A Technique in Implementations of Riemannian quasi-Newton Methods

Speaker: Wen Huang

Xiamen University

April 20, 2019

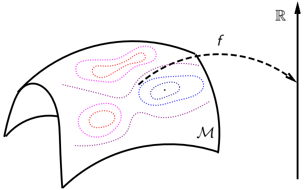
Joint work with Pierre-Antoine Absil, and Kyle Gallivan

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.

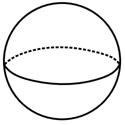


Unconstrained optimization problem on a constrained space.

Riemannian manifold = manifold + Riemannian metric

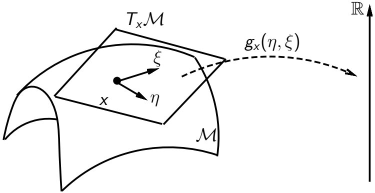
Riemannian Manifold

Manifolds:



- Stiefel manifold: $\mathrm{St}(p, n) = \{X \in \mathbb{R}^{n\times p} | X^T X = I_p\}$;
- Grassmann manifold $\mathrm{Gr}(p, n)$: all p-dimensional subspaces of $\mathbb{R}^n$;
- And many more.

Riemannian metric:



A Riemannian metric, denoted by g, is a smoothly-varying inner product on the tangent spaces;

BFGS Quasi-Newton Algorithm: from Euclidean to Riemannian

- Update formula:

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

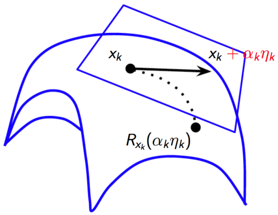
- Search direction:

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

- B_k update:

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

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



Optimization on a Manifold

BFGS Quasi-Newton Algorithm: from Euclidean to Riemannian

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

- Update formula:

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

- Search direction:

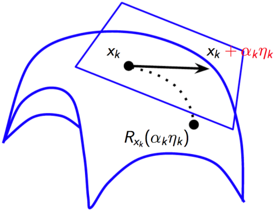
$$\eta_k = -B_k^{-1} \operatorname{grad} f(x_k)$$

- B_k update:

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

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

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



Optimization on a Manifold

BFGS Quasi-Newton Algorithm: from Euclidean to Riemannian

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

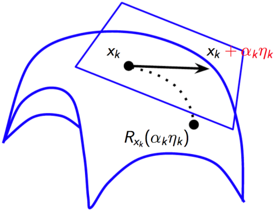
- Update formula:

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

- Search direction:

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

- B_k update:



Optimization on a Manifold

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}, \qquad \leftarrow$$ use vector transport

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

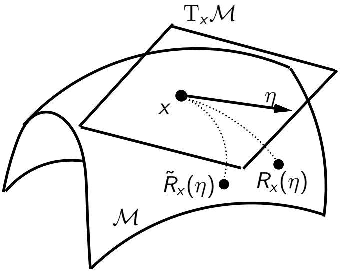
replaced by $R_{x_k}^{-1}(x_{k+1})$

use vector transport

Retraction and Vector Transport

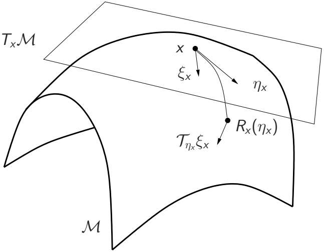
Retraction: $R : \mathrm{T}\mathcal{M} \to \mathcal{M}$

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



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

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



BFGS Quasi-Newton Algorithm: from Euclidean to Riemannian

- Update formula:

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

- Search direction:

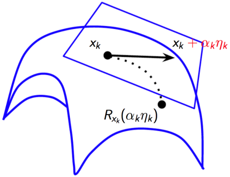
$$\eta_k = -B_k^{-1} \operatorname{grad} f(x_k)$$

- B_k update:

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

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

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



Optimization on a Manifold

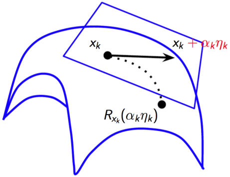
BFGS Quasi-Newton Algorithm: from Euclidean to Riemannian

- Update formula:

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

- Search direction:

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



Optimization on a Manifold

- B_k update:

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

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

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

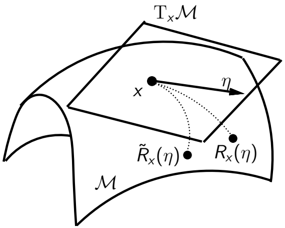
$\uparrow$ matrix vector multiplication

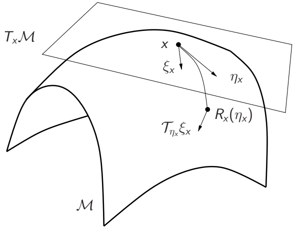
$\uparrow$ matrix vector multiplication

Extra cost on vector transports!

