# Riemannian Optimization with its Application to Blind Deconvolution Problem

Wen Huang

Rice University

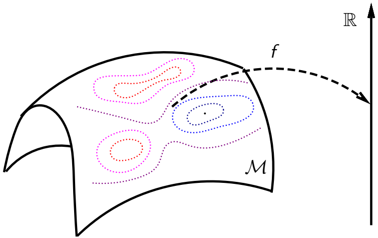
April 23, 2018

# Riemannian Optimization

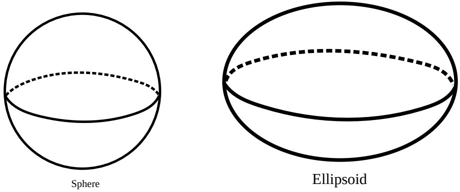
**Problem:** Given $f(x) : \mathcal{M} \to \mathbb{R}$, solve

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

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



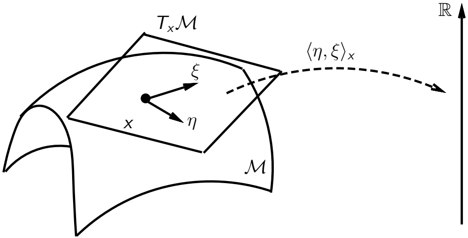
# Examples of Manifolds




- Stiefel manifold: $\mathrm{St}(p, n) = \{X \in \mathbb{R}^{n \times p} | X^T X = I_p\}$
- Grassmann manifold: Set of all $p$-dimensional subspaces of $\mathbb{R}^n$
- Set of fixed rank $m$-by-$n$ matrices
- And many more

# Riemannian Manifolds

Roughly, a Riemannian manifold $\mathcal{M}$ is a smooth set with a smoothly-varying inner product on the tangent spaces.

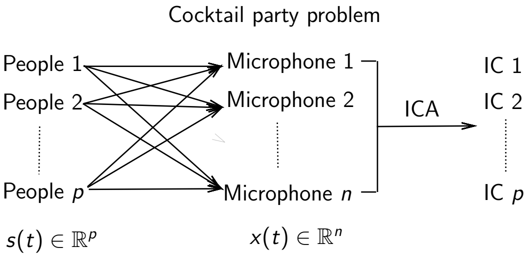


# Applications

Three applications are used to demonstrate the importance of the Riemannian optimization:

- Independent component analysis [CS93]
- Matrix completion problem [Van13, HAGH16]
- Elastic shape analysis of curves [SKJJ11, HGSA15]

# Application: Independent Component Analysis




- Observed signal is $x(t) = As(t)$
- One approach:
  - Assumption: $E\{s(t)s(t+\tau)\}$ is diagonal for all $\tau$
  - $C_\tau(x) := E\{x(t)x(x+\tau)^T\} = AE\{s(t)s(t+\tau)^T\}A^T$

# Application: Independent Component Analysis

- Minimize joint diagonalization cost function on the Stiefel manifold [TI06]:

$$f : \mathrm{St}(p, n) \to \mathbb{R} : V \mapsto \sum_{i=1}^{N} \|V^T C_i V - \mathrm{diag}(V^T C_i V)\|_F^2.$$

- $C_1, \ldots, C_N$ are covariance matrices and $\mathrm{St}(p, n) = \{X \in \mathbb{R}^{n \times p} | X^T X = I_p\}$.

