Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 54-9 (2021) 572–577

A limited-memory Riemannian symmetric rank-one trust-region method with an efficient algorithm for its subproblem ⋆

Wen Huang * **Kyle A. Gallivan** **

* *School of Mathematical Sciences, Xiamen University, Xiamen, China (e-mail: wen.huang@xmu.edu.cn)*

** *Department of Mathematics, Florida State University, Tallahassee, FL, USA (e-mail: kgallivan@fsu.edu)*

Abstract: Limited-memory versions of quasi-Newton methods are efficient methods for large-scale optimization problems in the Euclidean space. In particular, a quasi-Newton symmetric rank-one update used in a trust-region setting has proven to be an effective method. In this paper, we present a Riemannian version of a limited-memory symmetric rank-one trust-region method with an efficient algorithm for solving its subproblem. Following a known standard result, we show the global convergence of the proposed method. The numerical experiments are performed to compare our method with other Riemannian optimization methods.

Keywords: Riemannian optimization; limited-memory quasi-Newton; trust-region; symmetric rank-one;

1. INTRODUCTION

Addressing the tradeoff between the convergence rate and the complexity of an individual iteration in order to achieve low total computational work for the required accuracy is a key task in designing and implementing efficient algorithms. For numerical optimization this usually involves considering the cost of assembling and solving the problem that determines the direction and scale of the update to the current iterate. For many methods this involves using information local to the current and recent iterates to construct a linear operator and solve a linear system for line search methods or solve a constrained optimization subproblem for trust-region methods.

In both the line-search and trust-region settings for Euclidean and Riemannian optimization problems, quasi-Netwon updates that enforce some version of a local secant condition, e.g., the Broyden family of updates, have been successfully used to efficiently construct each iterate's linear operator, see e.g., Liu and Vander Wiel (2007); Huang et al. (2015b). If a method is to be used for problems defined on high-dimensional spaces then additional design techniques such as the use of limited-memory versions of the updates are usually employed. Efficient and effective limited-memory line search methods have been designed for both Euclidean and Riemannian optimization problems. For trust-region methods in the Euclidean setting, recent works in Brust et al. (2017); Burdakov et al. (2017) have combined a quasi-Newton update with a low-complexity solution technique for the constrained optimization subproblem solved for every iteration.

In this paper, we build on our earlier work on quasi-Newton-based Riemannian trust-region methods (Huang et al. (2015a)) and combine it with this recent work on the Euclidean constrained subproblem to produce an efficient quasi-Newton-based limited-memory Riemannian trust-region algorithm. In numerical experiments, the joint diagonalization problem in independent component analysis and the matrix completion problem are used to demonstrate that the new algorithm is competitive to the state-of-the-art limited-memory line search quasi-Newton algorithm in Huang et al. (2018b).

This paper is organized as follows. Section 2 gives notation used in this paper. Section 3 describes the limited-memory quasi-Newotn-based Riemannian trust-region method and presents an efficient algorithm for its subproblem. Numerical experiments are reported in Section 4 and finally the conclusion and future work are discussed in Section 5.

2. NOTATION

The notation in this paper follows the standard literature Absil et al. (2008). Let the function f denote a continuously differentiable cost function defined on a d-dimensional Riemannian manifold $\mathcal{M}$ with the Riemannian metric $g : (\eta_x, \xi_x) \mapsto g_x(\eta_x, \xi_x) \in \mathbb{R}$. $\mathrm{T}_x\mathcal{M}$ denotes the tangent space of $\mathcal{M}$ at x and $\mathrm{T}\mathcal{M}$ denotes the tangent bundle, i.e., the union of all tangent spaces. The Riemannian gradient of f at x is denoted by $\mathrm{grad} f(x)$. Given $\eta_x \in \mathrm{T}_x\mathcal{M}$, $\eta_x^\flat$ represents the flat of η_x, i.e., $\eta_x^\flat : \mathrm{T}_x\mathcal{M} \to \mathbb{R} : \xi_x \mapsto g_x(\eta_x, \xi_x)$. The norm of η_x is defined by $\|\eta_x\|_x = \sqrt{g_x(\eta_x, \eta_x)}$. Let $\mathbb{B}_x(a)$ denote $\{\eta_x \in \mathrm{T}_x\mathcal{M} : \|\eta_x\|_x \leq a\}$. Given a linear operator $\mathcal{A}_x$ on $\mathrm{T}_x\mathcal{M}$ and $a \in \mathbb{R}$,

⋆ The author WH was partially supported by the Fundamental Research Funds for the Central Universities (NO. 20720190060) and National Natural Science Foundation of China (NO. 12001455). The work of KAG is partially supported by the U.S. National Science Foundation under grant DBI 1934157.

10.1016/j.ifacol.2021.06.118

let $\mathcal{L}_a(\mathcal{A}_x)$ denote $\mathcal{L}_a(\mathcal{A}_x) = \sum_{i=1}^d \min(a, \sigma_i) u_i v_i^\flat$, where $\mathcal{A}_x = \sum_{i=1}^d \sigma_i u_i v_i^\flat$ is the singular value decomposition, $u_i \in \mathrm{T}_x\mathcal{M}$ and $v_i \in \mathrm{T}_x\mathcal{M}, i = 1, \dots, d$ are left and right singular vectors, respectively. Given a matrix $A \in \mathbb{R}^{d\times d}$, let $L_a(A)$ denote $L_a(A) = \sum_{i=1}^d \min(\sigma_i, a) u_i v_i^T$, where $A = \sum_{i=1}^d \sigma_i u_i v_i^T$ is the singular value decomposition of the matrix A.

