Nonconvex Demixing from Bilinear Measurements

Yuanming Shi

信息科学与技术学院
School of Information Science and Technology

Outline

- **Motivations**
  - Blind deconvolution meets blind demixing
- **Two Vignettes:**
  - **Implicitly regularized Wirtinger flow**
    - Why nonconvex optimization?
    - Implicitly regularized Wirtinger flow
  - **Matrix optimization over manifolds**
    - Why manifold optimization?
    - Riemannian optimization for blind demixing

Motivations: Blind deconvolution meets blind demixing

Blind deconvolution

- In many science and engineering problems, the observed signal can be modeled as:

$$z(t) = f(t) * g(t)$$

where $*$ is the convolution operator

  - $f(t)$ is a physical signal of interest
  - $g(t)$ is the impulse response of the sensory system

- **Applications:** astronomy, neuroscience, image processing, computer vision, wireless communications, microscopy data processing,…
- **Blind deconvolution:** estimate $f(t)$ and $g(t)$ given $z(t)$

Image deblurring

- Blurred images due to camera shake can be modeled as a convolution of the *latent sharp image* and a *kernel* capturing the motion of the camera

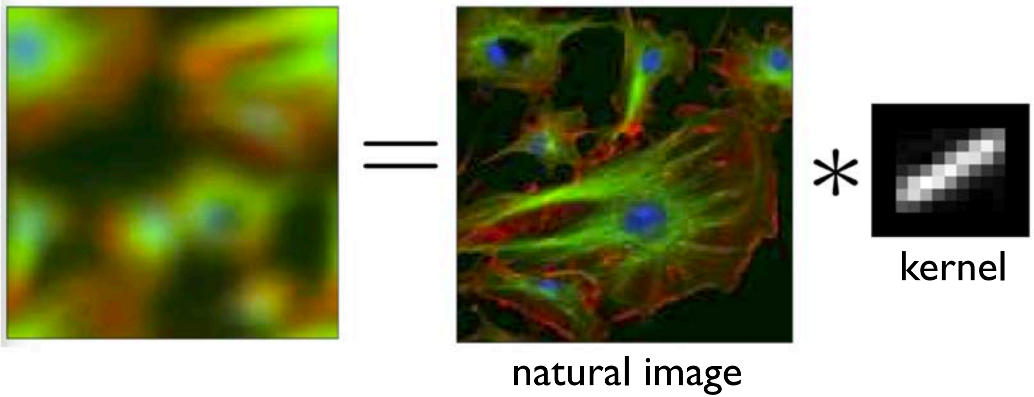
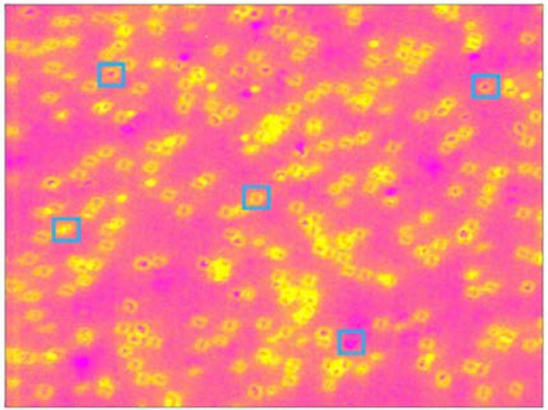


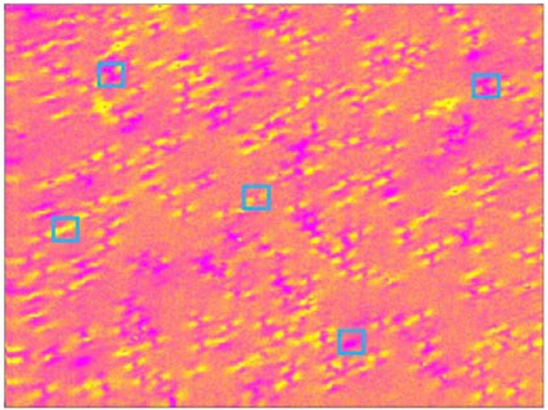
Fig. credit: Chi

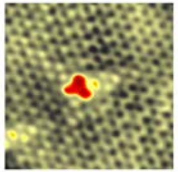
How to find the high-resolution image and the blurring kernel simultaneously?

Microscopy data analysis

- **Defects:** the electronic structure of the material is contaminated by randomly and sparsely distributed "defects"







Doped Graphene

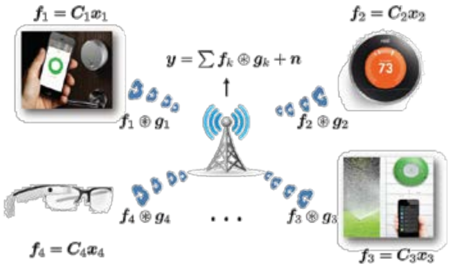
Fig. credit: Wright

How to determine the locations and characteristic signatures of the defects?

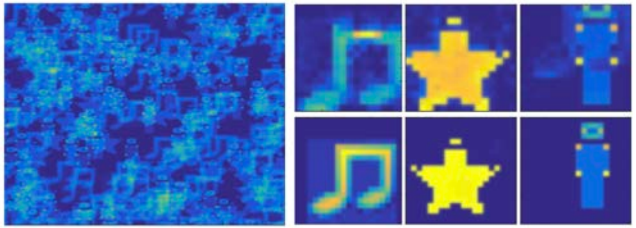
Blind demixing

- The received measurement consists of the sum of all convolved signals

$$z(t) = \sum_{i=1}^{s} f_i(t) * g_i(t)$$



low-latency communication for IoT



convolutional dictionary learning (multi kernel)

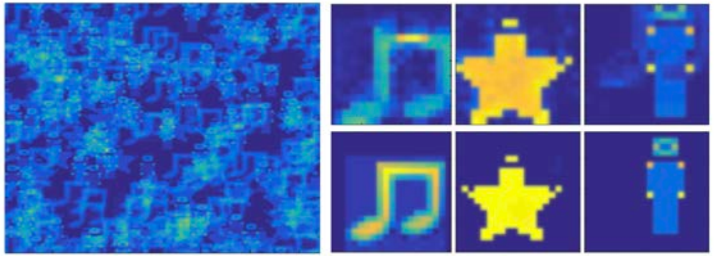
- **Applications:** IoT, dictionary learning, neural spike sorting,…
- **Blind demixing:** estimate $\{f_i(t)\}$ and $\{g_i(t)\}$ given $z(t)$

Convolutional dictionary learning

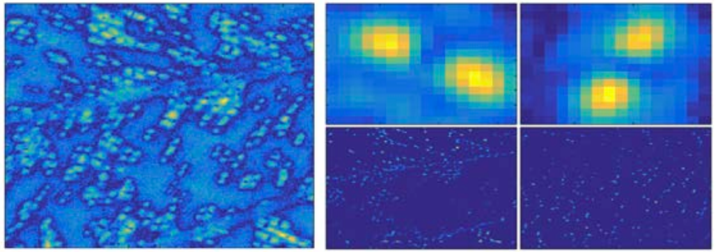
- The observation signal is the superposition of several convolutions

$$z(t) = \sum_{i=1}^{s} f_i(t) * g_i(t)$$

Fig. credit: Wright



experiment on synthetic image



experiment on microscopy image

How to recover multiple kernels and the corresponding activation signals?

Low-latency communications for IoT

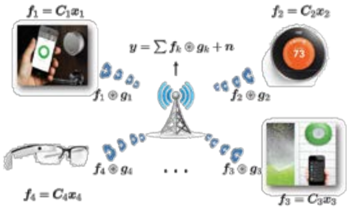
- **Packet structure:** metadata (preamble (PA) and header (H)) and data

long data packet in current wireless systems

short data packet in IoT

- **Proposal:** transmitters just send overhead-free signals, and the receiver can still extract the information

$$z(t) = \sum_{i=1}^{s} f_i(t) * g_i(t)$$



How to detect data without channel estimation in multi-user environments?