# Application: Matrix Completion Problem

Matrix completion problem

$$
\begin{array}{c}
\begin{array}{cccccc}
\text{Movie } 1 & \text{Movie } 2 & & & & \text{Movie } n
\end{array}
\\
\begin{array}{c}
\text{User } 1 \\ \text{User } 2 \\ \\ \text{User } m
\end{array}
\begin{pmatrix}
 & 1 & & & 4 & \\
3 & 5 & & & 4 & \\
 & & 5 & 1 & & \\
 & 2 & & & 5 & 3
\end{pmatrix}
\end{array}
$$

Rate matrix $M$

- The matrix $M$ is sparse
- The goal: complete the matrix $M$

# Application: Matrix Completion Problem

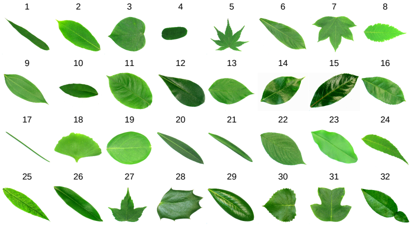
$$
\begin{array}{c}
\text{movies} \\
\begin{pmatrix}
a_{11} & & & a_{14} \\
 & & & a_{24} \\
 & & a_{33} & \\
a_{41} & & & \\
 & a_{52} & a_{53} &
\end{pmatrix}
\end{array}
=
\begin{array}{c}
\text{meta-user} \\
\begin{pmatrix}
b_{11} & b_{12} \\
b_{21} & b_{22} \\
b_{31} & b_{32} \\
b_{41} & b_{42} \\
b_{51} & b_{52}
\end{pmatrix}
\end{array}
\begin{array}{c}
\text{meta-movie} \\
\begin{pmatrix}
c_{11} & c_{12} & c_{13} & c_{14} \\
c_{21} & c_{22} & c_{23} & c_{24}
\end{pmatrix}
\end{array}
$$

- Minimize the cost function

$$f : \mathbb{R}_r^{m \times n} \to \mathbb{R} : X \mapsto f(X) = \|P_\Omega M - P_\Omega X\|_F^2.$$

- $\mathbb{R}_r^{m \times n}$ is the set of $m$-by-$n$ matrices with rank $r$. It is known to be a Riemannian manifold.

# Application: Elastic Shape Analysis of Curves




- Classification [LKS+12, HGSA15]
- Face recognition [DBS+13]

# Application: Elastic Shape Analysis of Curves

- Elastic shape analysis invariants:
    - Rescaling
    - Translation
    - Rotation
    - Reparametrization
- The shape space is a quotient space

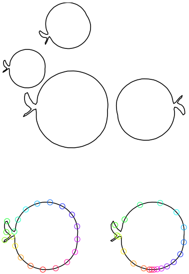
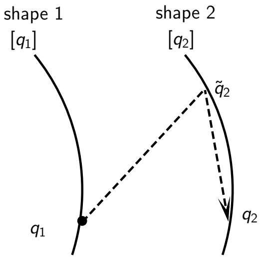


Figure: All are the same shape.

# Application: Elastic Shape Analysis of Curves



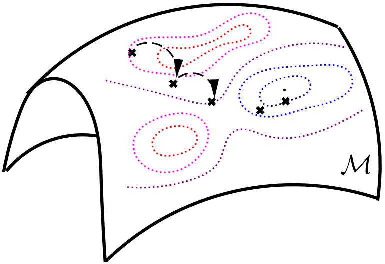

- Optimization problem $\min_{q_2 \in [q_2]} \mathrm{dist}(q_1, q_2)$ is defined on a Riemannian manifold
- Computation of a geodesic between two shapes
- Computation of Karcher mean of a population of shapes

# More Applications

- Role model extraction [MHB+16]
- Computations on SPD matrices [YHAG17]
- Phase retrieval problem [HGZ17]
- Blind deconvolution [HH17]
- Synchronization of rotations [Hua13]
- Computations on low-rank tensor
- Low-rank approximate solution for Lyapunov equation

# Comparison with Constrained Optimization

- All iterates on the manifold
- Convergence properties of unconstrained optimization algorithms
- No need to consider Lagrange multipliers or penalty functions
- Exploit the structure of the constrained set



# Iterations on the Manifold

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

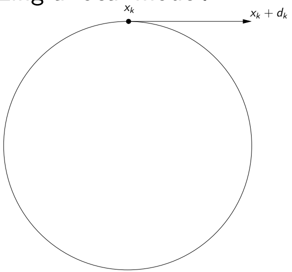
$$x_{k+1} = x_k + \Delta x_k = x_k + \alpha_k s_k \ .$$

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: $\Delta x_k$ 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



# Riemannian gradient and Riemannian Hessian

## Definition

The Riemannian gradient of $f$ at $x$ is the unique tangent vector in $T_x M$ satisfying $\forall \eta \in T_x M$, the directional derivative

$$\mathrm{D} f(x)[\eta] = \langle \operatorname{grad} f(x), \eta \rangle$$

and $\operatorname{grad} f(x)$ is the direction of steepest ascent.