A retraction R is a C^1 map from the tangent bundle to the manifold such that (i) $R(0_x) = x$ for all $x \in \mathcal{M}$ (where 0_x denotes the origin of $\mathrm{T}\mathcal{M}$) and (ii) $\frac{d}{dt}R(t\xi_x)|_{t=0} = \xi_x$ for all $\xi_x \in \mathrm{T}_x\mathcal{M}$. The notation R_x denotes the retraction R restricting its domain to $\mathrm{T}_x\mathcal{M}$. A vector transport $\mathcal{T} : \mathrm{T}\mathcal{M} \oplus \mathrm{T}\mathcal{M} \to \mathrm{T}\mathcal{M} : (\eta_x, \xi_x) \mapsto \mathcal{T}_{\eta_x}\xi_x$ with associated retraction R is a C^1 map such that, for all (x, η_x) in the domain of R and all $\xi_x \in \mathrm{T}_x\mathcal{M}$, the following two conditions hold (i) $\mathcal{T}_{\eta_x}\xi_x \in \mathrm{T}_{R_x(\eta_x)}\mathcal{M}$ and (ii) $\mathcal{T}_{\eta_x}$ is a linear map. An isometric vector transport $\mathcal{T}_\mathrm{S}$ additionally satisfies $g_{R_x(\eta_x)}(\mathcal{T}_{\mathrm{S}_{\eta_x}}\xi_x, \mathcal{T}_{\mathrm{S}_{\eta_x}}\zeta_x) = g_x(\xi_x, \zeta_x)$, for all $\zeta_x, \xi_x, \eta_x \in \mathrm{T}_x\mathcal{M}$.

3. A LIMITED-MEMORY RIEMANNIAN TRUST-REGION METHOD WITH SYMMETRIC RANK-ONE UPDATE

The proposed limited-memory Riemannian trust-region method with symmetric rank-one update is described in Algorithm 1. The algorithm statement is discussed in Section 3.1. The algorithm for solving the subproblem in Algorithm 1 is given in Section 3.2, and the convergence result is stated in Section 3.3.

3.1 A guide to Algorithm 1

The trust-region method relies on minimizing an approximation of the objective function in a neighborhood of the current iterate. In this paper, the approximation of f at k-th iterate x_k is

$$m_k(s) = f(x_k) + g(\mathrm{grad} f(x_k), s) + \frac{1}{2} g(s, \mathcal{B}_k s), \quad (3.1)$$

where $s \in \mathbb{B}_x(\Delta_k)$, $\Delta_k > 0$ is the radius of the trust region, and $\mathcal{B}_k$ is a linear operator on $\mathrm{T}_{x_k}\mathcal{M}$. There are many choices for the linear operator $\mathcal{B}_k$. For example, the standard Riemannian trust-region method uses the Hessian operator in Absil et al. (2007) and a symmetric rank-one update (SR1) is used in Huang et al. (2015a) to define the Riemannian symmetric rank-one trust-region method.

In this paper, we use the operator used in Huang et al. (2015a) to develop an efficient limited-memory method. To describe the limited-memory approach, let s_k be a global solution of $\arg\min_{s\in\mathbb{B}_{x_k}(\Delta_k)} m_k(s)$ and

$$y_k = \mathcal{T}_{s_k}^{-1}\mathrm{grad} f(R_{x_k}(s_k)) - \mathrm{grad} f(x_k)$$

and let $S_{k,m}$ and $Y_{k,m}$ contain the (at most) m latest pairs of (s_k, y_k), which in the Riemannian setting are all transported to the tangent space $\mathrm{T}_{x_k}\mathcal{M}$, yielding $S_{k,m} = \{s_{k-\ell}^{(k)}, s_{k-\ell+1}^{(k)}, \dots, s_{k-1}^{(k)}\}$ and $Y_{k,m} = \{y_{k-\ell}^{(k)}, y_{k-\ell+1}^{(k)}, \dots, y_{k-1}^{(k)}\}$, where $\ell = \min\{m, k\}$ and where $s^{(k)}$ denotes s transported to $\mathrm{T}_{x_k}\mathcal{M}$. Moreover, let $P_{k,m} = D_{k,m} + L_{k,m} + L_{k,m}^T$, where

$$D_{k,m} = \mathrm{diag}\{g(s_{k-\ell}, y_{k-\ell}), \dots, g(s_{k-1}, y_{k-1})\},$$

and

$$(L_{k,m})_{i,j} = \begin{cases} g(s_{k-\ell+i-1}^{(k-\ell+i-1)}, y_{k-\ell+j-1}^{(k-\ell+i-1)}), & \text{if } i > j; \\ 0, & \text{otherwise.} \end{cases}$$

The limited-memory version of Riemannian SR1 is then given by

$$\tilde{\mathcal{B}}_k = \gamma_k \mathrm{id} + \Psi_{k,m} M_{k,m}^\dagger \Psi_{k,m}^\flat, \quad (3.2a)$$

$$\mathcal{B}_k = \mathcal{L}_\alpha\left(\tilde{\mathcal{B}}_k\right) \quad (3.2b)$$

where

$$\gamma_k = g(y_{k-1}, y_{k-1})/g(s_{k-1}, y_{k-1}), \quad (3.3)$$

$$\Psi_{k,m} = Y_{k,m} - \gamma_k S_{k,m}, \quad (3.4)$$

$$M_{k,m} = P_{k,m} - \gamma_k Q_{k,m}, \quad (3.5)$$

$Q_{k,m} = S_{k,m}^\flat S_{k,m}$, $\dagger$ denotes the pseudo-inverse operator, and α is a given constant.

