Landscape Analysis of Non-Convex Optimizations in Phase Retrieval

Jian-Feng Cai

Department of Mathematics,
The Hong Kong University of Science and Technology

joint work with
Zhenzhen Li (Caltech), Ke Wei (Fudan),
Meng Huang, Dong Li, Yang Wang (HKUST).

August 14, 2020

Non-Convex Optimizations

Non-Convex optimizations are powerful tools for solving many problems.

- Matrix singular value / eigenvalue problems
- Matrix low-rank approximation, low-rank matrix completion / recovery
- Deep neural network training

Non-Convex Optimizations

Non-Convex optimizations are powerful tools for solving many problems.

- Matrix singular value / eigenvalue problems
- Matrix low-rank approximation, low-rank matrix completion / recovery
- Deep neural network training

Simple algorithms (e.g., gradient descent, stochastic gradient descent, project gradient descent, alternating minimization) often provide good solutions efficiently and effectively, despite possible non-optimal critical points.

A classical example: Power Iteration

To find the top singular value and vectors $(\sigma_1, \boldsymbol{u}_1, \boldsymbol{v}_1)$ of $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ (assume $\sigma_1 > \sigma_2$), the power iteration is: starting from $(\boldsymbol{u}_{init}, \boldsymbol{v}_{init})$,

$$\begin{cases} \boldsymbol{u} \leftarrow \boldsymbol{A}\boldsymbol{v}/\|\boldsymbol{A}\boldsymbol{v}\|_2, \\ \boldsymbol{v} \leftarrow \boldsymbol{A}^T\boldsymbol{u}/\|\boldsymbol{A}^T\boldsymbol{u}\|_2. \end{cases}$$

A classical example: Power Iteration

To find the top singular value and vectors $(\sigma_1, \boldsymbol{u}_1, \boldsymbol{v}_1)$ of $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ (assume $\sigma_1 > \sigma_2$), the power iteration is: starting from $(\boldsymbol{u}_{init}, \boldsymbol{v}_{init})$,

$$\begin{cases} \boldsymbol{u} \leftarrow \boldsymbol{A}\boldsymbol{v}/\|\boldsymbol{A}\boldsymbol{v}\|_2, \\ \boldsymbol{v} \leftarrow \boldsymbol{A}^T\boldsymbol{u}/\|\boldsymbol{A}^T\boldsymbol{u}\|_2. \end{cases}$$

- It is well known that, provided $\boldsymbol{u}_{init} \not\perp \boldsymbol{u}_1$ and $\boldsymbol{v}_{init} \not\perp \boldsymbol{v}_1$,

$$\sin\angle(\boldsymbol{u}, \boldsymbol{u}_1) \to 0, \quad \sin\angle(\boldsymbol{v}, \boldsymbol{v}_1) \to 0, \quad \text{linearly.}$$

A classical example: Power Iteration

To find the top singular value and vectors $(\sigma_1, \boldsymbol{u}_1, \boldsymbol{v}_1)$ of $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ (assume $\sigma_1 > \sigma_2$), the power iteration is: starting from $(\boldsymbol{u}_{init}, \boldsymbol{v}_{init})$,

$$\begin{cases} \boldsymbol{u} \leftarrow \boldsymbol{A}\boldsymbol{v}/\|\boldsymbol{A}\boldsymbol{v}\|_2, \\ \boldsymbol{v} \leftarrow \boldsymbol{A}^T\boldsymbol{u}/\|\boldsymbol{A}^T\boldsymbol{u}\|_2. \end{cases}$$

- It is well known that, provided $\boldsymbol{u}_{init} \not\perp \boldsymbol{u}_1$ and $\boldsymbol{v}_{init} \not\perp \boldsymbol{v}_1$,

$$\sin\angle(\boldsymbol{u}, \boldsymbol{u}_1) \to 0, \quad \sin\angle(\boldsymbol{v}, \boldsymbol{v}_1) \to 0, \quad \text{linearly.}$$

- The power iteration is an alternating minimization algorithm for solving the non-convex optimization

$$\min_{\|\boldsymbol{u}\|_2=1, \|\boldsymbol{v}\|_2=1} -\boldsymbol{u}^T\boldsymbol{A}\boldsymbol{v} \qquad \left(\Longleftrightarrow \qquad \max_{\|\boldsymbol{u}\|_2=1, \|\boldsymbol{v}\|_2=1} \boldsymbol{u}^T\boldsymbol{A}\boldsymbol{v}\right)$$

A classical example: Power Iteration

To find the top singular value and vectors $(\sigma_1, \boldsymbol{u}_1, \boldsymbol{v}_1)$ of $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ (assume $\sigma_1 > \sigma_2$), the power iteration is: starting from $(\boldsymbol{u}_{init}, \boldsymbol{v}_{init})$,

$$\begin{cases} \boldsymbol{u} \leftarrow \boldsymbol{A}\boldsymbol{v}/\|\boldsymbol{A}\boldsymbol{v}\|_2, \\ \boldsymbol{v} \leftarrow \boldsymbol{A}^T\boldsymbol{u}/\|\boldsymbol{A}^T\boldsymbol{u}\|_2. \end{cases}$$

- It is well known that, provided $\boldsymbol{u}_{init} \not\perp \boldsymbol{u}_1$ and $\boldsymbol{v}_{init} \not\perp \boldsymbol{v}_1$,

$$\sin\angle(\boldsymbol{u}, \boldsymbol{u}_1) \to 0, \quad \sin\angle(\boldsymbol{v}, \boldsymbol{v}_1) \to 0, \quad \text{linearly.}$$

- The power iteration is an alternating minimization algorithm for solving the non-convex optimization

$$\min_{\|\boldsymbol{u}\|_2=1, \|\boldsymbol{v}\|_2=1} -\boldsymbol{u}^T\boldsymbol{A}\boldsymbol{v} \qquad \left(\Longleftrightarrow \qquad \max_{\|\boldsymbol{u}\|_2=1, \|\boldsymbol{v}\|_2=1} \boldsymbol{u}^T\boldsymbol{A}\boldsymbol{v}\right)$$

- Why does the power iteration almost surely find $(\boldsymbol{u}_1, \boldsymbol{v}_1)$?

A classical example: Power Iteration

To find the top singular value and vectors $(\sigma_1, \boldsymbol{u}_1, \boldsymbol{v}_1)$ of $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ (assume $\sigma_1 > \sigma_2$), the power iteration is: starting from $(\boldsymbol{u}_{init}, \boldsymbol{v}_{init})$,

$$\begin{cases} \boldsymbol{u} \leftarrow \boldsymbol{A}\boldsymbol{v}/\|\boldsymbol{A}\boldsymbol{v}\|_2, \\ \boldsymbol{v} \leftarrow \boldsymbol{A}^T\boldsymbol{u}/\|\boldsymbol{A}^T\boldsymbol{u}\|_2. \end{cases}$$

- It is well known that, provided $\boldsymbol{u}_{init} \not\perp \boldsymbol{u}_1$ and $\boldsymbol{v}_{init} \not\perp \boldsymbol{v}_1$,

$$\sin\angle(\boldsymbol{u}, \boldsymbol{u}_1) \to 0, \quad \sin\angle(\boldsymbol{v}, \boldsymbol{v}_1) \to 0, \quad \text{linearly.}$$

- The power iteration is an alternating minimization algorithm for solving the non-convex optimization

$$\min_{\|\boldsymbol{u}\|_2=1, \|\boldsymbol{v}\|_2=1} -\boldsymbol{u}^T\boldsymbol{A}\boldsymbol{v} \qquad \left(\Longleftrightarrow \qquad \max_{\|\boldsymbol{u}\|_2=1, \|\boldsymbol{v}\|_2=1} \boldsymbol{u}^T\boldsymbol{A}\boldsymbol{v}\right)$$

- Why does the power iteration almost surely find $(\boldsymbol{u}_1, \boldsymbol{v}_1)$?
- Moreover, variants of the power iteration are successful in SVD computation, despite of the non-convex essence of SVD as an optimization. Why?