## Definition

The Riemannian Hessian of $f$ at $x$ is a symmetric linear operator from $T_x M$ to $T_x M$ defined as

$$\operatorname{Hess} f(x) : T_x M \to T_x M : \eta \to \nabla_\eta \operatorname{grad} f,$$

where $\nabla$ is the affine connection.

# Retractions

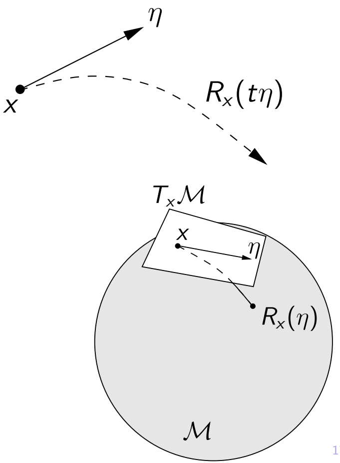
| Euclidean | Riemannian |
|---|---|
| $x_{k+1} = x_k + \alpha_k d_k$ | $x_{k+1} = R_{x_k}(\alpha_k \eta_k)$ |

## 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 $\xi_x$ to tangent space of $R_x(\eta_x)$. $R$ is a retraction associated with $\mathcal{T}$

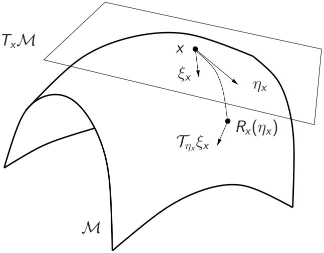



Figure: Vector transport.

# 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

## Benefits

- Increased generality does not compromise the important theory
- Less expensive than or similar to previous approaches
- May provide theory to explain behavior of algorithms specifically developed for a particular application – or closely related ones

## Possible Problems

- May be inefficient compared to algorithms that exploit application details

# Some History of Optimization On Manifolds (I)

Luenberger (1973), *Introduction to linear and nonlinear programming*. Luenberger mentions the idea of performing line search along geodesics, "which we would use if it were computationally feasible (which it definitely is not)". Rosen (1961) essentially anticipated this but was not explicit in his Gradient Projection Algorithm.

Gabay (1982), *Minimizing a differentiable function over a differential manifold*. Steepest descent along geodesics; Newton's method along geodesics; Quasi-Newton methods along geodesics. On Riemannian submanifolds of $\mathbb{R}^n$.

Smith (1993-94), *Optimization techniques on Riemannian manifolds*. Levi-Civita connection $\nabla$; Riemannian exponential mapping; parallel translation.

# Some History of Optimization On Manifolds (II)

The "pragmatic era" begins:

Manton (2002), *Optimization algorithms exploiting unitary constraints* "The present paper breaks with tradition by not moving along geodesics". The geodesic update $\mathrm{Exp}_x \eta$ is replaced by a projective update $\pi(x+\eta)$, the *projection* of the point $x+\eta$ onto the manifold.

Adler, Dedieu, Shub, et al. (2002), *Newton's method on Riemannian manifolds and a geometric model for the human spine*. The exponential update is relaxed to the general notion of *retraction*. The geodesic can be replaced by any (smoothly prescribed) curve tangent to the search direction.

Absil, Mahony, Sepulchre (2007) Nonlinear conjugate gradient using retractions.

