Blind Deconvolution by Optimizing Over a Quotient Manifold

Wen Huang, Paul Hand

Rice University

May 7, 2018
SIAM Conference on Applied Linear Algebra

Outline

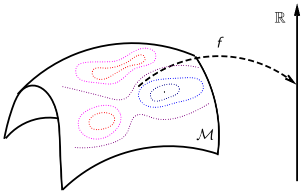
- Riemannian Optimization
- Blind Deconvolution
    - Problem Statement
    - Related Work
    - Riemannian Approach
    - Numerical Results

Riemannian Optimization

Problem: Given $f(x) : \mathcal{M} \to \mathbb{R}$,
solve

$$\min_{x \in \mathcal{M}} f(x)$$

where $\mathcal{M}$ is a Riemannian manifold.



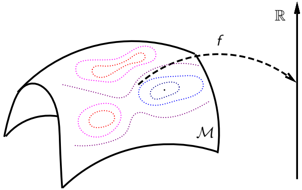
What is a Riemannian manifold? manifold + Riemannian metric

Riemannian Optimization

Problem: Given $f(x) : \mathcal{M} \to \mathbb{R}$, solve

$$\min_{x \in \mathcal{M}} f(x)$$

where $\mathcal{M}$ is a Riemannian manifold.



What is a Riemannian manifold? manifold + Riemannian metric

Examples of manifolds:

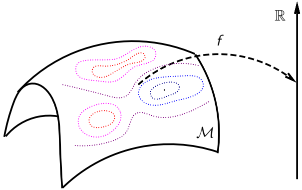
- $\{X \in \mathbb{R}^{n \times p} \mid X^T X = I_p\}$
- $\{X \in \mathbb{R}^{m \times n} \mid \mathrm{rank}(X) = p\}$
- $\{X \in \mathbb{R}^{n \times n} \mid X = X^T, X \succ 0\}$
- All p-dimensional linear subspaces of $\mathbb{R}^n$

Riemannian Optimization

Problem: Given $f(x) : \mathcal{M} \to \mathbb{R}$, solve

$$\min_{x \in \mathcal{M}} f(x)$$

where $\mathcal{M}$ is a Riemannian manifold.



What is a Riemannian manifold? manifold + Riemannian metric

Examples of manifolds:

- $\{X \in \mathbb{R}^{n \times p} \mid X^T X = I_p\}$
- $\{X \in \mathbb{R}^{m \times n} \mid \mathrm{rank}(X) = p\}$
- $\{X \in \mathbb{R}^{n \times n} \mid X = X^T, X \succ 0\}$
- All p-dimensional linear subspaces of $\mathbb{R}^n$

Riemannian metric:

- Inner product on tangent spaces
- Define angles and lengths

Iterations on the Manifold

Consider the following generic update for an iterative Euclidean optimization algorithm:

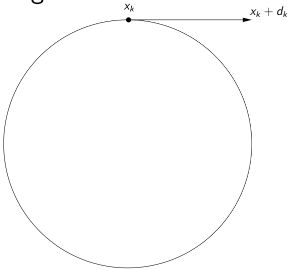
$$x_{k+1} = x_k + \Delta x_k = x_k + \alpha_k s_k \; .$$

This iteration is implemented in numerous ways, e.g.:

- Steepest descent: $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$
- Newton's method: $x_{k+1} = x_k - \left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k)$
- Trust region method: Δx_k is set by optimizing a local model.

Riemannian Manifolds Provide

- Riemannian concepts describing directions and movement on the manifold
- Riemannian analogues for gradient and Hessian



Retractions

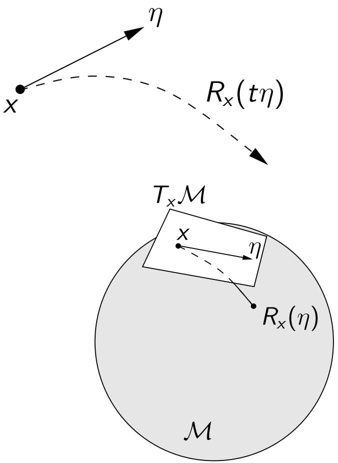
Euclidean	Riemannian
$x_{k+1} = x_k + \alpha_k d_k$	$x_{k+1} = R_{x_k}(\alpha_k \eta_k)$

Definition

A retraction is a mapping R from TM to M satisfying the following:

- R is continuously differentiable
- $R_x(0) = x$
- $\mathrm{D}\, R_x(0)[\eta] = \eta$

- maps tangent vectors back to the manifold
- defines curves in a direction



Categories of Riemannian optimization methods

Retraction-based: local information only

Line search-based: use local tangent vector and $R_x(t\eta)$ to define line

- Steepest decent
- Newton

Local model-based: series of flat space problems

- Riemannian trust region Newton (RTR)
- Riemannian adaptive cubic overestimation (RACO)

Categories of Riemannian optimization methods

Retraction and transport-based: information from multiple tangent spaces

- Nonlinear conjugate gradient: multiple tangent vectors
- Quasi-Newton e.g. Riemannian BFGS: transport operators between tangent spaces

Additional element required for optimizing a cost function (M, g):

- formulas for combining information from multiple tangent spaces.

Vector Transports

Vector Transport

- Vector transport: Transport a tangent vector from one tangent space to another
- $\mathcal{T}_{\eta_x}\xi_x$, denotes transport of ξ_x to tangent space of $R_x(\eta_x)$. R is a retraction associated with $\mathcal{T}$

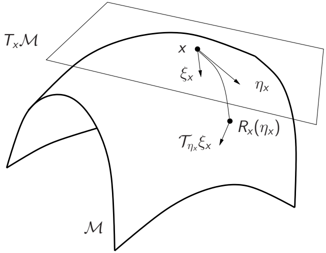


Figure: Vector transport.

Retraction/Transport-based Riemannian Optimization

Given a retraction and a vector transport, we can generalize many Euclidean methods to the Riemannian setting. Do the Riemannian versions of the methods work well?

Retraction/Transport-based Riemannian Optimization

Given a retraction and a vector transport, we can generalize many Euclidean methods to the Riemannian setting. Do the Riemannian versions of the methods work well?

No

- Lose many theoretical results and important properties;
- Impose restrictions on retraction/vector transport;

Retraction/Transport-based Riemannian Optimization

Elements required for optimizing a cost function (M, g):

- an representation for points x on M, for tangent spaces $T_x M$;
- choice of an inner products $g_x(\cdot, \cdot)$ on $T_x M$;
- choice of a retraction $R_x : T_x M \to M$;
- formulas for $f(x), \mathrm{grad}\ f(x)$ and $\mathrm{Hess}\ f(x)$ (or its action);

Blind deconvolution

[Blind deconvolution]

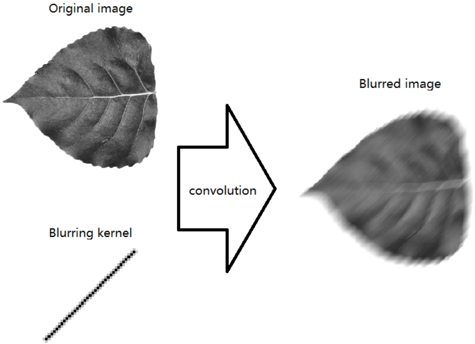
Blind deconvolution is to recover two unknown signals from their convolution.

Blurred image

Blind deconvolution

[Blind deconvolution]

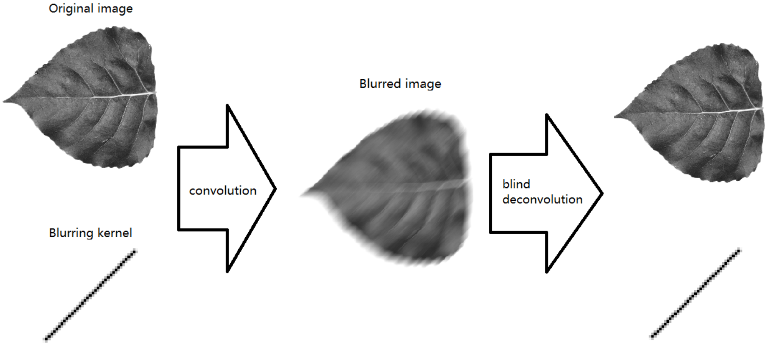
Blind deconvolution is to recover two unknown signals from their convolution.



