.

— *Nonconvex Optimization Meets Low-Rank Matrix Factorization: An Overview*  
*Chi, Lu, Chen ’18*
Example: low-rank matrix recovery

\[
\minimize_{U \in \mathbb{R}^{n \times r}} \quad f(U) := \sum_{i=1}^{m} \left( \langle A_i, UU^\top \rangle - \langle A_i, \star \star^\top \rangle \right)^2
\]

where entries of \( A_i \) are i.i.d. Gaussian
Example: low-rank matrix recovery

\[
\text{minimize } \quad f(U) := \sum_{i=1}^{m} \left( \langle A_i, UU^\top \rangle - \langle A_i, U^*U^*\rangle \right)^2
\]

where entries of \( A_i \) are i.i.d. Gaussian

- **no spurious local minima** under large enough sample size (Bhojanapalli et al. ’16)
Separation of landscape analysis and generic algorithm design

landscape analysis (statistics)

- 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 minimization (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)
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)
- 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 minimization (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)
- ...

### Generic Algorithms (Optimization)
- 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)
Separation of landscape analysis and generic algorithm design

landscape analysis (statistics)

- 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 minimization (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)
- ...

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

generic algorithms (optimization)

- 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)
- ...

7/ 44
This talk: blending landscape and convergence analysis
This talk: blending landscape and convergence analysis

Even **simplest** possible nonconvex methods can be surprisingly **efficient** under suitable statistical models.

A case study: solving random quadratic systems of equations
Solving quadratic systems of equations

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

$$y_k = (\mathbf{a}_k^\top \mathbf{x}^*)^2 + \text{noise}, \quad k = 1, \ldots, m$$

*assume w.l.o.g. $\|\mathbf{x}^*\|_2 = 1*
Motivation: phase retrieval

Detectors record **intensities** of diffracted rays

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

*Fig credit: Stanford SLAC*

intensity of electrical field: $|\hat{x}(f_1, f_2)|^2 = \left|\int x(t_1, t_2)e^{-i2\pi(f_1 t_1 + f_2 t_2)} dt_1 dt_2\right|^2$
Motivation: phase retrieval

Detectors record intensities of diffracted rays

• electric field \(x(t_1, t_2)\) \(\rightarrow\) Fourier transform \(\hat{x}(f_1, f_2)\)

\[\text{Fig credit: Stanford SLAC}\]

intensity of electrical field:
\[
|\hat{x}(f_1, f_2)|^2 = \left| \int x(t_1, t_2)e^{-i2\pi(f_1 t_1 + f_2 t_2)} \, dt_1 dt_2 \right|^2
\]

Phase retrieval: recover signal \(x(t_1, t_2)\) from intensity \(|\hat{x}(f_1, f_2)|^2\)
Motivation: learning neural nets with quadratic activation

— Soltanolkotabi, Javanmard, Lee ’17, Li, Ma, Zhang ’17

input features: $\mathbf{a}$; weights: $X^* = [x_1^*, \ldots, x_r^*]$

output: $y = \sum_{i=1}^{r} \sigma(\mathbf{a}^\top x_i^*)$
Motivation: learning neural nets with quadratic activation

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

input features: \( \mathbf{a} \); weights: \( \mathbf{X}^* = [\mathbf{x}_1^*, \ldots, \mathbf{x}_r^*] \)

output: \( y = \sum_{i=1}^{r} \sigma(\mathbf{a} \top \mathbf{x}_i^*) \sigma(z) = z^2 = \sum_{i=1}^{r} (\mathbf{a} \top \mathbf{x}_i^*)^2 \)
Motivation: learning neural nets with quadratic activation

— Soltanolkotabi, Javanmard, Lee ’17, Li, Ma, Zhang ’17

input features: \( \mathbf{a} \); weights: \( X^* = [\mathbf{x}_1^*, \cdots, \mathbf{x}_r^*] \)

output: \( y = \sum_{i=1}^{r} \sigma(\mathbf{a}^\top \mathbf{x}_i^*) \sigma(z) = z^2 = \sum_{i=1}^{r} (\mathbf{a}^\top \mathbf{x}_i^*)^2 \)