# Some History of Optimization On Manifolds (III)

Theory, efficiency, and library design improve dramatically:

Absil, Baker, Gallivan (2004-07), Theory and implementations of Riemannian Trust Region method. Retraction-based approach. Matrix manifold problems, software repository

`http://www.math.fsu.edu/~cbaker/GenRTR`

Anasazi Eigenproblem package in Trilinos Library at Sandia National Laboratory

Absil, Gallivan, Qi (2007-10), Basic theory and implementations of Riemannian BFGS and Riemannian Adaptive Cubic Overestimation. Parallel translation and Exponential map theory, Retraction and vector transport empirical evidence.

# Some History of Optimization On Manifolds (IV)

Ring and With (2012), combination of differentiated retraction and isometric vector transport for convergence analysis of RBFGS

Absil, Gallivan, Huang (2009-2017), Complete theory of Riemannian Quasi-Newton and related transport/retraction conditions, Riemannian SR1 with trust-region, RBFGS on partly smooth problems, A C++ library: `http://www.math.fsu.edu/~whuang2/ROPTLIB`

Sato, Iwai (2013-2015), Zhu (2017), Global convergence analysis for Riemannian conjugate gradient methods

Bonnabel (2011), Sato, Kasai, Mishra(2017) Riemannian stochastic gradient descent method.

Many people Application interests increase noticeably

# Current UCL/FSU Methods

- Riemannian Steepest Descent [AMS08]
- Riemannian conjugate gradient [AMS08]
- Riemannian Trust Region Newton [ABG07]: global, quadratic convergence
- Riemannian Broyden Family [HGA15, HAG18] : global (convex), superlinear convergence
- Riemannian Trust Region SR1 [HAG15]: global, $(d+1)-$superlinear convergence
- For large problems
  - Limited memory RTRSR1
  - Limited memory RBFGS

# Current UCL/FSU Methods

Riemannian manifold optimization library (ROPTLIB) is used to optimize a function on a manifold.

- Most state-of-the-art methods;
- Commonly-encountered manifolds;
- Written in C++;
- Interfaces with Matlab, Julia and R;
- BLAS and LAPACK;
- `www.math.fsu.edu/~whuang2/Indices/index_ROPTLIB.html`

# Current/Future Work on Riemannian methods

- Manifold and inequality constraints
- Discretization of infinite dimensional manifolds and the convergence/accuracy of the approximate minimizers – specific to a problem and extracting general conclusions
- Partly smooth cost functions on Riemannian manifold
- Limited-memory quasi-Newton methods on manifolds

# Blind deconvolution

## [Blind deconvolution]

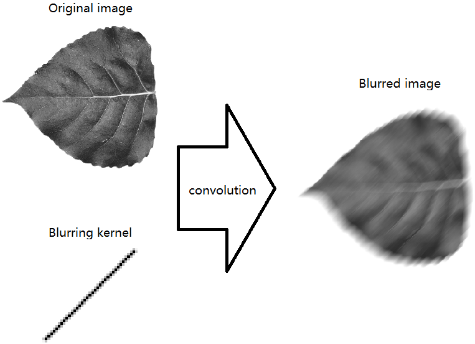
Blind deconvolution is to recover two unknown signals from their convolution.

Blurred image

# Blind deconvolution

## [Blind deconvolution]

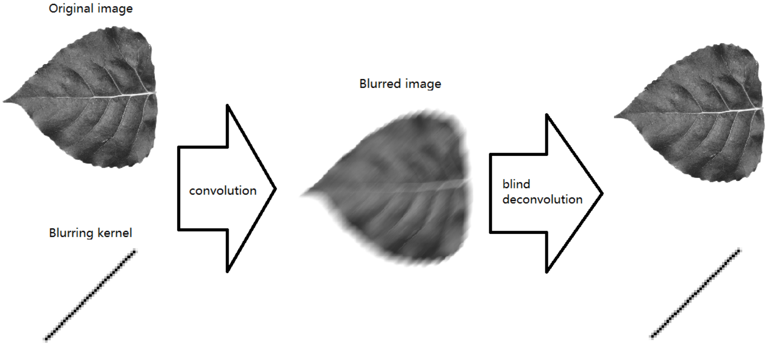
Blind deconvolution is to recover two unknown signals from their convolution.