Demixing from bilinear model?

$$z(t) = \sum_{i=1}^{s} f_i(t) * g_i(t)$$

Bilinear model

- Translate into the frequency domain…

$$z = \sum_{i=1}^{s} f_i \odot g_i \in \mathbb{C}^m$$

- **Subspace assumptions:** f_i and g_i lie in some known low-dimensional subspaces

$$f_i = A_i x_i^\natural \in \mathbb{C}^m \qquad g_i = B h_i^\natural \in \mathbb{C}^m$$

where $A_i = [a_{i1}, \cdots, a_{im}]^* \in \mathbb{C}^{m \times L}$, $B = [b_1, \cdots, b_m]^* \in \mathbb{C}^{m \times K}$ and $L, K \ll m$

$$a_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, I) \qquad \{b_j\} : \text{partial Fourier basis}$$

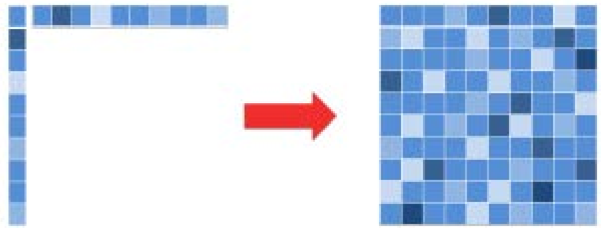
- **Demixing from bilinear measurements:**

$$\text{find} \quad \{x_i\}, \{h_i\} \quad \text{subject to} \quad z_j = \sum_{i=1}^{s} b_j^* h_i x_i^* a_{ij}, \quad 1 \le j \le m$$

An equivalent view: low-rank factorization

- **Lifting:** introduce $\boldsymbol{M}_k^\natural = \boldsymbol{h}_k^\natural \boldsymbol{x}_k^{\natural *}$ to linearize constraints

$$z_j = \sum_{i=1}^{s} \boldsymbol{b}_i^* \boldsymbol{h}_i^\natural \boldsymbol{x}_i^{\natural *} \boldsymbol{a}_{ij} = \sum_{i=1}^{s} \boldsymbol{b}_i^* \underbrace{(\boldsymbol{h}_i^\natural \boldsymbol{x}_i^{\natural *})}_{\boldsymbol{M}_i^\natural \in \mathbb{C}^{K \times L}} \boldsymbol{a}_{ij}$$



- **Low-rank matrix optimization problem**

$$\begin{aligned} \text{find} \quad & \{\boldsymbol{M}_i\} \\ \text{subject to} \quad & z_j = \sum_{i=1}^{s} \boldsymbol{b}_i^* \boldsymbol{M}_i \boldsymbol{a}_{ij}, \quad j = 1, \cdots, m \\ & \operatorname{rank}(\boldsymbol{M}_i) = 1, \ i = 1, \cdots, s, \end{aligned}$$

Convex relaxation

- Ling and Strohmer (TIT'2017) proposed to solve the **nuclear norm minimization problem:**

$$\begin{aligned} &\text{minimize} \quad \sum_{k=1}^{s} \|\boldsymbol{M}_k\|_* \\ &\text{subject to} \quad z_j = \sum_{k=1}^{s} \boldsymbol{b}_k^* \boldsymbol{M}_k \boldsymbol{a}_{kj}, \quad j = 1, \cdots, m \end{aligned}$$

$\boldsymbol{a}_{kj} \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(\mathbf{0}, \boldsymbol{I})$

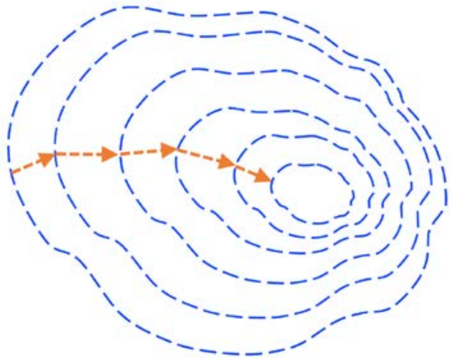
$\{\boldsymbol{b}_j\}$: partial Fourier basis

➢ **Sample-efficient:** $m \gtrsim s^2 \max\{K, L\} \log^2 m$ samples for exact recovery if $\{\boldsymbol{b}_j\}$ is incoherent w.r.t. $\{\boldsymbol{h}_k^\natural\}$

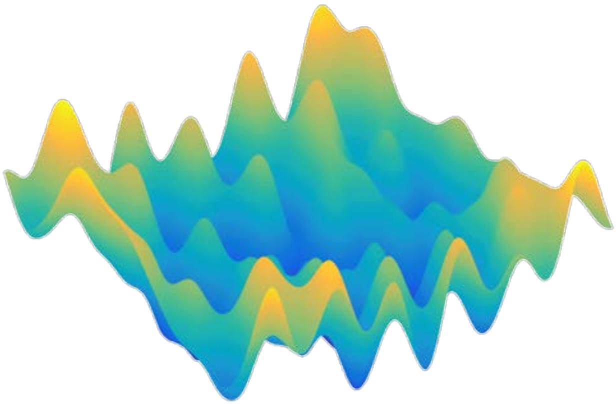
➢ **Computational-expensive:** SDP in the lifting space

Can we solve the nonconvex matrix optimization problem directly?

Vignettes A: Implicitly regularized Wirtinger flow



Why nonconvex optimization?

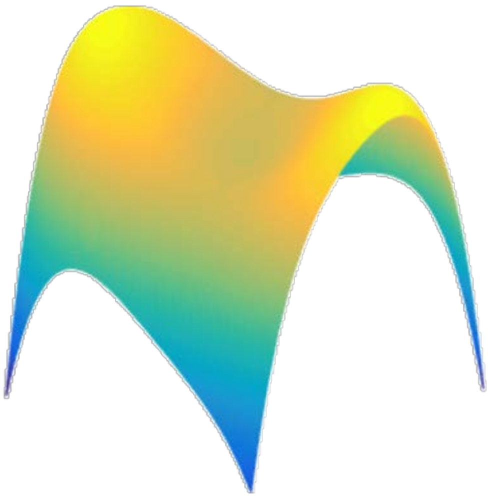


Nonconvex problems are everywhere

- Empirical risk minimization is usually nonconvex

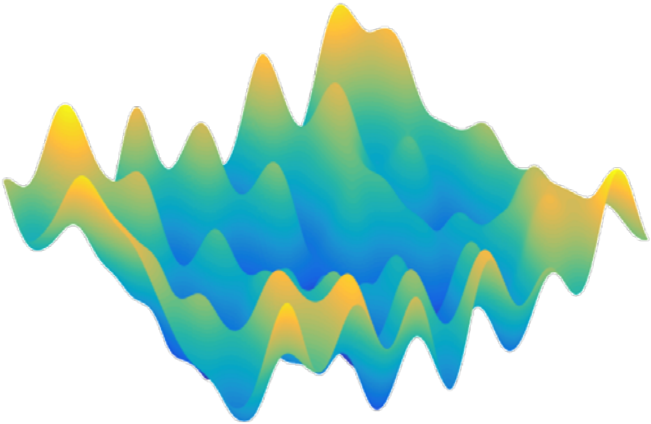
$$\underset{\boldsymbol{x}}{\text{minimize}} \quad f(\boldsymbol{x}; \boldsymbol{\theta})$$

  - low-rank matrix completion
  - blind deconvolution/demixing
  - dictionary learning
  - phase retrieval
  - mixture models
  - deep learning
  - …



Nonconvex optimization may be super scary

- **Challenges:** saddle points, local optima, bumps,…



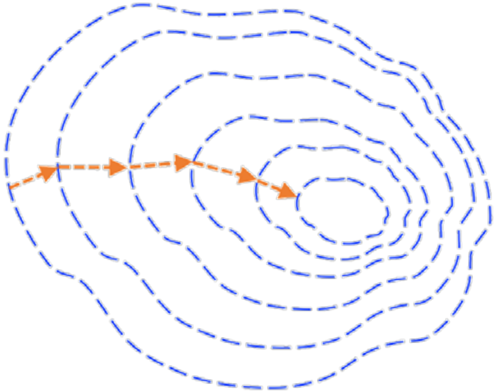


Fig. credit: Chen

- **Fact:** they are usually solved on a daily basis via simple algorithms like (stochastic) gradient descent

Statistical models come to rescue

- **Blessings:** when data are generated by certain statistical models, problems are often much nicer than worst-case instances

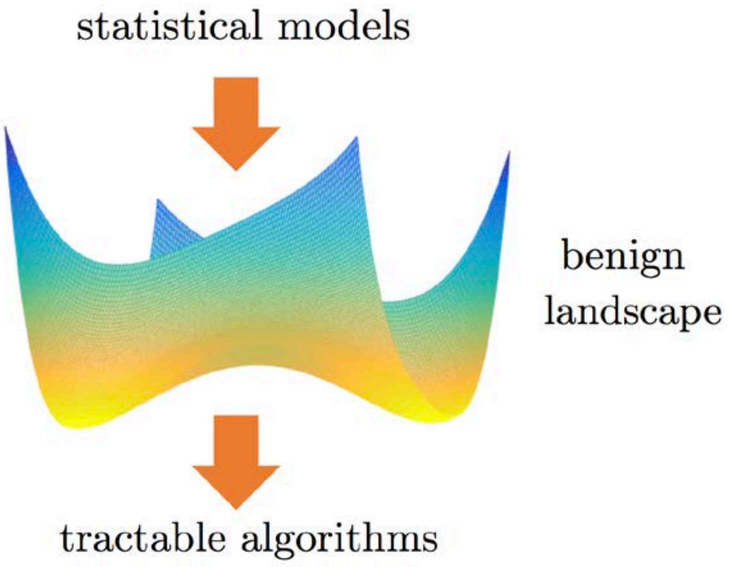
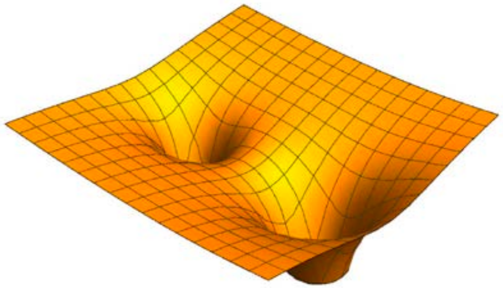