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 $y = \mathbf{F}\mathbf{y}$, $w = \mathbf{F}\mathbf{w}$, and $x = \mathbf{F}\mathbf{x}$, 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][1]
  - 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 $X = hm^*$;

[1] 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][1]
  - 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;

[1] 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][1]
  - 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;

[1] 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. [LLSW16][2]
  - Nonconvex problem[3]:

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

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

[3] 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. [LLSW16][2]
  - Nonconvex problem[3]:

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

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

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

[3] 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. [LLSW16][2]
  - Nonconvex problem[3]:

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

  - (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.

---

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

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

Manifold Approach

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

- 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

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

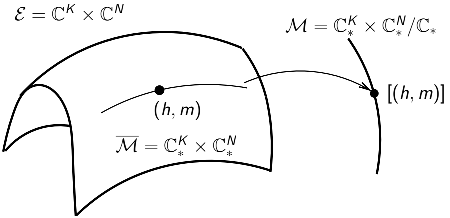
- The problem is defined on the set of rank-one matrices (denoted by $\mathbb{C}_1^{K \times N}$), neither $\mathbb{C}^{K \times N}$ nor $\mathbb{C}^K \times \mathbb{C}^N$; Why not work on the manifold directly?
- Optimization on manifolds: A Riemannian steepest descent method;
    - Representation of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

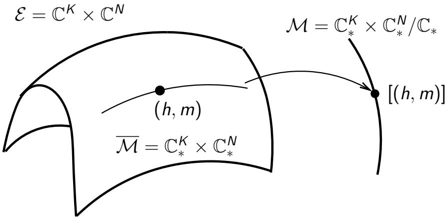
Cost function[4]

- Riemannian approach:

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

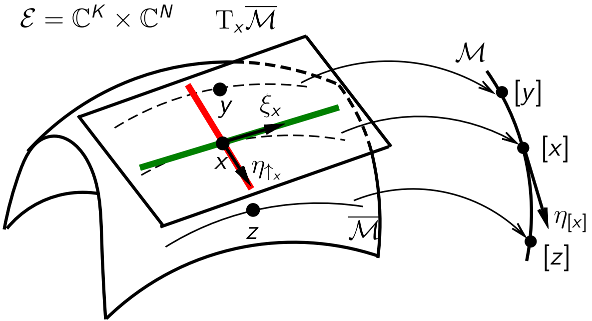
- Approach in [LLSW16]:

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



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

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



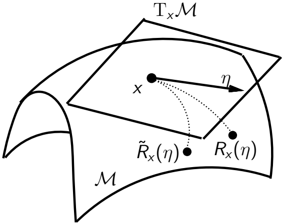
- x denotes (h, m);
- Green line: the tangent space of $[x]$;
- Red line (horizontal space at x): orthogonal to the green line;
- Horizontal space at x: a representation of the tangent space of $\mathcal{M}$ at $[x]$;

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

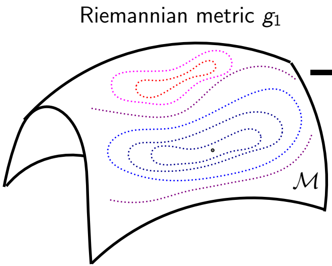


Figure: Changing metric may influence the difficulty of a problem.

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:

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

Riemannian gradient

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

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

A Riemannian steepest descent method (RSD)

An implementation of a Riemannian steepest descent method[5]

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.

[5]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[5]

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 [LLSW16]

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.

[5] 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[5]

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 [LLSW16]

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.

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

Penalty

Penalty term for (i) Riemannian method, (ii) Wirtinger flow [LLSW16]

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

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

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 $\|m\|_2$;

Penalty

Penalty term for (i) Riemannian method, (ii) Wirtinger flow [LLSW16]

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

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

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

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

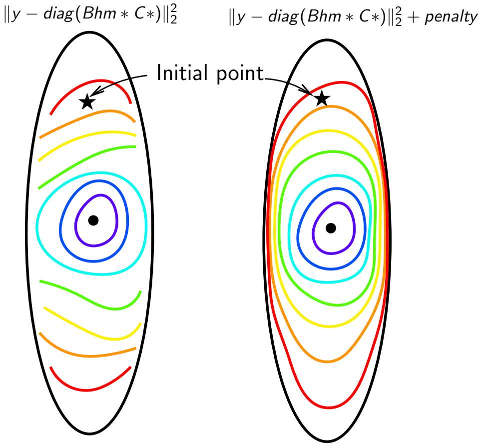
- Coherence at the true solution $[(h_\sharp, m_\sharp)]$
  - influences the probability of recovery
  - Small coherence is preferred