# 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 $\xlongequal{\text{high probability}}$ 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

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

- 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

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

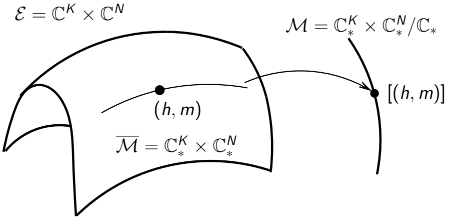
- The problem is defined on the set of rank-one matrices (denoted by $\mathbb{C}_1^{K \times N}$), neither $\mathbb{C}^{K \times N}$ nor $\mathbb{C}^K \times \mathbb{C}^N$; Why not work on the manifold directly?
- Optimization on manifolds: A Riemannian steepest descent method;
  - Representation of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

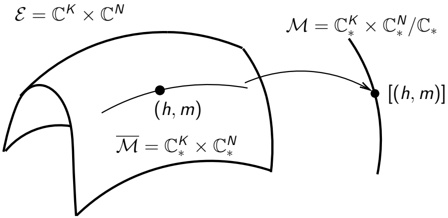
Cost function[4]

- Riemannian approach:

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

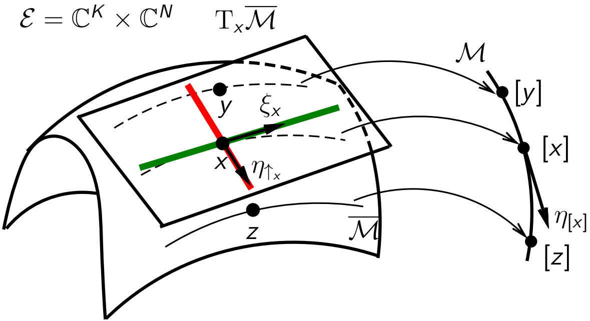
- Approach in [LLSW16]:

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




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

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



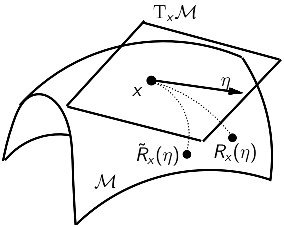

- $x$ denotes $(h, m)$;
- Green line: the tangent space of $[x]$;
- Red line (horizontal space at $x$): orthogonal to the green line;
- Horizontal space at $x$: a representation of the tangent space of $\mathcal{M}$ at $[x]$;

# 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

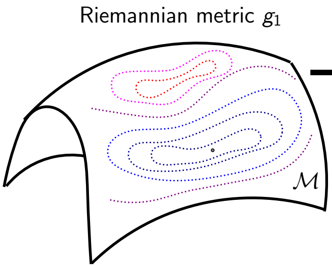



Figure: Changing metric may influence the difficulty of a problem.

# 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

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

## 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:

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

# 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

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

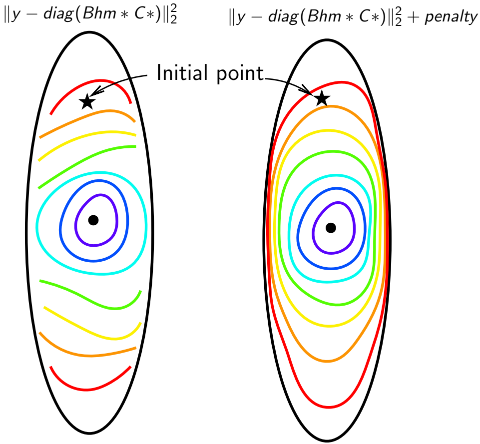
- Coherence at the true solution $[(h_\sharp, m_\sharp)]$
  - influences the probability of recovery
  - Small coherence is preferred