Fig. credit: Chen

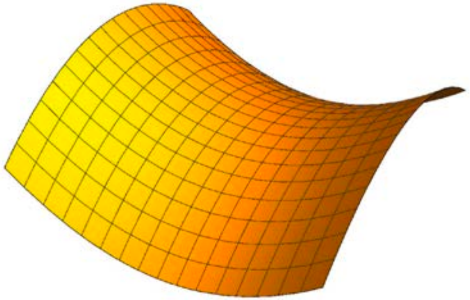
First-order stationary points

- Saddle points and local minima:

$$\lambda_{\min}(\nabla^2 f(\boldsymbol{z})) \begin{cases} > 0 & \text{local minimum} \\ = 0 & \text{local minimum or saddle point} \\ < 0 & \text{strict saddle point} \end{cases}$$



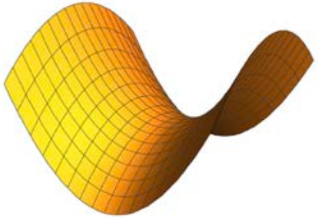
Local minima



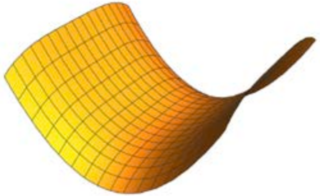
Saddle points/local maxima

First-order stationary points

- **Applications:** PCA, matrix completion, dictionary learning etc.
  - **Local minima:** either all local minima **are** global minima or all local minima **as good as** global minima
  - **Saddle points:** **very poor** compared to global minima; **several** such points



Strict saddle point



Non-strict saddle point

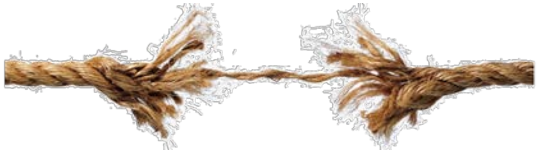
- **Bottomline:** local minima much more desirable than saddle points

How to escape saddle points efficiently?

Statistics meets optimization

- **Proposal:** separation of landscape analysis and generic algorithm design

landscape analysis (statistics)



generic algorithms (optimization)

all local minima are global minima

- dictionary learning (Sun et al. '15)
- phase retrieval (Sun et al. '16)
- matrix completion (Ge et al. '16)
- synchronization (Bandeira et al. '16)
- inverting deep neural nets (Hand et al. '17)
- ...

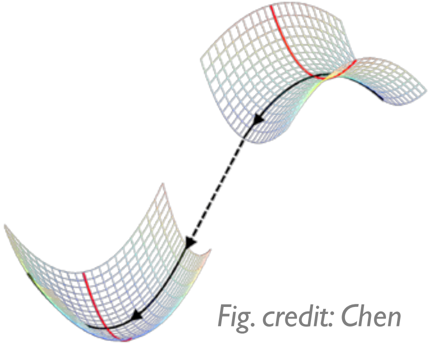


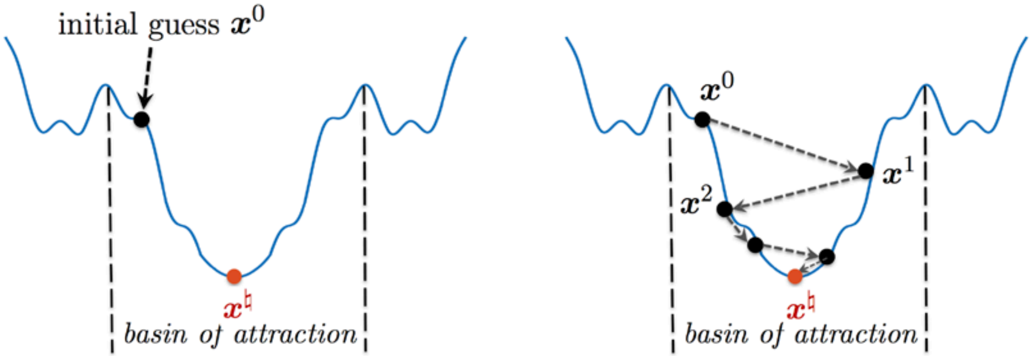
Fig. credit: Chen

all the saddle points can be escaped

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

Issue: conservative computational guarantees for specific problems (e.g., phase retrieval, blind deconvolution, matrix completion)

Solution: blending landscape and convergence analysis



implicitly regularized Wirtinger flow

A natural least-squares formulation

- **Goal:** demixing from bilinear measurements

Given: $$y_j = \sum_{i=1}^{s} \boldsymbol{b}_j^* \boldsymbol{h}_i^\natural \boldsymbol{x}_i^{\natural *} \boldsymbol{a}_{ij}, \quad 1 \le j \le m$$

$$\underset{\{\boldsymbol{h}_k\},\{\boldsymbol{x}_k\}}{\text{minimize}} \; f(\boldsymbol{h}, \boldsymbol{x}) := \sum_{j=1}^{m} \sum_{k=1}^{s} \left(\boldsymbol{b}_j^* \boldsymbol{h}_k \boldsymbol{x}_k^* \boldsymbol{a}_{kj} - y_j \right)^2$$

  - **Pros:** computational-efficient in the natural parameter space
  - **Cons:** $f(\cdot)$ is nonconvex: bilinear constraint, scaling ambiguity

Wirtinger flow

- Least-square minimization via Wirtinger flow (Candes, Li, Soltanolkotabi '14)

$$\underset{\{\boldsymbol{h}_k\},\{\boldsymbol{x}_k\}}{\text{minimize}} \quad f(\boldsymbol{h}, \boldsymbol{x}) := \sum_{j=1}^{m} \sum_{k=1}^{s} \left(\boldsymbol{b}_j^* \boldsymbol{h}_k \boldsymbol{x}_k^* \boldsymbol{a}_{kj} - y_j\right)^2$$

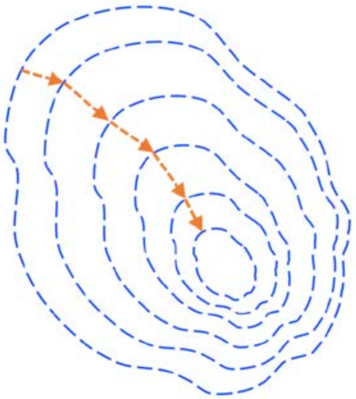
  - **Spectral initialization by top eigenvector of**

$$\boldsymbol{M}_k := \sum_{j=1}^{m} \boldsymbol{y}_j \boldsymbol{b}_j \boldsymbol{a}_{kj}^*, \quad k = 1, \cdots, s$$

  - **Gradient iterations**

$$\boldsymbol{h}_k^{t+1} = \boldsymbol{h}_k^t - \eta \frac{1}{\|\boldsymbol{x}_k^t\|_2^2} \nabla_{\boldsymbol{h}_k} f(\boldsymbol{h}^t, \boldsymbol{x}^t)$$

$$\boldsymbol{x}_k^{t+1} = \boldsymbol{x}_k^t - \eta \frac{1}{\|\boldsymbol{h}_k^t\|_2^2} \nabla_{\boldsymbol{x}_k} f(\boldsymbol{h}^t, \boldsymbol{x}^t)$$



Two-stage approach

- **Initialize** within local basin sufficiently close to ground-truth (i.e., strongly convex, no saddle points/ local minima)
- **Iterative refinement** via some iterative optimization algorithms