We consider simplest model when \( r = 1 \)
A natural least squares formulation

\[
\text{minimize}_{x \in \mathbb{R}^n} \quad f(x) = \frac{1}{4m} \sum_{k=1}^{m} \left[ (a_k^T x)^2 - y_k \right]^2
\]
A natural least squares formulation

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

- **Issue:** \( f(\cdot) \) is highly nonconvex
  \[ \rightarrow \text{computationally challenging!} \]
Wirtinger flow (Candès, Li, Soltanolkotabi ’14)

\[
\text{minimize } x \quad f(x) = \frac{1}{4m} \sum_{k=1}^{m} \left[ (a_k^\top x)^2 - y_k \right]^2
\]
Wirtinger flow (Candès, Li, Soltanolkotabi ’14)

\[
\text{minimize}_x \quad f(x) = \frac{1}{4m} \sum_{k=1}^{m} \left[ (a_k^T x)^2 - y_k \right]^2
\]

- **spectral initialization:** \( x^0 \leftarrow \) leading eigenvector of certain data matrix
Wirtinger flow (Candès, Li, Soltanolkotabi ’14)

\[
\text{minimize}_x \quad f(x) = \frac{1}{4m} \sum_{k=1}^{m} \left[ (a_k^\top x)^2 - y_k \right]^2
\]

- **spectral initialization**: \( x^0 \leftarrow \) leading eigenvector of certain data matrix

- **gradient descent**:

\[
x^{t+1} = x^t - \eta_t \nabla f(x^t), \quad t = 0, 1, \cdots
\]
1. initialize within local basin sufficiently close to $x^*$

(restricted) strongly convex; no saddles / spurious local mins
Rationale of two-stage approach

1. initialize within local basin sufficiently close to \( x^* \)
   (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
- ...
Is carefully-designed initialization necessary for fast convergence?
• spectral initialization gets us to (restricted) strongly convex region
• 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

\[ \eta = 0.1, \ a_i \sim \mathcal{N}(0, I_n), \ m = 10n, \ x^0 \sim \mathcal{N}(0, n^{-1} I_n) \]
Numerical efficiency of randomly initialized GD

\[ \eta = 0.1, \ a_i \sim N(0, I_n), \ m = 10n, \ x^0 \sim N(0, n^{-1} I_n) \]

Randomly initialized GD enters local basin within tens of iterations
Numerical efficiency of randomly initialized GD

\[ \eta = 0.1, \quad a_i \sim \mathcal{N}(0, I_n), \quad m = 10n, \quad x^0 \sim \mathcal{N}(0, n^{-1} I_n) \]

Randomly initialized GD enters local basin within tens of iterations
Exponential growth of signal strength in Stage 1

relative $\ell_2$ error

Numerically, a few iterations suffice for entering local region
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 $a_i \sim \mathcal{N}(0, I_n)$:
Our theory: noiseless case

These numerical findings can be formalized when $a_i \overset{i.i.d.}{\sim} \mathcal{N}(0, I_n)$:

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

**Theorem 1 (Chen, Chi, Fan, Ma ’18)**

*Under i.i.d. Gaussian design, GD with $x^0 \sim \mathcal{N}(0, n^{-1} I_n)$ achieves*
Our theory: noiseless case

These numerical findings can be formalized when $a_i \overset{i.i.d.}{\sim} \mathcal{N}(0, I_n)$:

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

**Theorem 1 (Chen, Chi, Fan, Ma ’18)**

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

$$\text{dist}(x^t, x^*) \leq \gamma(1 - \rho)^{t - T_\gamma} \|x^*\|_2, \quad t \geq T_\gamma$$

