A Series of Talks on Riemannian Optimization

Smooth Optimization: Influence of Metrics

Wen Huang

Xiamen University

September 16, 2025

Hunan University

Content

1. Geometric Mean of SPD Matrices
    - Motivations;
    - Averaging on a Riemannian manifold;
    - Algorithms and manifold geometry;
2. Signal Recovery on Low-rank Matrices
    - Motivations;
    - Problem formulations;
    - Algorithms and manifold geometry;
3. Rank Overestimation (Hermitian PSD low-rank Constraints);
    - Problem formulation;
    - Riemannian metrics;
    - Condition number for nearly low-rank solutions;

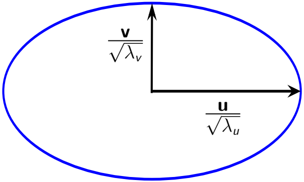
Symmetric Positive Definite (SPD) Matrix

Definition

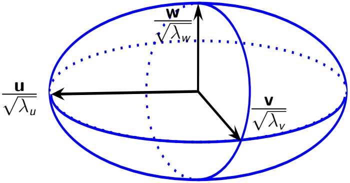
A symmetric matrix A is called positive definite $A \succ 0$ iff all its eigenvalues are positive.

$$\mathcal{S}_{++}^{\mathrm{n}} = \{A \in \mathbb{R}^{n \times n} : A = A^T, A \succ 0\}$$

2×2 SPD matrix

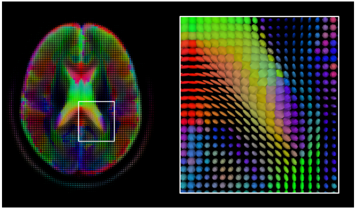


3×3 SPD matrix



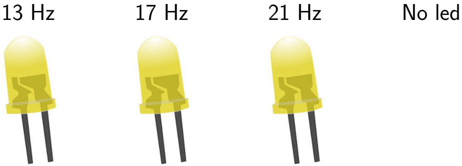
Motivation of Averaging SPD Matrices

- Possible applications of SPD matrices
  - Diffusion tensors in medical imaging [CSV12, FJ07, RTM07]
  - Describing images and video [LWM13, SFD02, ASF^{+}05, TPM06, HWSC15]
- Motivation of averaging SPD matrices
  - denoising / interpolation
  - clustering / classification



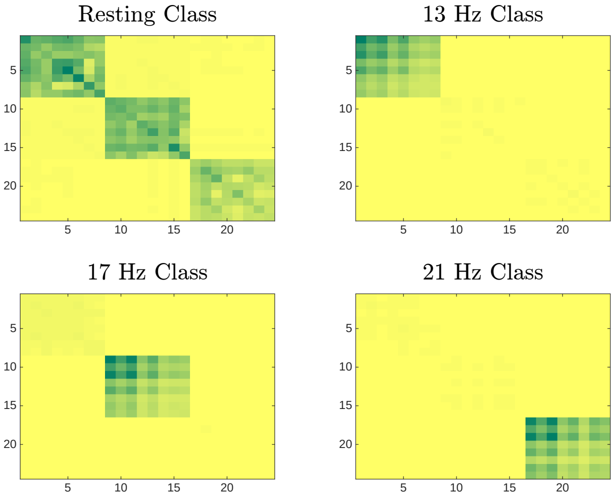
Motivation of Averaging SPD Matrices

Application: Electroencephalography (EEG) Classification



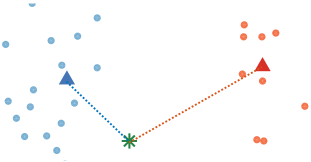
- The subject is either asked to focus on one specific blinking LED or a location without LED
- EEG system is used to record brain signals
- Covariance matrices of size 24 × 24 are used to represent EEG recordings [KCB$^+$15, MC17]
- Covariance matrices in $\mathcal{S}_{++}^{\mathrm{n}} = \{A \in \mathbb{R}^{n \times n} : A = A^T, A \succ 0\}$

EEG Classification: Examples of Covariance Matrices



EEG Classification: Minimum Distance to Mean classier

Goal: classify new covariance matrix using Minimum Distance to Mean Classifier



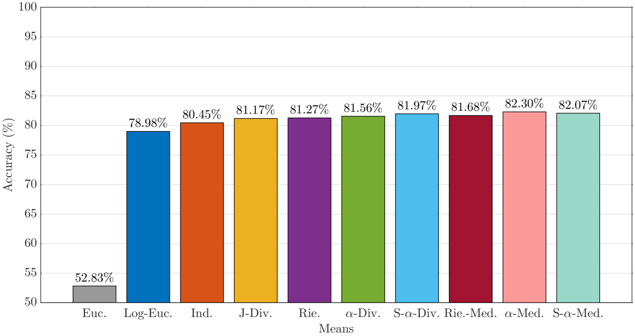
- For each class $k = 1, \ldots, K$, compute the center μ_k of the covariance matrices in the training set that belong to class k

- Classify a new covariance matrix X according to

$$\hat{k} = \operatorname*{argmin}_{1 \leq k \leq K} \delta(X, \mu_k)$$

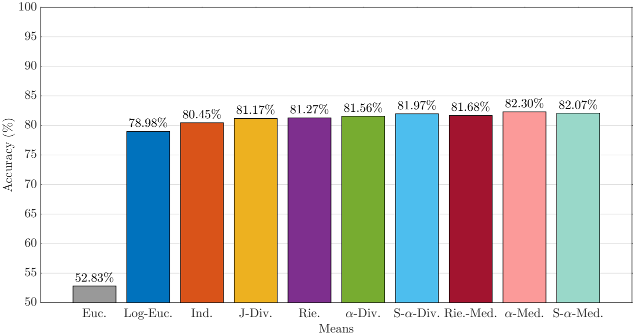
EEG Classfification: Accuracy

- Accuracy comparison



EEG Classfification: Accuracy

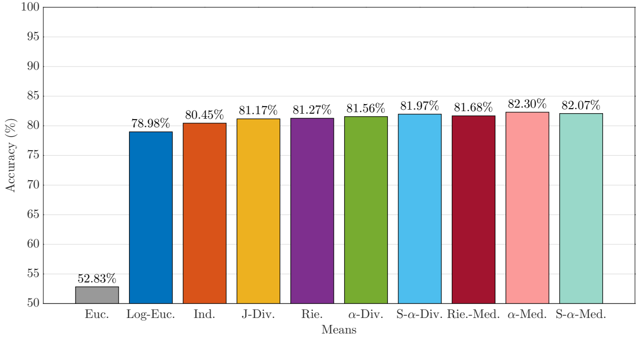
- Accuracy comparison



Euclidean metric is not appropriate to define the problem!

EEG Classfification: Accuracy

- Accuracy comparison



Euclidean metric is not appropriate to define the problem!

Is Euclidean metric appropriate for optimization? Averaging SPD matrices.

Averaging Schemes: from Scalars to Matrices

Let $A_1, \ldots, A_K$ be SPD matrices.

- Generalized arithmetic mean: $\frac{1}{K} \sum_{i=1}^{K} A_i$

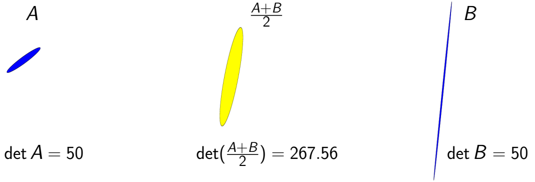
  → Not appropriate in many practical applications

Averaging Schemes: from Scalars to Matrices

Let $A_1, \dots, A_K$ be SPD matrices.

- Generalized arithmetic mean: $\frac{1}{K}\sum_{i=1}^{K} A_i$

  → Not appropriate in many practical applications

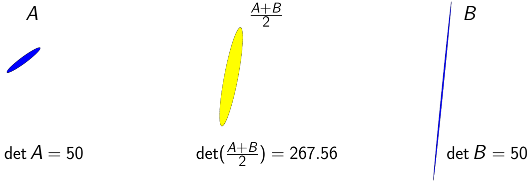


Averaging Schemes: from Scalars to Matrices

Let $A_1, \dots, A_K$ be SPD matrices.

- Generalized arithmetic mean: $\frac{1}{K} \sum_{i=1}^{K} A_i$

  $\rightarrow$ Not appropriate in many practical applications



- Generalized geometric mean: $(A_1 \cdots A_K)^{1/K}$

  $\rightarrow$ Not appropriate due to non-commutativity

  $\rightarrow$ How to define a matrix geometric mean?

Desired Properties of a Matrix Geometric Mean

The desired properties are given in the ALM list[1], some of which are:

- $G(A_{\pi(1)}, \ldots, A_{\pi(K)}) = G(A_1, \ldots, A_K)$ with π a permutation of $(1, \ldots, K)$
- if $A_1, \ldots, A_K$ commute, then $G(A_1, \ldots, A_K) = (A_1, \ldots, A_K)^{1/K}$
- $G(A_1, \ldots, A_K)^{-1} = G(A_1^{-1}, \ldots, A_K^{-1})$
- $\det(G(A_1, \ldots, A_K)) = (\det(A_1) \cdots \det(A_K))^{1/K}$

---

[1] T. Ando, C.-K. Li, and R. Mathias, *Geometric means*, Linear Algebra and Its Applications, 385:305-334, 2004

Geometric Mean of SPD Matrices

- A well-known mean on the manifold of SPD matrices is the **Karcher mean** [Kar77]:

$$G(A_1, \ldots, A_K) = \arg \min_{X \in \mathcal{S}_{++}^n} \frac{1}{2K} \sum_{i=1}^{K} \delta^2(X, A_i), \tag{1}$$

where $\delta(X, Y) = \| \log(X^{-1/2} Y X^{-1/2}) \|_F$ is the geodesic distance under the affine-invariant metric

$$g(\eta_X, \xi_X) = \mathrm{trace}(\eta_X X^{-1} \xi_X X^{-1})$$

- The Karcher mean defined in (1) satisfies all the geometric properties in the ALM list [LL11]

Geometric Mean of SPD Matrices

Optimization problem:

$$G(A_1, \ldots, A_K) = \arg \min_{X \in \mathcal{S}_{++}^n} \frac{1}{2K} \sum_{i=1}^{K} \| \log(X^{-1/2} Y X^{-1/2}) \|_F^2,$$

- Derived from Riemannian manifold;
- An optimization problem on an open set (cone);
- What algorithms are preferred?

Algorithms

$$G(A_1, \ldots, A_k) = \operatorname*{argmin}_{X \in \mathcal{S}_{++}^n} \frac{1}{2K} \sum_{i=1}^{K} \| \log(X^{-1/2} Y X^{-1/2}) \|_F^2,$$

Existing algorithms:

- Riemannian steepest descent [RA11, Ren13]
- Riemannian Barzilai-Borwein method [IP15]
- Riemannian Newton method [RA11]
- Richardson-like iteration [BI13]
- Riemannian steepest descent, conjugate gradient, BFGS, and trust region Newton methods [JVV12]
- Limited-memory Riemannian BFGS method [YHAG19]

Algorithms

$$G(A_1, \dots, A_k) = \operatorname*{argmin}_{X \in \mathcal{S}^n_{++}} \frac{1}{2K} \sum_{i=1}^{K} \| \log(X^{-1/2} Y X^{-1/2}) \|_F^2,$$

Existing algorithms:

- Riemannian steepest descent [RA11, Ren13]
- Riemannian Barzilai-Borwein method [IP15]
- Riemannian Newton method [RA11]
- Richardson-like iteration [BI13]
- Riemannian steepest descent, conjugate gradient, BFGS, and trust region Newton methods [JVV12]
- Limited-memory Riemannian BFGS method [YHAG19]

Riemannian gradient is used in all the above methods!