The update (3.2a) is a Riemannian generalization of the Euclidean SR1 in the sense that if the manifold $\mathcal{M}$ is the Euclidean space, then (3.2a) is the Euclidean version of SR1 in (Brust et al., 2017, (4)). Moreover, if the vector transport is isometric, then (3.2a) is the update (Huang et al., 2015a, (46)). Equation (3.2b) guarantees that the norm of the Hessian approximation $\mathcal{B}_k$ is uniformly bounded from above by the constant α. This condition is used in the global convergence in Section 3.3.

Note that the fact that the vector transport in Huang et al. (2015a) is an isometry is used for the local superlinear convergence analysis of the trust region method with symmetric rank-one update. In this paper, we do not require the vector transport to be an isometry to make the method easier to use in practice since isometric vector transports may not be easy to construct or efficient in all cases.

By solving Problem (3.1), we obtain s_k and the candidate for the next iterate x_{k+1} is $R_{x_k}(s_k)$. The quality of the candidate is measured by the value of $\rho_k = \frac{f(x_k) - f(R_{x_k}(s_k))}{m_k(0) - m_k(s_k)}$. When ρ_k is sufficiently large the candidate sufficiently reduces the objective function, and the candidate is accepted as the next iterate. Otherwise, the candidate is rejected. The radius Δ_k of the trust region is then adjusted accordingly, i.e., shrinking Δ_k if ρ_k is small, enlarging Δ_k if ρ_k is large. The details are stated in Algorithm 1.

3.2 Algorithm for the subproblem

In the Euclidean setting, the subproblem in the trust-region method with SR1 has been investigated and efficient methods have also been proposed, see e.g., Burdakov et al. (2017); Brust et al. (2017). In this section, we reformulate the subproblem in Algorithm 1

$$s_k = \operatorname*{arg\,min}_{s\in\mathbb{B}_{x_k}(\Delta_k)} f(x_k) + g(\mathrm{grad} f(x_k), s) + \frac{1}{2} g(s, \mathcal{B}_k s), \quad (3.6)$$

into appropriate matrix expressions by exploiting the intrinsic representation proposed in Huang et al. (2016). The resulting problem has the same form as the subproblem in the Euclidean setting thereby enabling the use of the algorithm for the subproblem in Brust et al. (2017).

Algorithm 1 Limited-memory Riemannian trust-region with symmetric rank-one update (LRTR-SR1)

Input: Riemannian manifold $\mathcal{M}$ with Riemannian metric g; retraction R; vector transport $\mathcal{T}$; smooth function f on $\mathcal{M}$; initial iterate $x_0 \in \mathcal{M}$;

1: Choose an integer $m > 0$ and real numbers $\Delta_0 > 0$, $\nu \in (0,1)$, $c \in (0, 0.1)$, $\alpha > 0$, $\tau_1 \in (0,1)$, and $\tau_2 > 1$; Set $k \leftarrow 0$, $\ell \leftarrow 0$, $\gamma_0 \leftarrow 1$;

2: Obtain $s_k \in \mathrm{T}_{x_k}\mathcal{M}$ by (approximately) solving

$$s_k = \operatorname*{arg\,min}_{s \in \mathbb{B}_{x_k}(\Delta_k)} f(x_k) + g(\mathrm{grad} f(x_k), s) + \frac{1}{2} g(s, \mathcal{B}_k s),$$

where $\mathcal{B}_k$ is defined in accordance with (3.2);

3: Set $\rho_k \leftarrow \frac{f(x_k) - f(R_{x_k}(s_k))}{m_k(0) - m_k(s_k)}$;

4: Set $y_k \leftarrow \mathcal{T}_{\eta_k}^{-1}\mathrm{grad} f(R_{x_k}(s_k)) - \mathrm{grad} f(x_k)$;

5: **if** $|g(s_k, y_k - \mathcal{B}_k s_k)| \geq \nu \|s_k\| \|y_k - \mathcal{B}_k s_k\|$ **then**

6: $\gamma_{k+1} \leftarrow \frac{g(y_k, y_k)}{g(s_k, y_k)}$; Add $s_k^{(k)}$ and $y_k^{(k)}$ into storage; If $\ell \geq m$, then discard vector pair $\{s_{k-\ell}^{(k)}, y_{k-\ell}^{(k)}\}$ from storage, else $\ell \leftarrow \ell + 1$; Compute matrices $P_{k,m}$ and $Q_{k,m}$ by updating $P_{k-1,m}$ and $Q_{k-1,m}$ if available;

7: **else**

8: Set $\gamma_{k+1} \leftarrow \gamma_k$, $P_{k+1,m} \leftarrow P_{k,m}$, $Q_{k+1,m} \leftarrow Q_{k,m}$

9: $\{s_k^{(k)}, y_k^{(k)}\} \leftarrow \{s_{k-1}^{(k)}, y_{k-1}^{(k)}\}, \ldots, \{s_{k-\ell+1}^{(k)}, y_{k-\ell+1}^{(k)}\} \leftarrow \{s_{k-\ell}^{(k)}, y_{k-\ell}^{(k)}\}$.