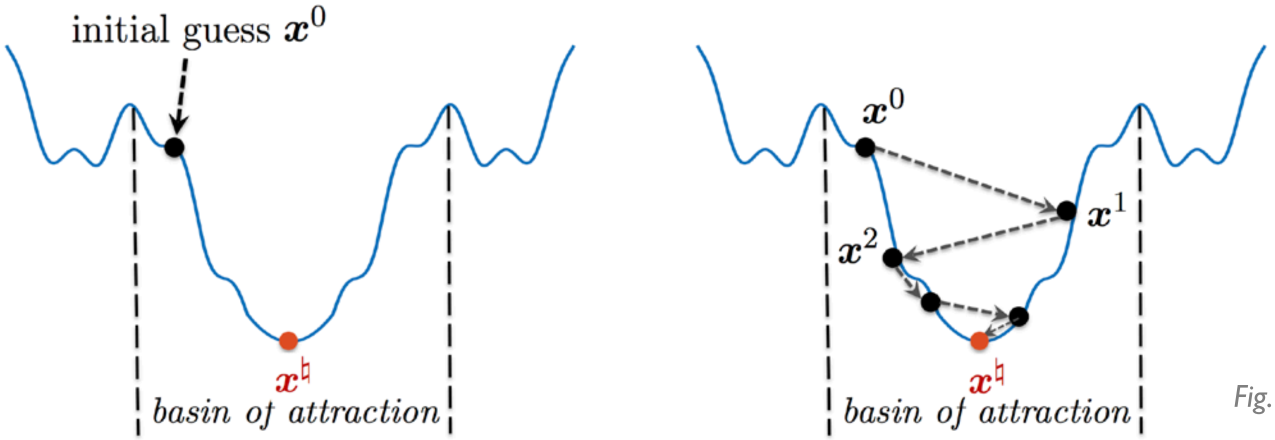
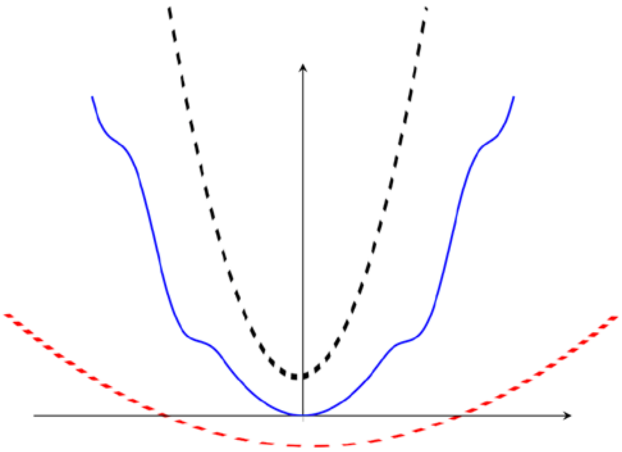


Fig. credit: Chen

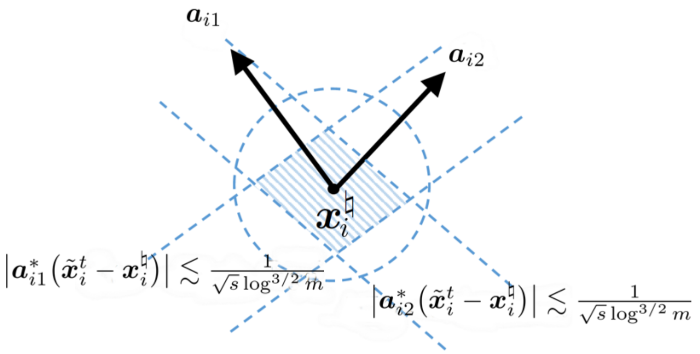
Gradient descent theory

- Two standard conditions that enable **geometric convergence** of GD
  - (local) restricted strong convexity
  - (local) smoothness



Gradient descent theory

- **Question:** which region enjoys both strong convexity and smoothness?



  - x is not far away from $x^\natural$ (convexity)
  - x is incoherent w.r.t. sampling vectors (incoherence region for smoothness)

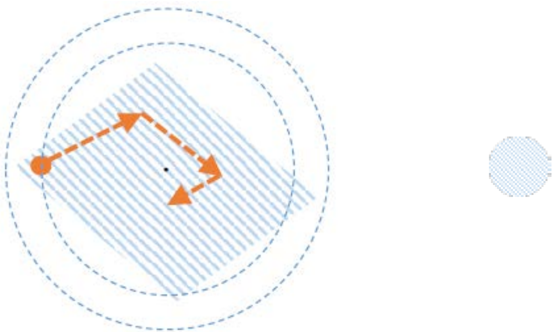
Prior works suggest enforcing *regularization* (e.g., regularized loss [Ling & Strohmer'17]) to promote incoherence

Our finding: WF is implicitly regularized

- WF (GD) implicitly forces iterates to remain **incoherent** with $\{\boldsymbol{a}_{ij}\}$

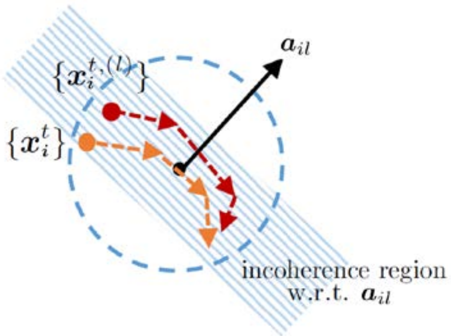
$$\max_{1 \le i \le s, 1 \le j \le m} \left| \boldsymbol{a}_{ij}^{*} \left(\alpha_i^t \boldsymbol{x}_i^t - \boldsymbol{x}_i^{\natural} \right) \right| \lesssim \frac{1}{\sqrt{s} \log^{3/2} m} \|\boldsymbol{x}_i^{\natural}\|_2$$

  - ➢ cannot be derived from generic optimization theory
  - ➢ relies on finer *statistical analysis* for entire trajectory of GD



region of local strong
convexity and smoothness

Key proof idea: leave-one-out analysis



- introduce leave-one-out iterates $\boldsymbol{x}_i^{t,(l)}$ by running WF without *l*-th sample
- leave-one-out iterate $\boldsymbol{x}_i^{t,(l)}$ is independent of $\boldsymbol{a}_{il}$
- leave-one-out iterate $\boldsymbol{x}_i^{t,(l)} \approx$ true iterate $\boldsymbol{x}_i^t$
- $\boldsymbol{x}_i^t$ is nearly independent of (i.e., nearly orthogonal to) $\boldsymbol{a}_{il}$

Theoretical guarantees

- With i.i.d. Gaussian design, WF **(regularization-free)** achieves
  - **Incoherence**

$$\max_{1\le i\le s, 1\le j\le m}\left|\boldsymbol{a}_{ij}^{*}\left(\alpha_i^t\boldsymbol{x}_i^t-\boldsymbol{x}_i^{\natural}\right)\right|\lesssim\frac{1}{\sqrt{s}\log^{3/2}m}\|\boldsymbol{x}_i^{\natural}\|_2$$

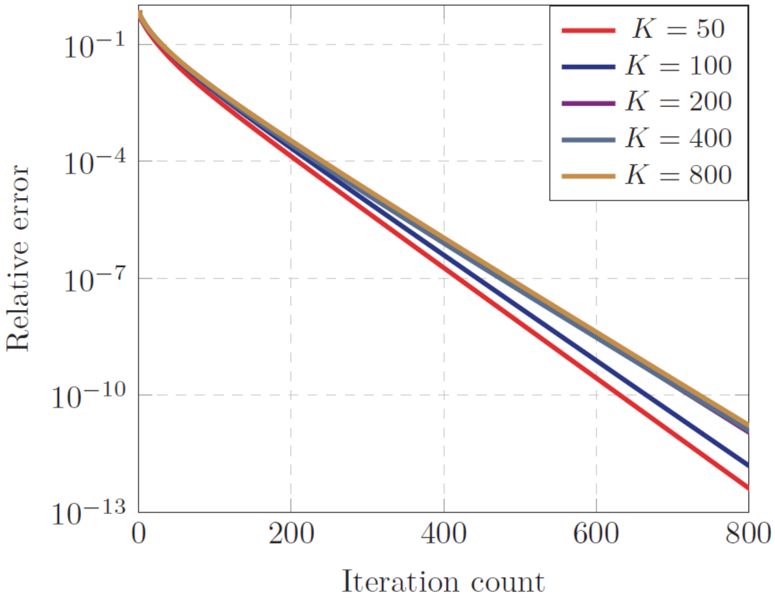
  - **Near-linear convergence rate**

$$\mathrm{dist}(\boldsymbol{z}^t,\boldsymbol{z}^{\natural})\lesssim\left(1-\frac{\eta}{16\kappa}\right)^t\frac{1}{\log^2 m}$$

- **Summary:**
  - **Sample size:** $m\gtrsim s^2\max\{K,L\}\mathrm{poly}\log m$
  - **Stepsize:** $\eta\asymp s^{-1}$ vs. $\eta\lesssim(sm)^{-1}$ [Ling & Strohmer'17]
  - **Computational complexity:** $\mathcal{O}(s\log\frac{1}{\varepsilon})$ vs. $\mathcal{O}(sm\log\frac{1}{\varepsilon})$ [Ling & Strohmer'17]

Numerical results

- stepsize: $\eta = 0.1$
- number of users: $s = 10$
- sample size: $m = 50K$



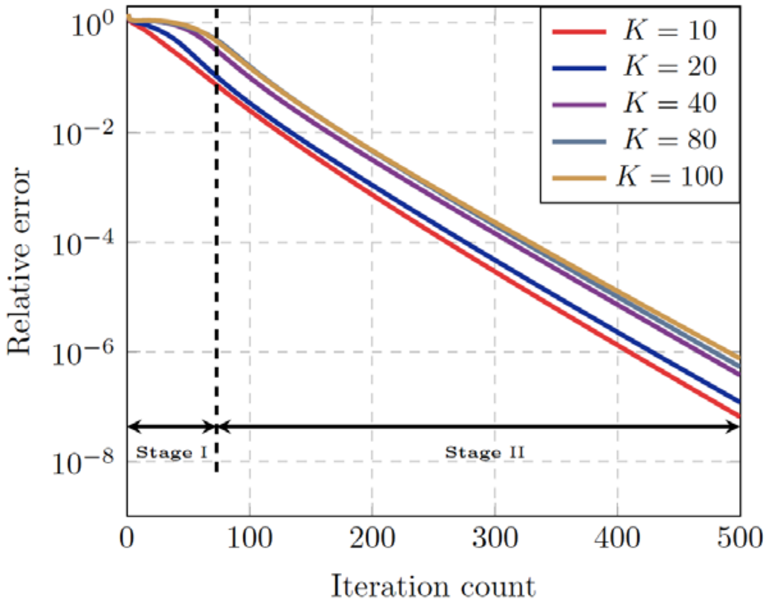
linear convergence:
WF attains ε- accuracy within $\mathcal{O}(s \log \frac{1}{\varepsilon})$ iterations

Is carefully-designed initialization necessary?