Algorithms

$$G(A_1, \dots, A_k) = \operatorname*{argmin}_{X \in \mathcal{S}_{++}^n} \frac{1}{2K} \sum_{i=1}^{K} \| \log(X^{-1/2} Y X^{-1/2}) \|_F^2,$$

Existing algorithms:

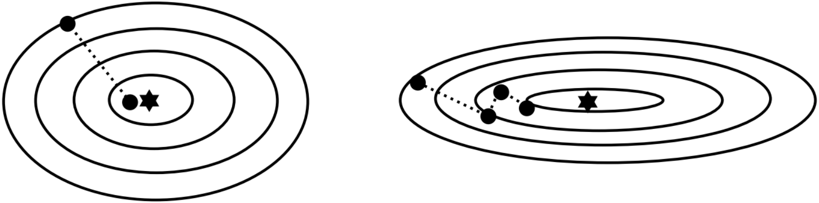
- Riemannian steepest descent [RA11, Ren13]
- Riemannian Barzilai-Borwein method [IP15]
- Riemannian Newton method [RA11]
- Richardson-like iteration [BI13]
- Riemannian steepest descent, conjugate gradient, BFGS, and trust region Newton methods [JVV12]
- Limited-memory Riemannian BFGS method [YHAG19]

Riemannian gradient is used in all the above methods!

The LRBFGS in [YHAG19] is preferred.

Conditioning of the Objective Function

Hemstitching phenomenon
for steepest descent



well-conditioned Hessian

ill-conditioned Hessian

- Small condition number ⇒ fast convergence
- Large condition number ⇒ slow convergence

Conditioning of the Karcher Mean Objective Function

- **Riemannian metric**:

$$g_X(\xi, \eta) = \mathrm{trace}(\xi X^{-1} \eta X^{-1})$$

- **Euclidean metric**:

$$g_X(\xi, \eta) = \mathrm{trace}(\xi \eta)$$

Condition number κ of Hessian at the minimizer μ:

- Hessian of Riemannian metric:
  - $\kappa(H^R) \leq 1 + \frac{\ln(\max \kappa_i)}{2}$, where $\kappa_i = \kappa(\mu^{-1/2} A_i \mu^{-1/2})$
  - $\kappa(H^R) \leq 20$ if $\max(\kappa_i) = 10^{16}$
- Hessian of Euclidean metric:
  - $\frac{\kappa^2(\mu)}{\kappa(H^{\mathrm{R}})} \leq \kappa(H^{\mathrm{E}}) \leq \kappa(H^{\mathrm{R}}) \kappa^2(\mu)$
  - $\kappa(H^E) \geq \kappa^2(\mu)/20$

Smooth Optimization Framework

Riemannian Metric

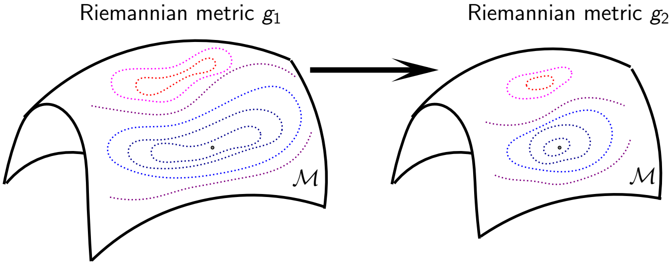


Figure: Changing metric may influence the difficulty of a problem.

Riemannian metric influences

- Riemannian gradient
- Riemannian Hessian

BFGS: from Euclidean to Riemannian

- Update formula:

$$x_{k+1} = \underline{x_k + \alpha_k \eta_k}$$

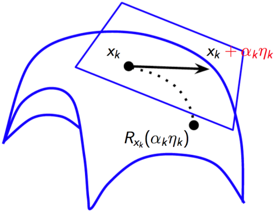
- Search direction:

$$\eta_k = -B_k^{-1} \operatorname{grad} f(x_k)$$

- B_k update:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k},$$

where $s_k = \underline{x_{k+1} - x_k}$, and $y_k = \underline{\operatorname{grad} f(x_{k+1}) - \operatorname{grad} f(x_k)}$



Optimization on a Manifold

BFGS: from Euclidean to Riemannian

replace by $R_{x_k}(\eta_k)$

↓

- Update formula:

$$x_{k+1} = \underline{x_k + \alpha_k \eta_k}$$

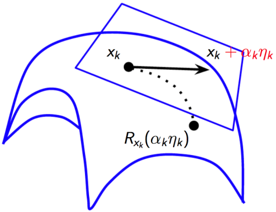
- Search direction:

$$\eta_k = -B_k^{-1} \operatorname{grad} f(x_k)$$

- B_k update:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k},$$

where $s_k = \underline{x_{k+1} - x_k}$, and $y_k = \underline{\operatorname{grad} f(x_{k+1}) - \operatorname{grad} f(x_k)}$



Optimization on a Manifold

BFGS: from Euclidean to Riemannian

replace by $R_{x_k}(\eta_k)$

$\downarrow$

- Update formula:

$$x_{k+1} = \underline{x_k + \alpha_k \eta_k}$$

- Search direction:

$$\eta_k = -B_k^{-1} \operatorname{grad} f(x_k)$$

- B_k update:

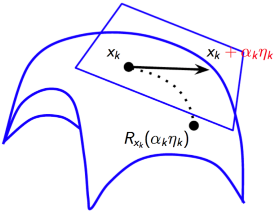
$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k},$$

where $s_k = \underline{x_{k+1} - x_k}$, and $y_k = \underline{\operatorname{grad} f(x_{k+1}) - \operatorname{grad} f(x_k)}$

$\uparrow$ $\uparrow$

replaced by $R_{x_k}^{-1}(x_{k+1})$

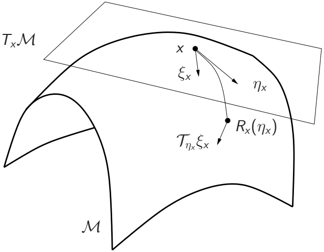
on different tangent spaces



Optimization on a Manifold

BFGS: from Euclidean to Riemannian

A vector transport: $\mathcal{T} : \mathrm{T}\,\mathcal{M} \times \mathrm{T}\,\mathcal{M} \to \mathrm{T}\,\mathcal{M} : (\eta_x, \xi_x) \mapsto \mathcal{T}_{\eta_x}\xi_x$:



- Euclidean: $y_k = \underline{\mathrm{grad}\, f(x_{k+1}) - \mathrm{grad}\, f(x_k)}$
- Riemannian: $y_k = \underline{\mathrm{grad}\, f(x_{k+1}) - \mathcal{T}_{\alpha_k\eta_k}\, \mathrm{grad}\, f(x_k)}$

BFGS: from Euclidean to Riemannian

- Update formula:

$$x_{k+1} = \underline{R_{x_k}(\alpha_k \eta_k)}$$

- Search direction:

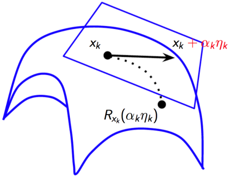
$$\eta_k = -B_k^{-1} \operatorname{grad} f(x_k)$$

- B_k update:

$$\tilde{B}_k = \underline{\mathcal{T}_{\alpha_k \eta_k} \circ B_k \circ \mathcal{T}_{\alpha_k \eta_k}^{-1}},$$

$$B_{k+1} = \underline{\tilde{B}_k - \frac{\tilde{B}_k s_k s_k^{\flat} \tilde{B}_k}{s_k^{\flat} \tilde{B}_k s_k} + \frac{y_k y_k^{\flat}}{y_k^{\flat} s_k}},$$

where $s_k = \underline{\mathcal{T}_{\alpha_k \eta_k}(\alpha_k \eta_k)}$, and $y_k = \underline{\operatorname{grad} f(x_{k+1}) - \mathcal{T}_{\alpha_k \eta_k} \operatorname{grad} f(x_k)}$;



Optimization on a Manifold

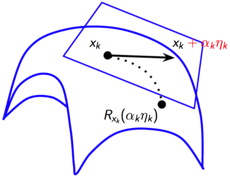
BFGS: from Euclidean to Riemannian

- Update formula:

$$x_{k+1} = \underline{R_{x_k}(\alpha_k \eta_k)}$$

- Search direction:

$$\eta_k = -B_k^{-1} \operatorname{grad} f(x_k)$$



Optimization on a Manifold

- B_k update:

$$\tilde{B}_k = \underline{\mathcal{T}_{\alpha_k\eta_k} \circ B_k \circ \mathcal{T}_{\alpha_k\eta_k}^{-1}}, \leftarrow$$

matrix matrix multiplication

$$B_{k+1} = \underline{\tilde{B}_k - \frac{\tilde{B}_k s_k s_k^\flat \tilde{B}_k}{s_k^\flat \tilde{B}_k s_k} + \frac{y_k y_k^\flat}{y_k^\flat s_k}},$$

where $s_k = \underline{\mathcal{T}_{\alpha_k\eta_k}(\alpha_k\eta_k)}$, and $y_k = \underline{\operatorname{grad} f(x_{k+1}) - \mathcal{T}_{\alpha_k\eta_k} \operatorname{grad} f(x_k)}$;

$\uparrow$ matrix vector multiplication

$\uparrow$ matrix vector multiplication

Extra cost on vector transports!

Limited-memory RBFGS (LRBFGS)

Riemannian BFGS:

- Let $\mathcal{H}_{k+1} = \mathcal{B}_{k+1}^{-1}$
- $\mathcal{H}_{k+1} = (\mathrm{id} - \rho_k y_k s_k^\flat)\tilde{\mathcal{H}}_k(\mathrm{id} - \rho_k y_k s_k^\flat) + \rho_k s_k s_k^\flat$
  where $s_k = \mathcal{T}_{\alpha_k \eta_k} \alpha_k \eta_k$, $y_k = \operatorname{grad} f(x_{k+1}) - \mathcal{T}_{\alpha_k \eta_k} \operatorname{grad} f(x_k)$,
  $\rho_k = 1/g(y_k, s_k)$ and $\tilde{\mathcal{H}}_k = \mathcal{T}_{\alpha_k \eta_k} \circ \mathcal{H}_k \circ \mathcal{T}_{\alpha_k \eta_k}^{-1}$

Limited-memory Riemannian BFGS:

- Stores only the m most recent s_k and y_k
- Transports these vectors to the new tangent space rather than $\mathcal{H}_k$
- Computational and storage complexity depends upon m

Implementations

- Retraction
  - Exponential mapping: $\mathrm{Exp}_X(\xi) = X^{1/2} \exp(X^{-1/2} \xi X^{-1/2}) X^{1/2}$
  - Second order approximation retraction [JVV12]:
    $$R_X(\xi) = X + \xi + \frac{1}{2} \xi X^{-1} \xi = \tfrac{1}{2}(\xi X^{-1/2} + X^{1/2})(\xi X^{-1/2} + X^{1/2})^T + \tfrac{1}{2} X$$
- Vector transport
  - Parallel translation: $\mathcal{T}_{p_\eta}(\xi) = Q \xi Q^T$, with $Q = X^{\frac{1}{2}} \exp(\frac{X^{-\frac{1}{2}} \eta X^{-\frac{1}{2}}}{2}) X^{-\frac{1}{2}}$
  - Vector transport by parallelization [HAG17] : essentially an identity

Implementation

Vector Transport by Parallelization

- Vector transport by parallelization:

$$\mathcal{T}_{\eta_x}\xi_x = B_y B_x^{\dagger}\xi_x;$$

where $y = R_x(\eta_x)$ and † denotes pseudo-inverse, has identity implementation [HAG17]:

$$\mathcal{T}_{\tilde{\eta}_x}\tilde{\xi}_x = \tilde{\xi}_x.$$

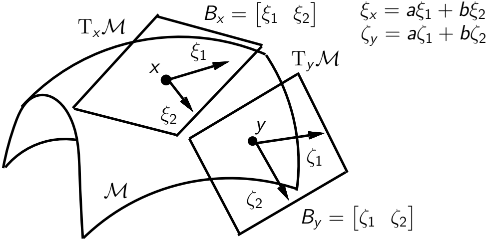
Example:

Extrinsic:

$$\zeta = \mathcal{T}_\eta \xi = B_y B_x^{\dagger}\xi$$

Intrinsic:

$$\begin{aligned}\tilde{\zeta} =& \widetilde{\mathcal{T}_\eta \xi}\\ =& B_y^{\dagger} B_y B_x^{\dagger} B_x \tilde{\xi}\\ =& \tilde{\xi}\end{aligned}$$



Implementations

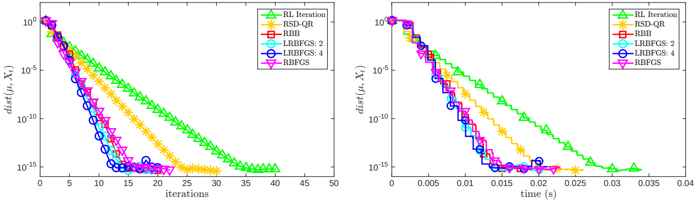
- Cholesky $X_k = L_k L_k^T$ assumed to be computed on each step
- B_X of $\mathrm{T}_X\,\mathcal{S}_{++}^n$, the orthonormal basis of $\mathrm{T}_X\,\mathcal{S}_{++}^n$

  $$B_X = \{L e_i e_i^T L^T : i = 1, \dots, n\} \cup \{\frac{1}{\sqrt{2}} L(e_i e_j^T + e_j e_i^T) L^T, \\ i < j,\ i = 1, \dots, n,\ j = 1, \dots, n\},$$

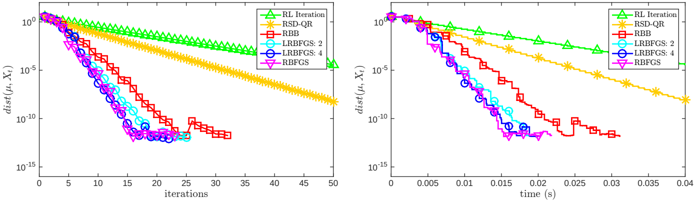
  where $\{e_i, \dots, e_n\}$ is the standard basis of n-dimensional Euclidean space.
- orthonormal under $g_X(\xi_X, \eta_X)$.
- $\xi_X = B_X \hat{\xi}_X \leftrightarrow \xi_X = LSL^T$, where S is symmetric and constains scale coefficients.
- intrinsic representation of tangent vectors is easily maintained.

Numerical Results: $K = 100$, size $= 3 \times 3$, $d = 6$

- $1 \leq \kappa(A_i) \leq 200$

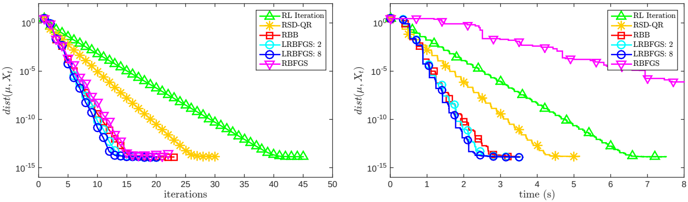


- $10^3 \leq \kappa(A_i) \leq 10^7$

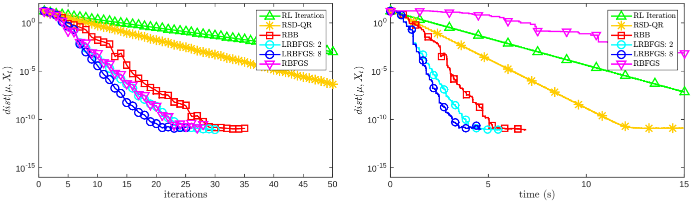


Numerical Results: $K = 30$, size $= 100 \times 100$, $d = 5050$

- $1 \leq \kappa(A_i) \leq 20$

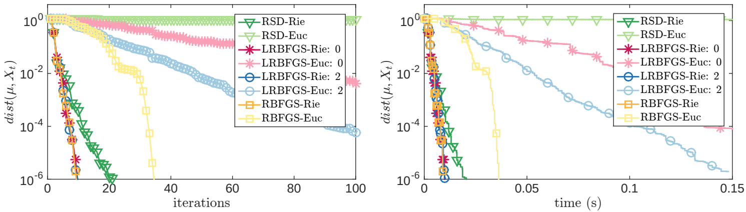


- $10^4 \leq \kappa(A_i) \leq 10^7$

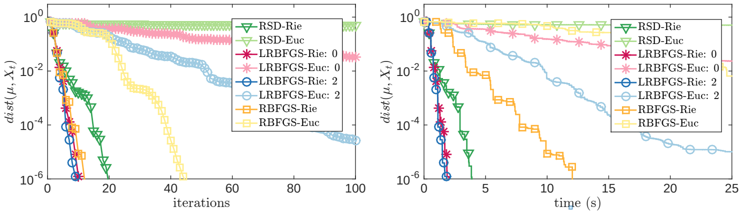


Numerical Results: Riemannian vs. Euclidean Metrics

- $K = 100$, $n = 3$, and $1 \leq \kappa(A_i) \leq 10^6$.



- $K = 30$, $n = 100$, and $1 \leq \kappa(A_i) \leq 10^5$.



Summary of SPD Mean

Non-Euclidean metric helps!

- Covariance matrices classification
- A geometric mean of SPD matrices
- Conditioner number of the Hessian
- Limited-memory Riemannian BFGS
- Numerical experiments

X. Yuan, W. Huang, P.-A. Absil, K. A. Gallivan. Computing the matrix geometric mean: Riemannian vs Euclidean conditioning, implementation techniques, and a Riemannian BFGS method, Numerical Linear Algebra with Applications, 27:5, 1-23, 2020

Discussions

About non-Euclidean metric

Questions: Riemannian algorithms versus preconditioned algorithms

---

A special case that may cause confusion:

Riemannian SD:

- Open submanifold of $\mathcal{L}$;
- Metric: $\langle u, v \rangle_x = u^T G_x v$
- Riemannian gradient:
  $$\mathrm{grad}\, f(x) = G_x^{-1} \nabla f(x);$$
- Riemannian SD:
  $$\begin{aligned} x_{k+1} &= R_{x_k}(-\alpha_k \,\mathrm{grad}\, f(x_k)) \\ &= R_{x_k}(-\alpha_k G_{x_k}^{-1} \nabla f(x_k)); \end{aligned}$$

Preconditioned SD:

- Metric: $\langle u, v \rangle_x = u^T v$
- Eucldean gradient $\nabla f(x)$;
- Preconditioner $P_x \approx \nabla^2 f(x)$;
- Preconditioned SD:
  $$x_{k+1} = x_k - \alpha_k P_{x_k}^{-1} \nabla f(x_k);$$

Same updates

Discussions

About non-Euclidean metric

Questions: Riemannian algorithms versus preconditioned algorithms

---

Differences:

- Very special case (open submanifold of $\mathcal{L}$);
- Retraction preferences;
- Riemannian conjugate gradient, Newton, quasi-Newton, preconditioned method, etc;

Discussions

Open submanifold of $\mathcal{L}$

Manifold has nice property, the metric is used for the landscape of the objective function;

Nonlinear manifold

The objective function has nice property and the metric is used for the nonlinearity of the manifold;

Discussions

Open submanifold of $\mathcal{L}$

Manifold has nice property, the metric is used for the landscape of the objective function;

Nonlinear manifold

The objective function has nice property and the metric is used for the nonlinearity of the manifold;

Example: signal recovery problems

Content

1. Geometric Mean of SPD Matrices
    - Motivations;
    - Averaging on a Riemannian manifold;
    - Algorithms and manifold geometry;
2. Signal Recovery on Low-rank Matrices
    - Motivations;
    - Problem formulations;
    - Algorithms and manifold geometry;
3. Rank Overestimation (Hermitian PSD low-rank Constraints);
    - Problem formulation;
    - Riemannian metrics;
    - Condition number for nearly low-rank solutions;

Signal Recovery on Low-rank Matrices

Observation y is a linear transformation of unknown signal x up to a noise, i.e., $y = \mathcal{A}(x) + e$;

- Matrix completion;
- Phase retrieval (The phase is also unknown);
- Blind deconvolution;
- etc;

Signal Recovery on Low-rank Matrices

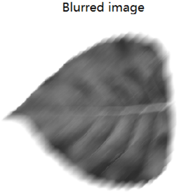
Observation y is a linear transformation of unknown signal x up to a noise, i.e., $y = \mathcal{A}(x) + e$;

- Matrix completion;
- Phase retrieval (The phase is also unknown);
- Blind deconvolution;
- etc;

Blind deconvolution

[Blind deconvolution]

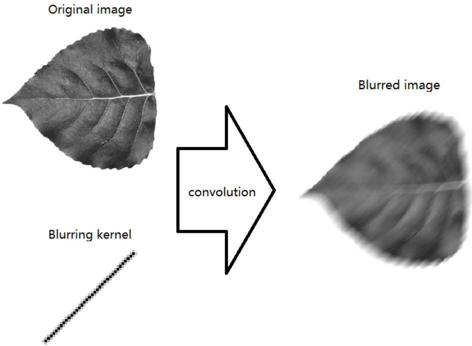
Blind deconvolution is to recover two unknown signals from their convolution.



Blind deconvolution

[Blind deconvolution]

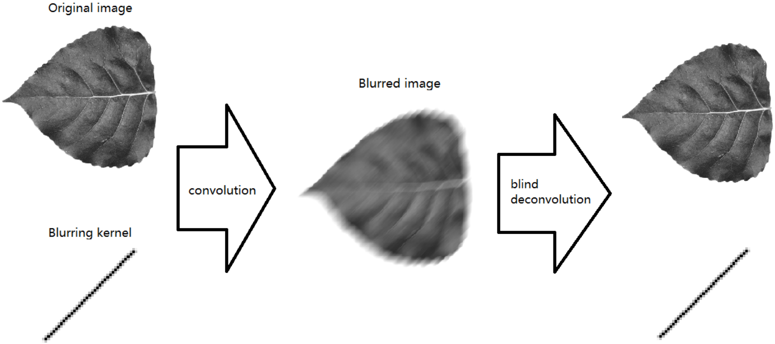
Blind deconvolution is to recover two unknown signals from their convolution.



Blind deconvolution

[Blind deconvolution]

Blind deconvolution is to recover two unknown signals from their convolution.



Problem Statement

[Blind deconvolution (Discretized version)]

Blind deconvolution is to recover two unknown signals $\mathbf{w} \in \mathbb{C}^L$ and $\mathbf{x} \in \mathbb{C}^L$ from their convolution $\mathbf{y} = \mathbf{w} * \mathbf{x} \in \mathbb{C}^L$.

- We only consider circular convolution:

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \\ \vdots \\ \mathbf{y}_L \end{bmatrix} = \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_L & \mathbf{w}_{L-1} & \dots & \mathbf{w}_2 \\ \mathbf{w}_2 & \mathbf{w}_1 & \mathbf{w}_L & \dots & \mathbf{w}_3 \\ \mathbf{w}_3 & \mathbf{w}_2 & \mathbf{w}_1 & \dots & \mathbf{w}_4 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{w}_L & \mathbf{w}_{L-1} & \mathbf{w}_{L-2} & \dots & \mathbf{w}_1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \vdots \\ \mathbf{x}_L \end{bmatrix}$$

- Let $y = \mathbf{Fy}$, $w = \mathbf{Fw}$, and $x = \mathbf{Fx}$, where $\mathbf{F}$ is the DFT matrix;
- $y = w \odot x$, where $\odot$ is the Hadamard product, i.e., $y_i = w_i x_i$.
- Equivalent question: Given y, find w and x.

Problem Statement

Problem: Given $y \in \mathbb{C}^L$, find $w, x \in \mathbb{C}^L$ so that $y = w \odot x$.

- An ill-posed problem. Infinite solutions exist;

Problem Statement

Problem: Given $y \in \mathbb{C}^L$, find $w, x \in \mathbb{C}^L$ so that $y = w \odot x$.