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 [LLSW16]

- $(d, \tilde{h}_0, \tilde{m}_0)$: 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

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

Table: Comparisons of efficiency

	$L = 400, K = N = 50$			$L = 600, K = N = 50$		
Algorithms	[LLSW16]	[LWB13]	R-SD	[LLSW16]	[LWB13]	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 $\frac{\|hm^* - h_\sharp m_\sharp^*\|_F}{\|h_\sharp\|_2 \|m_\sharp\|_2}$

[LLSW16]: X. Li et. al., Rapid, robust, and reliable blind deconvolution via nonconvex optimization, *preprint arXiv:1606.04933*, 2016
[LWB13]: 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 $\frac{\|hm^* - h_\sharp m_\sharp^*\|_F}{\|h_\sharp\|_2 \|m_\sharp\|_2} \leq 10^{-2}$

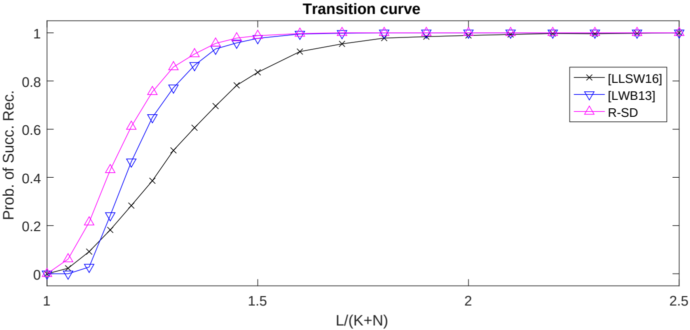
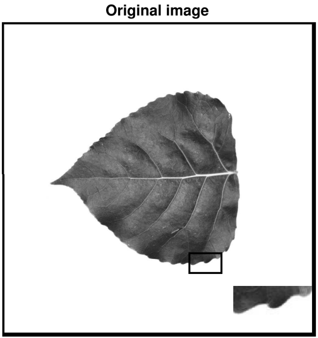


Figure: Empirical phase transition curves for 1000 random runs.

[LLSW16]: X. Li et. al., Rapid, robust, and reliable blind deconvolution via nonconvex optimization, *preprint arXiv:1606.04933*, 2016
[LWB13]: 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

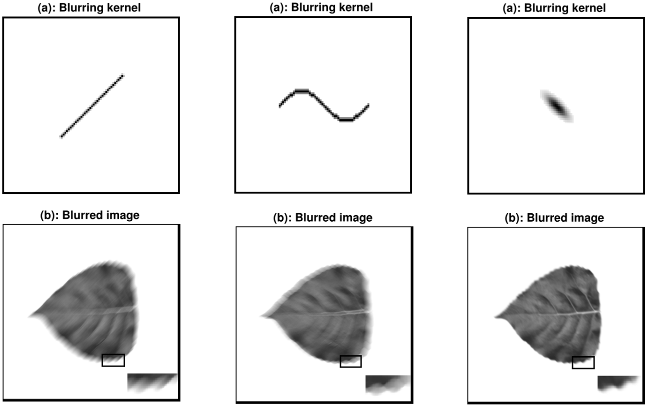


Figure: Left: Motion kernel by Matlab function "fspecial('motion', 50, 45)"; Middle: Kernel like function "sin"; Right: Gaussian kernel with covariance $[1, 0.8; 0.8, 1]$;

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

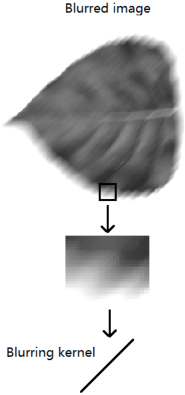


Image deblurring with various kernels

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

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

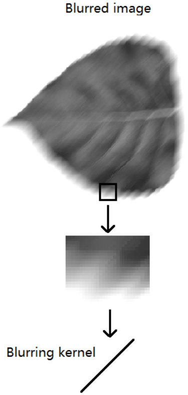


Image deblurring with various kernels

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

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**.

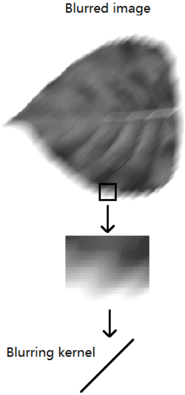


Image deblurring with various kernels

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

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**.