A classical example: Power Iteration

Landscape of the non-convex optimization:

$$\min_{\|\boldsymbol{u}\|_2=1, \|\boldsymbol{v}\|_2=1} -\boldsymbol{u}^T \boldsymbol{A}\boldsymbol{v}$$

A classical example: Power Iteration

Landscape of the non-convex optimization:

$$\min_{\|\boldsymbol{u}\|_2=1, \|\boldsymbol{v}\|_2=1} -\boldsymbol{u}^T \boldsymbol{A} \boldsymbol{v}$$

- The (Lagrange) critical points $(\boldsymbol{u}, \boldsymbol{v}, \lambda)$ must satisfy

  $$\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{u}, \quad \boldsymbol{A}^T \boldsymbol{u} = \lambda \boldsymbol{v}, \quad \lambda = \boldsymbol{u}^T \boldsymbol{A} \boldsymbol{v} = \boldsymbol{v}^T \boldsymbol{A}^T \boldsymbol{u}, \quad \|\boldsymbol{u}\|_2 = \|\boldsymbol{v}\|_2 = 1$$

  So all the critical points are

  $$(\pm \boldsymbol{u}_i, \pm \boldsymbol{v}_i, \sigma_i), \quad (\pm \boldsymbol{u}_i, \mp \boldsymbol{v}_i, -\sigma_i), \quad i = 1, \ldots, n.$$

- Among all critical points (assume all singular values are distinct):
  - Only $(\pm \boldsymbol{u}_1, \pm \boldsymbol{v}_1, \sigma_1)$ are local minimizers, which are also global.
  - Only $(\pm \boldsymbol{u}_1, \mp \boldsymbol{v}_1, -\sigma_1)$ are local maximizers, which are also global.
  - All other critical points are strict saddle points.

A classical example: Power Iteration

Landscape of the non-convex optimization:

$$\min_{\|\boldsymbol{u}\|_2=1,\|\boldsymbol{v}\|_2=1} -\boldsymbol{u}^T\boldsymbol{A}\boldsymbol{v}$$

- The (Lagrange) critical points $(\boldsymbol{u}, \boldsymbol{v}, \lambda)$ must satisfy

  $$\boldsymbol{A}\boldsymbol{v} = \lambda\boldsymbol{u}, \quad \boldsymbol{A}^T\boldsymbol{u} = \lambda\boldsymbol{v}, \quad \lambda = \boldsymbol{u}^T\boldsymbol{A}\boldsymbol{v} = \boldsymbol{v}^T\boldsymbol{A}^T\boldsymbol{u}, \quad \|\boldsymbol{u}\|_2 = \|\boldsymbol{v}\|_2 = 1$$

  So all the critical points are

  $$(\pm\boldsymbol{u}_i, \pm\boldsymbol{v}_i, \sigma_i), \quad (\pm\boldsymbol{u}_i, \mp\boldsymbol{v}_i, -\sigma_i), \quad i = 1, \ldots, n.$$

- Among all critical points (assume all singular values are distinct):
  - Only $(\pm\boldsymbol{u}_1, \pm\boldsymbol{v}_1, \sigma_1)$ are local minimizers, which are also global.
  - Only $(\pm\boldsymbol{u}_1, \mp\boldsymbol{v}_1, -\sigma_1)$ are local maximizers, which are also global.
  - All other critical points are strict saddle points.

  Thus, the non-convex optimization is not as difficult as in general cases — any algorithms that converge to a local minimum find $(\pm\boldsymbol{u}_1, \pm\boldsymbol{v}_1)$.

This talk

This talk present a similar, recent, and more complicated example — Phase retrieval via non-convex optimization, and its landscape analysis.

Outline

Phase Retrieval

Solving a length-n vector $\boldsymbol{x}$ from its phaseless measurements

$$|\boldsymbol{a}_r^* \boldsymbol{x}|^2 = y_r, \quad r = 1, 2, \ldots m.$$

Phase Retrieval

Solving a length-n vector $\boldsymbol{x}$ from its phaseless measurements

$$|\boldsymbol{a}_r^* \boldsymbol{x}|^2 = y_r, \quad r = 1, 2, \ldots m.$$

In matrix form, we need to recover $\boldsymbol{x}$ from

$$|\boldsymbol{A}\boldsymbol{x}|^2 = \boldsymbol{y}, \quad \boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1^* \\ \vdots \\ \boldsymbol{a}_m^* \end{bmatrix}$$

- We need to solve m **quadratic** equations with n unknowns.
- An application is Phase Retrieval, which are used widely in X-ray crystallography, coherent diffractive imaging, astronomial imaging, biomedical imaging, quantum mechanics, etc.

Phase Retrieval

In a diffraction imaging, only magnitudes are observed, and phases are missing.

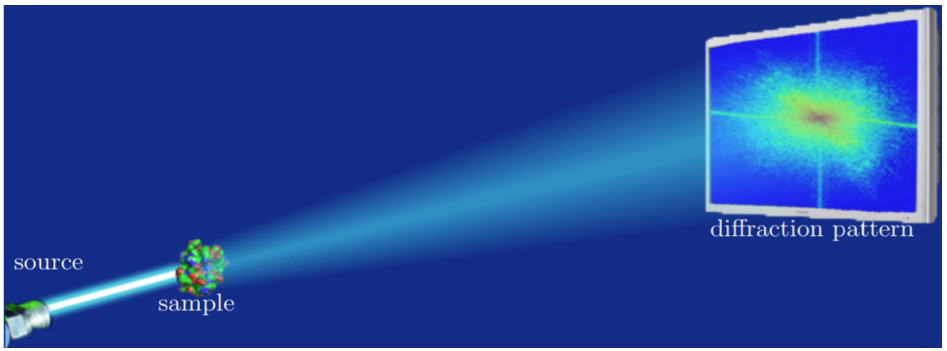


Figure: Diffraction Pattern

Phase Retrieval

In a diffraction imaging, only magnitudes are observed, and phases are missing.

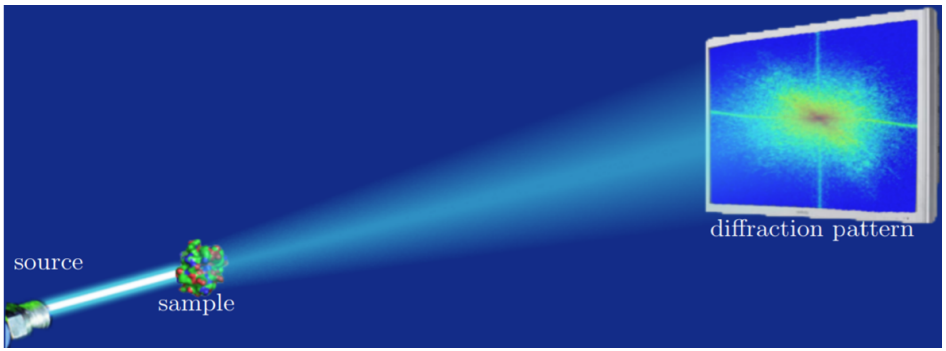


Figure: Diffraction Pattern

Let $\boldsymbol{x}$ be the unknown image. By Fresnel Diffraction Principle, the observed data $\boldsymbol{y}$ is

$$|\boldsymbol{A}\boldsymbol{x}|^2 = \boldsymbol{y}$$

where $\boldsymbol{A} = [\boldsymbol{F} \cdot \mathrm{Diag}(\boldsymbol{d}_i) \ \ldots \ \boldsymbol{F} \cdot \mathrm{Diag}(\boldsymbol{d}_L)]^T$ with $\boldsymbol{F}$ the Fourier transform and $\boldsymbol{d}_i$'s coded diffraction patterns.