- An ill-posed problem. Infinite solutions exist;
- Assumption: w and x are in known subspaces, i.e., $w = Bh$ and $x = \overline{Cm}$, $B \in \mathbb{C}^{L \times K}$ and $C \in \mathbb{C}^{L \times N}$;

Problem Statement

Problem: Given $y \in \mathbb{C}^L$, find $w, x \in \mathbb{C}^L$ so that $y = w \odot x$.

- An ill-posed problem. Infinite solutions exist;
- Assumption: w and x are in known subspaces, i.e., $w = Bh$ and $x = \overline{Cm}$, $B \in \mathbb{C}^{L \times K}$ and $C \in \mathbb{C}^{L \times N}$;
  - Reasonable in various applications;
  - Leads to mathematical rigor; $(L/(K + N)$ reasonably large)

Problem Statement

Problem: Given $y \in \mathbb{C}^L$, find $w, x \in \mathbb{C}^L$ so that $y = w \odot x$.

- An ill-posed problem. Infinite solutions exist;
- Assumption: w and x are in known subspaces, i.e., $w = Bh$ and $x = \overline{Cm}$, $B \in \mathbb{C}^{L \times K}$ and $C \in \mathbb{C}^{L \times N}$;
  - Reasonable in various applications;
  - Leads to mathematical rigor; $(L/(K+N)$ reasonably large)

Problem under the assumption

Given $y \in \mathbb{C}^L$, $B \in \mathbb{C}^{L \times K}$ and $C \in \mathbb{C}^{L \times N}$, find $h \in \mathbb{C}^K$ and $m \in \mathbb{C}^N$ so that

$$y = Bh \odot \overline{Cm} = \mathrm{diag}(Bhm^* C^*).$$

Related work

$$\boxed{\text{Find } h, m, \text{ s. t. } y = \mathrm{diag}(Bhm^* C^*);}$$

- Ahmed et al. [ARR14][2]
    - Convex problem:

$$\min_{X \in \mathbb{C}^{K \times N}} \|X\|_n, \text{ s. t. } y = \mathrm{diag}(BXC^*),$$

where $\| \cdot \|_n$ denotes the nuclear norm, and $X = hm^*$;

---

[2] A. Ahmed, B. Recht, and J. Romberg, Blind deconvolution using convex programming, *IEEE Transactions on Information Theory*, 60:1711-1732, 2014

Related work

$$\boxed{\text{Find } h, m, \text{ s. t. } y = \mathrm{diag}(Bhm^* C^*);}$$

- Ahmed et al. [ARR14][2]
  - Convex problem:

$$\min_{X \in \mathbb{C}^{K \times N}} \|X\|_n, \text{ s. t. } y = \mathrm{diag}(BXC^*),$$

where $\| \cdot \|_n$ denotes the nuclear norm, and $X = hm^*$;

  - (Theoretical result): the unique minimizer $\xlongequal{\text{high probability}}$ the true solution;

---

[2] A. Ahmed, B. Recht, and J. Romberg, Blind deconvolution using convex programming, *IEEE Transactions on Information Theory*, 60:1711-1732, 2014

Related work

$$\boxed{\text{Find } h, m, \text{ s. t. } y = \mathrm{diag}(Bhm^* C^*);}$$

- Ahmed et al. [ARR14][2]
  - Convex problem:

    $$\min_{X \in \mathbb{C}^{K \times N}} \|X\|_n, \text{ s. t. } y = \mathrm{diag}(BXC^*),$$

    where $\| \cdot \|_n$ denotes the nuclear norm, and $X = hm^*$;
  - (Theoretical result): the unique minimizer $\overset{\text{high probability}}{=\!=\!=\!=\!=}$ the true solution;
  - The convex problem is expensive to solve;

---

[2] A. Ahmed, B. Recht, and J. Romberg, Blind deconvolution using convex programming, *IEEE Transactions on Information Theory*, 60:1711-1732, 2014

Related work

$$\boxed{\text{Find } h, m, \text{ s. t. } y = \mathrm{diag}(Bhm^* C^*);}$$

- Li et al. [LLSW18][3]
    - Nonconvex problem[4]:

$$\min_{(h,m)\in\mathbb{C}^K \times \mathbb{C}^N} \|y - \mathrm{diag}(Bhm^* C^*)\|_2^2;$$

---

[3] X. Li et. al., Rapid, robust, and reliable blind deconvolution via nonconvex optimization, *preprint arXiv:1606.04933*, 2016

[4] The penalty in the cost function is not added for simplicity

Related work

$$\boxed{\text{Find } h, m, \text{ s. t. } y = \mathrm{diag}(Bhm^* C^*);}$$

- Li et al. [LLSW18][3]
    - Nonconvex problem[4]:

$$\min_{(h,m)\in\mathbb{C}^K\times\mathbb{C}^N} \|y - \mathrm{diag}(Bhm^* C^*)\|_2^2;$$

    - (Theoretical result):
        - A good initialization
        - (Wirtinger flow method + a good initialization) $\xrightarrow{\text{high probability}}$ the true solution;

---

[3] X. Li et. al., Rapid, robust, and reliable blind deconvolution via nonconvex optimization, *preprint arXiv:1606.04933*, 2016

[4] The penalty in the cost function is not added for simplicity

Related work

$$\boxed{\text{Find } h, m, \text{ s. t. } y = \mathrm{diag}(Bhm^* C^*);}$$

- Li et al. [LLSW18][3]
  - Nonconvex problem[4]:

$$\min_{(h,m)\in\mathbb{C}^K\times\mathbb{C}^N} \|y - \mathrm{diag}(Bhm^* C^*)\|_2^2;$$

  - (Theoretical result):
    - A good initialization
    - (Wirtinger flow method + a good initialization) $\xrightarrow{\text{high probability}}$ the true solution;
  - Lower successful recovery probability than alternating minimization algorithm empirically.

---

[3] X. Li et. al., Rapid, robust, and reliable blind deconvolution via nonconvex optimization, *preprint arXiv:1606.04933*, 2016

[4] The penalty in the cost function is not added for simplicity

Manifold Approach

$$\boxed{\text{Find } h, m, \text{ s. t. } y = \mathrm{diag}(Bhm^* C^*);}$$

- The problem is defined on the set of rank-one matrices (denoted by $\mathbb{C}_1^{K\times N}$), neither $\mathbb{C}^{K\times N}$ nor $\mathbb{C}^K \times \mathbb{C}^N$; Why not work on the manifold directly?

Manifold Approach

$$\boxed{\text{Find } h, m, \text{ s. t. } y = \mathrm{diag}(Bhm^* C^*);}$$

- The problem is defined on the set of rank-one matrices (denoted by $\mathbb{C}_1^{K\times N}$), neither $\mathbb{C}^{K\times N}$ nor $\mathbb{C}^K \times \mathbb{C}^N$; Why not work on the manifold directly?
- A representative Riemannian method: Riemannian steepest descent method (RSD)
    - A good initialization
    - (RSD + the good initialization) $\xRightarrow{\text{high probability}}$ the true solution;
    - The Riemannian Hessian at the true solution is well-conditioned;

Manifold Approach

$$\boxed{\text{Find } h, m, \text{ s. t. } y = \mathrm{diag}(Bhm^* C^*);}$$

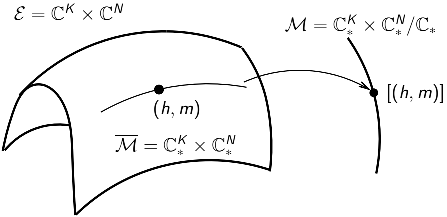
- The problem is defined on the set of rank-one matrices (denoted by $\mathbb{C}_1^{K \times N}$), neither $\mathbb{C}^{K \times N}$ nor $\mathbb{C}^K \times \mathbb{C}^N$; Why not work on the manifold directly?
- Optimization on manifolds: A Riemannian steepest descent method;
    - Representation of $\mathbb{C}_1^{K \times N}$;
    - Representation of directions (tangent vectors);
    - Riemannian metric;
    - Riemannian gradient;

A Representation of $\mathbb{C}_1^{K\times N}$: $\mathbb{C}_*^K \times \mathbb{C}_*^N / \mathbb{C}_*$

- Given $X \in \mathbb{C}_1^{K\times N}$, there exists (h, m), $h \neq 0$ and $m \neq 0$ such that $X = hm^*$;
- (h, m) is not unique;
- The equivalent class: $[(h, m)] = \{(ha, ma^{-*}) \mid a \neq 0\}$;
- Quotient manifold: $\mathbb{C}_*^K \times \mathbb{C}_*^N / \mathbb{C}_* = \{[(h, m)] \mid (h, m) \in \mathbb{C}_*^K \times \mathbb{C}_*^N\}$

A Representation of $\mathbb{C}_1^{K\times N}$: $\mathbb{C}_*^K \times \mathbb{C}_*^N/\mathbb{C}_*$

- Given $X \in \mathbb{C}_1^{K\times N}$, there exists (h, m), $h \neq 0$ and $m \neq 0$ such that $X = hm^*$;
- (h, m) is not unique;
- The equivalent class: $[(h, m)] = \{(ha, ma^{-*}) \mid a \neq 0\}$;
- Quotient manifold: $\mathbb{C}_*^K \times \mathbb{C}_*^N/\mathbb{C}_* = \{[(h, m)] \mid (h, m) \in \mathbb{C}_*^K \times \mathbb{C}_*^N\}$



$$\mathbb{C}_*^K \times \mathbb{C}_*^N/\mathbb{C}_* \simeq \mathbb{C}_1^{K\times N}$$

A Representation of $\mathbb{C}_1^{K\times N}$: $\mathbb{C}_*^K \times \mathbb{C}_*^N/\mathbb{C}_*$

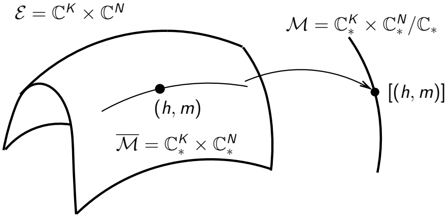
Cost function[5]

- Riemannian approach:

  $f : \mathbb{C}_*^K \times \mathbb{C}_*^N/\mathbb{C}_* \to \mathbb{R} : [(h,m)] \mapsto \|y - \mathrm{diag}(Bhm^* C^*)\|_2^2.$

- Approach in [LLSW18]:

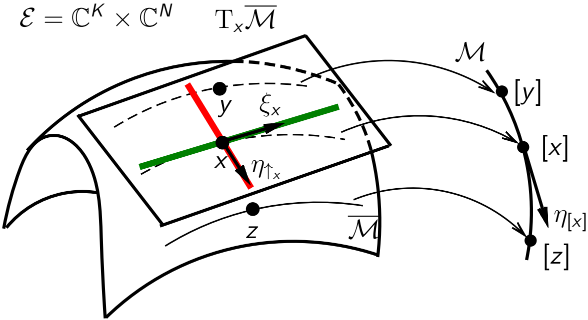
  $\mathfrak{f} : \mathbb{C}^K \times \mathbb{C}^N \to \mathbb{R} : (h,m) \mapsto \|y - \mathrm{diag}(Bhm^* C^*)\|_2^2.$



---

[5] The penalty in the cost function is not added for simplicity.