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

Blurred image

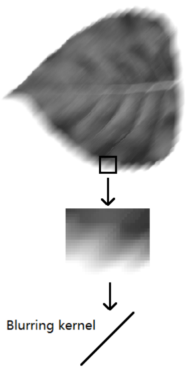


Image deblurring with various kernels

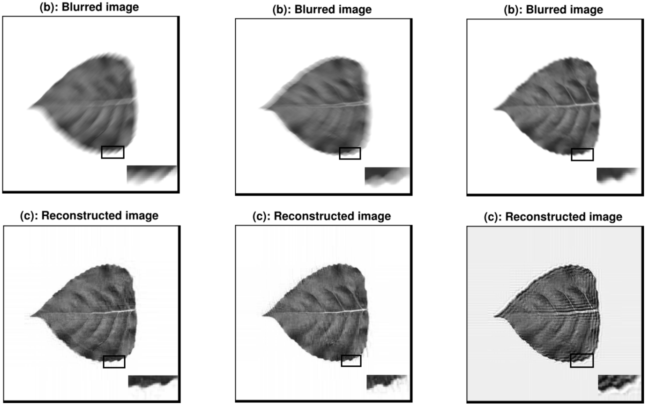


Figure: The number of iterations is 80; Computational times are about 48s; Relative errors $\left\| \hat{\mathbf{y}} - \frac{\|\mathbf{y}\|}{\|\mathbf{y}_f\|} \mathbf{y}_f \right\| / \|\hat{\mathbf{y}}\|$ are 0.038, 0.040, and 0.089 from left to right.

Image deblurring with unknown supports

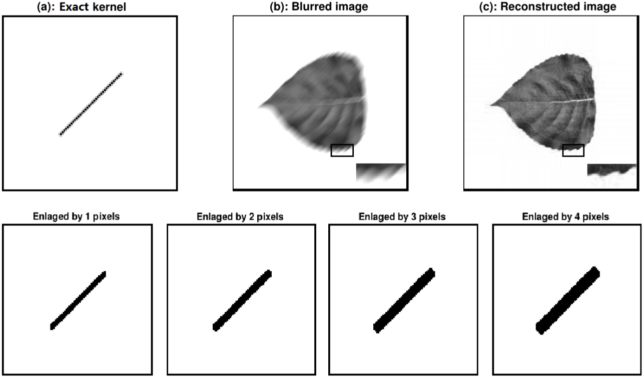


Figure: Top: reconstructed image using the exact support; Bottom: estimated supports with the numbers of nonzero entries: $K_1 = 183$, $K_2 = 265$, $K_3 = 351$, and $K_4 = 441$;

Image deblurring with inexact supports

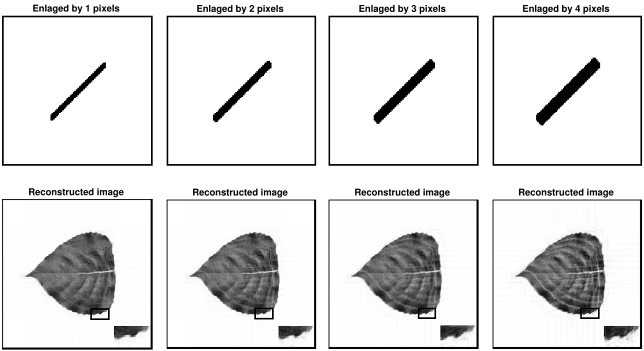


Figure: Relative errors $\left\| \hat{\mathbf{y}} - \frac{\|\mathbf{y}\|}{\|\mathbf{y}_f\|} \mathbf{y}_f \right\| / \|\hat{\mathbf{y}}\|$ are 0.044, 0.048, 0.052, and 0.067 from left to right.

Summary

- 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 [LLSW16]
- RSD method works well for the tested imaging debluring problems

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.

Xiaodong Li, Shuyang Ling, Thomas Strohmer, and Ke Wei.
Rapid, robust, and reliable blind deconvolution via nonconvex optimization.
CoRR, abs/1606.04933, 2016.

K. Lee, Y. Wu, and Y. Bresler.
Near Optimal Compressed Sensing of a Class of Sparse Low-Rank Matrices via Sparse Power Factorization.
pages 1–80, 2013.

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.