*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 $m \gtrsim n \text{ polylog } m$*
Our theory: noiseless case

\[ \text{dist}(\mathbf{x}^t, \mathbf{x}^*) \leq \gamma (1 - \rho)^{t - T\gamma}\|\mathbf{x}^*\|_2, \quad t \geq T\gamma \asymp \log n \]
Our theory: noiseless case

\[ \text{dist}(x_t, x^*) \leq \gamma (1 - \rho)^{t - T_{\gamma}} \| x^* \|_2, \quad t \geq T_{\gamma} \asymp \log n \]

- **Stage 1**: takes \( O(\log n) \) iterations to reach \( \text{dist}(x_t, x^*) \leq \gamma \)
  (e.g. \( \gamma = 0.1 \))
Our theory: noiseless case

\[
\text{dist}(\mathbf{x}^t, \mathbf{x}^*) \leq \gamma (1 - \rho)^{t-T\gamma} \|\mathbf{x}^*\|_2, \quad t \geq T\gamma \approx \log n
\]

- **Stage 1**: takes \(O(\log n)\) iterations to reach \(\text{dist}(\mathbf{x}^t, \mathbf{x}^*) \leq \gamma\) (e.g. \(\gamma = 0.1\))
- **Stage 2**: linear (geometric) convergence
Our theory: noiseless case

\[ \text{dist}(\mathbf{x}^t, \mathbf{x}^*) \leq \gamma(1 - \rho)^{t - T\gamma} \|\mathbf{x}^*\|_2, \quad t \geq T\gamma \approx \log n \]

- near-optimal computational cost:
  - \( O(\log n + \log \frac{1}{\varepsilon}) \) iterations to yield \( \varepsilon \) accuracy
Our theory: noiseless case

\[
\text{dist}(x^t, x^*) \leq \gamma (1 - \rho)^{t - T\gamma} \|x^*\|_2, \quad t \geq T\gamma \simeq \log n
\]

- **near-optimal computational cost:**
  - \(O(\log n + \log \frac{1}{\varepsilon})\) iterations to yield \(\varepsilon\) accuracy

- **near-optimal sample size:** \(m \gtrsim n\text{poly} \log m\)
Stability vis-a-vis noise

\[ y_k = |a_k^\top x^\star|^2 + \epsilon_k, \quad \epsilon_k \sim \mathcal{N}(0, \sigma^2) \quad k = 1, \ldots, m \]
Stability vis-a-vis noise

\[ y_k = |a_k^\top \mathbf{x}^*|^2 + \epsilon_k, \quad \epsilon_k \sim \mathcal{N}(0, \sigma^2) \quad k = 1, \ldots, m \]

- randomly initialized GD converges to maximum likelihood estimate in \( O(\log n + \log \frac{1}{\varepsilon}) \) iterations
Stability vis-a-vis noise

\[ y_k = |a_k^\top x^*|^2 + \epsilon_k, \quad \epsilon_k \sim \mathcal{N}(0, \sigma^2) \quad k = 1, \ldots, m \]

- randomly initialized GD converges to maximum likelihood estimate in \( O(\log n + \log \frac{1}{\varepsilon}) \) iterations
- minimax optimal
Experiments on images

- coded diffraction patterns
- $\mathbf{x}^* \in \mathbb{R}^{256 \times 256}$
- $m/n = 12$
GD with random initialization

\[ x^t \]
GD iterate

*use Adobe to see animation*
GD with random initialization

\[ x^t \]
GD iterate

\[ \langle x^t, x^\ast \rangle x^\ast \]
signal component

\[ x^t - \langle x^t, x^\ast \rangle x^\ast \]
perpendicular component

use Adobe to see animation
Stage 1: random initialization $\rightarrow$ local region

<table>
<thead>
<tr>
<th>iteration complexity</th>
<th>prior theory based on global landscape</th>
<th>our theory</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>almost surely (Lee et al. '16)</td>
<td>$O(\log n)$</td>
</tr>
</tbody>
</table>
What if we have infinite samples?