10: **end if**

11: **if** $\rho_k > c$ **then**

12: $x_{k+1} \leftarrow R_{x_k}(s_k)$; Transport $s_{k-\ell+1}^{(k)}, s_{k-\ell+2}^{(k)}, \ldots, s_k^{(k)}$ and $y_{k-\ell+1}^{(k)}, y_{k-\ell+2}^{(k)}, \ldots, y_k^{(k)}$ from $\mathrm{T}_{x_k}\mathcal{M}$ to $\mathrm{T}_{x_{k+1}}\mathcal{M}$ by $\mathcal{T}$;

13: **else**

14: $x_{k+1} \leftarrow x_k$;

15: **end if**

16: **if** $\rho_k > \frac{3}{4}$ **then**

17: **if** $\|\eta_k\| \geq 0.8\Delta_k$ **then**

18: $\Delta_{k+1} \leftarrow \tau_2 \Delta_k$;

19: **else**

20: $\Delta_{k+1} \leftarrow \Delta_k$;

21: **end if**

22: **else if** $\rho_k < 0.1$ **then**

23: $\Delta_{k+1} \leftarrow \tau_1 \Delta_k$;

24: **else**

25: $\Delta_{k+1} \leftarrow \Delta_k$;

26: **end if**

27: $k \leftarrow k + 1$, goto 2 until convergence.

The intrinsic representation of tangent vector relies on a field of tangent bases B for all x, i.e., $B : x \mapsto B_x$, where B_x is a basis of $\mathrm{T}_x\mathcal{M}$. Here, it is assumed, without loss of generality, that B_x is an orthonormal basis for all x. For any tangent vector $\eta_x \in \mathrm{T}_x\mathcal{M}$, its intrinsic representation under the basis B_x is a vector $c_x \in \mathbb{R}^d$ such that $\eta_x = B_x c_x$. Since B_x is orthonormal, it follows that $c_x = B_x^\flat \eta_x$. Converting a tangent vector to its intrinsic representation and the other way round can be computed in reasonable computational cost for many manifolds, such as the Stiefel manifold, the manifold of fixed-rank matrices, and the manifold of symmetric positive definite matrices, see details in Huang et al. (2016); Yuan et al. (2020).

Note that in Huang et al. (2016) and Yuan et al. (2020), the field of tangent bases B is required to be a smooth function at least locally since it is used to define a specific vector transport – vector transport by parallelization $\mathcal{T}_{\eta_x}\xi_x = B_{R_{x(\eta_x)}} B_x^\flat$. In this paper, smoothness is not required.

When the intrinsic representation is used, the linear operator of $\mathrm{T}_x\mathcal{M}$ is a d-by-d matrix. Let $H \in \mathbb{R}^{d\times d}$ denote $B_{x_k}^\flat \mathcal{B}_k B_{x_k}$. It follows from (3.2) that

$$\begin{aligned} H =& B_{x_k}^\flat \mathcal{L}_\alpha \left(\gamma_k \mathrm{id} + \Psi_{k,m} M_{k,m}^\dagger \Psi_{k,m}^\flat\right) B_{x_k} \\ =& L_\alpha \left(\gamma_k I_d + B_{x_k}^\flat \Psi_{k,m} M_{k,m}^\dagger \Psi_{k,m}^\flat B_{x_k}\right) \\ =& L_\alpha \left(\gamma_k I_d + \tilde{\Psi}_{k,m} M_{k,m}^\dagger \tilde{\Psi}_{k,m}^T\right), \end{aligned}$$

where $\tilde{\Psi}_{k,m} = B_{x_k}^\flat \Psi_{k,m} \in \mathbb{R}^{d\times \ell}$. It follows that the subproblem (3.6) in the intrinsic representation is

$$c_* = \operatorname*{arg\,min}_{c\in\mathbb{R}^d, \|c\|_2 \leq \Delta_k} f(x_k) + c^T w + \frac{1}{2} c^T H c \tag{3.7}$$

where $w = B_{x_k}^\flat \mathrm{grad} f(x_k) \in \mathbb{R}^d$. The solution s_k in (3.6) is therefore $s_k = B_{x_k} c_*$.

In order to solve (3.7) efficiently, the spectral decomposition of H is used. Let $\tilde{\Psi}_{k,m} = QR$ be the thin QR factorization of $\tilde{\Psi}$ and let $RM_{k,m}^\dagger R^T = USU^T \in \mathbb{R}^{\ell\times\ell}$ be the spectral decomposition of $RM_{k,m}^\dagger R^T$, where $S = \mathrm{diag}(s_1, s_2, \ldots, s_\ell)$ and $s_1 \leq s_2 \ldots \leq s_\ell$. We have

$$H = L_\alpha \left(\gamma_k I_d + QUSU^T Q^T\right).$$

Let $P_\perp \in \mathbb{R}^{d\times(d-\ell)}$ denote the orthonormal complement of QU, i.e., $PP^T = I_p$, where $P = [QU\ P_\perp]$. Let $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$ denote the diagonal matrix

$$L_\alpha(\mathrm{diag}(\gamma_k + s_1, \ldots, \gamma_k + s_\ell, \gamma_k, \ldots \gamma_k)). \tag{3.8}$$

Let Λ_1 and Λ_2 denote the first ℓ-by-ℓ block and the last $(d-\ell)$-by-$(d-\ell)$ block of Λ, i.e.,

$$\Lambda = \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix}. \tag{3.9}$$

The spectral decomposition of H is therefore

$$\begin{aligned} H =& P\Lambda P^T = P\mathrm{diag}(\lambda_1, \ldots, \lambda_d) P^T \\ =& [QU\ P_\perp] \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix} [QU\ P_\perp]^T. \end{aligned}$$

The necessary and sufficient optimality condition of (3.7) can be found in Nocedal and Wright (2006) and is stated in Lemma 3.1.