Numerical results of randomly initialized WF

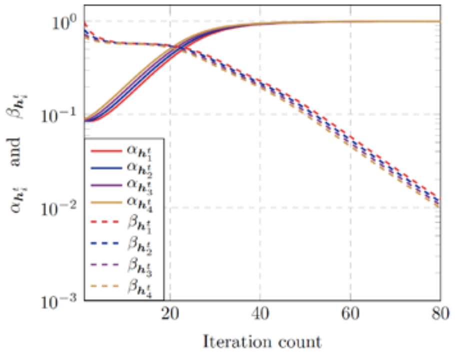
- stepsize: $\eta = 0.1$
- number of users: $s = 10$
- sample size: $m = 50K$
- initial point:

$$\boldsymbol{h}_i^0 \sim \mathcal{N}(\boldsymbol{0}, \tfrac{1}{L}\boldsymbol{I}_L), \boldsymbol{x}_i^0 \sim \mathcal{N}(\boldsymbol{0}, \tfrac{1}{K}\boldsymbol{I}_K),$$
$$i = 1, \cdots, s, \ (K = L)$$



Randomly initialized WF enters local basin within $\mathcal{O}(s \log K)$ iterations

Analysis: population dynamics



Population level (infinite sample)

$$\boldsymbol{h}_i^{t+1} = \boldsymbol{h}_i^t - \eta \frac{1}{\|\boldsymbol{x}_i^t\|_2^2} \nabla_{\boldsymbol{h}_i} F(\boldsymbol{h}^t, \boldsymbol{x}^t)$$

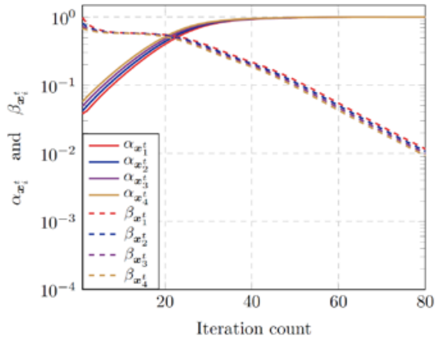
$$\nabla_{\boldsymbol{h}_i} F(\boldsymbol{h}, \boldsymbol{x}) := \mathbb{E}[\nabla_{\boldsymbol{h}_i} f(\boldsymbol{h}, \boldsymbol{x})] = \|\boldsymbol{x}_i\|_2^2 \boldsymbol{h}_i - (\boldsymbol{x}_i^{\natural *} \boldsymbol{x}_i) \boldsymbol{h}_i^{\natural},$$

- Signal strength: $\alpha_{\boldsymbol{h}_i^t} := \langle \boldsymbol{h}_i^{\natural}, 1/\overline{\omega_i^t} \boldsymbol{h}_i^t \rangle \|\boldsymbol{h}_i^{\natural}\|_2$, ω_i^t is the alignment parameter
- Size of residual component: $\beta_{\boldsymbol{h}_i^t} := \left\| \boldsymbol{h}_i^t - \langle \boldsymbol{h}_i^{\natural}, 1/\overline{\omega_i^t} \boldsymbol{h}_i^t \rangle \boldsymbol{h}_i^{\natural} \right\|_2$
- **State evolution**

$$\left.\begin{aligned} \alpha_{\boldsymbol{h}_i^{t+1}} &= (1-\eta)\alpha_{\boldsymbol{h}_i^t} + \eta \alpha_{\boldsymbol{x}_i^t} / (\alpha_{\boldsymbol{x}_i^t}^2 + \beta_{\boldsymbol{x}_i^t}^2) \\ \beta_{\boldsymbol{h}_i^{t+1}} &= (1-\eta)\beta_{\boldsymbol{h}_i^t} \end{aligned}\right\} \longrightarrow \begin{aligned} T_\gamma &= \mathcal{O}(s \log K) \\ \mathrm{dist}(\boldsymbol{h}_i^{T_\gamma}, \boldsymbol{h}_i^{\natural}) &\le \gamma \end{aligned}$$

local basin

Analysis: population dynamics



Population level (infinite sample)

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t - \eta \frac{1}{\|\boldsymbol{h}_i^t\|_2^2} \nabla_{\boldsymbol{x}_i} F(\boldsymbol{h}^t, \boldsymbol{x}^t)$$

$$\nabla_{\boldsymbol{x}_i} F(\boldsymbol{h}, \boldsymbol{x}) := \mathbb{E}[\nabla_{\boldsymbol{x}_i} f(\boldsymbol{h}, \boldsymbol{x})] = \|\boldsymbol{h}_i\|_2^2 \boldsymbol{x}_i - (\boldsymbol{h}_i^{\natural *} \boldsymbol{h}_i) \boldsymbol{x}_i^{\natural}$$

- Signal strength: $\alpha_{\boldsymbol{x}_i^t} := \langle \boldsymbol{x}_i^{\natural}, \omega_i^t \boldsymbol{x}_i^t \rangle \|\boldsymbol{x}_i^{\natural}\|_2$, ω_i^t is the alignment parameter
- Size of residual component: $\beta_{\boldsymbol{x}_i^t} := \left\| \boldsymbol{x}_i^t - \langle \boldsymbol{x}_i^{\natural}, \omega_i^t \boldsymbol{x}_i^t \rangle \boldsymbol{x}_i^{\natural} \right\|_2$
- **State evolution**

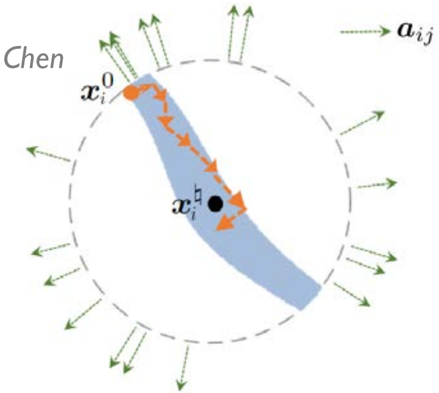
$$\left.\begin{aligned} \alpha_{\boldsymbol{x}_i^{t+1}} &= (1-\eta)\alpha_{\boldsymbol{x}_i^t} + \eta \alpha_{\boldsymbol{h}_i^t} / (\alpha_{\boldsymbol{h}_i^t}^2 + \beta_{\boldsymbol{h}_i^t}^2) \\ \beta_{\boldsymbol{x}_i^{t+1}} &= (1-\eta)\beta_{\boldsymbol{x}_i^t} \end{aligned}\right\} \longrightarrow \begin{aligned} & T_\gamma = \mathcal{O}(s \log K) \\ & \mathrm{dist}(\boldsymbol{x}_i^{T_\gamma}, \boldsymbol{x}_i^{\natural}) \le \gamma \end{aligned}$$

local basin

Analysis: finite-sample analysis

$$z_i^{t+1} = \begin{bmatrix} h_i^{t+1} \\ x_i^{t+1} \end{bmatrix} = \underbrace{\begin{bmatrix} h_i^t - \eta/\|x_i^t\|_2 \cdot \nabla_{h_i} F(z) \\ x_i^t - \eta/\|x_i^t\|_2 \cdot \nabla_{x_i} F(z) \end{bmatrix}}_{:=m(z_i^t)} - \underbrace{\begin{bmatrix} \eta/\|x_i^t\|_2 \cdot (\nabla_{h_i} f(z) - \nabla_{h_i} F(z)) \\ \eta/\|h_i^t\|_2 \cdot (\nabla_{x_i} f(z) - \nabla_{x_i} F(z)) \end{bmatrix}}_{:=r(z_i^t)}$$