Gaussian designs: $a_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_n), \quad 1 \leq k \leq m$

Population level (infinite samples)

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

where

$$\nabla F(x) := \mathbb{E}[\nabla f(x)] = (3\|x\|_2^2 - 1)x - 2(x^\top x)x^*$$
Let \( \alpha_t := \left\| \langle x^t, x^* \rangle \right\| \) and \( \beta_t = \left\| x^t - \langle x^t, x^* \rangle x^* \right\|_2 \), then
Let $\alpha_t := \langle x^t, x^* \rangle$ and $\beta_t = \|x^t - \langle x^t, x^* \rangle x^*\|_2$, 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

\[ x^{t+1} = x^t - \eta \nabla f(x^t) \]
Back to finite-sample analysis

\[ x^{t+1} = x^t - \eta \nabla f(x^t) = x^t - \eta \nabla F(x^t) - \eta (\nabla f(x^t) - \nabla F(x^t)) \]

residual
Back to finite-sample analysis

\[ x^{t+1} = x^t - \eta \nabla f(x^t) = x^t - \eta \nabla F(x^t) - \eta \left( \nabla f(x^t) - \nabla F(x^t) \right) \]

— take one term in \( x^*^\top \left( \nabla f(x^t) - \nabla F(x^t) \right) \) as example:

\[ \frac{1}{m} \sum_{i=1}^{m} \left( a_i^\top x^t \right)^3 a_i^\top x^* \]
Back to finite-sample analysis

\[ \mathbf{x}^{t+1} = \mathbf{x}^t - \eta \nabla f(\mathbf{x}^t) = \mathbf{x}^t - \eta \nabla F(\mathbf{x}^t) - \eta (\nabla f(\mathbf{x}^t) - \nabla F(\mathbf{x}^t)) \]

— take one term in \( \mathbf{x}^* \top (\nabla f(\mathbf{x}^t) - \nabla F(\mathbf{x}^t)) \) as example:

\[
\frac{1}{m} \sum_{i=1}^{m} (\mathbf{a}_i \top \mathbf{x}^t)^3 \mathbf{a}_i \top \mathbf{x}^* \]

- population-level analysis holds approximately if \( \mathbf{x}^t \) is independent of \( \{\mathbf{a}_i\} \)

a region with well-controlled residual
Back to finite-sample analysis

\[ x^{t+1} = x^t - \eta \nabla f(x^t) = x^t - \eta \nabla F(x^t) - \eta (\nabla f(x^t) - \nabla F(x^t)) \]

— take one term in \( x^* \top (\nabla f(x^t) - \nabla F(x^t)) \) as example:

\[ \frac{1}{m} \sum_{i=1}^{m} (a_i^\top x^t)^3 a_i^\top x^* \]

- population-level analysis holds approximately if \( x^t \) is independent of \( \{a_l\} \)

- key analysis ingredient: show \( x^t \) is “nearly-independent” of each \( a_l \)

a region with well-controlled residual
Stage 2: local refinement (implicit regularization)

<table>
<thead>
<tr>
<th>iteration complexity</th>
<th>prior theory</th>
<th>our theory</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$O(n \log \frac{1}{\varepsilon})$ (Candès et al. ’14)</td>
<td>$O(\log \frac{1}{\varepsilon})$</td>
</tr>
</tbody>
</table>
Gradient descent theory revisited

Two standard conditions that enable geometric convergence of GD
Gradient descent theory revisited

Two standard conditions that enable geometric convergence of GD

- (local) restricted strong convexity
Gradient descent theory revisited

Two standard conditions that enable geometric convergence of GD

- (local) restricted strong convexity
- (local) smoothness
Gradient descent theory revisited

\( f \) is said to be \( \alpha \)-strongly convex and \( \beta \)-smooth if

\[
0 \leq \alpha I \preceq \nabla^2 f(x) \preceq \beta I, \quad \forall x
\]
Gradient descent theory revisited

\( f \) is said to be \( \alpha \)-strongly convex and \( \beta \)-smooth if

\[
0 \preceq \alpha I \preceq \nabla^2 f(x) \preceq \beta I, \quad \forall x
\]