Existing Generic Riemannian BFGS methods

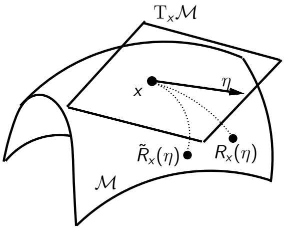
- Qi [Qi11]: exponential mapping, parallel translation;
  - Idea: imitate the Euclidean setting;

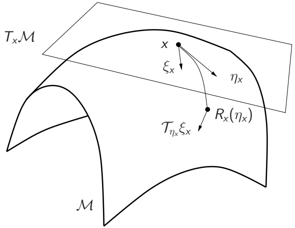




Existing Generic Riemannian BFGS methods

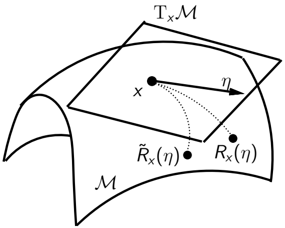
- Qi [Qi11]: exponential mapping, parallel translation;
    - Idea: imitate the Euclidean setting;
    - Exponential mapping: along the geodesic;

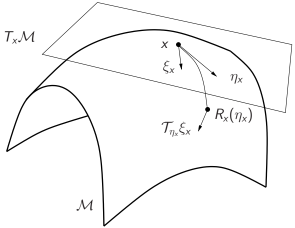




Existing Generic Riemannian BFGS methods

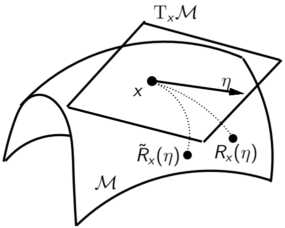
- Qi [Qi11]: exponential mapping, parallel translation;
  - Idea: imitate the Euclidean setting;
  - Exponential mapping: along the geodesic;
  - Parallel translation: move tangent vector parallelly;

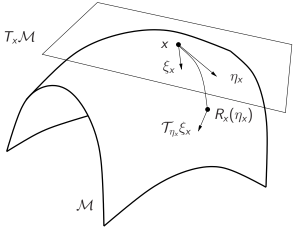




Existing Generic Riemannian BFGS methods

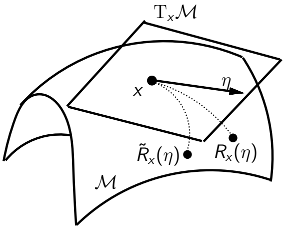
- Qi [Qi11]: exponential mapping, parallel translation;
  - Idea: imitate the Euclidean setting;
  - Exponential mapping: along the geodesic;
  - Parallel translation: move tangent vector parallelly;
  - Problem: maybe unknown to users, maybe expensive to compute;

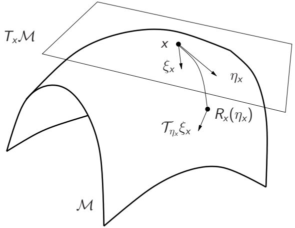




Existing Generic Riemannian BFGS methods

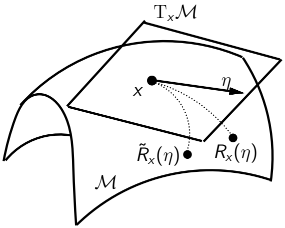
- Ring and Wirth [RW12]: retraction, vector transport by differentiated retraction, isometric vector transport;
    - Idea: quasi-Newton update in tangent space, then transport to new tangent space isometrically;

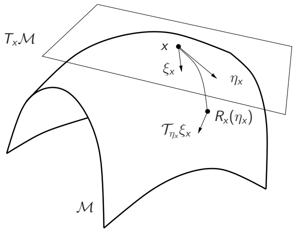




Existing Generic Riemannian BFGS methods

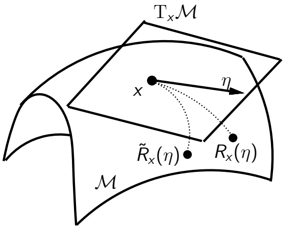
- Ring and Wirth [RW12]: retraction, vector transport by differentiated retraction, isometric vector transport;
  - Idea: quasi-Newton update in tangent space, then transport to new tangent space isometrically;
  - Retraction: no constraints;