Representation of directions on $\mathbb{C}_*^K \times \mathbb{C}_*^N / \mathbb{C}_*$



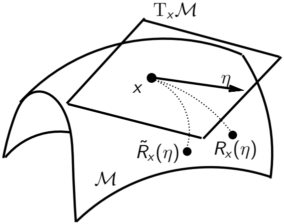
- x denotes (h, m);
- Green line: the tangent space of $[x]$;
- Red line (horizontal space at x): orthogonal to the green line;
- Horizontal space at x: a representation of the tangent space of $\mathcal{M}$ at $[x]$;

Retraction

Euclidean	Riemannian
$x_{k+1} = x_k + \alpha_k d_k$	$x_{k+1} = R_{x_k}(\alpha_k \eta_k)$

- Retraction: $R : \mathrm{T}\,\mathcal{M} \to \mathcal{M}$
- $R(0_{[x]}) = [x]$
- $\frac{dR(t\eta_{[x]})}{dt}\big|_{t=0} = \eta_{[x]}$;
- Retraction on $\mathbb{C}_*^K \times \mathbb{C}_*^N / \mathbb{C}_*$:

$$R_{[(h,m)]}(\eta_{[(h,m)]}) = [(h + \eta_h, m + \eta_m)] .$$



Two retractions: R and $\tilde{R}$

A Riemannian metric

Riemannian metric:

- Inner product on tangent spaces
- Define angles and lengths

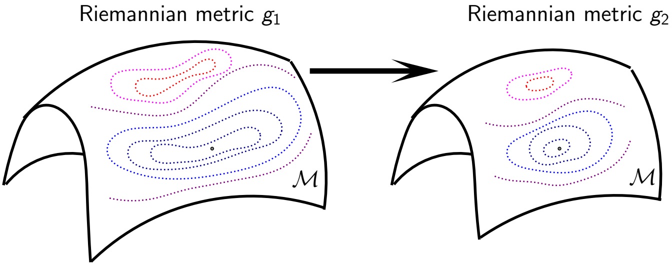


Figure: Changing metric may influence the difficulty of a problem.

A Riemannian metric

$$\boxed{\min_{[(h,m)]} \|y - \mathrm{diag}(Bhm^* C^*)\|_2^2}$$

Idea for choosing a Riemannian metric

The block diagonal terms in the Euclidean Hessian are used to choose the Riemannian metric.

A Riemannian metric

$$\boxed{\min_{[(h,m)]} \|y - \mathrm{diag}(Bhm^* C^*)\|_2^2}$$

Idea for choosing a Riemannian metric

The block diagonal terms in the Euclidean Hessian are used to choose the Riemannian metric.

- Let $\langle u, v \rangle_2 = \mathrm{Re}(\mathrm{trace}(u^* v))$:

$$\frac{1}{2}\langle \eta_h, \mathrm{Hess}_h f[\xi_h] \rangle_2 = \langle \mathrm{diag}(B\eta_h m^* C^*), \mathrm{diag}(B\xi_h m^* C^*) \rangle_2 \approx \langle \eta_h m^*, \xi_h m^* \rangle_2$$
$$\frac{1}{2}\langle \eta_m, \mathrm{Hess}_m f[\xi_m] \rangle_2 = \langle \mathrm{diag}(Bh\eta_m^* C^*), \mathrm{diag}(Bh\xi_m^* C^*) \rangle_2 \approx \langle h\eta_m^*, h\xi_m^* \rangle_2,$$

where $\approx$ can be derived from some assumptions;

A Riemannian metric

$$\boxed{\min_{[(h,m)]} \|y - \mathrm{diag}(Bhm^* C^*)\|_2^2}$$

Idea for choosing a Riemannian metric

The block diagonal terms in the Euclidean Hessian are used to choose the Riemannian metric.

- Let $\langle u, v\rangle_2 = \mathrm{Re}(\mathrm{trace}(u^* v))$:

$$\frac{1}{2}\langle \eta_h, \mathrm{Hess}_h f[\xi_h]\rangle_2 = \langle \mathrm{diag}(B\eta_h m^* C^*), \mathrm{diag}(B\xi_h m^* C^*)\rangle_2 \approx \langle \eta_h m^*, \xi_h m^*\rangle_2$$
$$\frac{1}{2}\langle \eta_m, \mathrm{Hess}_m f[\xi_m]\rangle_2 = \langle \mathrm{diag}(Bh\eta_m^* C^*), \mathrm{diag}(Bh\xi_m^* C^*)\rangle_2 \approx \langle h\eta_m^*, h\xi_m^*\rangle_2,$$

  where $\approx$ can be derived from some assumptions;

- The Riemannian metric:

$$g\left(\eta_{[x]}, \xi_{[x]}\right) = \langle \eta_h, \xi_h m^* m\rangle_2 + \langle \eta_m^*, \xi_m^* h^* h\rangle_2;$$

A Riemannian metric

$$\boxed{\min_{[(h,m)]} \|y - \mathrm{diag}(Bhm^* C^*)\|_2^2}$$

RIP

Restricted Isometry Property for a linear operator $\mathcal{A}$ holds uniformly for all X satisfying $\mathrm{rank}(X) \leq 2$ if

$$\frac{3}{4}\|X\|_F^2 \leq \|\mathcal{A}(X)\|_2^2 \leq \frac{5}{4}\|X\|_F^2.$$

In BD problem, we have $\mathcal{A}(Z) = \mathrm{diag}(BZC^*)$.

This is a nice property of the objective function around the minimizer.

A Riemannian metric

Discussions

$$\begin{aligned}
&\langle(\eta_h,\eta_m),\mathrm{Hess}\, f([(h,m)])[(\eta_h,\eta_m)]\rangle_{[(h,m)]}\\
=&\underbrace{\|\mathcal{A}(h\eta_m^*+\eta_h m^*)\|_2^2}_{\text{well conditioned by RIP}}+\underbrace{\langle h\eta_m^*+\eta_h m^*,\mathrm{P}_{(\eta_h,\eta_m)}\mathrm{P}_{\mathrm{N}}^{\perp}\mathcal{A}^*(\mathcal{A}(hm^*)-y)\rangle_2}_{\text{Geometry}}\\
\approx&\|h\eta_m^*+\eta_h m^*\|_2^2+\text{geometry}\\
=&\underbrace{\langle\eta_h,\eta_h m^* m\rangle_2+\langle\eta_m,\eta_m h^* h\rangle_2}_{\text{metric}}+2\langle h\eta_m^*,\eta_h m^*\rangle_2+\text{geometry}
\end{aligned}$$

Note that the left hand side is independent of Riemannian metric and geometry.

Riemannian gradient

- Riemannian gradient
  - A tangent vector: $\mathrm{grad}\,\mathbf{f}([x]) \in \mathrm{T}_{[x]}\,\mathcal{M}$;
  - Satisfies: $\mathrm{D} f([x])[\eta_{[x]}] = g(\mathrm{grad}\, f([x]), \eta_{[x]}), \quad \forall \eta_{[x]} \in \mathrm{T}_{[x]}\,\mathcal{M}$;
- Represented by a vector in a horizontal space;
- Riemannian gradient:

$$(\mathrm{grad}\, f([(h, m)]))_{\uparrow_{(h,m)}} = \left(\nabla_h f(h, m)(m^* m)^{-1}, \nabla_m f(h, m)(h^* h)^{-1}\right);$$

A Riemannian steepest descent method (RSD)

An implementation of a Riemannian steepest descent method[6]

0. Given (h_0, m_0), step size $\alpha > 0$, and set $k = 0$
1. $d_k = \|h_k\|_2 \|m_k\|_2$, $h_k \leftarrow \sqrt{d_k} \frac{h_k}{\|h_k\|_2}$; $m_k \leftarrow \sqrt{d_k} \frac{m_k}{\|m_k\|_2}$;
2. $(h_{k+1}, m_{k+1}) = (h_k, m_k) - \alpha \left(\frac{\nabla_{h_k} f(h_k, m_k)}{d_k}, \frac{\nabla_{m_k} f(h_k, m_k)}{d_k} \right)$;
3. If not converge, goto Step 2.

---

[6] The penalty in the cost function is not added for simplicity

A Riemannian steepest descent method (RSD)

An implementation of a Riemannian steepest descent method[6]

0. Given (h_0, m_0), step size $\alpha > 0$, and set $k = 0$
1. $d_k = \|h_k\|_2 \|m_k\|_2,\ h_k \leftarrow \sqrt{d_k} \frac{h_k}{\|h_k\|_2};\ m_k \leftarrow \sqrt{d_k} \frac{m_k}{\|m_k\|_2};$
2. $(h_{k+1}, m_{k+1}) = (h_k, m_k) - \alpha \left(\frac{\nabla_{h_k} f(h_k, m_k)}{d_k}, \frac{\nabla_{m_k} f(h_k, m_k)}{d_k} \right);$
3. If not converge, goto Step 2.

Wirtinger flow Method in [LLSW18]

0. Given (h_0, m_0), step size $\alpha > 0$, and set $k = 0$
1. $(h_{k+1}, m_{k+1}) = (h_k, m_k) - \alpha (\nabla_{h_k} f(h_k, m_k), \nabla_{m_k} f(h_k, m_k));$
2. If not converge, goto Step 2.

---

[6] The penalty in the cost function is not added for simplicity

A Riemannian steepest descent method (RSD)

An implementation of a Riemannian steepest descent method[6]

0. Given (h_0, m_0), step size $\alpha > 0$, and set $k = 0$
1. $d_k = \|h_k\|_2 \|m_k\|_2, \; h_k \leftarrow \sqrt{d_k} \frac{h_k}{\|h_k\|_2}; \; m_k \leftarrow \sqrt{d_k} \frac{m_k}{\|m_k\|_2};$
2. $(h_{k+1}, m_{k+1}) = (h_k, m_k) - \alpha \left(\frac{\nabla_{h_k} f(h_k, m_k)}{d_k}, \frac{\nabla_{m_k} f(h_k, m_k)}{d_k} \right);$
3. If not converge, goto Step 2.

Wirtinger flow Method in [LLSW18]

0. Given (h_0, m_0), step size $\alpha > 0$, and set $k = 0$
1. $(h_{k+1}, m_{k+1}) = (h_k, m_k) - \alpha (\nabla_{h_k} f(h_k, m_k), \nabla_{m_k} f(h_k, m_k));$
2. If not converge, goto Step 2.

---

[6] The penalty in the cost function is not added for simplicity

Penalty

Penalty term for (i) Riemannian method, (ii) Wirtinger flow [LLSW18]

$$\text{(i): } \rho \sum_{i=1}^{L} G_0 \left(\frac{L|b_i^* h|^2 \|m\|_2^2}{8 d^2 \mu^2} \right)$$

$$\text{(ii): } \rho \left[G_0 \left(\frac{\|h\|_2^2}{2d} \right) + G_0 \left(\frac{\|m\|_2^2}{2d} \right) + \sum_{i=1}^{L} G_0 \left(\frac{L|b_i^* h|^2}{8 d \mu^2} \right) \right],$$

where $G_0(t) = \max(t - 1, 0)^2$, $[b_1 b_2 \dots b_L]^* = B$.

- The first two terms in (ii) penalize large values of $\|h\|_2$ and $\|m\|_2$;

Penalty

Penalty term for (i) Riemannian method, (ii) Wirtinger flow [LLSW18]

$$\text{(i):}\ \rho \sum_{i=1}^{L} G_0 \left(\frac{L|b_i^* h|^2 \|m\|_2^2}{8d^2\mu^2} \right)$$

$$\text{(ii):}\ \rho \left[G_0 \left(\frac{\|h\|_2^2}{2d} \right) + G_0 \left(\frac{\|m\|_2^2}{2d} \right) + \sum_{i=1}^{L} G_0 \left(\frac{L|b_i^* h|^2}{8d\mu^2} \right) \right],$$

where $G_0(t) = \max(t-1, 0)^2$, $[b_1 b_2 \dots b_L]^* = B$.

- The first two terms in (ii) penalize large values of $\|h\|_2$ and $\|m\|_2$;
- The other terms promote a small coherence;
- The one in (i) is defined in the quotient space whereas the one in (ii) is not.