Lemma 3.1. A vector c_* is a global solution of the trust-region subproblem (3.7) if and only if $\|c_*\|_2 \leq \Delta_k$ and there exists a unique $\sigma_* \geq 0$ such that $H + \sigma_* I_d$ is positive semidefinite,

$$(H + \sigma_* I_d) c_k = -w, \text{ and} \tag{3.10}$$

$$\sigma_*(\Delta_k - \|c_*\|_2) = 0. \tag{3.11}$$

Moreover, if $H + \sigma_* I_d$ is positive definite, then the global minimizer is unique.

Two functions $\varphi(\sigma) = -(H + \sigma I_d)^\dagger w$ and $\phi(\sigma) = 1/\|\varphi(\sigma)\|_2 - 1/\Delta_k$ play important roles in finding the σ_* and c_* in Lemma 3.1. It can be shown that

$$\varphi(\sigma) = \begin{cases} \kappa(\sigma), & \text{if } \sigma = -\gamma_k; \\ \kappa(\sigma) - \dfrac{1}{\gamma_k + \sigma}(I_d - QUU^T Q^T) w, & \text{otherwise;} \end{cases}$$

where $\kappa(\sigma) = -QU(\Lambda_1 + \sigma I_\ell)^\dagger U^T Q^T w$, and

$$\phi(\sigma) = \begin{cases} -\dfrac{1}{\Delta_k} & \text{if } \sigma = -\lambda_i \text{ and } a_i \neq 0, \\ \dfrac{1}{\sqrt{\sum\limits_{a_i \neq 0} \frac{a_i^2}{(\lambda_i+\sigma)^2}}} - \dfrac{1}{\Delta_k} & \text{otherwise;} \end{cases}$$

where $a = [a_1 \ a_2 \ \dots \ a_{\ell+1}] \in \mathbb{R}^{\ell+1}$,

$$a_j = (U^T Q^T w)_j \text{ for } j = 1, \dots, \ell, \tag{3.12a}$$

$$a_{\ell+1} = \sqrt{\|w\|_2^2 - \|U^T Q^T w\|_2^2}. \tag{3.12b}$$

The resulting algorithm for the Riemannian trust-region symmetric rank-one subproblem (3.6) is given in Algorithm 2.

Algorithm 2 Solving Riemannian SR1 trust-region subproblem (3.6)

Input: Riemannian manifold $\mathcal{M}$ with Riemannian metric g; two maps $\theta : \mathrm{T}_x\mathcal{M} \to \mathbb{R}^d : \eta_x \mapsto B_x^\flat \eta_x$ and $\vartheta : \mathbb{R}^d \to \mathrm{T}_x\mathcal{M} : c_x \mapsto B_x c_x$, where $B_x = [b_1 \ b_2 \ \dots \ b_d]$ is an orthonormal basis of $\mathrm{T}_x\mathcal{M}$ with respect to the Riemannian metric g; $S_{k,m} = \{s_{k-\ell}^{(k)}, s_{k-\ell+1}^{(k)}, \dots, s_{k-1}^{(k)}\}$ and $Y_{k,m} = \{y_{k-\ell}^{(k)}, y_{k-\ell+1}^{(k)}, \dots, y_{k-1}^{(k)}\}$;

Output: A global solution s_k;

1: Compute γ_k and $M_{k,m}$ by (3.3) and (3.5);
2: Compute $\tilde{\Psi}_{k,m} \in \mathbb{R}^{d \times \ell} = B_x^\flat \Psi_{k,m}$ by
$$\left(\tilde{\Psi}_{k,m}\right)_{:,j} = \theta(y_{k-\ell+j-1}^{(k)} - \gamma_k s_{k-\ell+j-1}^{(k)}),$$
for $j = 1, \dots, \ell$, where $M_{:,j}$ denotes the j-th column of the matrix M, and $\Psi_{k,m}$ is defined in (3.4);
3: Compute $QR = \tilde{\Psi}$ by the thin QR factorization;
4: Compute $RM_{k,m}^\dagger R^T = USU^T$ by the spectral decomposition;
5: Let λ_i, $i = 1, \dots, d$, Λ_1 and Λ_2 be defined in (3.8) and (3.9);
6: Compute $a \in \mathbb{R}^{\ell+1}$ by (3.12);
7: Let $\lambda_{\min} = \min(\lambda_1, \gamma_k)$ and let r be the algebraic multiplicity of $\lambda_{\min}$ in Λ;
8: **if** $\lambda_{\min} > 0$ and $\phi(0) \geq 0$ **then**
9: $\quad \sigma_* = 0$ and compute $c_* = \varphi(0)$;
10: **else if** $\lambda_{\min} \leq 0$ and $\phi(-\lambda_{\min}) \geq 0$ **then**
11: $\quad \sigma_* = -\lambda_{\min}$ and compute $\tilde{c}_* = \varphi(-\lambda_{\min})$;
12: $\quad$ **if** $\lambda_{\min} < 0$ **then**
13: $\quad\quad$ Let u be the first column of QU if $\lambda_{\min} = \lambda_1$; and u be any column of $P_\perp$ otherwise;
14: $\quad\quad$ Compute $c_* = \tilde{c}_* + tu$, where $t \in \mathbb{R}$ such that $\|c_*\|_2 = \Delta_k$.
15: $\quad$ **else**
16: $\quad\quad$ $c_* = \tilde{c}_*$;
17: $\quad$ **end if**
18: **else**
19: $\quad$ Use Newton's method to find σ_*, a root of ϕ, in $(\max(-\lambda_{\min}, 0), \infty)$ with initial iterate
$$\sigma_0 = \max\left(0, \max_{1 \leq i \leq \ell+1} \{\|a_i\|/\Delta_k - \lambda_i\}\right);$$
20: $\quad$ Compute $c_* = \varphi(\sigma_*)$;
21: **end if**
22: $s_k = \vartheta(c_*)$;