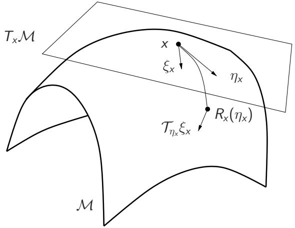




Existing Generic Riemannian BFGS methods

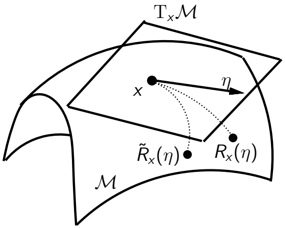
- Ring and Wirth [RW12]: retraction, vector transport by differentiated retraction, isometric vector transport;
  - Idea: quasi-Newton update in tangent space, then transport to new tangent space isometrically;
  - Retraction: no constraints;
  - Two vector transport: VT by differentiated retraction, and isometric VT;

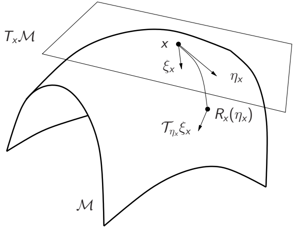




Existing Generic Riemannian BFGS methods

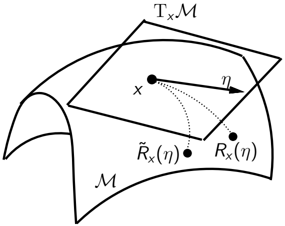
- Ring and Wirth [RW12]: retraction, vector transport by differentiated retraction, isometric vector transport;
  - Idea: quasi-Newton update in tangent space, then transport to new tangent space isometrically;
  - Retraction: no constraints;
  - Two vector transport: VT by differentiated retraction, and isometric VT;
  - Problem: vector transport by differentiated retraction maybe unknown to users, maybe expensive to compute;

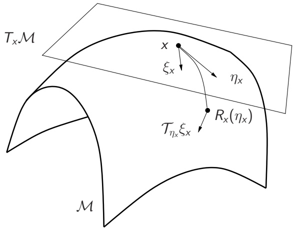




Existing Generic Riemannian BFGS methods

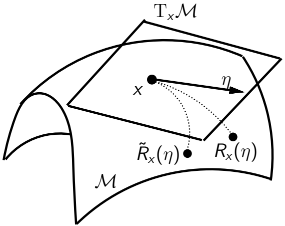
- Huang, Absil, Gallivan [HGA15, HAG18]: retraction, isometric vector transport
  - Idea: quasi-Newton update in tangent space, then transport to new tangent space isometrically;

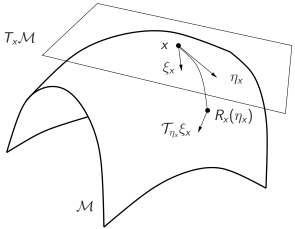




Existing Generic Riemannian BFGS methods

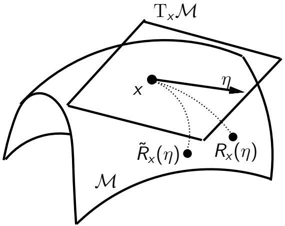
- Huang, Absil, Gallivan [HGA15, HAG18]: retraction, isometric vector transport
  - Idea: quasi-Newton update in tangent space, then transport to new tangent space isometrically;
  - Retraction: no constraints;

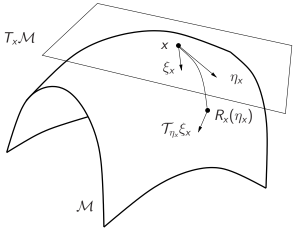




Existing Generic Riemannian BFGS methods

- Huang, Absil, Gallivan [HGA15, HAG18]: retraction, isometric vector transport
  - Idea: quasi-Newton update in tangent space, then transport to new tangent space isometrically;
  - Retraction: no constraints;
  - Vector transport: Isometric VT or VT by differentiated retraction along a direction;





Implementation of an isometric vector transport

Summary:

- Isometric vector transport is needed [RW12, HGA15, HAG18];
- An efficient vector transport is crucial;

Implementation of an isometric vector transport

Summary:

- Isometric vector transport is needed [RW12, HGA15, HAG18];
- An efficient vector transport is crucial;

An efficient isometric vector transport:

- Representative manifold:
  - the Stiefel manifold $\mathrm{St}(p, n) = \{X \in \mathbb{R}^{n\times p} | X^T X = I_p\}$;
  - canonical metric: $g(\eta_X, \xi_X) = \mathrm{trace}\left(\eta_X^T \left(I_n - \frac{1}{2} X X^T\right) \xi_X\right)$;
- The idea in this talk can be used for more algorithms and many commonly-encountered manifolds.