Phase Retrieval

In a diffraction imaging, only magnitudes are observed, and phases are missing.

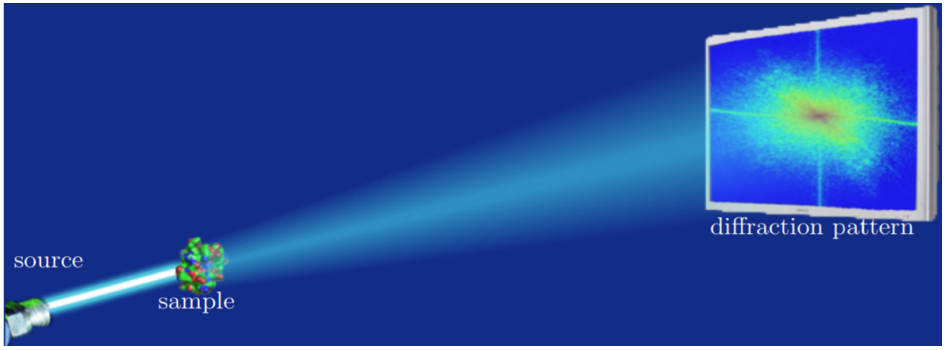


Figure: Diffraction Pattern

Let $\boldsymbol{x}$ be the unknown image. By Fresnel Diffraction Principle, the observed data $\boldsymbol{y}$ is

$$|\boldsymbol{A}\boldsymbol{x}|^2 = \boldsymbol{y}$$

where $\boldsymbol{A} = [\boldsymbol{F} \cdot \mathrm{Diag}(\boldsymbol{d}_i) \ \ldots \ \boldsymbol{F} \cdot \mathrm{Diag}(\boldsymbol{d}_L)]^T$ with $\boldsymbol{F}$ the Fourier transform and $\boldsymbol{d}_i$'s coded diffraction patterns.

Phase retrieval is a fundamental problem in many other imaging techniques as well.

When phase retrieval is solvable?

- Obviously, phase retrieval cannot have a unique solution because $|\boldsymbol{A}\boldsymbol{z}|^2 = |\boldsymbol{A}(c\boldsymbol{z})|^2$ for a global sign c satisfying $|c| = 1$.

When phase retrieval is solvable?

- Obviously, phase retrieval cannot have a unique solution because $|\boldsymbol{A}\boldsymbol{z}|^2 = |\boldsymbol{A}(c\boldsymbol{z})|^2$ for a global sign c satisfying $|c| = 1$.
- When phase retrieval has a unique solution up to a global sign?

Solvability ([Wang & Xu, 2015])

- For the real case $\boldsymbol{x} \in \mathbb{R}^n$, if $m \geq 2n - 1$, then phase retrieval has a unique solution up to a global sign almost surely for all $\boldsymbol{A}$.
- For the complex case $\boldsymbol{x} \in \mathbb{C}^n$, if $m \geq 4n - 4$, then phase retrieval has a unique solution up to a global sign almost surely for all $\boldsymbol{A}$.

Solving Phase Retrieval Is NOT Easy

- Consider the stone problem, which is to separate n stones into two groups with equal weights. It can be recast into a special case of our phase retrieval problem.
- Let w_i, $i = 1, \ldots, n$, be the weight of stones. Let $\boldsymbol{x} \in \{\pm 1\}^n$ be an indicator vector of the two groups.

$$\begin{cases} |\boldsymbol{e}_i^* \boldsymbol{x}|^2 = 1, & i = 1, 2, \ldots n, \\ |\boldsymbol{w}^* \boldsymbol{x}|^2 = 0. \end{cases}$$

- The stone problem is NP hard.

Solving Phase Retrieval Is NOT Easy

- Consider the stone problem, which is to separate n stones into two groups with equal weights. It can be recast into a special case of our phase retrieval problem.
- Let w_i, $i = 1, \ldots, n$, be the weight of stones. Let $\boldsymbol{x} \in \{\pm 1\}^n$ be an indicator vector of the two groups.

$$\begin{cases} |\boldsymbol{e}_i^* \boldsymbol{x}|^2 = 1, & i = 1, 2, \ldots n, \\ |\boldsymbol{w}^* \boldsymbol{x}|^2 = 0. \end{cases}$$

- The stone problem is NP hard.

For simplicity, in the rest of the talk we assume all vectors are real and $\{\boldsymbol{a}_r\}_{r=1}^m$ are i.i.d. random Gaussian.

Outline

Intensity equations

We may solve the squared equations, called intensity equations, directly.

$$|\boldsymbol{a}_r^* \boldsymbol{x}|^2 = y_r, \quad r = 1, \ldots, m.$$

Intensity equations

We may solve the squared equations, called intensity equations, directly.

$$|\boldsymbol{a}_r^* \boldsymbol{x}|^2 = y_r, \quad r = 1, \ldots, m.$$

- Convex solvers.
  - The intensity equations are linear equations on the rank-1 matrix $\boldsymbol{x}\boldsymbol{x}^*$

$$|\boldsymbol{a}_r^* \boldsymbol{x}|^2 = y_r \quad \Longleftrightarrow \quad \langle \boldsymbol{a}_r \boldsymbol{a}_r^*, \boldsymbol{x}\boldsymbol{x}^* \rangle = y_r, \qquad r = 1, \ldots, m$$

    Then convex low-rank matrix recovery techniques are applied.

Intensity equations

We may solve the squared equations, called intensity equations, directly.

$$|\boldsymbol{a}_r^* \boldsymbol{x}|^2 = y_r, \quad r = 1, \dots, m.$$

- Convex solvers.
  - The intensity equations are linear equations on the rank-1 matrix $\boldsymbol{x}\boldsymbol{x}^*$

    $$|\boldsymbol{a}_r^* \boldsymbol{x}|^2 = y_r \quad \Longleftrightarrow \quad \langle \boldsymbol{a}_r \boldsymbol{a}_r^*, \boldsymbol{x}\boldsymbol{x}^* \rangle = y_r, \qquad r = 1, \dots, m$$

    Then convex low-rank matrix recovery techniques are applied.
  - Nice theories of recovery guarantee [Candes, Strohmer, 2013; Candes, Li, 2015], e.g., $m = O(n)$ equations are sufficient.
  - Slow computation: The number of unknowns are n^2 instead of n, unnecessarily large.