Penalty/Coherence

- Coherence is defined as

$$\mu_h^2 = \frac{L\|Bh\|_\infty^2}{\|h\|_2^2} = \frac{L \max\left(|b_1^* h|^2, |b_2^* h|^2, \ldots, |b_L^* h|^2\right)}{\|h\|_2^2};$$

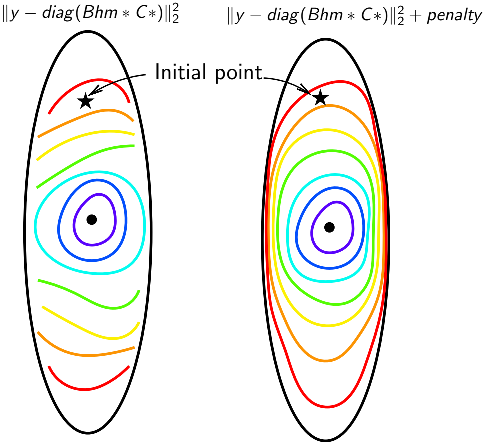
- Coherence at the true solution $[(h_\sharp, m_\sharp)]$
  - influences the probability of recovery
  - Small coherence is preferred

Penalty

Promote low coherence:

$$\rho \sum_{i=1}^{L} G_0 \left(\frac{L |b_i^* h|^2 \|m\|_2^2}{8 d^2 \mu^2} \right),$$

where $G_0(t) = \max(t-1,0)^2$;



Initialization

Initialization method [LLSW18]

- $(d, \tilde{h}_0, \tilde{m}_0)$: SVD of $B^* \operatorname{diag}(y) C$;
- $h_0 = \operatorname{argmin}_z \|z - \sqrt{d}\tilde{h}_0\|_2^2$, subject to $\sqrt{L}\|Bz\|_\infty \leq 2\sqrt{d}\mu$;
- $m_0 = \sqrt{d}\tilde{m}_0$;
- Initial iterate $[(h_0, m_0)]$;

Numerical Results

- Synthetic tests
  - Efficiency
  - Probability of successful recovery
- Image deblurring
  - Kernels with known supports
  - Motion kernel with inexact supports

Efficiency

$$\boxed{\min \|y - \mathrm{diag}(Bhm^* C^*)\|_2^2}$$

Table: Comparisons of efficiency

	$L = 400, K = N = 50$			$L = 600, K = N = 50$		
Algorithms	[LLSW18]	[LWB18]	R-SD	[LLSW18]	[LWB18]	R-SD
nBh/nCm	351	718	208	162	294	122
$nFFT$	870	1436	518	401	588	303
$RMSE$	2.22_{-8}	3.67_{-8}	2.20_{-8}	1.48_{-8}	2.34_{-8}	1.42_{-8}

- An average of 100 random runs
- nBh/nCm: the numbers of Bh and Cm multiplication operations respectively
- nFFT: the number of Fourier transform
- RMSE: the relative error $\frac{\|hm^* - h_\sharp m_\sharp^*\|_F}{\|h_\sharp\|_2 \|m_\sharp\|_2}$

[LLSW18]: X. Li et. al., Rapid, robust, and reliable blind deconvolution via nonconvex optimization, *preprint arXiv:1606.04933*, 2016

[LWB18]: K. Lee et. al., Near Optimal Compressed Sensing of a Class of Sparse Low-Rank Matrices via Sparse Power Factorization *preprint arXiv:1312.0525*, 2013

Probability of successful recovery

- Success if $\frac{\|hm^* - h_\sharp m_\sharp^*\|_F}{\|h_\sharp\|_2 \|m_\sharp\|_2} \leq 10^{-2}$

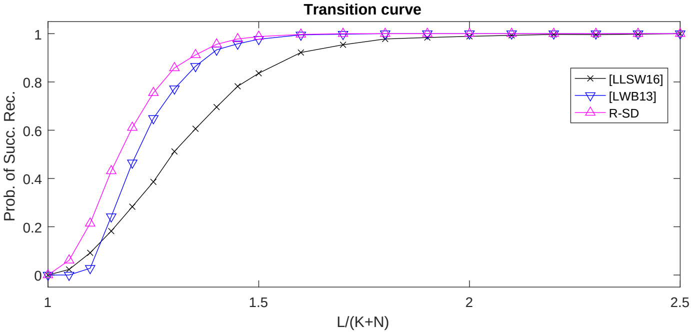
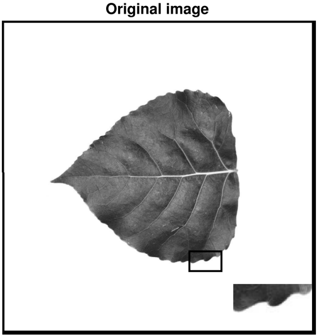


Figure: Empirical phase transition curves for 1000 random runs.

[LLSW18]: X. Li et. al., Rapid, robust, and reliable blind deconvolution via nonconvex optimization, *preprint arXiv:1606.04933*, 2016

[LWB18]: K. Lee et. al., Near Optimal Compressed Sensing of a Class of Sparse Low-Rank Matrices via Sparse Power Factorization *preprint arXiv:1312.0525*, 2013

Image deblurring



- Image [WBX$^+$07]: 1024-by-1024 pixels

Image deblurring with various kernels

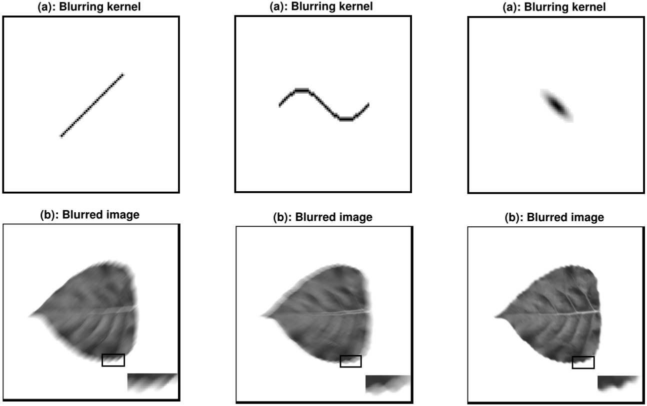


Figure: Left: Motion kernel by Matlab function "fspecial('motion', 50, 45)"; Middle: Kernel like function "sin"; Right: Gaussian kernel with covariance $[1, 0.8; 0.8, 1]$;

Image deblurring with various kernels

$$\boxed{\min \|y - \mathrm{diag}(Bhm^* C^*)\|_2^2}$$

What subspaces are the two unknown signals in?

- Image is approximately sparse in the Haar wavelet basis

- Support of the blurring kernel is learned from the blurred image

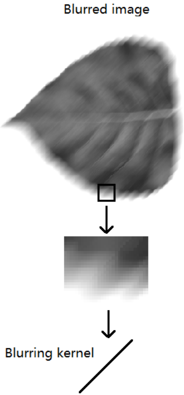


Image deblurring with various kernels

$$\boxed{\min \|y - \mathrm{diag}(Bhm^* C^*)\|_2^2}$$

What subspaces are the two unknown signals in?

- Image is approximately sparse in the Haar wavelet basis

  Use the blurred image to learn the dominated basis vectors: **C**.

- Support of the blurring kernel is learned from the blurred image

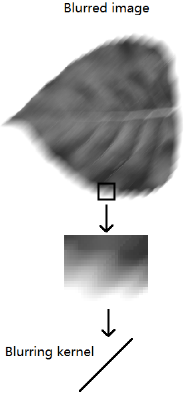


Image deblurring with various kernels

$$\boxed{\min \|y - \mathrm{diag}(Bhm^* C^*)\|_2^2}$$

What subspaces are the two unknown signals in?

- Image is approximately sparse in the Haar wavelet basis

  Use the blurred image to learn the dominated basis vectors: **C**.

- Support of the blurring kernel is learned from the blurred image

  Suppose the supports of the blurring kernels are known: **B**.

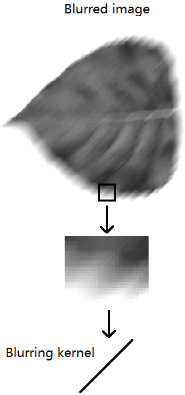


Image deblurring with various kernels

$$\boxed{\min \|y - \mathrm{diag}(Bhm^* C^*)\|_2^2}$$

What subspaces are the two unknown signals in?

- Image is approximately sparse in the Haar wavelet basis

  Use the blurred image to learn the dominated basis vectors: **C**.

- Support of the blurring kernel is learned from the blurred image

  Suppose the supports of the blurring kernels are known: **B**.

- $L = 1048576$, $N = 20000$, $K_{motion} = 109$, $K_{sin} = 153$, $K_{Gaussian} = 181$;

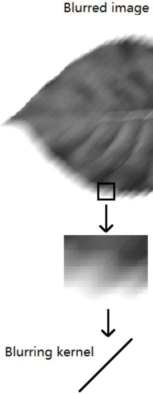


Image deblurring with various kernels

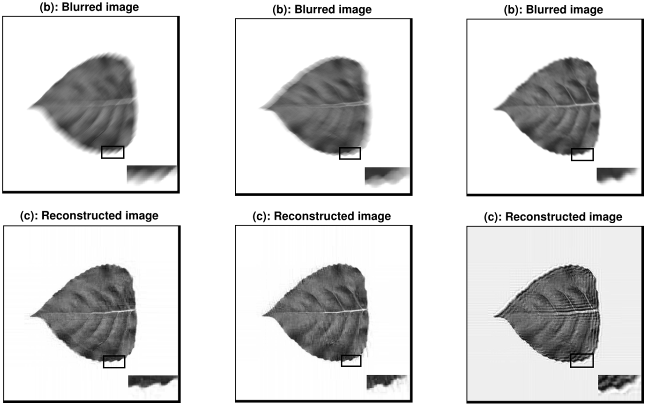


Figure: The number of iterations is 80; Computational times are about 48s; Relative errors $\left\|\hat{\mathbf{y}} - \frac{\|\mathbf{y}\|}{\|\mathbf{y}_f\|}\mathbf{y}_f\right\| / \|\hat{\mathbf{y}}\|$ are 0.038, 0.040, and 0.089 from left to right.

Image deblurring with unknown supports

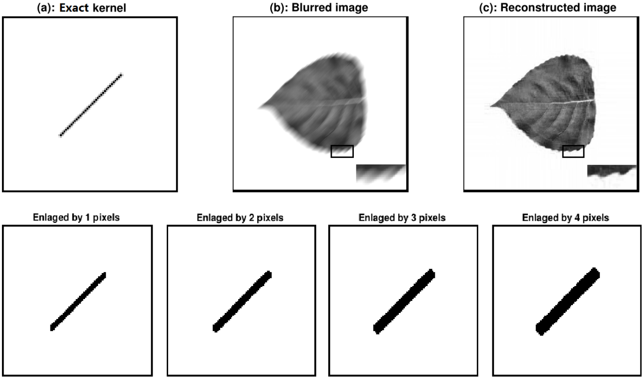


Figure: Top: reconstructed image using the exact support; Bottom: estimated supports with the numbers of nonzero entries: $K_1 = 183$, $K_2 = 265$, $K_3 = 351$, and $K_4 = 441$;

Image deblurring with inexact supports

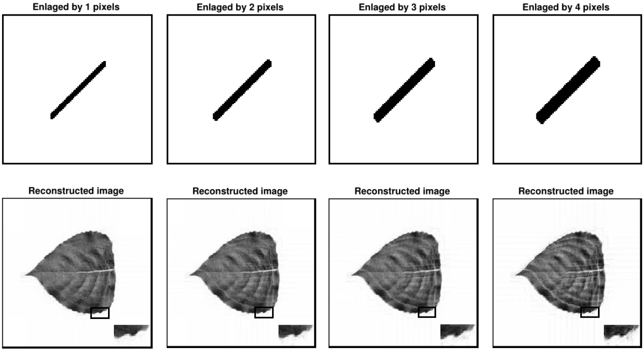


Figure: Relative errors $\left\| \hat{\mathbf{y}} - \frac{\|\mathbf{y}\|}{\|\mathbf{y}_f\|} \mathbf{y}_f \right\| / \|\hat{\mathbf{y}}\|$ are 0.044, 0.048, 0.052, and 0.067 from left to right.

Summary of BD

- Introduce rectraction and transport-based Riemannian optimization
- RSD has efficient implementation for solving blind deconvolution problem
- RSD method has recovery guarantee
- RSD is faster and has higher probability of successful recovery compared to the alternating minimization method and the approach in [LLSW18]
- RSD method works well for the tested imaging debluring problems

W. Huang, P. Hand. Blind Deconvolution by a Steepest Descent Algorithm on a Quotient Manifold, SIAM Journal on Imaging Sciences, 11:4, pp. 2757-2785, 2018.