\( \ell_2 \) error contraction: GD with \( \eta = 1/\beta \) obeys

\[
\|x^{t+1} - x^*\|_2 \leq \left(1 - \frac{\alpha}{\beta}\right) \|x^t - x^*\|_2
\]
Gradient descent theory revisited

$f$ is said to be $\alpha$-strongly convex and $\beta$-smooth if

$$0 \preceq \alpha I \preceq \nabla^2 f(x) \preceq \beta I, \quad \forall x$$

\textit{$\ell_2$ error contraction:} GD with $\eta = 1/\beta$ obeys

$$\|x^{t+1} - x^*\|_2 \leq \left(1 - \frac{\alpha}{\beta}\right) \|x^t - x^*\|_2$$

- Condition number $\beta/\alpha$ determines rate of convergence
Gradient descent theory revisited

$f$ is said to be $\alpha$-strongly convex and $\beta$-smooth if

$$0 \preceq \alpha I \preceq \nabla^2 f(x) \preceq \beta I, \quad \forall x$$

**$l_2$ error contraction:** GD with $\eta = 1/\beta$ obeys

$$\|x^{t+1} - x^*\|_2 \leq \left(1 - \frac{\alpha}{\beta}\right)\|x^t - x^*\|_2$$

- Condition number $\beta/\alpha$ determines rate of convergence
- Attains $\varepsilon$-accuracy within $O\left(\frac{\beta}{\alpha} \log \frac{1}{\varepsilon}\right)$ iterations
What does this optimization theory say about GD?

Gaussian designs: \( a_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_n), \quad 1 \leq k \leq m \)
What does this optimization theory say about GD?

Gaussian designs: \( a_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_n), \quad 1 \leq k \leq m \)

Finite-sample level \( (m \asymp n \log n) \)

\[ \nabla^2 f(x) \succ 0.5I \]
What does this optimization theory say about GD?

Gaussian designs: \( a_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_n), \quad 1 \leq k \leq m \)

Finite-sample level \((m \asymp n \log n)\)

\[ \nabla^2 f(x) \succ 0.5I \quad \text{but ill-conditioned} \quad \text{(even locally)} \]

\text{condition number} \asymp n
What does this optimization theory say about GD?

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

Finite-sample level \( (m \asymp n \log n) \)

\[ \nabla^2 f(\mathbf{x}) \succ 0.5 \mathbf{I} \quad \text{but ill-conditioned} \quad \text{(even locally)} \]

\[ \text{condition number} \asymp n \]

Consequence (Candès et al. ’14): WF attains \( \varepsilon \)-accuracy within \( O(n \log \frac{1}{\varepsilon}) \) iterations if \( m \asymp n \log n \)
What does this optimization theory say about GD?

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

Finite-sample level \( (m \approx n \log n) \)

\[ \nabla^2 f(x) \succ 0.5 \mathbf{I} \quad \text{but ill-conditioned} \quad (\text{even locally}) \]

condition number \( \asymp n \)

Consequence (Candès et al. ’14): WF attains \( \varepsilon \)-accuracy within \( O(n \log \frac{1}{\varepsilon}) \) iterations if \( m \approx n \log n \)

— optimization theory based on generic landscape conditions implies slow convergence ...
A second look at gradient descent theory

Which local region enjoys both strong convexity and smoothness?
A second look at gradient descent theory

Which local region enjoys both strong convexity and smoothness?

\[ \nabla^2 f(x) = \frac{1}{m} \sum_{k=1}^{m} 3(a_k^\top x)^2 a_k a_k^\top - \frac{1}{m} \sum_{k=1}^{m} (a_k^\top x^*)^2 a_k a_k^\top \]
A second look at gradient descent theory

Which local region enjoys both strong convexity and smoothness?

\[ \nabla^2 f(x) = \frac{1}{m} \sum_{k=1}^{m} 3 (a_k^\top x)^2 a_k a_k^\top - \frac{1}{m} \sum_{k=1}^{m} (a_k^\top x^*)^2 a_k a_k^\top \]

- Not sufficiently smooth if \( x \) and \( a_k \) are too close
A second look at gradient descent theory

Which local region enjoys both strong convexity and smoothness?