- Non-convex solvers.
  - Minimize the least squares fitting to intensity equations directly [Candes, Li, Soltanolkotabi, 2015; Chen, Candes, 2017]

$$\min_{\boldsymbol{z}} f(\boldsymbol{z}), \qquad f(\boldsymbol{z}) = \frac{1}{2m} \sum_{r=1}^{m} \left((\boldsymbol{a}_r^T \boldsymbol{z})^2 - y_r \right)^2$$

- Non-convex solvers.
  - Minimize the least squares fitting to intensity equations directly [Candes, Li, Soltanolkotabi, 2015; Chen, Candes, 2017]

$$\min_{\boldsymbol{z}} f(\boldsymbol{z}), \qquad f(\boldsymbol{z}) = \frac{1}{2m}\sum_{r=1}^{m}\left((\boldsymbol{a}_r^T\boldsymbol{z})^2 - y_r\right)^2$$

  - Or minimize the least squares fitting to the linear equations on the manifold of rank-1 matrices (denoted by $\mathbb{M}_1$) [Cai, Wei, 2018]

$$\min_{\boldsymbol{Z}\in\mathbb{M}_1} F(\boldsymbol{Z}), \qquad F(\boldsymbol{Z}) = \frac{1}{2m}\sum_{r=1}^{m}\left(\langle \boldsymbol{a}_r\boldsymbol{a}_r^T, \boldsymbol{Z}\rangle - y_r\right)^2$$

- Non-convex solvers.
  - Minimize the least squares fitting to intensity equations directly [Candes, Li, Soltanolkotabi, 2015; Chen, Candes, 2017]

$$\min_{\boldsymbol{z}} f(\boldsymbol{z}), \qquad f(\boldsymbol{z}) = \frac{1}{2m}\sum_{r=1}^{m}\left((\boldsymbol{a}_r^T\boldsymbol{z})^2 - y_r\right)^2$$

  - Or minimize the least squares fitting to the linear equations on the manifold of rank-1 matrices (denoted by $\mathbb{M}_1$) [Cai, Wei, 2018]

$$\min_{\boldsymbol{Z}\in\mathbb{M}_1} F(\boldsymbol{Z}), \qquad F(\boldsymbol{Z}) = \frac{1}{2m}\sum_{r=1}^{m}\left(\langle \boldsymbol{a}_r\boldsymbol{a}_r^T, \boldsymbol{Z}\rangle - y_r\right)^2$$

  - Usually in two stages:
    Initialization (by spectral initialization) + Refinement (by gradient-type algorithms).
  - Fast computation: only n unknowns.
  - Nice theories of recovery guarantee: $m = O(n)$ equations are sufficient.
  - The computation and analysis of f (respectively F) were both carried out in a neighbourhood of $\pm\boldsymbol{x}$ (respectively $\boldsymbol{x}\boldsymbol{x}^T$).

Global Landscape

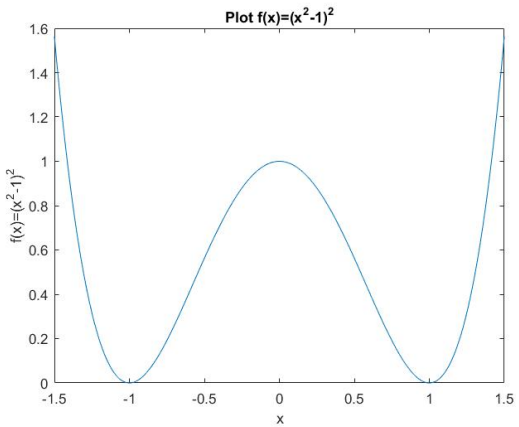
- Though non-convex algorithms need a special initialization in theory, they work well even with a random initialization.

Global Landscape

- Though non-convex algorithms need a special initialization in theory, they work well even with a random initialization.
- This reminds us to check the non-convex optimization globally.

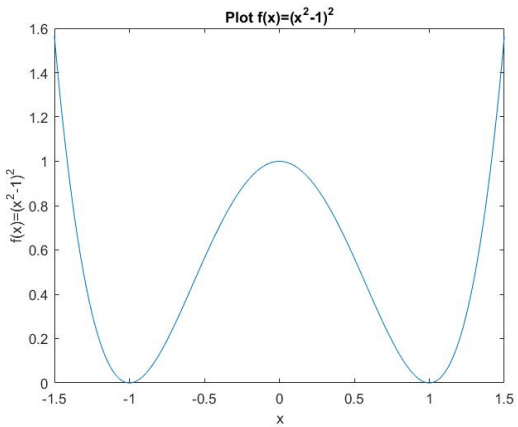
Global Landscape

- Though non-convex algorithms need a special initialization in theory, they work well even with a random initialization.
- This reminds us to check the non-convex optimization globally.
- Consider a 1D example: $f(z) = (|z|^2 - 1)^2$.



Global Landscape

- Though non-convex algorithms need a special initialization in theory, they work well even with a random initialization.
- This reminds us to check the non-convex optimization globally.
- Consider a 1D example: $f(z) = (|z|^2 - 1)^2$.



- For this 1-D example:
  - No spurious local minimum: all local minima are global.
  - The function is strongly convex in a small neighborhood of any global minimizer.
  - Therefore, any algorithm finding a local minimum will give a global minimum. (e.g. gradient descent with random initialization)

Favourable landscape

Theorem ([Sun, Qu, & Wright, 2018])

Assume $\boldsymbol{A}$ *is random Gaussian. Then, if*

$$m \geq Cn \log^3 n,$$

then with overwhelming probability $f(\boldsymbol{z})$ *satisfies:*

- *There are no spurious local minimum: all local min are global min.*
- $\nabla^2 f(z)$ *has a negative eigenvalue if* $\nabla f(\boldsymbol{z}) = \boldsymbol{0}$ *and* $\boldsymbol{z} \neq c\boldsymbol{x}$ *with* $|c| = 1$.

- The sampling complexity $m = O(n \log^3 n)$ is not optimal.

Favourable landscape

Theorem ([Sun, Qu, & Wright, 2018])

Assume $\boldsymbol{A}$ is random Gaussian. Then, if

$$m \geq Cn \log^3 n,$$

then with overwhelming probability $f(\boldsymbol{z})$ satisfies:

- *There are no spurious local minimum: all local min are global min.*
- *$\nabla^2 f(z)$ has a negative eigenvalue if $\nabla f(\boldsymbol{z}) = \boldsymbol{0}$ and $\boldsymbol{z} \neq c\boldsymbol{x}$ with $|c| = 1$.*

- The sampling complexity $m = O(n \log^3 n)$ is not optimal.
- We improved the result to $m = O(n \log n)$ recently [Cai, Huang, Li, Wang, forthcoming].
- There is still a gap to the optimal sampling complexity.

Activated Least Squares Fitting

$$f(\boldsymbol{z}) = \frac{1}{2m} \sum_{r=1}^{m} \left((\boldsymbol{a}_r^T \boldsymbol{z})^2 - y_r \right)^2.$$

The gap is due to the 4-th moment of Gaussians involved in f.

Activated Least Squares Fitting

$$f(\boldsymbol{z}) = \frac{1}{2m} \sum_{r=1}^{m} \left((\boldsymbol{a}_r^T \boldsymbol{z})^2 - y_r \right)^2.$$

The gap is due to the 4-th moment of Gaussians involved in f.

- Given a large β, $y_r := (\boldsymbol{a}_r^T \boldsymbol{x})^2 \leq \beta \|\boldsymbol{x}\|_2^2$ should hold true for most r, since $\boldsymbol{a}_r^T \boldsymbol{x}$ is random Gaussian.
- Those outliers $y_r > \beta \|\boldsymbol{x}\|_2^2$ will have a very large weight in f, which will make f deviate too much from its expectation.
- To avoid this, we activate only those well-behaved equations in the least squares fitting.