3.3 Global convergence

It is proven in Absil et al. (2007); Huang et al. (2015a) that under the assumption that there exists a constant β such that $\|\mathcal{B}_k\| < \beta$ for all k and the assumption that the subproblem is solved accurate enough in the sense that

$$m_k(0) - m_k(\mathfrak{s}_k) \geq \sigma_1 \|\mathrm{grad} f(x_k)\| \min\left(\Delta_k, \sigma_2 \frac{\|\mathrm{grad} f(x_k)\|}{\|\mathcal{B}_k\|}\right)$$

holds for some positive constant σ_1 and σ_2, the Riemannian version of the trust region method converges in the sense that $\lim_{k\to} \mathrm{grad} f(x_k) = 0$. Here, the first assumption is not required since we explicitly force the norm of $\mathcal{B}_k$ to be smaller than the given constant α. We assume the second assumption holds since Algorithm 2 aims to find high accuracy solution and the root of the scalar function $\phi(\sigma)$ is found in high accuracy ($|\phi(\sigma)| < 10^{-10}$ in our experiments). The global convergence is stated in Theorem 3.1 for completeness.

Theorem 3.1. Let Ω be the sublevel set $\{x \in \mathcal{M} : f(x) \leq f(x_0)\}$. Suppose there exists $\mu > 0$ and $\delta_\mu > 0$ such that

$$\|\xi\| \geq \mu \mathrm{dist}(x, R_x(\xi))$$

for all $x \in \Omega$, for all $\xi \in \mathrm{T}_x\mathcal{M}, \|\xi\| \leq \delta_\mu$. Then (i) if $f \in C^2$ is bounded below on the sublevel set Ω, then $\lim_{k\to\infty} \mathrm{grad} f(x_k) = 0$; (ii) if $f \in C^2$, $\mathcal{M}$ is compact, then $\lim_{k\to\infty} \mathrm{grad} f(x_k) = 0$, $\{x_k\}$ has at least one limit point, and every limit point of $\{x_k\}$ is a stationary point of f; and (iii) if $f \in C^2$, the sublevel set Ω is compact, f has a unique stationary point x^* in Ω, then $\{x_k\}$ converges to x^*.

4. NUMERICAL EXPERIMENTS

In this section, LRTR-SR1 is compared to the limited-memory Riemannian line-search method with BFGS update (LRBFGS) developed in Huang et al. (2018b). LRTR-SR1 is also compared to a variant of the quasi-Newton method that discards all the pairs of (s_k, y_k) when their number in memory reaches the given maximum number m, rather than only discarding the oldest pair on each iteration. The resulting "restarted" methods of LRBFGS and LRTR-SR1 are denoted by LRBFGS-R and LRTRSR1-R, respectively. The problem of joint diagonalization in independent component analysis from Theis et al. (2009) and the matrix completion problem from Vandereycken (2013) are used to investigate the performance of these methods. The experiments are performed in Matlab R2018b on a 64 bit Ubuntu platform with 3.5Ghz CPU (Intel Core i7-7800X). The C++ package ROPTLIB from Huang et al. (2018a) with its Matlab interface is used.

4.1 Joint diagonalization

The objective function in the joint diagonalization problem is

$$f : \mathrm{St}(p, n) \to \mathbb{R} : X \mapsto -\sum_{i=1}^{N} \left\|\mathrm{diag}(X^T C_i X)\right\|_2^2,$$

where $C_1, \dots, C_N$ are given symmetric matrices, $\mathrm{diag}(M)$ denotes the vector formed by the diagonal entries of M, and $\mathrm{St}(p, n) = \{X \in \mathbb{R}^{n \times p} : X^T X = I_p\}$ is the compact Stiefel manifold.

The Riemannian metric of $\mathrm{St}(p,n)$ endowed from its embedding space $\mathbb{R}^{n\times p}$, as in (Edelman et al., 1998, Section 2.2) is used. The corresponding Riemannian gradient is given in Theis et al. (2009). The vector transport by parallelization defined in Huang et al. (2016) is used. The retraction R on $\mathrm{St}(p,n)$ is based on the QR factorization defined in (Absil et al., 2008, (4.8)).