Fig. credit: Chen



$r(z_i^t)$ is well-controlled in this region

- Population-level analysis holds approximately if

  $r(z_i^t) \ll m(z_i^t)$

- $r(z_i^t)$ is well-controlled if x_i^t is independent of $\{a_{ij}\}$
- **Key analysis ingredient:** show x_i^t is "nearly independent" of each $\{a_{ij}\}$

Theoretical guarantees

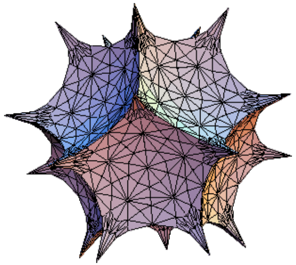
- With i.i.d. Gaussian design, WF with random initialization achieves

$$\mathrm{dist}(\boldsymbol{z}^t, \boldsymbol{z}^\natural) \lesssim \gamma \left(1 - \frac{\eta}{16\kappa}\right)^{t-T_\gamma} \|\boldsymbol{z}^\natural\|_2, \quad t \geq T_\gamma$$

Summary:

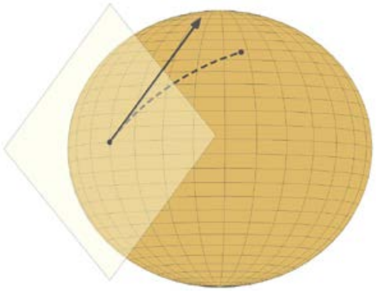
- **Stepsize:** $\eta \asymp s^{-1}$
- **Sample size:** $m \gtrsim s^2 \max\{K, L\} \mathrm{poly} \log m$
- **Stage I:** reach local basin $\mathrm{dist}(\boldsymbol{z}^t, \boldsymbol{z}^\natural) \leq \gamma$ within $T_\gamma = \mathcal{O}(s \log K)$ iterations
- **Stage II:** linear convergence $\mathcal{O}(s \log \frac{1}{\varepsilon})$
- **Computational complexity:** $\mathcal{O}(s \log K + s \log \frac{1}{\varepsilon})$

Vignettes B: ***Matrix optimization*** *over* ***manifolds***



Optimization over Riemannian Manifolds (non-Euclidean geometry)

Why manifold optimization?

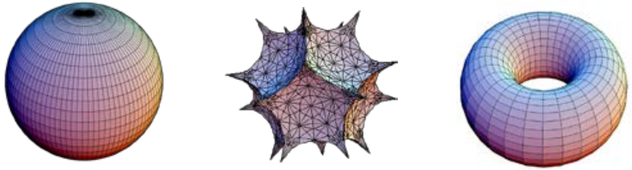


What is manifold optimization?

- Manifold (or manifold-constrained) optimization problem

$$\underset{M \in \mathbb{C}^{m \times n}}{\text{minimize}} \quad f(M) \quad \text{subject to} \quad M \in \mathcal{M}$$

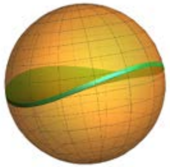
  - $f : \mathbb{R}^{m \times n} \to \mathbb{R}$ is a smooth function
  - $\mathcal{M}$ is a **Riemannian manifold:** spheres, orthonormal bases (Stiefel), rotations, positive definite matrices, ***fixed-rank matrices***, Euclidean distance matrices, semidefinite fixed-rank matrices, linear subspaces (Grassmann), phases, essential matrices, fixed-rank tensors, Euclidean spaces...



Convergence results of manifold optimization

- Convergence guarantees for Riemannian **trust regions**
  - Global convergence to **second-order critical points**
  - Quadratic convergence rate locally
  - Reach ϵ-second order stationary point $\|\mathrm{grad} f(\boldsymbol{z})\| \leq \epsilon$ and $\nabla^2 f(\boldsymbol{z}) \succeq -\epsilon \boldsymbol{I}$ in $\mathcal{O}(1/\epsilon^3)$ iterations under Lipschitz assumptions [Cartis & Absil'16]

Escape *strict* saddle points via finding second-order stationary point



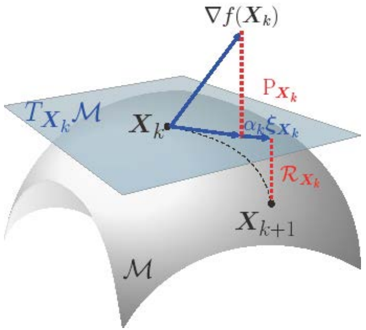
Recent applications of manifold optimization

- **High-dimensional data analysis:** matrix/tensor completion/recovery: [Vandereycken'13], [Boumal-Absil'15], [Kasai-Mishra'16]; phase retrieval: [Sun-Qu-Wright'17]; community detection: [Boumal'16], [Bandeira-Boumal-Voroninski'16],…
- **Machine and deep learning:** Gaussian mixture models: [Hosseini-Sra'15]; dictionary learning: [Sun-Qu-Wright'17]; deep metric learning: [Roy-Mhammedi-Harandi'18],…
- ***Wireless transceivers design:*** [Shi-Zhang-Letaief'16], [Yu-Shen-Zhang-K. B. Letaief'16], [Shi-Mishra-Chen'17],…

Exploit manifold geometry to address non-convex problems

The power of manifold optimization paradigms

- Generalize Euclidean gradient (Hessian) to *Riemannian gradient (Hessian)*



$$\nabla_{\mathcal{M}} f(\mathbf{X}^{(k)}) = P_{\mathbf{X}^{(k)}}(\nabla f(\mathbf{X}^{(k)}))$$

Riemannian Gradient   Euclidean Gradient

$$\mathbf{X}^{(k+1)} = \mathcal{R}_{\mathbf{X}^{(k)}}(-\alpha^{(k)} \nabla_{\mathcal{M}} f(\mathbf{X}^{(k)}))$$

Retraction Operator

- We need Riemannian geometry: 1) linearize search space $\mathcal{M}$ into a tangent space $T_{\mathbf{X}}\mathcal{M}$; 2) pick a metric on $T_{\mathbf{X}}\mathcal{M}$ to give intrinsic notions of gradient and Hessian