Activated Least Squares Fitting

$$f(\boldsymbol{z}) = \frac{1}{2m}\sum_{r=1}^{m}\left((\boldsymbol{a}_r^T\boldsymbol{z})^2 - y_r\right)^2.$$

The gap is due to the 4-th moment of Gaussians involved in f.

- Given a large β, $y_r := (\boldsymbol{a}_r^T\boldsymbol{x})^2 \le \beta\|\boldsymbol{x}\|_2^2$ should hold true for most r, since $\boldsymbol{a}_r^T\boldsymbol{x}$ is random Gaussian.
- Those outliers $y_r > \beta\|\boldsymbol{x}\|_2^2$ will have a very large weight in f, which will make f deviate too much from its expectation.
- To avoid this, we activate only those well-behaved equations in the least squares fitting.
- Similar activation scheme is applied to $\boldsymbol{a}_r^T\boldsymbol{z}$.
- We consider the minimization of

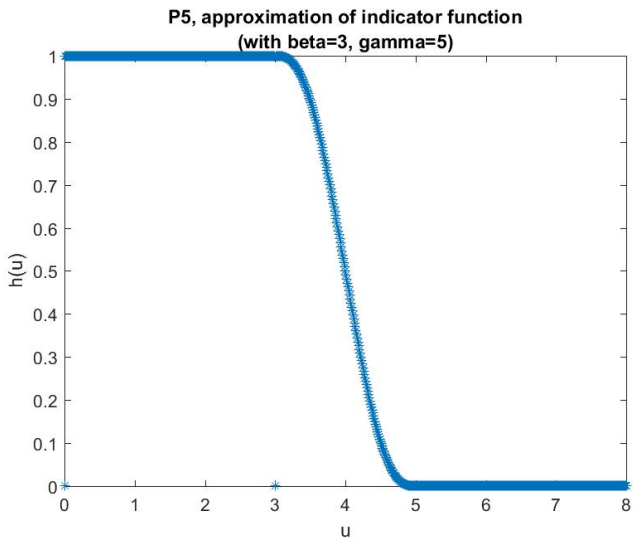
$$\tilde{f}(\boldsymbol{z}) = \frac{1}{2m}\sum_{r=1}^{m}\left(\left|\boldsymbol{a}_r^T\boldsymbol{z}\right|^2 - y_r\right)^2 \cdot h\left(\frac{\left|\boldsymbol{a}_r^*\boldsymbol{z}\right|^2}{\|\boldsymbol{z}\|_2^2}\right) h\left(\frac{my_r}{\|\boldsymbol{y}\|_1}\right),$$

where $h(\cdot)$ is a smooth activation function with bounded derivatives to approximate $1_{[0,\beta]}$.

Example of activation functions

- $h\left(\frac{|\boldsymbol{a}_r^* \boldsymbol{z}|^2}{\|\boldsymbol{z}\|_2^2}\right)$ is to activate only those measurements $|\boldsymbol{a}_r^T \boldsymbol{z}|^2 \le \beta \|\boldsymbol{z}\|_2^2$.
- Since $\|\boldsymbol{y}\|_1/m \approx \|\boldsymbol{x}\|_2^2$, $h\left(\frac{m y_r}{\|\boldsymbol{y}\|_1}\right)$ is to activate only those equations $|\boldsymbol{a}_r^T \boldsymbol{x}|^2 \lessapprox \beta \|\boldsymbol{x}\|_2^2$.
- Example:

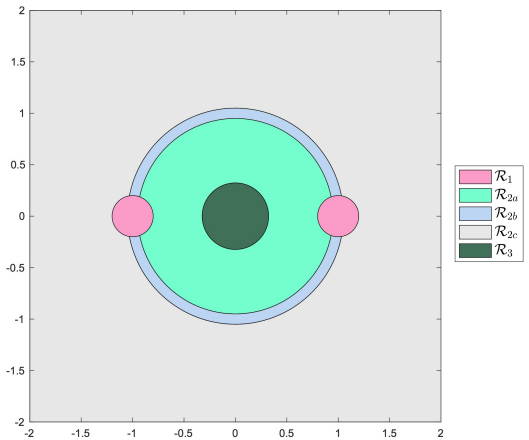
$$h(u) = \begin{cases} 1, & 0 < u \le \beta \\ -6v^5 + 15v^4 - 10v^3 + 1, & v = \frac{u-\beta}{\gamma-\beta}, \quad \beta < u < \gamma \\ 0, & u \ge \gamma \end{cases}$$



Landscape with optimal m

For any fixed $\delta \in (0, \frac{1}{100}]$, we partition

$$\mathbb{R}^n := \mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3 \cup \{\mathbf{0}\}$$



- $\mathcal{R}_1 = \left\{ \boldsymbol{z} \;\middle|\; \min_{\boldsymbol{v}=\pm\boldsymbol{x}} \|\boldsymbol{z} - \boldsymbol{v}\|_2 < \frac{1}{5}\|\boldsymbol{x}\|_2 \right\}$ — Strongly convex.
- $\mathcal{R}_3 = \left\{ \boldsymbol{z} \middle| 0 < \frac{\|\boldsymbol{z}\|^2}{\|\boldsymbol{x}\|^2} \leq \frac{1}{3} - \delta \right\}$ — Negative radial derivarive. So no critical points, and $\mathbf{0}$ is local max.
- $\mathcal{R}_2$ the rest — Possible critical points in a sub-region where the Hessian matrix have both negative and positive eigenvalues. Therefore, any critical points in this region must be strict saddle.

Landscape with optimal m

Theorem ([Li, Cai, Wei, IEEE TIT, 2020])

Assume $\boldsymbol{A}$ *is random Gaussian. For any* $\delta \in \left(0, \frac{1}{100}\right]$, *if* $m > C_1 n$, *then with probability at least* $1 - e^{-C_2 n}$ *we have the following*

1. *For any* $\boldsymbol{z} \in \mathcal{R}_1$ *and any unit* $\boldsymbol{u} \in \mathbb{R}^n$,
$$\boldsymbol{u}^T \nabla^2 \tilde{f}(\boldsymbol{z}) \boldsymbol{u} \geq \|\boldsymbol{x}\|_2^2/25.$$
2. *If* $\boldsymbol{z} \in \mathcal{R}_2$ *and* $\nabla \tilde{f}(\boldsymbol{z}) = \boldsymbol{0}$, *then* $\boldsymbol{z}$ *can only be in*
$$\mathcal{R}_2^0 = \left\{ \boldsymbol{z} \;\middle|\; \|\boldsymbol{z}\|^2 \in (1/3 - \delta, 1/3 + \delta)\|\boldsymbol{x}\|^2,\ |\langle \boldsymbol{z}, \boldsymbol{x} \rangle| < \delta \|\boldsymbol{x}\|^2 \right\},$$
which satisfies, for any $\boldsymbol{z} \in \mathcal{R}_2^0 \cap \mathcal{R}_2$,
$$\lambda_{\min}\left(\nabla^2 f(\boldsymbol{z})\right) \leq -3\|\boldsymbol{x}\|^2 \text{ and } \lambda_{\max}\left(\nabla^2 \tilde{f}(\boldsymbol{z})\right) \geq 2\|\boldsymbol{x}\|^2.$$
3. *For any* $\boldsymbol{z} \in \mathcal{R}_3$,
$$\langle \boldsymbol{z}, \nabla \tilde{f}(\boldsymbol{z}) \rangle \leq -5\delta \|\boldsymbol{z}\|^2 \|\boldsymbol{x}\|^2.$$

Here C_1, C_2 *are constants depending only on* δ, γ, β.