Discussions

- RIP also appears in Phase Retrieval;
- Preconditioned the manifold (use a non-Euclidean space) has been proposed for many problems on the fixed-rank manifold;
- Fixed rank manifold has multiple representation, which yields different metrics;
- Preferred retraction;

Content

1. Geometric Mean of SPD Matrices
    - Motivations;
    - Averaging on a Riemannian manifold;
    - Algorithms and manifold geometry;
2. Signal Recovery on Low-rank Matrices
    - Motivations;
    - Problem formulations;
    - Algorithms and manifold geometry;
3. Rank Overestimation (Hermitian PSD low-rank Constraints);
    - Problem formulation;
    - Riemannian metrics;
    - Condition number for nearly low-rank solutions;

Content

Problem of interest:

$$\begin{array}{ll} \underset{X}{\text{minimize}} & f(X) = \frac{1}{2}\|\mathcal{A}(X) - b\|_F^2 \\ \text{subject to} & X \in \mathcal{H}_+^{n,p} \end{array},$$

where $\mathcal{H}_+^{n,p}$ denotes the set of n-by-n Hermitian PSD matrices of fixed rank $p \ll n$.

Approximating solutions to a minimization with a convex PSD constraint:

$$\begin{array}{ll} \underset{X \in \mathbb{C}^{n \times n}}{\text{minimize}} & f(X) = \frac{1}{2}\|\mathcal{A}(X) - b\|_F^2 \\ \text{subject to} & X \succcurlyeq 0 \end{array}. \qquad (2)$$

Applications: Phase retrieval by PhaseLift [HGZ17], interferometry recovery problem, etc.

Problem formulation

$$\begin{array}{ll} \underset{X}{\text{minimize}} & f(X) = \frac{1}{2}\|\mathcal{A}(X) - b\|_F^2 \\ \text{subject to} & X \in \mathcal{H}_+^{n,p} \end{array},$$

Multiple approaches:

- Burer-Monteiro method: $\min_{Y \in \mathbb{C}^{n \times p}} F(Y) := f(YY^*)$.
- Regard $\mathcal{H}_+^{n,p}$ as an embedded submanifold of $\mathbb{C}^{n \times n}$;
- Consider the quotient manifold $\mathbb{C}_*^{n \times p} / \mathcal{O}_p$.

Which approach is preferred when p is greater than the rank of the true solution?

Problem formulation

Why using p greater than the rank of the true solution?

Theorem

Suppose $Y = K_s Q^*$ is a rank deficient minimizer of F, where $K_s \in \mathcal{C}_*^{n \times s}$ with $s < p$ and $Q \in \mathrm{St}(s, p)$. Then $\mathrm{grad}\, f(Y_p Y_p^*)$ is a positive semidefinite matrix and, therefore, $X = YY^*$ is a stationary point of f. If furthermore f is convex, then X is a global minimizer.

Problem formulation

Multiple approaches:

- Burer-Monteiro method: $\min_{Y \in \mathbb{C}^{n \times p}} F(Y) := f(YY^*)$.
- Regard $\mathcal{H}_+^{n,p}$ as an embedded submanifold of $\mathbb{C}^{n \times n}$;
- Consider the quotient manifold $\mathbb{C}_*^{n \times p} / \mathcal{O}_p$.

The three approaches can be equivalently reformulated into quotient manifold with different Riemannian metric on $\mathbb{C}_*^{n \times p}$:

$$g_Y^1(A, B) = \langle A, B \rangle_{\mathbb{C}^{n \times p}} = \Re(\mathrm{tr}(A^* B)) \quad \text{(Bures-Wasserstein metric)}$$

$$g_Y^2(A, B) = \langle AY^*, BY^* \rangle_{\mathbb{C}^{n \times n}} = \Re(\mathrm{tr}((Y^* Y) A^* B))$$

$$\begin{aligned} g_Y^3(A, B) = {} & \langle YA^* + AY^*, YB^* + BY^* \rangle_{\mathbb{C}^{n \times n}} \\ & + \left\langle Y \,\mathrm{Skew}\left((Y^* Y)^{-1} Y^* A\right) Y^*, Y \,\mathrm{Skew}\left((Y^* Y)^{-1} Y^* B\right) Y^* \right\rangle_{\mathbb{C}^{n \times n}}, \end{aligned}$$

where $\mathrm{Skew}(X) = (X - X^*)/2$.

Interesting findings

The gradient descent (GD) and nonlinear conjugate gradient (CG) applied to Burer–Monteiro form are equivalent to Riemannian GD and Riemannian CG on quotient manifold $(\mathbb{C}_*^{n \times p}/\mathcal{O}_p, g^1)$, which is counter intuitive because they look quite different.

Theorem

There exists a retraction and a vector transport such that when the Bures-Wasserstein metric g^1 is used, the Riemannian conjugate gradient algorithm is equivalent to the Euclidean conjugate gradient method in the sense that they produce exactly the same iterates if started from the same initial point.

Interesting findings

Riemannian GD and Riemannian CG on the embedded manifold $\mathcal{H}_+^{n,p}$ are exactly equivalent to Riemannian GD and Riemannian CG on quotient manifold $(\mathbb{C}_*^{n\times p}/\mathcal{O}_p, g^3)$.

Theorem

Let R^E and $\mathcal{T}^E$ denote any retraction and vector transport used in Algorithms with embedded geometry $\mathcal{H}_+^{n,p}$. Using the diffeomorphism $\tilde{\beta}$ between $\mathbb{C}_*^{n\times p}/\mathcal{O}_p$ and $\mathcal{H}_+^{n,p}$ and isomorphism $L_{\pi(Y)}$ between $T_{\pi(Y)}\mathbb{C}_*^{n\times p}/\mathcal{O}_p$ and $T_{YY^*}\mathcal{H}_+^{n,p}$, define the retraction R^Q and vector transport $\mathcal{T}^Q$ on the quotient manifold $(\mathbb{C}_*^{n\times p}/\mathcal{O}_p, g^3)$ as $R^Q_{\pi(Y)}(\xi_{\pi(Y)}) := \tilde{\beta}^{-1}\left(R^E_{\tilde{\beta}(\pi(Y))}\left(L(\xi_{\pi(Y)})\right)\right)$, and $\mathcal{T}^Q_{\eta_{\pi(Y)}}(\xi_{\pi(Y)}) := L^{-1}_{\pi(Y_2)}\left(\mathcal{T}^E_{L(\eta_{\pi(Y)})}\left(L(\xi_{\pi(Y)})\right)\right)$, where $\pi(Y_2)$ is in $\mathbb{C}_*^{n\times p}/\mathcal{O}_p$ such that $\tilde{\beta}(\pi(Y_2))$ denotes the foot of the tangent vector $\mathcal{T}^E_{L(\eta_{\pi(Y)})}\left(L(\xi_{\pi(Y)})\right)$. Using R^Q and $\mathcal{T}^Q$ as the retraction and vector transport in Algorithm with quotient geometry and g^3, and assume the initial step t_k is be chosen to be the same, then Algorithm with quotient geometry and g^3 is equivalent to Algorithm with embedded geometry in the sense that they produce exactly the same iterates if started from the same initial point.

Interesting findings

Main results: Conditioning of Riemannian Hessians

The Rayleigh quotient of the Riemannian Hessian of h on $(\mathbb{C}_*^{n \times p}/\mathcal{O}_p, g^i)$ is defined by

$$\rho^i(\pi(Y), \xi_{\pi(Y)}) = \frac{g^i_{\pi(Y)}\left(\mathrm{Hess}\, h(\pi(Y))[\xi_{\pi(Y)}], \xi_{\pi(Y)}\right)}{g^i_{\pi(Y)}(\xi_{\pi(Y)}, \xi_{\pi(Y)})}, \quad \forall \xi_{\pi(Y)} \in T_{\pi(Y)} \mathbb{C}_*^{n \times p}/\mathcal{O}_p.$$

If the Rayleigh quotient has a low bound μ and an upper bound L, then L/μ is an upper bound of the condition number of the Riemannian Hessian.

Interesting findings

Main results: Conditioning of Riemannian Hessians

Assumption

Let $\hat{X}$ be the global minimizer of f. For a fixed $\epsilon > 0$, there exist constants $A > 0$ and $B > 0$ such that for all X with $\|X - \hat{X}\|_F < \epsilon$, the following inequality holds,

$$A\|\zeta_X\|_F^2 \leq \left\langle \nabla^2 f(X)[\zeta_X], \zeta_X \right\rangle_{\mathbb{C}^{n\times n}} \leq B\|\zeta_X\|_F^2, \quad \forall \zeta_X \in T_X \mathcal{H}_+^{n,p}.$$

Interesting findings

Theorem

Let $\hat{X} = \hat{Y}\hat{Y}^*$ *be the global minimizer of* (2) *with rank* $r = p$*. For* $X = YY^* = U\Sigma U^*$ *with singular values* σ_i, $Y \in \mathbb{C}_*^{n\times p}$, *and* X *near* $\hat{X}$*, under the above Assumption, for any arbitrary tangent vectors* ζ_X *and* $\xi_{\pi(Y)}$*, the following hold:*

1. $A - \frac{2}{\sigma_p}\|\nabla f(X)\| \leq \rho^E(X, \zeta_X) \leq B + \frac{2}{\sigma_p}\|\nabla f(X)\|$,
2. $2A\sigma_p - 2\|\nabla f(YY^*)\| \leq \rho^1(\pi(Y), \xi_{\pi(Y)}) \leq B \cdot D^1_{\pi(Y)} + 2\|\nabla f(YY^*)\|$,
3. $2A - \frac{4(\sqrt{p}+1)}{\sigma_p}\|\nabla f(YY^*)\| \leq \rho^2(\pi(Y), \xi_{\pi(Y)}) \leq 4B + \frac{4(\sqrt{p}+1)}{\sigma_p}\|\nabla f(YY^*)\|$,
4. $A - \frac{1}{\sigma_p}\|\nabla f(YY^*)\| \leq \rho^3(\pi(Y), \xi_{\pi(Y)}) \leq B + \frac{1}{\sigma_p}\|\nabla f(YY^*)\|$,

where $D^1_{\pi(Y)}$ *satisfies* $2\sigma_1 \leq D^1_{\pi(Y)} \leq 2\left(\frac{\sigma_1^2}{\sigma_p} + \sigma_1\right)$.

