Riemannian Optimization with its Application to Blind Deconvolution Problem

Wen Huang
Rice University
April 23, 2018
**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**: $\text{St}(p, n) = \{ X \in \mathbb{R}^{n \times p} | X^T X = I_p \}$
- **Grassmann manifold**: Set of all $p$-dimensional subspaces of $\mathbb{R}^n$
- Set of fixed rank $m$-by-$n$ matrices
- And many more
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

Cocktail party problem

People 1 → Microphone 1 → IC 1
People 2 → Microphone 1 → IC 1
People 2 → Microphone 2 → IC 2
People p → Microphone n → IC p

s(t) ∈ ℝ^p  x(t) ∈ ℝ^n

- Observed signal is x(t) = As(t)
- One approach:
  - Assumption: E{s(t)s(t + τ)} is diagonal for all τ
  - Cτ(x) := E{x(t)x(x + τ)^T} = AE{s(t)s(t + τ)^T}A^T
Application: Independent Component Analysis

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

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

- \( C_1, \ldots, C_N \) are covariance matrices and
  \( \text{St}(p, n) = \{ X \in \mathbb{R}^{n\times p} | X^T X = I_p \} \).
**Application: Matrix Completion Problem**

Matrix completion problem

<table>
<thead>
<tr>
<th></th>
<th>Movie 1</th>
<th>Movie 2</th>
<th>Movie n</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>1</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>User 2</td>
<td>3</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>User m</td>
<td>2</td>
<td></td>
<td>5 3</td>
</tr>
</tbody>
</table>

Rate matrix $M$

- The matrix $M$ is sparse
- The goal: complete the matrix $M$
Application: Matrix Completion Problem

\[
\begin{pmatrix}
a_{11} & a_{14} \\
a_{24} & a_{33} \\
a_{41} \\
a_{52} & a_{53}
\end{pmatrix}
= 
\begin{pmatrix}
b_{11} & b_{12} \\
b_{21} & b_{22} \\
b_{31} & b_{32} \\
b_{41} & b_{42} \\
b_{51} & b_{52}
\end{pmatrix}
\begin{pmatrix}
c_{11} & c_{12} & c_{13} & c_{14} \\
c_{21} & c_{22} & c_{23} & c_{24}
\end{pmatrix}
\]

- Minimize the cost function

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

- $\mathbb{R}^{m \times n}_r$ 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 $\min_{q_2 \in [q_2]} \text{dist}(q_1, q_2)$ 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 $T_xM$ satisfying $\forall \eta \in T_xM$, the directional derivative

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

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

Definition

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

$$\text{Hess} f(x) : T_xM \rightarrow T_xM : \eta \rightarrow \nabla_\eta \text{grad} f,$$

where $\nabla$ is the affine connection.
Retractions

<table>
<thead>
<tr>
<th>Euclidean</th>
<th>Riemannian</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{k+1} = x_k + \alpha_k d_k$</td>
<td>$x_{k+1} = R_{x_k}(\alpha_k \eta_k)$</td>
</tr>
</tbody>
</table>

**Definition**

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

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

- 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 $\mathbb{R}^n$.

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 $\exp_x \eta$ 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:


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

Anasazi Eigenproblem package in Trilinos Library at Sandia National Laboratory

Some History of Optimization On Manifolds (IV)

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


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
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](http://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 is to recover two unknown signals from their convolution.
Blind deconvolution is to recover two unknown signals from their convolution.
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 \( w \in \mathbb{C}^L \) and \( x \in \mathbb{C}^L \) from their convolution \( y = w \ast x \in \mathbb{C}^L \).

- We only consider circular convolution:

\[
\begin{bmatrix}
y_1 \\
y_2 \\
y_3 \\
\vdots \\
y_L
\end{bmatrix} =
\begin{bmatrix}
w_1 & w_L & w_{L-1} & \ldots & w_2 \\
w_2 & w_1 & w_L & \ldots & w_3 \\
w_3 & w_2 & w_1 & \ldots & w_4 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
w_L & w_{L-1} & w_{L-2} & \ldots & w_1
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
\vdots \\
x_L
\end{bmatrix}
\]

- Let \( y = Fy \), \( w = Fw \), and \( x =Fx \), where \( F \) 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 $x = Cm$, $B \in \mathbb{C}^{L \times K}$ 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 \( x = Cm \), \( B \in \mathbb{C}^{L \times K} \) 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 \( x = \bar{C}m \), \( B \in \mathbb{C}^{L \times K} \) 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

\[
y = Bh \odot \bar{C}m = \text{diag}(Bhm^*C^*).
\]
Related work

- Ahmed et al. [ARR14]\(^1\)
  - Convex problem:
    \[
    \min_{X \in \mathbb{C}^{K \times N}} \|X\|_n, \text{ s. t. } y = \text{diag}(B hm^* C^*),
    \]
    where \(\|\cdot\|_n\) denotes the nuclear norm, and \(X = hm^*\);