Implication of the well-behaved Landscape

Our results imply: with $m = O(n)$, the activated least squares fitting to the intensity equations

- has no spurious local minima. Any local min is global.
- is strongly convex around global minimizers.
- $\mathbf{0}$ is a local max.
- any other critical points are strict saddle.
- Therefore, any algorithm for a local min will give an exact phase retrieval by minimizing $\tilde{f}$.

Implication of the well-behaved Landscape

Our results imply: with $m = O(n)$, the activated least squares fitting to the intensity equations

- has no spurious local minima. Any local min is global.
- is strongly convex around global minimizers.
- $\mathbf{0}$ is a local max.
- any other critical points are strict saddle.
- Therefore, any algorithm for a local min will give an exact phase retrieval by minimizing $\tilde{f}$.

The results are presented in:

- Z. Li, J.-F. Cai, K. Wei, Towards the Optimal Construction of a Loss Function without Spurious Local Minima for Solving Quadratic Equations, *IEEE Transactions on Information Theory*, 66(5): 3242–3260, 2020.

On-going projects

Landscape of the least square fitting on $\mathbb{M}_1$, the manifold of all rank-1 matrices

$$F(\boldsymbol{Z}) = \frac{1}{2m} \sum_{r=1}^{m} \left(\langle \boldsymbol{a}_r \boldsymbol{a}_r^T, \boldsymbol{Z} \rangle - y_r \right)^2.$$

On-going projects

Landscape of the least square fitting on $\mathbb{M}_1$, the manifold of all rank-1 matrices

$$F(\boldsymbol{Z}) = \frac{1}{2m} \sum_{r=1}^{m} \left(\langle \boldsymbol{a}_r \boldsymbol{a}_r^T, \boldsymbol{Z} \rangle - y_r \right)^2.$$

The main advantage is that there is no equivalent critical points of F on $\mathbb{M}_1$.

On-going projects

Landscape of the least square fitting on $\mathbb{M}_1$, the manifold of all rank-1 matrices

$$F(\boldsymbol{Z}) = \frac{1}{2m}\sum_{r=1}^{m}\left(\langle \boldsymbol{a}_r\boldsymbol{a}_r^T, \boldsymbol{Z}\rangle - y_r\right)^2.$$

The main advantage is that there is no equivalent critical points of F on $\mathbb{M}_1$. This is significant especially in the complex case. For example, in the complex case,

- The global minimizer of F on $\mathbb{M}_1$ is unique.
- Global minimizers of f on $\mathbb{C}^n$ is one dimensional and connected. Thus, f cannot be storngly convex around global minimizers.

On-going projects

Landscape of the least square fitting on $\mathbb{M}_1$, the manifold of all rank-1 matrices

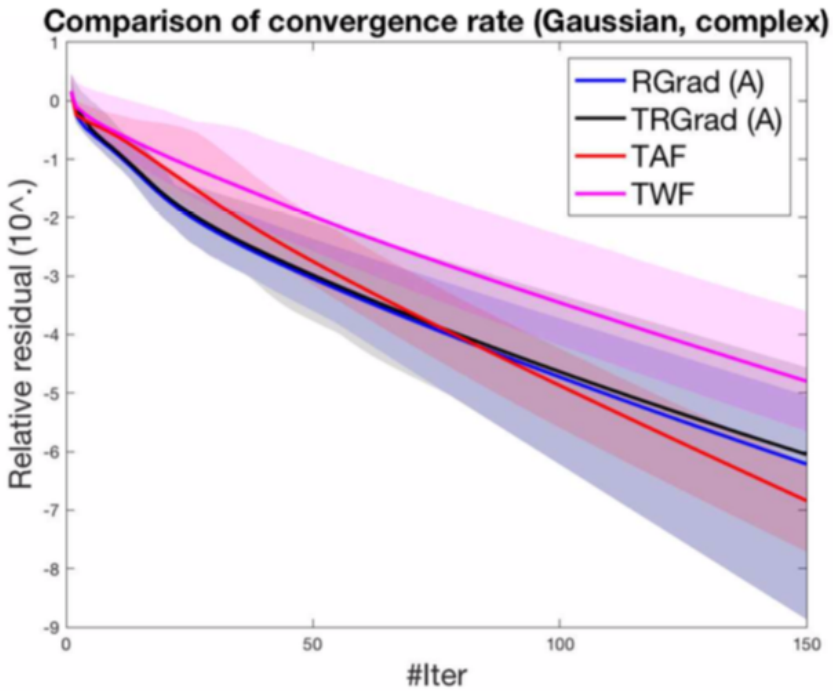
$$F(\boldsymbol{Z}) = \frac{1}{2m}\sum_{r=1}^{m}\left(\langle \boldsymbol{a}_r\boldsymbol{a}_r^T, \boldsymbol{Z}\rangle - y_r\right)^2.$$

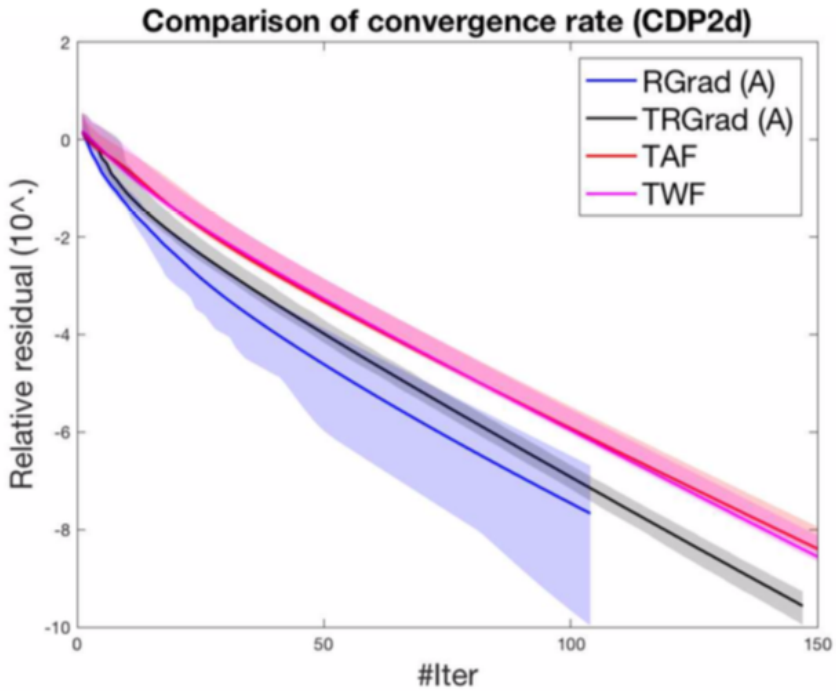
The main advantage is that there is no equivalent critical points of F on $\mathbb{M}_1$. This is significant especially in the complex case. For example, in the complex case,

- The global minimizer of F on $\mathbb{M}_1$ is unique.
- Global minimizers of f on $\mathbb{C}^n$ is one dimensional and connected. Thus, f cannot be storngly convex around global minimizers.

Practically, the optimization of F on $\mathbb{M}_1$ is faster than the optimization of f on $\mathbb{R}^n$ or $\mathbb{C}^n$.