Representations of Tangent Vectors

- $\mathcal{E} = \mathbb{R}^w$;
- Dimension of $\mathcal{M}$ is d;
- Stiefel manifold: $\mathcal{E} = \mathbb{R}^{n \times p}$;
- Stiefel manifold: $d = np - p(p+1)/2$;

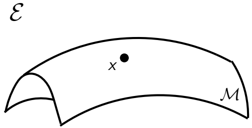


Figure: An embedded submanifold

- Extrinsic: $\eta_x \in \mathbb{R}^w$; $\left(T_X = \{X\Omega + X_\perp K \mid \Omega^T = -\Omega, X^T X_\perp = 0\}\right)$
- Intrinsic: $\tilde{\eta}_x \in \mathbb{R}^d$ such that $\eta_x = B_x \tilde{\eta}_x$, where B_x is smooth;

Representations of Tangent Vectors

- $\mathcal{E} = \mathbb{R}^w$;
- Dimension of $\mathcal{M}$ is d;

- Stiefel manifold: $\mathcal{E} = \mathbb{R}^{n\times p}$;
- Stiefel manifold: $d = np - p(p+1)/2$;

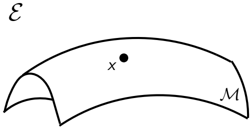


Figure: An embedded submanifold

- Extrinsic: $\eta_x \in \mathbb{R}^w$; $(T_X = \{X\Omega + X_\perp K \mid \Omega^T = -\Omega, X^T X_\perp = 0\})$
- Intrinsic: $\tilde{\eta}_x \in \mathbb{R}^d$ such that $\eta_x = B_x \tilde{\eta}_x$, where B_x is smooth;

How to find a basis B?

Extrinsic Representation and Intrinsic Representation on the Stiefel Manifold

$$\mathrm{T}_X \, \mathrm{St}(p,n) = \{X\Omega + X_\perp K \mid \Omega^T = -\Omega, X^T X_\perp = 0\};$$

$$B_x = \left\{ \begin{bmatrix} X & X_\perp \end{bmatrix} \left[\begin{array}{cccc} 0 & 1 & \ldots & 0 \\ -1 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 0 \\ \hline 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 0 \end{array}\right], \quad \ldots, \quad \begin{bmatrix} X & X_\perp \end{bmatrix} \left[\begin{array}{cccc} 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 0 \\ \hline 1 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 0 \end{array}\right] \right\}$$

Extrinsic Representation and Intrinsic Representation on the Stiefel Manifold

$$\mathrm{T}_X \mathrm{St}(p,n) = \{X\Omega + X_\perp K \mid \Omega^T = -\Omega, X^T X_\perp = 0\};$$

Extrinsic η_X:

$$\eta_X = \begin{bmatrix} X & X_\perp \end{bmatrix} \begin{bmatrix} \Omega \\ K \end{bmatrix}$$

$$= \begin{bmatrix} X & X_\perp \end{bmatrix} \begin{bmatrix} 0 & a_{12} & \dots & a_{1p} \\ -a_{12} & 0 & \dots & a_{2p} \\ \dots & \dots & \dots & \dots \\ -a_{1p} & -a_{2p} & \dots & 0 \\ b_{11} & b_{12} & \dots & b_{1p} \\ b_{21} & b_{22} & \dots & b_{2p} \\ \dots & \dots & \dots & \dots \\ b_{(n-p)1} & b_{(n-p)2} & \dots & b_{(n-p)p} \end{bmatrix}$$

Intrinsic $\tilde{\eta}_X$:

$$\tilde{\eta}_X = \begin{bmatrix} a_{12} \\ a_{13} \\ a_{23} \\ \vdots \\ a_{(p-1)p} \\ b_{11} \\ b_{21} \\ \vdots \\ b_{(n-p)p} \end{bmatrix}$$