Related work

- Ahmed et al. [ARR14]\(^1\)
  - Convex problem:
    \[
    \min_{X \in \mathbb{C}^{K \times N}} \|X\|_n, \text{ s. t. } y = \text{diag}(BXC^*),
    \]

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

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

---

Related work

- Ahmed et al. [ARR14]\(^1\)
  - Convex problem:
    \[
    \min_{X \in \mathbb{C}^{K \times N}} \|X\|_n, \text{ s. t. } y = \text{diag}(BXC^*),
    \]
    where \(\|\cdot\|_n\) denotes the nuclear norm, and \(X = hm^*\);
  - (Theoretical result): the unique minimizer \(\text{high probability}\) the true solution;
  - The convex problem is expensive to solve;

Related work

- Li et al. [LLSW16]$^2$
  - Nonconvex problem$^3$: 

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

---

$^3$The penalty in the cost function is not added for simplicity
Related work

- Li et al. [LLSW16]²
  - Nonconvex problem³:
    \[
    \min_{(h, m) \in \mathbb{C}^K \times \mathbb{C}^N} \|y - \text{diag}(Bhm^* C^*)\|^2_2;
    \]

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

³The penalty in the cost function is not added for simplicity
Related work

- Li et al. [LLSW16]²
  - Nonconvex problem³:
    $$\min_{(h,m) \in \mathbb{C}^K \times \mathbb{C}^N} \|y - \text{diag}(Bhm^* C^*)\|_2^2;$$

- (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.

---


³The penalty in the cost function is not added for simplicity
Manifold Approach

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?

Find $h, m$, s. t. $y = \text{diag}(Bhm^* C^*)$;
Problem Statement and Methods

Manifold Approach

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

- The problem is defined on the set of rank-one matrices (denoted by \( \mathbb{C}^{K \times N}_1 \)), 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
  - \( (\text{RSD} + \text{the good initialization}) \xrightarrow{\text{high probability}} \text{the true solution} \);
  - The Riemannian Hessian at the true solution is well-conditioned;
The problem is defined on the set of rank-one matrices (denoted by $\mathbb{C}^{K \times N}_1$), 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 $\mathbb{C}^{K \times N}_1$;
- Representation of directions (tangent vectors);
- Riemannian metric;
- Riemannian gradient;

Find $h, m$, s. t. $y = \text{diag}(Bhm^* C^*)$;
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: $[(h, m)] = \{(ha, ma^{-*}) | a \neq 0\}$;
- Quotient manifold: $\mathbb{C}^K \times \mathbb{C}^N / \mathbb{C}_* = \{[(h, m)] | (h, m) \in \mathbb{C}^K \times \mathbb{C}^N\}$
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: $\mathbb{C}_*^K \times \mathbb{C}_*^N / \mathbb{C}_* = \{[(h, m)] \mid (h, m) \in \mathbb{C}_*^K \times \mathbb{C}_*^N\}$

\[ \mathcal{E} = \mathbb{C}^K \times \mathbb{C}^N \]

\[ \mathcal{M} = \mathbb{C}_*^K \times \mathbb{C}_*^N / \mathbb{C}_* \]

\[ \mathcal{M} = \mathbb{C}_*^K \times \mathbb{C}_*^N \]

\[ \mathbb{C}_*^K \times \mathbb{C}_*^N / \mathbb{C}_* \cong \mathbb{C}_1^{K \times N} \]
A Representation of $\mathbb{C}^{K \times N}_1: \mathbb{C}^*_K \times \mathbb{C}^*_N / \mathbb{C}_*$

Cost function\(^4\)

- **Riemannian approach:**
  \[
  f : \mathbb{C}^*_K \times \mathbb{C}^*_N / \mathbb{C}_* \to \mathbb{R} : [(h, m)] \mapsto \|y - \text{diag}(Bhm^{*}C^{*})\|_2^2.
  \]

- **Approach in [LLSW16]:**
  \[
  f : \mathbb{C}^*_K \times \mathbb{C}^*_N \to \mathbb{R} : (h, m) \mapsto \|y - \text{diag}(Bhm^{*}C^{*})\|_2^2.
  \]