Experiments





Outline

Amplitude equations

Instead of intensity equations, we may also solve the following amplitude equations

$$|\boldsymbol{a}_r^* \boldsymbol{x}| = \sqrt{y}_r, \quad r = 1, \ldots, m.$$

- Solving amplitude equations is often faster than intensity equations.

Amplitude equations

Instead of intensity equations, we may also solve the following amplitude equations

$$|\boldsymbol{a}_r^* \boldsymbol{x}| = \sqrt{y}_r, \quad r = 1, \dots, m.$$

- Solving amplitude equations is often faster than intensity equations.
- Convex solvers: Relax amplitude equations to convex constraints

$$-\sqrt{y}_r \le \boldsymbol{a}_r^* \boldsymbol{x} \le \sqrt{y}_r, \quad r = 1, \dots, m.$$

  - Maximize the correlation with an anchor vector $\boldsymbol{z}_0$, which is a rough estimation of $\boldsymbol{x}$ [Goldstein, Studer, 2018; Bahmani, Romberg, 2017].

$$\max_{\boldsymbol{z}} \ \langle \boldsymbol{z}_0, \boldsymbol{z} \rangle, \quad \text{s.t.} \quad -\sqrt{y}_r \le \boldsymbol{a}_r^* \boldsymbol{z} \le \sqrt{y}_r, \quad r = 1, \dots, m,$$

  - Fast computation: only n unknowns.
  - Nice theory of recovery guarantee: $m = O(n)$ is sufficient for an exact phase retrieval.
  - Drawback: Need a good $\boldsymbol{z}_0$.

- Non-convex solvers.
  - Minimize the least squares fitting of amplitude equations [Wang, Giannakis, Eldar, 2017]

$$\min_{\boldsymbol{z}} g(\boldsymbol{z}), \quad g(\boldsymbol{z}) = \frac{1}{2m} \sum_{r=1}^{m} (|\boldsymbol{a}_r^T \boldsymbol{z}| - \sqrt{y}_r)^2$$

- Non-convex solvers.
  - Minimize the least squares fitting of amplitude equations [Wang, Giannakis, Eldar, 2017]

$$\min_{\boldsymbol{z}} g(\boldsymbol{z}), \quad g(\boldsymbol{z}) = \frac{1}{2m} \sum_{r=1}^{m} (|\boldsymbol{a}_r^T \boldsymbol{z}| - \sqrt{y}_r)^2$$

  - Or guess a phase for amplitude equations, and then solve the resulting linear equations [Wei, 2015; Tan, Vershynin, 2019; Netrapalli, Jain, Sanghavi, 2013].

- Non-convex solvers.
  - Minimize the least squares fitting of amplitude equations [Wang, Giannakis, Eldar, 2017]

$$\min_{\boldsymbol{z}} g(\boldsymbol{z}), \quad g(\boldsymbol{z}) = \frac{1}{2m} \sum_{r=1}^{m} (|\boldsymbol{a}_r^T \boldsymbol{z}| - \sqrt{y}_r)^2$$

  - Or guess a phase for amplitude equations, and then solve the resulting linear equations [Wei, 2015; Tan, Vershynin, 2019; Netrapalli, Jain, Sanghavi, 2013].
  - Usually in two stages: Initialization + Refinement. To overcome non-smoothness, at each step, equations with a small $|\boldsymbol{a}_r^T \boldsymbol{z}|$ will be de-activated.

- Non-convex solvers.
  - Minimize the least squares fitting of amplitude equations [Wang, Giannakis, Eldar, 2017]

$$\min_{\boldsymbol{z}} g(\boldsymbol{z}), \quad g(\boldsymbol{z}) = \frac{1}{2m} \sum_{r=1}^{m} (|\boldsymbol{a}_r^T \boldsymbol{z}| - \sqrt{y}_r)^2$$

  - Or guess a phase for amplitude equations, and then solve the resulting linear equations [Wei, 2015; Tan, Vershynin, 2019; Netrapalli, Jain, Sanghavi, 2013].
  - Usually in two stages: Initialization + Refinement. To overcome non-smoothness, at each step, equations with a small $|\boldsymbol{a}_r^T \boldsymbol{z}|$ will be de-activated.
  - Fast computation: Only matrix-vector products involved.
  - Nice theory of recovery guarantee: Usually $m = O(n)$ is sufficient for an exact phase retrieval.
  - The computation and analysis is done locally in a neighbourhood of $\boldsymbol{x}$.

Least squares fitting of amplitude equations

$$g(\boldsymbol{z}) = \frac{1}{2m}\sum_{r=1}^{m}(|\boldsymbol{a}_r^T\boldsymbol{z}| - \sqrt{y}_r)^2$$

- It is a piecewise quadratic function.

Least squares fitting of amplitude equations

$$g(\boldsymbol{z}) = \frac{1}{2m}\sum_{r=1}^{m}(|\boldsymbol{a}_r^T\boldsymbol{z}| - \sqrt{y}_r)^2$$

- It is a piecewise quadratic function.
- When g is smooth at $\boldsymbol{z}$,

$$\nabla^2 g(\boldsymbol{z}) = \sum_{r=1}^{m}\boldsymbol{a}_r\boldsymbol{a}_r^T \approx \boldsymbol{I}$$

  - That's why amplitude equations based solvers are faster.
  - However, g may have a spurious local minimum, because if g is smooth at a critical point then it must be a local minimum.

Least squares fitting of amplitude equations

$$g(\boldsymbol{z}) = \frac{1}{2m}\sum_{r=1}^{m}(|\boldsymbol{a}_r^T\boldsymbol{z}| - \sqrt{y}_r)^2$$

- It is a piecewise quadratic function.
- When g is smooth at $\boldsymbol{z}$,

$$\nabla^2 g(\boldsymbol{z}) = \sum_{r=1}^{m}\boldsymbol{a}_r\boldsymbol{a}_r^T \approx \boldsymbol{I}$$

  - That's why amplitude equations based solvers are faster.
  - However, g may have a spurious local minimum, because if g is smooth at a critical point then it must be a local minimum.
- But the expectation of g has a good landscape

$$Eg(\boldsymbol{z}) = \frac{1}{2}\left(\|\boldsymbol{z}\|_2^2 - \frac{4}{\pi}\left(\tau + \sigma\arctan\frac{\sigma}{\tau}\right) + \|\boldsymbol{x}\|_2^2\right),$$

where $\sigma = \frac{\boldsymbol{z}^T\boldsymbol{x}}{\|\boldsymbol{z}\|_2\|\boldsymbol{x}\|_2}$ and $\tau = \sqrt{1-\sigma^2}$.

Smoothed amplitude equations least squares fitting

- How to modify g such that the gradient converges fast and the landscape is good?

Smoothed amplitude equations least squares fitting

- How to modify g such that the gradient converges fast and the landscape is good?
- We consider

$$\tilde{g}(\boldsymbol{z}) = \sum_{r=1}^{m} \frac{y_r}{2} \cdot \left(\phi\left(\frac{\boldsymbol{a}_r^T \boldsymbol{z}}{\sqrt{y_r}} \right) - 1 \right)^2 .$$

  - If $\phi(t) = |t|$, then $\tilde{g} = g$.
  - If $\phi(t) = |t|$ for $|t| > \alpha$ and $\phi(t) = \alpha$ for $|t| \leq \alpha$, then gradient descent for $\tilde{g}$ is the truncated amplitude flow algorithm.
  - When ϕ is smooth, $\tilde{g}$ is a smooth approximation to g.