# 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 unknown supports

# Synthetic tests

- The matrix $B$ is the first $K$ column of the unitary DFT matrix;
- The matrix $C$ is a Gaussian random matrix;
- The measurement $y = \mathrm{diag}(B h_\sharp m_\sharp^* C^*)$, where entries in the true solution $(h_\sharp, m_\sharp)$ are drawn from Gaussian distribution;
- All tested algorithms use the same initial point;
- Stop when $\|y - \mathrm{diag}(Bhm^* C^*)\|_2 / \|y\|_2 \leq 10^{-8}$;

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

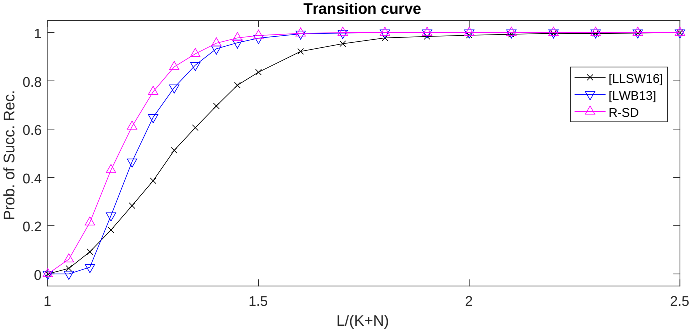


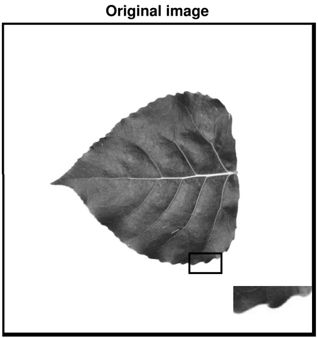

Figure: Empirical phase transition curves for 1000 random runs.

---

[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

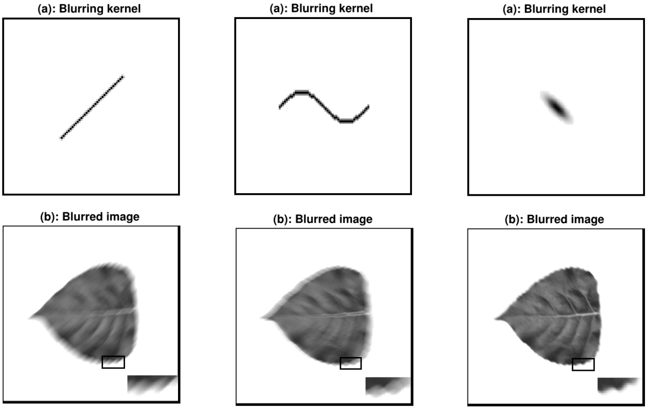


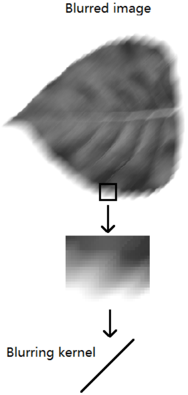

Figure: Left: Motion kernel by Matlab function "fspecial('motion', 50, 45)"; Middle: Kernel like function "sin"; Right: Gaussian kernel with covariance $[1, 0.8; 0.8, 1]$;

# Image deblurring with various kernels

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

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

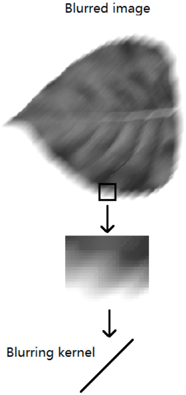
$$\boxed{\min \|y - \mathrm{diag}(Bhm^* C^*)\|_2^2}$$

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

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

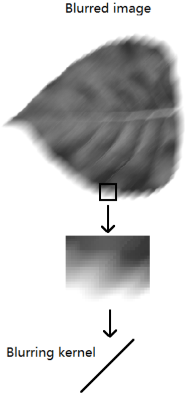
What subspaces are the two unknown signals in?

- 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

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

What subspaces are the two unknown signals in?