- $x$ is incoherent w.r.t. sampling vectors $\{a_k\}$ (incoherence region)

\[ |a_1^T (x - x^\dagger)| \lesssim \sqrt{\log n} \]
A second look at gradient descent theory

Which local region enjoys both strong convexity and smoothness?

- $x$ is incoherent w.r.t. sampling vectors $\{a_k\}$ (incoherence region)
A second look at gradient descent theory

Which local region enjoys both strong convexity and smoothness?

- $x$ is incoherent w.r.t. sampling vectors $\{a_k\}$ (incoherence region)

Prior works suggest enforcing regularization (e.g. truncation, projection, regularized loss) to promote incoherence
Aside: regularized methods

- Phase retrieval
  - Regularized trimming

- Matrix completion
  - Regularized cost projection

- Blind deconvolution
  - Regularized cost projection
Aside: regularized vs. unregularized methods

- Phase retrieval
  - Regularized
  - Unregularized
  - Trimming
  - Suboptimal comput. cost

- Matrix completion
  - Regularized
  - Unregularized
  - Regularized cost
  - Projection
  - ?

- Blind deconvolution
  - Regularized
  - Unregularized
  - Regularized cost
  - Projection
  - ?

Are unregularized methods suboptimal for nonconvex estimation?
Aside: regularized vs. unregularized methods

- **phase retrieval**
  - regularized
  - unregularized
  - trimming
  - suboptimal comput. cost

- **matrix completion**
  - regularized
  - unregularized
  - regularized cost projection

- **blind deconvolution**
  - regularized
  - unregularized
  - regularized cost projection

---

**Are unregularized methods suboptimal for nonconvex estimation?**
Our findings: GD is implicitly regularized

region of local strong convexity + smoothness
Our findings: GD is implicitly regularized

region of local strong convexity + smoothness
Our findings: GD is implicitly regularized

- region of local strong convexity + smoothness
Our findings: GD is implicitly regularized

region of local strong convexity + smoothness
Our findings: GD is implicitly regularized

region of local strong convexity + smoothness

\[ \max_l |a_l^\top x_t| \lesssim \sqrt{\log m} \|x_t\|_2, \quad \forall t \]

GD implicitly forces iterates to remain incoherent with \( \{a_l\} \)
Our findings: GD is implicitly regularized

- region of local strong convexity + smoothness

GD implicitly forces iterates to remain incoherent with \( \{a_l\} \)

\[
\max_l |a_l^\top x^t| \lesssim \sqrt{\log m} \|x^t\|_2, \quad \forall t
\]

- cannot be derived from generic optimization theory; relies on finer statistical analysis for entire trajectory of GD
Key proof idea: leave-one-out analysis

Leave out a small amount of information from data and run GD
Key proof idea: leave-one-out analysis

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

e.g. introduce leave-one-out iterates $x^{t,(l)}$ by running GD without $l$th sample
Key proof idea: leave-one-out analysis

- Leave-one-out iterate $\mathbf{x}^{t,(l)}$ is independent of $\mathbf{a}_l$
Key proof idea: leave-one-out analysis

- Leave-one-out iterate $x^{t,(l)}$ is independent of $a_l$
- Leave-one-out iterate $x^{t,(l)} \approx$ true iterate $x^t$
Key proof idea: leave-one-out analysis

- Leave-one-out iterate $x^{t,(l)}$ is independent of $a_l$
- Leave-one-out iterate $x^{t,(l)} \approx$ true iterate $x^t$

$\implies x^t$ is nearly independent of $a_l$

nearly orthogonal to
Key proof ingredient: random-sign sequences

\[ A^{\text{sgn}} \quad y = |A^{\text{sgn}}x^*|^2 \]