Smoothed amplitude equations least squares fitting

- How to modify g such that the gradient converges fast and the landscape is good?
- We consider

$$\tilde{g}(\boldsymbol{z}) = \sum_{r=1}^{m} \frac{y_r}{2} \cdot \left(\phi\left(\frac{\boldsymbol{a}_r^T \boldsymbol{z}}{\sqrt{y_r}} \right) - 1 \right)^2 .$$

  - If $\phi(t) = |t|$, then $\tilde{g} = g$.
  - If $\phi(t) = |t|$ for $|t| > \alpha$ and $\phi(t) = \alpha$ for $|t| \leq \alpha$, then gradient descent for $\tilde{g}$ is the truncated amplitude flow algorithm.
  - When ϕ is smooth, $\tilde{g}$ is a smooth approximation to g.
- We choose $\phi(t) = |t|$ for $|t| > \alpha$ and $\phi(t) = at^2 + b$ for $|t| \leq \alpha$. Then, we suitable parameters a, b, α
  - The gradient descent for $\tilde{g}$ is even faster than the truncated amplitude flow.

Well-behaved landscape

The modified function

$$\tilde{g}(\boldsymbol{z}) = \sum_{r=1}^{m} \frac{y_r}{2} \cdot \left(\phi\left(\frac{\boldsymbol{a}_r^T \boldsymbol{z}}{\sqrt{y_r}} \right) - 1 \right)^2$$

has a well-behaved landscape: Provided $m = O(n)$, with high probability,

- There is no spurious local minima: all local minima are global.
- $\tilde{g}$ is strongly convex in a neighbourhood of the global minima.
- $\mathbf{0}$ is a local maximum.
- The Hessian at all other critical points has both positive and negative eigenvalues.

Well-behaved landscape

The modified function

$$\tilde{g}(\boldsymbol{z}) = \sum_{r=1}^{m} \frac{y_r}{2} \cdot \left(\phi\left(\frac{\boldsymbol{a}_r^T \boldsymbol{z}}{\sqrt{y_r}} \right) - 1 \right)^2$$

has a well-behaved landscape: Provided $m = O(n)$, with high probability,

- There is no spurious local minima: all local minima are global.
- $\tilde{g}$ is strongly convex in a neighbourhood of the global minima.
- $\mathbf{0}$ is a local maximum.
- The Hessian at all other critical points has both positive and negative eigenvalues.

J.-F. Cai, M. Huang, D. Li, Y. Wang, Solving Phase Retrieval with Random Initial Guess Is Nearly as Good as Spectral Initialization, Forthcoming.

Outline

Phase retrieval and artificial neurons

- Our phase retrieval model can be viewed as one artificial neuron.
  - Intensity equations:

$$y_r = \sigma(\boldsymbol{a}_r^T \boldsymbol{x}), \quad r = 1, \dots, m,$$

    where $\{(\boldsymbol{a}_r, y_r)\}_{r=1}^m$ are input-output pairs, $\boldsymbol{x}$ is the weight, $\sigma(t) = t^2$ is the activation.
  - Amplitude equations:

$$\sqrt{y_r} = \sigma(\boldsymbol{a}_r^T \boldsymbol{x}), \quad r = 1, \dots, m,$$

    where $\{(\boldsymbol{a}_r, \sqrt{y_r})\}_{r=1}^m$ are input-output pairs, $\boldsymbol{x}$ is the weight, and $\sigma(t) = |t|$ is the activation.

Phase retrieval and artificial neurons

- Our phase retrieval model can be viewed as one artificial neuron.
  - Intensity equations:

$$y_r = \sigma(\boldsymbol{a}_r^T \boldsymbol{x}), \quad r = 1, \ldots, m,$$

    where $\{(\boldsymbol{a}_r, y_r)\}_{r=1}^m$ are input-output pairs, $\boldsymbol{x}$ is the weight, $\sigma(t) = t^2$ is the activation.
  - Amplitude equations:

$$\sqrt{y_r} = \sigma(\boldsymbol{a}_r^T \boldsymbol{x}), \quad r = 1, \ldots, m,$$

    where $\{(\boldsymbol{a}_r, \sqrt{y_r})\}_{r=1}^m$ are input-output pairs, $\boldsymbol{x}$ is the weight, and $\sigma(t) = |t|$ is the activation.

Our landscape analysis may be extended from phase retrieval to neural network training, where the input data are i.i.d. Gaussian.

Landscape of neural network training

Consider a neural network with one hidden layer and the coefficients c_i of the output layer fixed. We train the network from data pairs $\{(\boldsymbol{a}_r, y_r)\}_{r=1}^m$, i.e.,

$$\ell(\boldsymbol{X}) = \sum_{r=1}^{m} \left(\sum_{i=1}^{k} c_i \cdot \sigma(\boldsymbol{a}_r^T \boldsymbol{x}_i) - y_r \right)^2$$

Landscape of neural network training

Consider a neural network with one hidden layer and the coefficients c_i of the output layer fixed. We train the network from data pairs $\{(\boldsymbol{a}_r, y_r)\}_{r=1}^m$, i.e.,

$$\ell(\boldsymbol{X}) = \sum_{r=1}^{m} \left(\sum_{i=1}^{k} c_i \cdot \sigma(\boldsymbol{a}_r^T \boldsymbol{x}_i) - y_r \right)^2$$

Assume $\{\boldsymbol{a}_r\}_{r=1}^m$ is i.i.d. random Gaussian.

- If $\sigma(t) = |t|$, then ℓ may have spurious local minima according to our previous argument.
- If $\sigma(t) = \max\{t, 0\}$, then ℓ with $(k, m) \in \{(8, 9), (10, 11), \ldots, (19, 20)\}$ have spurious local minima. [Safran, Shamir, 2018].
- If σ is a polynomial, then the expectation of ℓ has no spurious local minima when over-parametrized. [Venturi, Bendeira, Bruna, 2018]
- Many other results available.

Landscape of neural network training

Consider a neural network with one hidden layer and the coefficients c_i of the output layer fixed. We train the network from data pairs $\{(\boldsymbol{a}_r, y_r)\}_{r=1}^m$, i.e.,

$$\ell(\boldsymbol{X}) = \sum_{r=1}^{m} \left(\sum_{i=1}^{k} c_i \cdot \sigma(\boldsymbol{a}_r^T \boldsymbol{x}_i) - y_r \right)^2$$

Assume $\{\boldsymbol{a}_r\}_{r=1}^m$ is i.i.d. random Gaussian.

- If $\sigma(t) = |t|$, then ℓ may have spurious local minima according to our previous argument.
- If $\sigma(t) = \max\{t, 0\}$, then ℓ with $(k, m) \in \{(8, 9), (10, 11), \ldots, (19, 20)\}$ have spurious local minima. [Safran, Shamir, 2018].
- If σ is a polynomial, then the expectation of ℓ has no spurious local minima when over-parametrized. [Venturi, Bendeira, Bruna, 2018]
- Many other results available.
- We are investigating: (1) the landscape with $\sigma(t) = t^2$ and finite samples. (2) the landscape with a smoothed ReLU and finite samples.

Outline

Conclusion

- Non-convex optimization is a powerful tool for solving many problems.
- Through landscape analysis, some non-convex optimizations are not as difficult as we suppose in the general case.
- Examples includes SVD computation, neural network training, and phase retrieval presented in this talk.
- Some other examples that has been revealed recently are matrix completion, robust principal component analysis, tensor decomposition, synchronization networks, etc.
- Similar to phase retrieval, many such examples are quadratic equations related.


THANK YOU
Any questions?