\(^4\)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 $[x]$;
Problem Statement and Methods

Retraction

<table>
<thead>
<tr>
<th>Euclidean</th>
<th>Riemannian</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_{k+1} = x_k + \alpha_k d_k )</td>
<td>( x_{k+1} = R_{x_k}(\alpha_k \eta_k) )</td>
</tr>
</tbody>
</table>

- Retraction: \( R : T\mathcal{M} \to \mathcal{M} \)
- \( R(0[x]) = [x] \)
- \( \frac{dR(t\eta[x])}{dt} \bigg|_{t=0} = \eta[x] \)
- Retraction on \( \mathbb{C}^K \times \mathbb{C}^N / \mathbb{C}^* \):
  \[
  R_{[(h,m)]}(\eta[(h,m)]) = [(h + \eta_h, m + \eta_m)].
  \]
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

Idea for choosing a Riemannian metric

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

\[
\min_{(h,m)} \| y - \text{diag}(B hm^* C^*) \|^2_2
\]
A Riemannian metric

Idea for choosing a Riemannian metric

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

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

$$
\frac{1}{2} \langle \eta_h, \text{Hess}_h f[\xi_h] \rangle_2 = \langle \text{diag}(B\eta_h m^* C^*), \text{diag}(B\xi_h m^* C^*) \rangle_2 \approx \langle \eta_h m^*, \xi_h m^* \rangle_2
$$

$$
\frac{1}{2} \langle \eta_m, \text{Hess}_m f[\xi_m] \rangle_2 = \langle \text{diag}(Bh\eta_m C^*), \text{diag}(Bh\xi_m C^*) \rangle_2 \approx \langle h\eta_m^*, h\xi_m^* \rangle_2,
$$

where $\approx$ can be derived from some assumptions;
A Riemannian metric

Idea for choosing a Riemannian metric

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

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

  $$\frac{1}{2} \langle \eta_h, \text{Hess}_h f[\xi_h] \rangle_2 = \langle \text{diag}(B\eta_h m^* C^*), \text{diag}(B\xi_h m^* C^*) \rangle_2 \approx \langle \eta_h m^*, \xi_h m^* \rangle_2$$

  $$\frac{1}{2} \langle \eta_m, \text{Hess}_m f[\xi_m] \rangle_2 = \langle \text{diag}(Bh\eta_m^* C^*), \text{diag}(Bh\xi_m^* C^*) \rangle_2 \approx \langle h\eta_m^*, h\xi_m^* \rangle_2,$$

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

- The Riemannian metric:

  $$g(\eta[x], \xi[x]) = \langle \eta_h, \xi_h m^* m \rangle_2 + \langle \eta_m^*, \xi_m^* h^* h \rangle_2;$$

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

- Riemannian gradient
  - A tangent vector: \( \nabla f([x]) \in T_{[x]} \mathcal{M} \);
  - Satisfies: \( Df([x])[\eta_{[x]}] = g(\nabla f([x]), \eta_{[x]}), \forall \eta_{[x]} \in T_{[x]} \mathcal{M} \);

- Represented by a vector in a horizontal space;

- Riemannian gradient:

  \[
  (\nabla f([((h, m)]) \uparrow_{(h, m)} = \text{Proj} \left( \nabla_h f(h, m)(m^* m)^{-1}, \nabla_m f(h, m)(h^* h)^{-1} \right);
  \]
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 \(k = 0\)

\[ d_k = \|h_k\|_2 \|m_k\|_2, \quad h_k \leftarrow \sqrt{d_k} \frac{h_k}{\|h_k\|_2}; \quad m_k \leftarrow \sqrt{d_k} \frac{m_k}{\|m_k\|_2}; \]

1 \((h_{k+1}, m_{k+1}) = (h_k, m_k) - \alpha \left( \frac{\nabla h_k f(h_k, m_k)}{d_k}, \frac{\nabla m_k f(h_k, m_k)}{d_k} \right)\);

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\(^5\)

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

Wirtinger flow Method in [LLSW16]

1. Given \((h_0, m_0)\), step size \(\alpha > 0\), and set \(k = 0\)
2. \((h_{k+1}, m_{k+1}) = (h_k, m_k) - \alpha \left( \nabla h_k f(h_k, m_k), \nabla m_k f(h_k, m_k) \right)\);
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

Given \((h_0, m_0)\), step size \(\alpha > 0\), and set \(k = 0\)