- \( x^{t,\text{sgn}} \): indep. of sign info of \( \{a_{i,1}\} \)
- \( x^{t,\text{sgn,(l)}} \): indep. of both sign info of \( \{a_{i,1}\} \) and \( a_l \)

- randomly flip signs of \( a_i^\top x^* \) and re-run GD
Key proof ingredient: random-sign sequences

\[ A^{\text{sgn}} \quad y = |A^{\text{sgn}}x^*|^2 \]

\[ A^{\text{sgn}} \quad y = |A^{\text{sgn}}x^*|^2 \]

\[ x^{t,\text{sgn}}: \text{indep. of sign info of} \quad \{a_{i,1}\} \]

\[ x^{t,\text{sgn,}(l)}: \text{indep. of both sign info of} \quad \{a_{i,1}\} \text{ and } a_l \]

- randomly flip signs of \( a_i^\top x^* \) and re-run GD
- crucial in controlling \[ \frac{1}{m} \sum_{i=1}^{m} \left( a_i^\top x^t \right)^3 \left( a_i^\top x^* \right) \]
  \[ |a_i^\top x^*| \text{sgn}(a_i^\top x^*) \]
analyzed in Step (ii), which arises from a key property concerning the “near-independence” between the design vectors; (iii) finally, we argue that the true trajectory is remarkably close to the one heuristically dynamics of the population gradient sequence (the case where we have infinite samples); (ii) we then turn iterative update.

Simistic) computational guarantees when analyzing a concrete algorithm like GD. In contrast, the current smoothness conditions. Such delicate geometric properties underlying the GD trajectory are not explained the same instance as plotted in Figure 3: The trajectory of $t$. The red dots represent the population-level saddle points.

Statistical observation plays a crucial role in characterizing the dynamics of the algorithm without the need of sample splitting.

In other words, when randomly initialized, initialization — results in suboptimal (or even pes-

Without loss of generality, we assume $\alpha_n = 0.2$. As we shall make precise in Section 5, $\alpha_n$'s are independent. Simple algebraic manipulation reveals the dynamics for both the signal and the perpendicular components:

$F_{t+1} = \alpha_{t+1} \leq F_t$.

Here, we do not take the absolute value of $e_t$, $?_{t+k}$, and $?_{t+k+1}$.

Assuming that $x_{t+k}$ and $x_{t+k+1}$ are of the same sign throughout the execution of the algorithm.

Valuable insights into algorithm designs with phenomena numerically. Set $x_{t+k}$ through and perpendicular to the signal direction. In what follows, the $x_{t+k}$'s are independent. Simple algebraic manipulation reveals the dynamics for both the signal and the perpendicular components:

$F_{t+1} = \alpha_{t+1} \leq F_t$.

Here, we do not take the absolute value of $e_t$, $?_{t+k}$, and $?_{t+k+1}$.
Other saddle-escaping schemes based on generic landscape analysis

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Iteration Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>trust-region (Sun et al. ’16)</td>
<td>( n^7 + \log \log \frac{1}{\varepsilon} )</td>
</tr>
<tr>
<td>perturbed GD (Jin et al. ’17)</td>
<td>( n^3 + n \log \frac{1}{\varepsilon} )</td>
</tr>
<tr>
<td>perturbed accelerated GD (Jin et al. ’17)</td>
<td>( n^{2.5} + \sqrt{n} \log \frac{1}{\varepsilon} )</td>
</tr>
<tr>
<td>GD (ours) (Chen et al. ’18)</td>
<td>( \log n + \log \frac{1}{\varepsilon} )</td>
</tr>
</tbody>
</table>

Generic optimization theory yields highly suboptimal convergence guarantees.
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.

<table>
<thead>
<tr>
<th>smart initialization</th>
<th>extra regularization</th>
<th>sample splitting</th>
<th>saddle escaping</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="NEED" alt="not needed" /></td>
<td><img src="NEED" alt="not needed" /></td>
<td><img src="NEED" alt="not needed" /></td>
<td><img src="NEED" alt="not needed" /></td>
</tr>
</tbody>
</table>