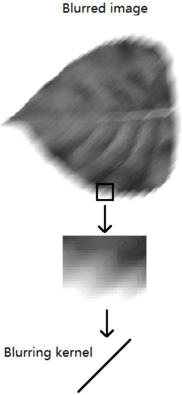
- Image is approximately sparse in the Haar wavelet basis

  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

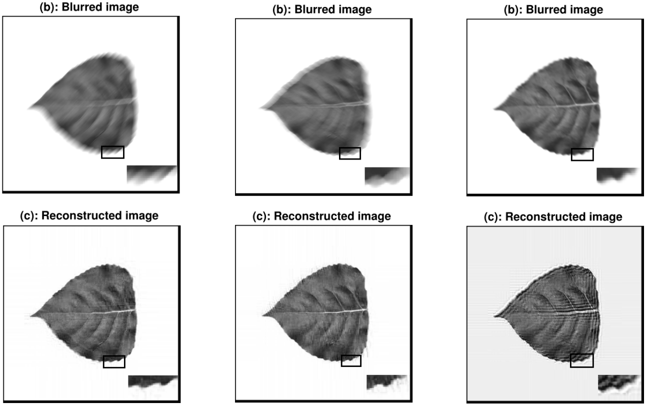



Figure: The number of iterations is 80; Computational times are about 48s; Relative errors $\left\|\hat{\mathbf{y}} - \frac{\|\mathbf{y}\|}{\|\mathbf{y}_f\|}\mathbf{y}_f\right\| / \|\hat{\mathbf{y}}\|$ are 0.038, 0.040, and 0.089 from left to right.

# Image deblurring with unknown supports

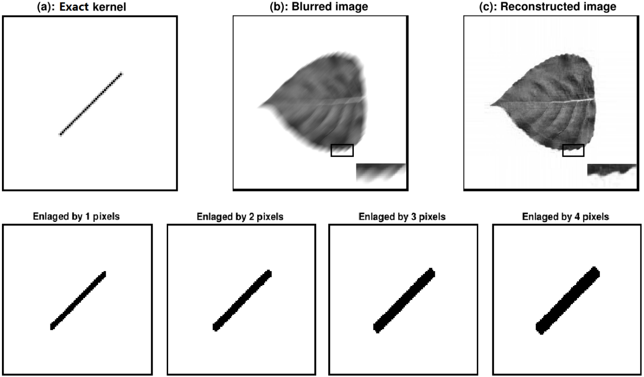



Figure: Top: reconstructed image using the exact support; Bottom: estimated supports with the numbers of nonzero entries: $K_1 = 183$, $K_2 = 265$, $K_3 = 351$, and $K_4 = 441$;

# Image deblurring with unknown supports

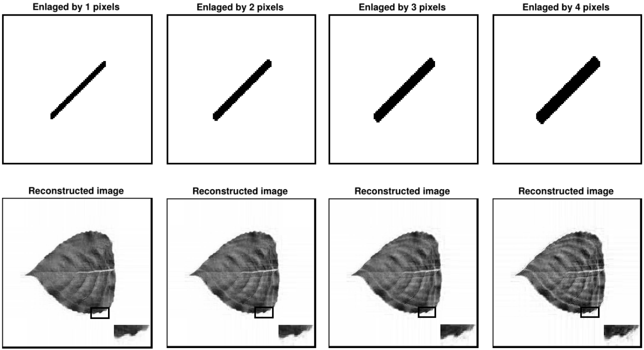



Figure: Relative errors $\left\|\hat{\mathbf{y}} - \frac{\|\mathbf{y}\|}{\|\mathbf{y}_f\|}\mathbf{y}_f\right\| / \|\hat{\mathbf{y}}\|$ are 0.044, 0.048, 0.052, and 0.067 from left to right.

# Summary

- Introduced the framework of Riemannian optimization
- Used applications to show the importance of Riemannian optimization
- Briefly reviewed the history of Riemannian optimization
- Introduced the blind deconvolution problem
- Reviewed related work
- Introduced a Riemannian steepest descent method
- Demonstrated the performance of the Riemannian steepest descent method

# Thank you

Thank you!