1. \[d_k = \|h_k\|_2 \|m_k\|_2, \quad h_k \leftarrow \sqrt{d_k} \frac{h_k}{\|h_k\|_2}; \quad m_k \leftarrow \sqrt{d_k} \frac{m_k}{\|m_k\|_2};\]

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]

Given \((h_0, m_0)\), step size \(\alpha > 0\), and set \(k = 0\)

1. \[(h_{k+1}, m_{k+1}) = (h_k, m_k) - \alpha \left( \nabla h_k f(h_k, m_k), \nabla m_k f(h_k, m_k) \right);\]

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]

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

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

- The first two terms in (ii) penalize large values of \(\|h\|_2\) and \(\|m\|_2\);
Penalty term for (i) Riemannian method, (ii) Wirtinger flow [LLSW16]

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

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

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

- 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.
Coherence is defined as

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

Coherence at the true solution \([(h^\dagger, m^\dagger)]\)

- influences the probability of recovery
- Small coherence is preferred
Penalty

Promote low coherence:

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

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

Initialization method [LLSW16]

- \((d, \tilde{h}_0, \tilde{m}_0)\): SVD of \(B^* \text{diag}(y)C\);
- \(h_0 = \arg\min_z \|z - \sqrt{d}\tilde{h}_0\|_2^2\), 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 $y = \text{diag}(B h_{\#}^* m_{\#}^* C^*)$, where entries in the true solution $(h_{\#}, m_{\#})$ are drawn from Gaussian distribution;
- All tested algorithms use the same initial point;
- Stop when $\|y - \text{diag}(B h m^* C^*)\|_2/\|y\|_2 \leq 10^{-8}$;
Efficiency

\[ \min \| y - \text{diag}(B h^* C^*) \|^2_2 \]

**Table:** Comparisons of efficiency

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>( L = 400, K = N = 50 )</th>
<th>( L = 600, K = N = 50 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( nFFT )</td>
<td>870</td>
<td>401</td>
</tr>
<tr>
<td>( RMSE )</td>
<td>2.22(_8)</td>
<td>1.48(_8)</td>
</tr>
</tbody>
</table>

- 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 \( \frac{\| h m^* - h^* m^* \|_F}{\| h^* \|_2 \| m^* \|_2} \)

Probability of successful recovery

Success if \[ \frac{\|h^*m^* - h^#m^#\|_F}{\|h^#\|_2 \|m^#\|_2} \leq 10^{-2} \]

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
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

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

\[
\min \|y - \text{diag}(Bhm^* C^*)\|_2^2
\]
Image deblurring with various kernels

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

$$\min \| y - \text{diag}(B h m^* C^*) \|^2_2$$
Image deblurring with various kernels

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: \( \mathbf{B} \).
Image deblurring with various kernels

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 $\frac{\|\hat{y} - \frac{\|y\|}{\|y_f\|} y_f\|}{\|\hat{y}\|}$ 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 \( \left\| \hat{y} - \frac{\|y\|}{\|y_f\|} y_f \right\| / \|\hat{y}\| \) 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!
Trust-region methods on Riemannian manifolds.

P.-A. Absil, R. Mahony, and R. Sepulchre.
*Optimization algorithms on matrix manifolds*.

A. Ahmed, B. Recht, and J. Romberg.
Blind deconvolution using convex programming.

J. F. Cardoso and A. Souloumiac.
Blind beamforming for non-gaussian signals.

3D face recognition under expressions, occlusions, and pose variations.

W. Huang, P.-A. Absil, and K. A. Gallivan.
A Riemannian symmetric rank-one trust-region method.
References II

Wen Huang, P.-A. Absil, and K. A. Gallivan.

ROPTLIB: an object-oriented C++ library for optimization on Riemannian manifolds.

W. Huang, K. A. Gallivan, and P.-A. Absil.
A Broyden Class of Quasi-Newton Methods for Riemannian Optimization. 

W. Huang, K. A. Gallivan, Anuj Srivastava, and P.-A. Absil.
Riemannian optimization for registration of curves in elastic shape analysis. 

Wen Huang, K. A. Gallivan, and Xiangxiong Zhang.
Solving PhaseLift by low rank Riemannian optimization methods for complex semidefinite constraints. 

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.

A Riemannian elastic metric for shape-based plant leaf classification.
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.

Shape analysis of elastic curves in Euclidean spaces.
doi:10.1109/TPAMI.2010.184.
References IV

F. J. Theis and Y. Inouye.
On the use of joint diagonalization in blind signal processing.

B. Vandereycken.
Low-rank matrix completion by Riemannian optimization—extended version.

A leaf recognition algorithm for plant classification using probabilistic neural network.

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.