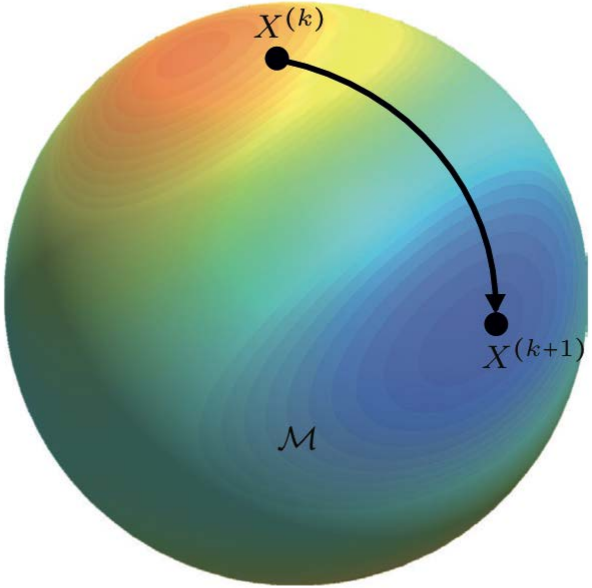
An excellent book

Optimization algorithms on matrix manifolds

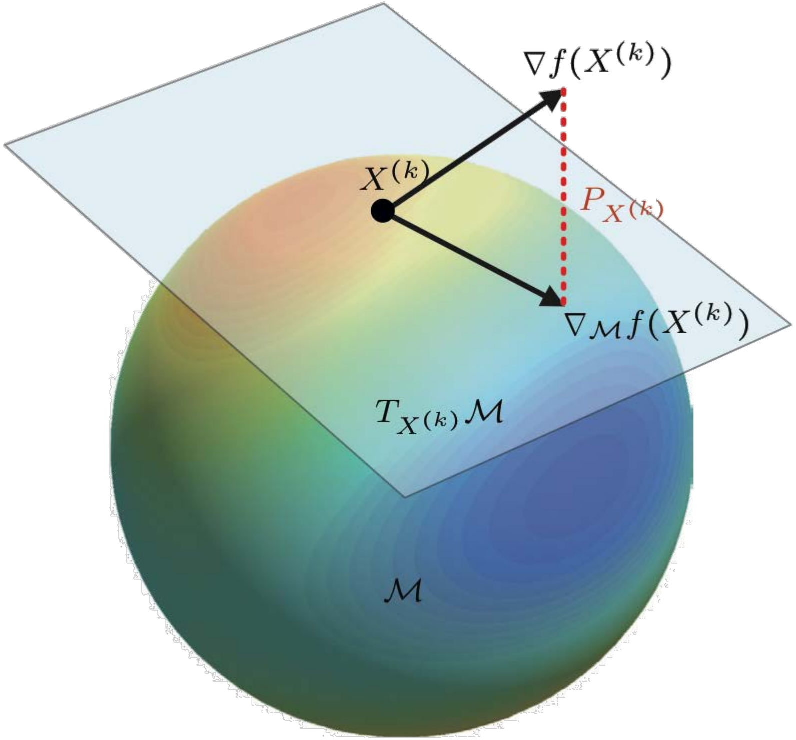
A Matlab toolbox

Taking a close look at ***gradient descent***

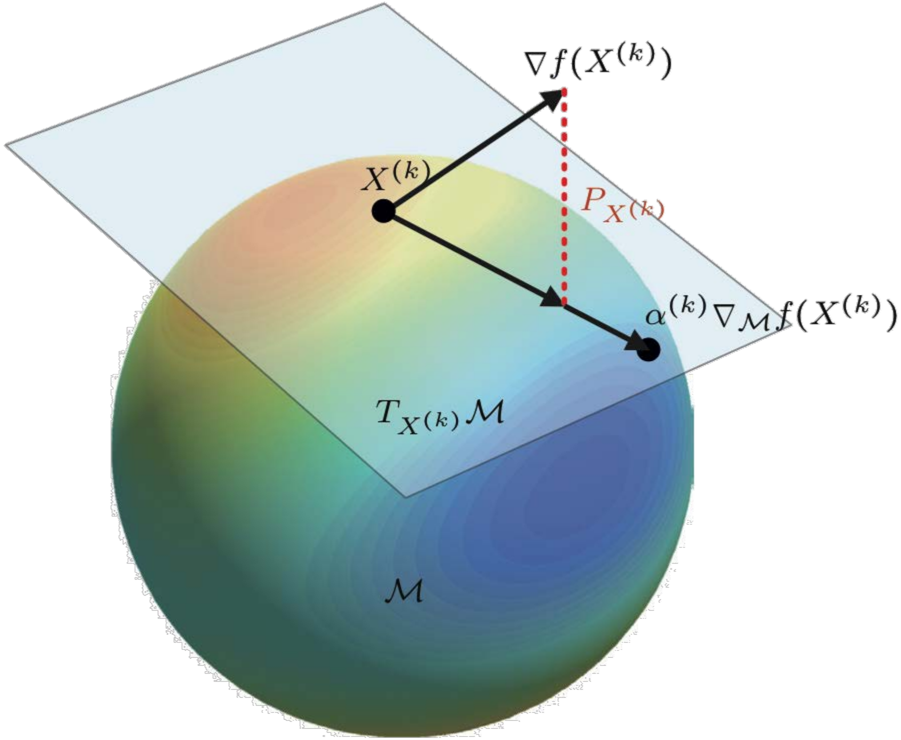
Optimization on the manifold: main idea



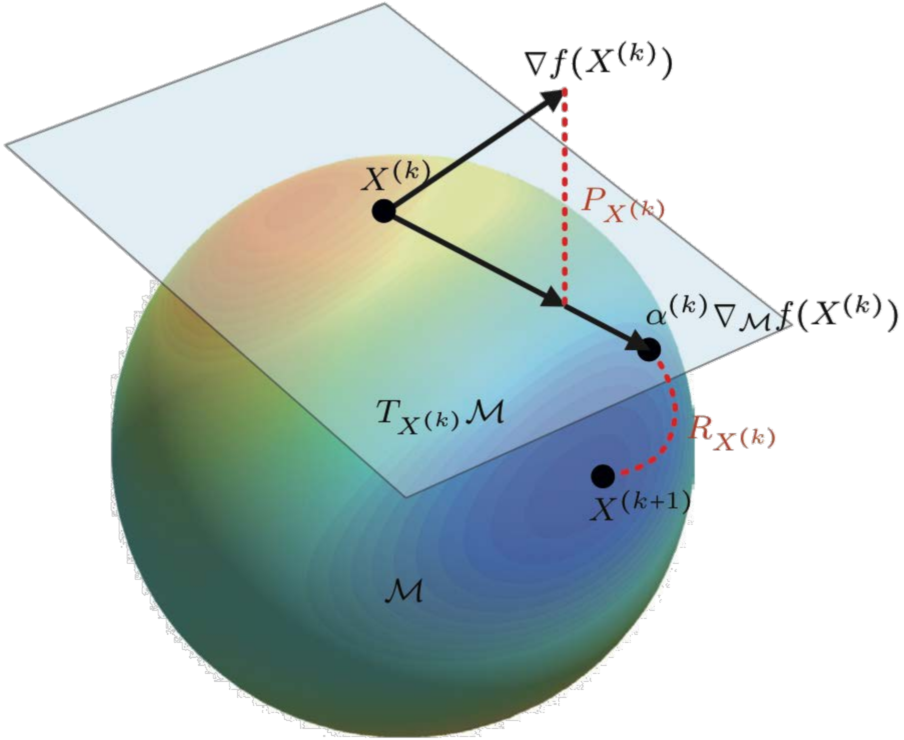
Optimization on the manifold: main idea



Optimization on the manifold: main idea



Optimization on the manifold: main idea



Example: Rayleigh quotient

- Optimization over (sphere) manifold $\mathbb{S}^{n-1} = \{x \in \mathbb{R}^n : x^T x = 1\}$

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \;\; f(x) = -x^T A x \quad \text{subject to} \;\; x^T x = 1$$

  - The cost function is smooth on $\mathbb{S}^{n-1}$, symmetric matrix $A \in \mathbb{R}^{n \times n}$

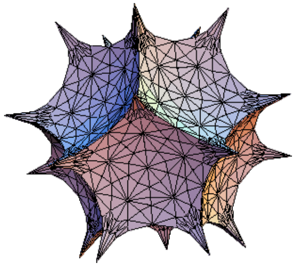
- Step 1: Compute the Euclidean gradient in $\mathbb{R}^n$

$$\nabla f(x) = -2Ax$$

- Step 2: Compute the Riemannian gradient on $\mathbb{S}^{n-1}$ via projecting $\nabla f(x)$ to the tangent space using the orthogonal projector $\text{Proj}_x u = (I - xx^T)u$

$$\text{grad} f(x) = \text{Proj}_x \nabla f(x) = -2(I - xx^T)Ax$$

Riemannian optimization for blind demixing



Blind demixing via low-rank optimization

- **Linear mapping:** from bilinear model to linear model

$$y_j = \sum_{i=1}^{s} \boldsymbol{b}_j^* \boldsymbol{h}_i^\natural \boldsymbol{x}_i^{\natural *} \boldsymbol{a}_{ij}, \quad 1 \le j \le m$$

  - ➢ $\boldsymbol{b}_j^* \boldsymbol{h}_i \boldsymbol{x}_i^* \boldsymbol{a}_{ij} = \langle \boldsymbol{b}_j \boldsymbol{a}_{ij}^*, \boldsymbol{h}_i \boldsymbol{x}_i^* \rangle$
  - ➢ $\mathcal{A}_i(\boldsymbol{X}_i) := \{\langle \boldsymbol{b}_j \boldsymbol{a}_{ij}^*, \boldsymbol{h}_i \boldsymbol{x}_i^* \rangle\}_{j=1}^{L} = \{\langle \boldsymbol{A}_{ij}, \boldsymbol{X}_k \rangle\}_{j=1}^{L}, \quad \boldsymbol{X}_i = \boldsymbol{h}_i \boldsymbol{x}_i^*$

- **Proposal:** (non-convex) low-rank optimization problem

$$\begin{aligned} \mathscr{P} : \underset{\boldsymbol{W}_k \in \mathbb{C}^{N \times K}}{\text{minimize}} \quad & \left\| \sum_{k=1}^{s} \mathcal{A}_k(\boldsymbol{W}_k) - \boldsymbol{y} \right\|^2 \\ \text{subject to} \quad & \operatorname{rank}(\boldsymbol{W}_k) = 1, \; k = 1, \cdots, s, \end{aligned}$$

  - ➢ Challenges: nonconvex constraints, complex asymmetric matrices

Blind demixing via Riemannian optimization

- Handle complex asymmetric matrices
  - Define linear map $\mathcal{J}_k : \mathbb{S}_+^{(N+K)\times(N+K)} \to \mathbb{C}^L$ as

$$[\mathcal{J}_k(\boldsymbol{Y}_k)]_i = \langle \boldsymbol{J}_{ki}, \boldsymbol{Y}_k \rangle,\ \boldsymbol{Y}_k \in \mathbb{S}_+^{(N+K)(N+K)} \qquad \boldsymbol{J}_{ki} = \begin{bmatrix} \boldsymbol{0}_{N\times N} & \boldsymbol{A}_{ki} \\ \boldsymbol{0}_{K\times N} & \boldsymbol{0}_{K\times K} \end{bmatrix}$$