Extrinsic Representation and Intrinsic Representation on the Stiefel Manifold

Question

Extrinsic representation $\eta_X \Longleftrightarrow$ Intrinsic representation $\tilde{\eta}_X$

- $\eta_X = \begin{bmatrix} X & X_\perp \end{bmatrix} \begin{bmatrix} \Omega \\ K \end{bmatrix} \Leftrightarrow \begin{bmatrix} \Omega \\ K \end{bmatrix} \Leftrightarrow \tilde{\eta}_X$

Extrinsic Representation and Intrinsic Representation on the Stiefel Manifold

Question

Extrinsic representation $\eta_X \Longleftrightarrow$ Intrinsic representation $\tilde{\eta}_X$

- $\eta_X = \begin{bmatrix} X & X_\perp \end{bmatrix} \begin{bmatrix} \Omega \\ K \end{bmatrix} \Leftrightarrow \begin{bmatrix} \Omega \\ K \end{bmatrix} \Leftrightarrow \tilde{\eta}_X$
- Apply Householder transformation to X, (Done in retraction)

$$Q_p^T Q_{p-1}^T \dots Q_1^T X = R = I_{n \times p}.$$

Extrinsic Representation and Intrinsic Representation on the Stiefel Manifold

Question

Extrinsic representation $\eta_X \Longleftrightarrow$ Intrinsic representation $\tilde{\eta}_X$

- $\eta_X = \begin{bmatrix} X & X_\perp \end{bmatrix} \begin{bmatrix} \Omega \\ K \end{bmatrix} \Leftrightarrow \begin{bmatrix} \Omega \\ K \end{bmatrix} \Leftrightarrow \tilde{\eta}_X$
- Apply Householder transformation to X, (Done in retraction)
$$Q_p^T Q_{p-1}^T \dots Q_1^T X = R = I_{n \times p}.$$
- $\begin{bmatrix} X & X_\perp \end{bmatrix} = Q_1 Q_2 \dots Q_p$ (Do not compute)

Extrinsic Representation and Intrinsic Representation on the Stiefel Manifold

Question

Extrinsic representation $\eta_X \Longleftrightarrow$ Intrinsic representation $\tilde{\eta}_X$

- $\eta_X = \begin{bmatrix} X & X_\perp \end{bmatrix} \begin{bmatrix} \Omega \\ K \end{bmatrix} \Leftrightarrow \begin{bmatrix} \Omega \\ K \end{bmatrix} \Leftrightarrow \tilde{\eta}_X$
- Apply Householder transformation to X, (Done in retraction)
$$Q_p^T Q_{p-1}^T \dots Q_1^T X = R = I_{n \times p}.$$
- $\begin{bmatrix} X & X_\perp \end{bmatrix} = Q_1 Q_2 \dots Q_p$ (Do not compute)
- Extrinsic to Intrinsic: $Q_p^T Q_{p-1}^T \dots Q_1^T \eta_X = \begin{bmatrix} \Omega \\ K \end{bmatrix}$ and reshape to $\tilde{\eta}_X$; $(4np^2 - 2p^3)$ flops
- Intrinsic to Extrinsic: reshape $\tilde{\eta}_X$ and $\eta_X = Q_1 Q_2 \dots Q_p \begin{bmatrix} \Omega \\ K \end{bmatrix}$; $(4np^2 - 2p^3)$ flops

Benefits of Intrinsic Representation

- Operations on tangent vectors are cheaper since $d \leq w$;
- If the basis is orthonormal, then the Riemannian metric reduces to the Euclidean metric:

$$g(\eta_x, \xi_x) = g(B_x \tilde{\eta}_x, B_x \tilde{\xi}_x) = \tilde{\eta}_x^T \tilde{\xi}_x.$$

Stiefel: $\mathrm{trace}\left(\eta_X^T \left(I_n - \frac{1}{2} X X^T\right) \xi_X\right) \quad \longrightarrow \quad \tilde{\eta}_X^T \tilde{\xi}_X$

- A vector transport has identity implementation, i.e., $\widetilde{\mathcal{T}}_\eta = \mathrm{id}$.