# References I


P.-A. Absil, C. G. Baker, and K. A. Gallivan.
Trust-region methods on Riemannian manifolds.
*Foundations of Computational Mathematics*, 7(3):303–330, 2007.

P.-A. Absil, R. Mahony, and R. Sepulchre.
*Optimization algorithms on matrix manifolds*.
Princeton University Press, Princeton, NJ, 2008.

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

J. F. Cardoso and A. Souloumiac.
Blind beamforming for non-gaussian signals.
*IEE Proceedings F Radar and Signal Processing*, 140(6):362, 1993.

H. Drira, B. Ben Amor, A. Srivastava, M. Daoudi, and R. Slama.
3D face recognition under expressions, occlusions, and pose variations.
*Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(9):2270–2283, 2013.

W. Huang, P.-A. Absil, and K. A. Gallivan.
A Riemannian symmetric rank-one trust-region method.
*Mathematical Programming*, 150(2):179–216, February 2015.

# References II


Wen Huang, P.-A. Absil, and K. A. Gallivan.
A Riemannian BFGS Method without Differentiated Retraction for Nonconvex Optimization Problems.
*SIAM Journal on Optimization*, 28(1):470–495, 2018.

Wen Huang, P.-A. Absil, K. A. Gallivan, and Paul Hand.
ROPTLIB: an object-oriented C++ library for optimization on Riemannian manifolds.
Technical Report FSU16-14, Florida State University, 2016.

W. Huang, K. A. Gallivan, and P.-A. Absil.
A Broyden Class of Quasi-Newton Methods for Riemannian Optimization.
*SIAM Journal on Optimization*, 25(3):1660–1685, 2015.

W. Huang, K. A. Gallivan, Anuj Srivastava, and P.-A. Absil.
Riemannian optimization for registration of curves in elastic shape analysis.
*Journal of Mathematical Imaging and Vision*, 54(3):320–343, 2015.
DOI:10.1007/s10851-015-0606-8.

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.

W. Huang and P. Hand.
Blind deconvolution by a steepest descent algorithm on a quotient manifold, 2017.
In preparation.

# References III


W. Huang.
*Optimization algorithms on Riemannian manifolds with applications*.
PhD thesis, Florida State University, Department of Mathematics, 2013.

H. Laga, S. Kurtek, A. Srivastava, M. Golzarian, and S. J. Miklavcic.
A Riemannian elastic metric for shape-based plant leaf classification.
*2012 International Conference on Digital Image Computing Techniques and Applications (DICTA)*, pages 1–7, December 2012.
doi:10.1109/DICTA.2012.6411702.

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.

Melissa Marchand, Wen Huang, Arnaud Browet, Paul Van Dooren, and Kyle A. Gallivan.
A riemannian optimization approach for role model extraction.
In *Proceedings of the 22nd International Symposium on Mathematical Theory of Networks and Systems*, pages 58–64, 2016.

A. Srivastava, E. Klassen, S. H. Joshi, and I. H. Jermyn.
Shape analysis of elastic curves in Euclidean spaces.
*IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1415–1428, September 2011.
doi:10.1109/TPAMI.2010.184.

# References IV

F. J. Theis and Y. Inouye.
On the use of joint diagonalization in blind signal processing.
*2006 IEEE International Symposium on Circuits and Systems*, (2):7–10, 2006.

B. Vandereycken.
Low-rank matrix completion by Riemannian optimization—extended version.
*SIAM Journal on Optimization*, 23(2):1214–1236, 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.

Xinru Yuan, Wen Huang, P.-A. Absil, and K. A. Gallivan.
A Riemannian quasi-newton method for computing the Karcher mean of symmetric positive definite matrices.
Technical Report FSU17-02, Florida State University, 2017.
www.math.fsu.edu/ whuang2/papers/RMKMSPDM.htm.