The initial radius Δ_0 is set to 1, ν is the square root of machine epsilon, c is set to 0.1, τ_1 to 0.25, and τ_2 to 2. The memory size m is set to 4. The upper bound α for $\mathcal{B}_k$ is set to $1000Nnp$. The algorithms terminate when the norm of the gradient is reduced by a factor of 10^6 compared to the norm of the initial gradient.

The matrices C_i are selected as $C_i = \mathrm{diag}(n, n-1, \ldots, 1) + R_i + R_i^T$, where the entries of $R_i \in \mathbb{R}^{n\times n}$ are independently drawn from the standard normal distribution. The initial iterate X_0 is computed by orthonormalizing a matrix M, i.e., $X_0 = \mathrm{orth}(M)$, where the entries of M are drawn from the standard normal distribution.

Table 1 reports an average result of 10 random runs. In the table, iter, nf, ng, $\|\mathrm{gf}_f\|/\|\mathrm{gf}_0\|$, f, and time, respectively, denote the number of iterations, the number of function evaluations, the number of gradient evaluations, the ratio of norm of final gradient to the norm of initial gradient, final function value, and computational time in seconds.

Table 1 shows that the line-search quasi-Newton methods LRBFGS and LRBFGS-R perform similarly in the sense that both methods use similar the number of function and gradient evaluations and computational time to get similar accuracy in their solutions. However, LRTRSR1-R outperforms LRTRSR1 implying that the restarting strategy is important here. Finally, quasi-Newton trust-region LRTRSR1-R is slower, but not catastrophically so, than LRBFGS and LRBFGS-R. A representative trajectory of the gradient norms is given in Figure 1 supporting these observations.

Table 1. An average result of 10 random runs for the joint diagonalization on the Stiefel manifold with $n = 12$, $p = 6$, $N = 5000$.

	LRBFGS	LRBFGS-R	LRTRSR1	LRTRSR1-R
iter	228	237	373	227
nf	258	273	374	228
ng	229	238	374	228
$\frac{\|\mathrm{gf}_f\|}{\|\mathrm{gf}_0\|}$	7.85_{-7}	9.10_{-7}	8.68_{-7}	8.66_{-7}
f	-4.23_6	-4.23_6	-4.23_6	-4.23_6
time	1.93	2.03	3.00	1.84

4.2 Matrix completion

For the matrix completion we consider the objective function proposed in Vandereycken (2013)

$$\min_{X\in\mathbb{R}_r^{m\times n}} f(X) := \frac{1}{2}\|P_\Omega(X - A)\|_F^2,$$

where $\mathbb{R}_r^{m\times n} = \{X \in \mathbb{R}^{m\times n} : \mathrm{rank}(X) = r\}$ is the manifold of fixed-rank matrices,

$$P_\Omega : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n} : X_{i,j} \mapsto \begin{cases} X_{i,j}, & \text{if } (i,j) \in \Omega; \\ 0, & \text{if } (i,j) \notin \Omega, \end{cases}$$

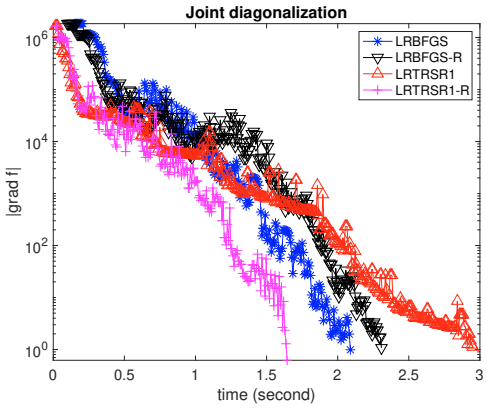

Fig. 1. A representative trajectory of the norms of the gradients versus the computational time in seconds for the joint diagonalization problem on the Stiefel manifold. The truncation for the Hessian approximation in LRTRSR1 and LRTRSR1-R are 5 and 0, respectively.

Ω is a given index set, and $P_\Omega(A)$ is the given sparse matrix.

The Riemannian metric is endowed from $\mathbb{R}^{m\times n}$, and is also used in Vandereycken (2013). The Riemannian gradient is also given therein. The default retraction and vector transport in ROPTLIB are used, i.e., retraction based on polar decomposition and vector transport by parallelization. The parameters in LRTRSR1 and LRTRSR1-R are the same as those in Section 4.1.

The matrix A is generated by GH^T, where $G \in \mathbb{R}^{m\times r}$, $H \in \mathbb{R}^{n\times r}$, and the entries of G and H are drawn from the standard normal distribution. The number of entries in Ω is fixed to be $3(m+n-k)k$. Note that $(m+n-k)k$ is the dimension of the manifold $\mathbb{R}_r^{m\times n}$. The set Ω is given by the first $3(m+n-k)k$ entries of a random permutation vector with size mn. The initial iterate is generated by $X_0 = UDV^T$, where the entries in D are drawn from the standard normal distribution and the entries of U and V obtained by orthonomalizing two matrices whose entries are drawn from the standard normal distribution. The upper bound α for $\mathcal{B}_k$ is set to 1000.