Vector Transport by Parallelization

- Vector transport by parallelization:

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

where $y = R_x(\eta_x)$ and $\dagger$ denotes pseudo-inverse, has identity implementation [HAG16]:

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

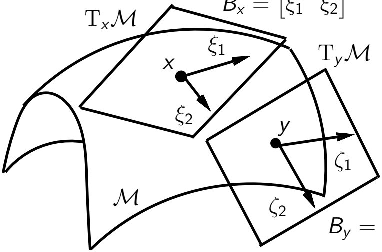
Example:

Extrinsic:

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

Intrinsic:

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



Sparse Eigenvalue Problem

Problem

Fine eigenvalues and eigenvectors of a sparse symmetric matrix A.

- The Brockett cost function:

$$f : \mathrm{St}(p, n) \to \mathbb{R} : X \mapsto \mathrm{trace}(X^T AXD);$$

- $D = \mathrm{diag}(\mu_1, \mu_2, \dots, \mu_p)$ with $\mu_1 > \cdots > \mu_p > 0$;
- Unique minimizer: X^* are eigenvectors for the p smallest eigenvalues.

Setting and Complexities

$$f : \mathrm{St}(p, n) \to \mathbb{R} : X \mapsto \mathrm{trace}(X^T AXD);$$

Setting

- $A = \mathrm{diag}(1, 2, \ldots, n) + B + B^T$, where entries of B has probability $1/n$ to be nonzero;
- $D = \mathrm{diag}(p, p - 1, \ldots, 1)$;

Complexities

- Function evaluation: $\approx 8np$
- Euclidean gradient evaluation: np (After function evaluation)
- Retraction evaluation (QR): $6np^2$

Extrinsic:

- $(10 + 8m)np^2 + O(p^3) + O(np)$;

Intrinsic:

- $14np^2 + O(p^3) + O(np)$;

Results

Table: An average of 100 random runs. Note that m is the upper bound of the limited-memory size m. $n = 1000$ and $p = 8$.

m	2		8		32	
	Extr	Intr	Extr	Intr	Extr	Intr
iter	1027	915	933	830	877	745
nf	1052	937	941	837	883	751
ng	1028	916	934	831	878	746
nR	1051	936	940	836	882	750
nV	1027	915	933	830	877	745
gf/gf0	9.00_{-7}	9.11_{-7}	9.24_{-7}	9.25_{-7}	9.52_{-7}	9.49_{-7}
t	2.94_{-1}	2.50_{-1}	4.84_{-1}	2.74_{-1}	1.27	4.31_{-1}
t/iter	2.86_{-4}	2.73_{-4}	5.18_{-4}	3.31_{-4}	1.45_{-3}	5.79_{-4}

Results

Table: An average of 100 random runs. Note that m is the upper bound of the limited-memory size m. $n = 1000$ and $p = 8$.

m	2		8		32	
	Extr	Intr	Extr	Intr	Extr	Intr
iter	1027	915	933	830	877	745
nf	1052	937	941	837	883	751
ng	1028	916	934	831	878	746
nR	1051	936	940	836	882	750
nV	1027	915	933	830	877	745
gf/gf0	9.00_{-7}	9.11_{-7}	9.24_{-7}	9.25_{-7}	9.52_{-7}	9.49_{-7}
t	2.94_{-1}	2.50_{-1}	4.84_{-1}	2.74_{-1}	1.27	4.31_{-1}
t/iter	2.86_{-4}	2.73_{-4}	5.18_{-4}	3.31_{-4}	1.45_{-3}	5.79_{-4}

Intrinsic representation yields faster LRBFGS implementation.

Conclusion

- Framework of Riemannian BFGS method;
- Review recent generic Riemannian BFGS methods;
- Intrinsic representation and vector transport by parallelization
- Numerical evidences of low complexity

Riemannian Manifold Optimization Library

- 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`

Users need only provide a cost function, gradient function, an action of Hessian (if a Newton method is used) in Matlab, Julia, R or C++ and parameters to control the optimization, e.g., the domain manifold, the algorithm, stopping criterion.

Thank you

Thank you!

References I

W. Huang, P.-A. Absil, and K. A. Gallivan.
Intrinsic representation of tangent vectors and vector transport on matrix manifolds.
Numerische Mathematik, 136(2):523–543, 2016.

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.

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.

C. Qi.
Numerical optimization methods on Riemannian manifolds.
PhD thesis, Florida State University, Department of Mathematics, 2011.

W. Ring and B. Wirth.
Optimization methods on Riemannian manifolds and their application to shape space.
SIAM Journal on Optimization, 22(2):596–627, January 2012.
doi:10.1137/11082885X.