Interesting findings

Theorem (Continue)

In particular, if $\hat{X} = \hat{Y}\hat{Y}^*$ *has rank p, we have the following limits, where* $X \to \hat{X}$ *and* $\pi(Y) \to \pi(\hat{Y})$ *are taken in the sense of* $\left\|X - \hat{X}\right\|_F \to 0$ *and* $\left\|YY^* - \hat{Y}\hat{Y}^*\right\|_F \to 0$*:*

1. $A - \frac{2}{\hat{\sigma}_p}\left\|\nabla f(\hat{X})\right\| \leq \lim_{X \to \hat{X}} \rho^E(X, \xi_X) \leq B + \frac{2}{\hat{\sigma}_p}\left\|\nabla f(\hat{X})\right\|,$
2. $2A\hat{\sigma}_p - 2\left\|\nabla f(\hat{X})\right\| \leq \lim_{\pi(Y) \to \pi(\hat{Y})} \rho^1(\pi(Y), \xi_{\pi(Y)}) \leq$ $B \cdot D^1_{\pi(\hat{Y})} + 2\left\|\nabla f(\hat{X})\right\|,$
3. $2A - \frac{4(\sqrt{p}+1)}{\hat{\sigma}_p}\left\|\nabla f(\hat{X})\right\| \leq \lim_{\pi(Y) \to \pi(\hat{Y})} \rho^2(\pi(Y), \xi_{\pi(Y)}) \leq$ $4B + \frac{4(\sqrt{p}+1)}{\hat{\sigma}_p}\left\|\nabla f(\hat{X})\right\|,$
4. $A - \frac{1}{\hat{\sigma}_p}\left\|\nabla f(\hat{X})\right\| \leq \lim_{\pi(Y) \to \pi(\hat{Y})} \rho^3(\pi(Y), \xi_{\pi(Y)}) \leq B + \frac{1}{\hat{\sigma}_p}\left\|\nabla f(\hat{X})\right\|,$

where $D^1_{\pi(\hat{Y})}$ *satisfies* $2\hat{\sigma}_1 \leq D^1_{\pi(\hat{Y})} \leq 2\left(\frac{\hat{\sigma}_1^2}{\hat{\sigma}_p} + \hat{\sigma}_1\right).$

Interesting findings

Assumption

For a sequence $\{X_k\}$ *with* $X_k \in \mathcal{H}_+^{n,p}$ *(or* $\pi(Y_k) \in \mathbb{C}_*^{n \times p} / \mathcal{O}_p$ *) that converges to the minimizer* $\hat{X}$ *(or* $\pi(\hat{Y})$*), let* $(\sigma_p)_k$ *be the smallest nonzero singular value of* $X_k = Y_k Y_k^*$*, assume the following limits hold.*

1. *For the embedded manifold,* $\lim_{k \to \infty} \frac{2}{(\sigma_p)_k} \|\nabla f(X_k)\| \leq \frac{A}{2}$.
2. *For the quotient manifold with metric* g^1, $\lim_{k \to \infty} \frac{1}{(\sigma_p)_k} \|\nabla f(Y_k Y_k^*)\| \leq \frac{A}{2}$.
3. *For the quotient manifold with metric* g^2, $\lim_{k \to \infty} \frac{4(\sqrt{p}+1)}{(\sigma_p)_k} \|\nabla f(Y_k Y_k^*)\| \leq A$.
4. *For the quotient manifold with metric* g^3, $\lim_{k \to \infty} \frac{1}{(\sigma_p)_k} \|\nabla f(Y_k Y_k^*)\| \leq \frac{A}{2}$.

Interesting findings

If $\hat{X}$ has rank $r < p$ and $\{X_k\}$ is a sequence that satisfies Assumption on the previous page, then Theorem implies

1. For the embedded manifold we have $\frac{A}{2} \leq \lim_{k\to\infty} \rho^E(X_k, \xi_{X_k}) \leq B + \frac{A}{2}$.
2. $A \leq \lim_{k\to\infty} \frac{\rho^1(\pi(Y_k),\xi_{\pi(Y_k)})}{(\sigma_p)_k} \leq B \lim_{k\to\infty} \frac{D^1_{\pi(Y_k)}}{(\sigma_p)_k} + 2A$,
3. $A \leq \lim_{k\to\infty} \rho^2(\pi(Y_k), \xi_{\pi(Y_k)}) \leq 4B + A$,
4. $\frac{A}{2} \leq \lim_{k\to\infty} \rho^3(\pi(Y_k), \xi_{\pi(Y_k)}) \leq B + \frac{A}{2}$,

where $\lim\limits_{k\to\infty} \frac{D^1_{\pi(Y_k)}}{(\sigma_p)_k} \geq \lim\limits_{k\to\infty} \frac{2(\sigma_1)_k}{(\sigma_p)_k} = +\infty$ since $\sigma_p \to \hat{\sigma}_p = 0$.

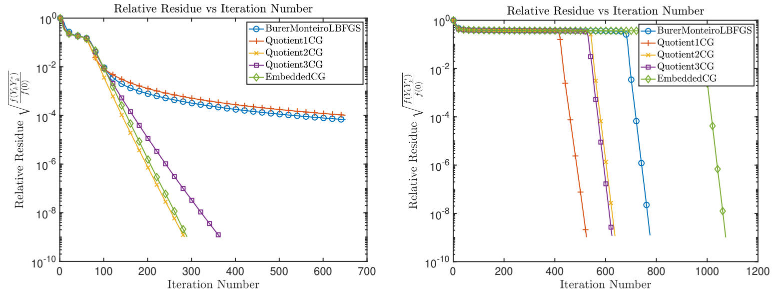
Numerical Experiments

Phase retrieval problem

$$f(X) = \frac{1}{2} \left\| \mathcal{A}(X) - b \right\|^2,$$

where $X = xx^*$ is a rank-one matrix and x represents an image, and $\mathcal{A} : \mathbb{C}^{n \times n} \to \mathbb{R}^{mn \times 1}, \quad X \mapsto [\mathrm{diag}(Z^1 X Z^{1^*}), \cdots, \mathrm{diag}(Z^m X Z^{m*})]^T$ with given $Z^i \in \mathbb{C}^{n \times n}$, see [CSV13].

Numerical Experiments



(a) The algorithms are solved on the rank 3 manifold

(b) The algorithms are solved on the rank 1 manifold

Figure: Phase retrieval of a 256-by-256 image with 6 Gaussian masks. A comparison of relative residue $\frac{\left\|\mathcal{A}(Y_kY_k^*)-b\right\|}{\|b\|}$ versus iteration number k when using L-BFGS approach, quotient CG method with metric $g^i, i = 1, 2, 3$ and embedded CG method. When the minimizer is rank deficient (the case in 8(a)), L-BFGS approach and CG method with metric g^1 is significantly slower.

Numerical Experiments

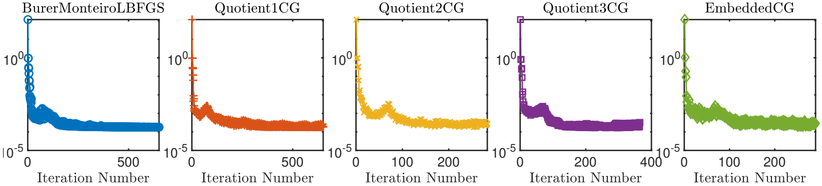


Figure: Numerical examination of Assumption 3.2 for the phase retrieval problem of a 256-by-256 image with 6 Gaussian masks solved on the rank 3 manifold (same setup as the numerical test shown in Fig 8(a)). Plots show the ratio term $\frac{\left\|\nabla f(Y_k Y_k^*)\right\|_F}{(\sigma_p)_k}$ in the Assumption 3.2 versus the iteration number k for L-BFGS approach, quotient CG method with metric $g^i, i = 1, 2, 3$ and embedded CG method.

Summary of Rank Overestimation

- Optimization over Hermitian PSD matrices of fixed rank;
- Multiple geometries to quotient geometry with multiple metrics;
- Rank overestimation accelerates convergence;
- Bures-Wasserstein metric is worse than the other two when minimizer is (almost) rank deficient.

S. Zheng, W. Huang, B. Vandereycken, and X. Zhang. Riemannian optimization using three different metrics for Hermitian PSD fixed-rank constraints, 91, 1135-1184, 2025.

Thank you

Thank you!

References I

A. Ahmed, B. Recht, and J. Romberg.
Blind deconvolution using convex programming.
IEEE Transactions on Information Theory, 60(3):1711–1732, March 2014.

Ognjen Arandjelovic, Gregory Shakhnarovich, John Fisher, Roberto Cipolla, and Trevor Darrell.
Face recognition with image sets using manifold density divergence.
In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 581–588. IEEE, 2005.

D. A. Bini and B. Iannazzo.
Computing the Karcher mean of symmetric positive definite matrices.
Linear Algebra and its Applications, 438(4):1700–1710, 2013.

Guang Cheng, Hesamoddin Salehian, and Baba Vemuri.
Efficient recursive algorithms for computing the mean diffusion tensor and applications to DTI segmentation.
Computer Vision–ECCV 2012, pages 390–401, 2012.

E. J. Candès, T. Strohmer, and V. Voroninski.
PhaseLift : Exact and stable signal recovery from magnitude measurements via convex programming.
Communications on Pure and Applied Mathematics, 66(8):1241–1274, 2013.

P. T. Fletcher and S. Joshi.
Riemannian geometry for the statistical analysis of diffusion tensor data.
Signal Processing, 87(2):250–262, 2007.

W. Huang, P.-A. Absil, and K. A. Gallivan.
Intrinsic representation of tangent vectors and vector transport on matrix manifolds.
Numerische Mathematik, 136(2):523–543, 2017.

References II

Wen Huang, K. A. Gallivan, and Xiangxiong Zhang.
Solving PhaseLift by low rank Riemannian optimization methods for complex semidefinite constraints.
SIAM Journal on Scientific Computing, 39(5):B840–B859, 2017.

Zhiwu Huang, Ruiping Wang, Shiguang Shan, and Xilin Chen.
Face recognition on large-scale video in the wild with hybrid Euclidean-and-Riemannian metric learning.
Pattern Recognition, 48(10):3113–3124, 2015.

Bruno Iannazzo and Margherita Porcelli.
The Riemannian Barzilai-Borwein method with nonmonotone line-search and the Karcher mean computation.
Optimization online, December, 2015.

B. Jeuris, R. Vandebril, and B. Vandereycken.
A survey and comparison of contemporary algorithms for computing the matrix geometric mean.
Electronic Transactions on Numerical Analysis, 39:379–402, 2012.

H. Karcher.
Riemannian center of mass and mollifier smoothing.
Communications on Pure and Applied Mathematics, 1977.

Emmanuel K Kalunga, Sylvain Chevallier, Quentin Barthélemy, Karim Djouani, Yskandar Hamam, and Eric Monacelli.
From Euclidean to Riemannian means: Information geometry for SSVEP classification.
In *International Conference on Networked Geometric Science of Information*, pages 595–604. Springer, 2015.

J. Lawson and Y. Lim.
Monotonic properties of the least squares mean.
Mathematische Annalen, 351(2):267–279, 2011.

References III

Xiaodong Li, Shuyang Ling, Thomas Strohmer, and Ke Wei.
Rapid, robust, and reliable blind deconvolution via nonconvex optimization.
Applied and Computational Harmonic Analysis, 2018.

K. Lee, Y. Wu, and Y. Bresler.
Near-Optimal Compressed Sensing of a Class of Sparse Low-Rank Matrices Via Sparse Power Factorization.
IEEE Transactions on Information Theory, 64(3):1666–1698, March 2018.

Jiwen Lu, Gang Wang, and Pierre Moulin.
Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning.
In *Proceedings of the IEEE International Conference on Computer Vision*, pages 329–336, 2013.

Estelle M Massart and Sylvain Chevallier.
Inductive means and sequences applied to online classication of EEG.
Technical report, ICTEAM Institute, Universite Catholiqué de Louvain, 2017.

Q. Rentmeesters and P.-A. Absil.
Algorithm comparison for Karcher mean computation of rotation matrices and diffusion tensors.
In *19th European Signal Processing Conference*, pages 2229–2233, Aug 2011.

Q. Rentmeesters.
Algorithms for data fitting on some common homogeneous spaces.
PhD thesis, Universite catholiqué de Louvain, 2013.

Y. Rathi, A. Tannenbaum, and O. Michailovich.
Segmenting images on the tensor manifold.
In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

References IV

Gregory Shakhnarovich, John W Fisher, and Trevor Darrell.
Face recognition from long-term observations.
In *European Conference on Computer Vision*, pages 851–865. Springer, 2002.

Oncel Tuzel, Fatih Porikli, and Peter Meer.
Region covariance: A fast descriptor for detection and classification.
In *European conference on computer vision*, pages 589–600. Springer, 2006.

S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang.
A leaf recognition algorithm for plant classification using probabilistic neural network.
2007 IEEE International Symposium on Signal Processing and Information Technology, pages 11–16, 2007.
arXiv:0707.4289v1.

Xinru Yuan, Wen Huang, P.-A. Absil, and K. A. Gallivan.
Computing the matrix geometric mean: Riemannian vs Euclidean conditioning, implementation techniques, and a Riemannian BFGS method.
Technical Report UCL-INMA-2019.05, U.C.Louvain, 2019.