The numerical results are reported in Table 2 and Figure 2. As with the numerical results in Section 4.1, the line-search methods LRBFGS-R and LRBFGS perform similarly and the trust-region method LRTRSR1-R outperforms LRTRSR1. Therefore, we conclude that restarting strategy plays an important role in LRTRSR1 method. However, unlike the results in Section 4.1, LRTRSR1-R is competitive with and, in fact, slightly less efficient, than LRBFGS and LRBFGS-R for this matrix completion problem. Therefore, whether LRBFGS(-r) or LRTRSR1-R is more efficient is problem-dependent but there is leading evidence that the use of limited-memory quasi-Newton updates the usual gap between the robust trust-region approach and the simpler line-search approach for large scale Riemannian and Euclidean optimization.

5. CONCLUSION AND FUTURE WORK

In this paper, we have used our work in Riemannian line-search and trust-region methods, quasi-Newton up-

Table 2. An average result of 10 random runs for the matrix completion problem on the manifold of fixed-rank matrices with $m = n = 4000$, $r = 20$.

	LRBFGS	LRBFGS-R	LRTRSR1	LRTRSR1-R
iter	76	76	95	87
nf	82	82	96	88
ng	77	77	96	88
$\frac{\|\mathrm{gf}_f\|}{\|\mathrm{gf}_0\|}$	8.35_{-7}	8.27_{-7}	8.02_{-7}	6.35_{-7}
f	3.06_{-6}	4.28_{-6}	3.78_{-6}	1.62_{-6}
time	1.95	1.90	3.99	2.53

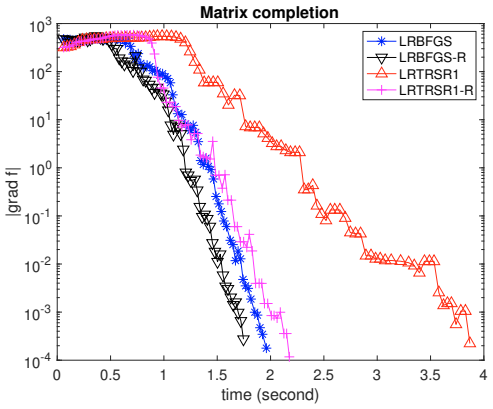

Fig. 2. A representative trajectory of the norms of the gradients versus the computational time in second for the matrix completion problem on manifold of fixed-rank matrices. The truncation for the Hessian approximation in LRTRSR1 and LRTRSR1-R are 0 and 5, respectively.

dates, limited-memory Riemannian method, vector transport and efficient Riemannian optimization algorithm development in combination with recent important innovative work on the efficient solution of the local optimization problem in limited-memory quasi-Newton updates in a Euclidean trust-region approach to improve the efficiency of Riemannian trust-region methods using a limited-memory symmetric rank-one quasi-Newton update. Leading evidence is provided on two standard Riemannian optimization problems that the performance gap between Riemannian line-search and trust-region methods can be reduced or reversed by the proposed Riemannian methods. The new method has fewer technical restrictions compared to our earlier limited-memory SR-1 work in that an isometric vector transport is no longer required.

Ongoing and future work on this topic includes the formal derivation of the limited-memory version from the full Riemannian SR-1 method, investigating guaranteeing convergence and boundedness of $\mathcal{B}_k$ without use of the spectral threshold guard α, and a systematic characterization of the strengths and weakness of the proposed approach to Riemannian line-search methods and restarting vs truncated versions of both.

REFERENCES

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

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

Brust, J., Erway, J.B., and Marcia, R.F. (2017). On solving L-SR1 trust-region subproblems. *Computational Optimization and Applications*, 66(2), 245–266. doi: 10.1007/s10589-016-9868-3.

Burdakov, O., Gong, L., Zikrin, S., and Yuan, Y.X. (2017). On efficiently combining limited-memory and trust-region techniques. *Mathematical Programming Computation*, 9(1), 101–134.

Edelman, A., Arias, T.A., and Smith, S.T. (1998). The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2), 303–353. Doi:10.1137/S0895479895290954.

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

Huang, W., Absil, P.A., and Gallivan, K.A. (2016). Intrinsic representation of tangent vectors and vector transport on matrix manifolds. *Numerische Mathematik*, 136(2), 523–543. doi:10.1007/s00211-016-0848-4.

Huang, W., Absil, P.A., Gallivan, K.A., and Hand, P. (2018a). ROPTLIB: an object-oriented C++ library for optimization on Riemannian manifolds. *ACM Transactions on Mathematical Software*, 4(44), 43:1–43:21.

Huang, W., Gallivan, K.A., and Absil, P.A. (2015b). A Broyden Class of Quasi-Newton Methods for Riemannian Optimization. *SIAM Journal on Optimization*, 25(3), 1660–1685. doi:10.1137/140955483.

Huang, W., Absil, P.A., and Gallivan, K.A. (2018b). A Riemannian BFGS method without differentiated retraction for nonconvex optimization problems. *SIAM Journal on Optimization*, 28(1), 470–495.

Liu, C. and Vander Wiel, S.A. (2007). Statistical quasi-Newton: a new look at least change. *SIAM Journal on Optimization*, 18(4), 1266–1285.

Nocedal, J. and Wright, S.J. (2006). *Numerical Optimization*. Springer, second edition.

Theis, F.J., Cason, T.P., and Absil, P.A. (2009). Soft dimension reduction for ICA by joint diagonalization on the Stiefel manifold. *Proceedings of the 8th International Conference on Independent Component Analysis and Signal Separation*, 5441, 354–361.

Vandereycken, B. (2013). Low-rank matrix completion by Riemannian optimization—extended version. *SIAM Journal on Optimization*, 23(2), 1214–1236.

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