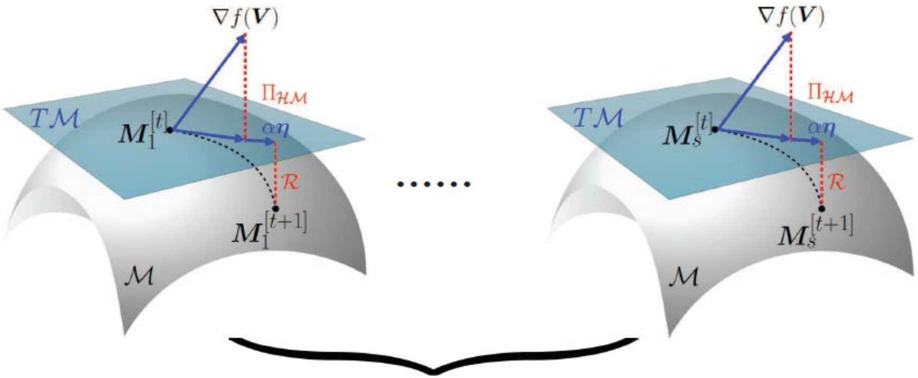
- Matrix optimization over the **product manifolds**

$$\begin{aligned} &\underset{\boldsymbol{M}_k \in \mathbb{S}_+^{(N+K)}}{\text{minimize}} && \left\| \sum\nolimits_{k=1}^{s} \mathcal{J}_k(\boldsymbol{M}_k) - \boldsymbol{y} \right\|^2 \\ &\text{subject to} && \operatorname{rank}(\boldsymbol{M}_k) = 1,\ k = 1, \cdots, s \end{aligned}$$

  - **Key observations:** rank-one Hermitian positive semidefinite matrices is a manifold; multiple rank-one constraints construct a manifold

Riemannian optimization over product manifolds

- **Elementwise extension principles**
  - The manifold topology of the product manifold is equivalent to the product topology



optimization over individual manifolds $\mathcal{M}$

Element-wise optimization-related ingredients

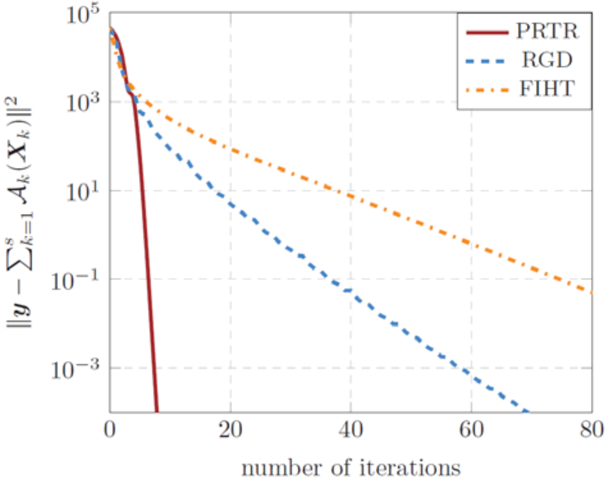
- Riemannian optimization for blind demixing

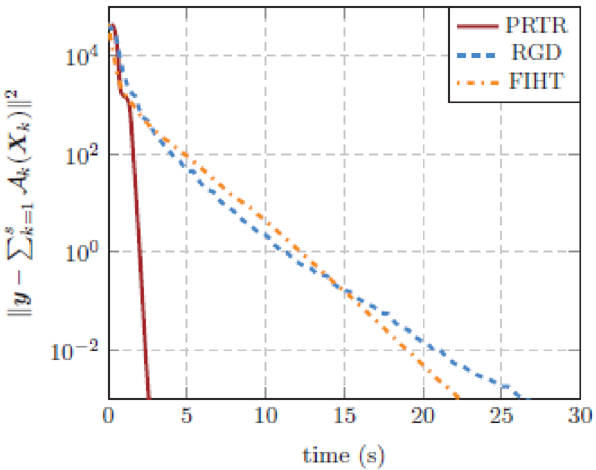
$$\begin{aligned} \underset{\boldsymbol{M}_k \in \mathbb{S}_+^{(N+K)}}{\text{minimize}} \quad & \left\| \sum_{k=1}^{s} \mathcal{J}_k(\boldsymbol{M}_k) - \boldsymbol{y} \right\|^2 \\ \text{subject to} \quad & \operatorname{rank}(\boldsymbol{M}_k) = 1, \; k = 1, \cdots, s \end{aligned}$$

	$\text{minimize}_{\boldsymbol{w}_k \in \mathcal{M}} \|\sum_{k=1}^{s} \mathcal{J}_k(\boldsymbol{w}_k \boldsymbol{w}_k^{\mathsf{H}}) - \boldsymbol{y}\|^2$
Computational space $\mathcal{M}$	$\mathbb{C}_*^{N+K}$
Quotient space $\mathcal{M}/\sim$	$\mathbb{C}_*^{N+K}/\mathrm{SU}(1)$
Riemannian metric $g_{\boldsymbol{w}_k}$	$g_{\boldsymbol{w}_k}(\boldsymbol{\zeta}_{\boldsymbol{w}_k}, \boldsymbol{\eta}_{\boldsymbol{w}_k}) = \mathrm{Tr}(\boldsymbol{\zeta}_{\boldsymbol{w}_k}^{\mathsf{H}} \boldsymbol{\eta}_{\boldsymbol{w}_k} + \boldsymbol{\eta}_{\boldsymbol{w}_k}^{\mathsf{H}} \boldsymbol{\zeta}_{\boldsymbol{w}_k})$
Horizontal space $\mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}$	$\boldsymbol{\eta}_{\boldsymbol{w}_k} \in \mathbb{C}^{N+K} : \boldsymbol{\eta}_{\boldsymbol{w}_k}^{\mathsf{H}} \boldsymbol{w}_k = \boldsymbol{w}_k^{\mathsf{H}} \boldsymbol{\eta}_{\boldsymbol{w}_k}$
Horizontal space projection	$\Pi_{\mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}}(\boldsymbol{\eta}_{\boldsymbol{w}_k}) = \boldsymbol{\eta}_{\boldsymbol{w}_k} - a\boldsymbol{w}_k,\ a = (\boldsymbol{w}_k^{\mathsf{H}} \boldsymbol{\eta}_{\boldsymbol{w}_k} - \boldsymbol{\eta}_{\boldsymbol{w}_k}^{\mathsf{H}} \boldsymbol{w}_k)/2\boldsymbol{w}^{\mathsf{H}}\boldsymbol{w}$
Riemannian gradient $\operatorname{grad}_{\boldsymbol{w}_k} f$	$\operatorname{grad}_{\boldsymbol{w}} f = \Pi_{\mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}}(\frac{1}{2}\nabla_{\boldsymbol{w}_k} f(\boldsymbol{v}))$
Riemannian Hessian $\operatorname{Hess}_{\boldsymbol{w}_k} f[\boldsymbol{\eta}_{\boldsymbol{w}_k}]$	$\operatorname{Hess}_{\boldsymbol{w}_k} f[\boldsymbol{\eta}_{\boldsymbol{w}_k}] = \Pi_{\mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}}(\frac{1}{2}\nabla_{\boldsymbol{w}_k}^2 f(\boldsymbol{v})[\boldsymbol{\eta}_{\boldsymbol{w}_k}])$
Retraction $\mathcal{R}_{\boldsymbol{w}_k} : T_{\boldsymbol{w}_k}\mathcal{M} \to \mathcal{M}$	$\mathcal{R}_{\boldsymbol{w}_k}(\boldsymbol{\eta}_{\boldsymbol{w}_k}) = \boldsymbol{w}_k + \boldsymbol{\eta}_{\boldsymbol{w}_k}$

Numerical results

- Optimize over the product of multiple rank-one Hermitian positive semidefinite matrices





Riemannian algorithms: 1) exploit the rank structure in a principled way; 2) develop second-order algorithms systematically; 3) scalable, SVD-free

Concluding remarks

- **Implicitly regularized Wirtinger flow**
  - **Implicit regularization:** vanilla gradient descent automatically forces iterates to stay *incoherent*
  - Even **simplest** nonconvex methods are remarkably efficient under suitable **statistical** models
- **Matrix optimization over manifolds**
  - Exploit the manifold geometry of multiple rank-one Hermitian positive semidefinite matrices
  - Develop second-order algorithms systematically: escape saddle points, quadratic convergence rate
- **Future works:** sparse blind demixing, convolutional dictionary learning [Wright, CVPR'17], convolutional neural network [Papyan, et al., SPM'18],…

Reference

- J. Dong and Y. Shi, "Nonconvex demixing from bilinear measurements," *IEEE Trans. Signal Process.*, vol. 66, no. 19, pp. 5152-5166, Oct., 2018.
- J. Dong, K. Yang, and Y. Shi, "Blind demixing for low-latency communication," *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 897-911, Feb., 2019.
- J. Dong, Y. Shi, and Z. Ding, "Blind over-the-air computation and data fusion via provable Wirtinger flow," https://arxiv.org/abs/1811.04644.
- J. Dong and Y. Shi, "Blind Demixing via Wirtinger Flow with Random initialization," in *Proc. Int. Conf. Artificial Intell. Stat. (AISTATS)*, 2019.

Thanks