Riemannian stochastic quasi-Newton algorithm with variance reduction and its convergence analysis

Hiroyuki Kasai
The University of Electro-Communications
kasai@is.uec.ac.jp

Hiroyuki Sato
Kyoto University
hsato@amp.i.kyoto-u.ac.jp

Bamdev Mishra
Amazon.com
bamdevm@amazon.com

Abstract

Stochastic variance reduction algorithms have recently become popular for minimizing the average value of a large but finite number of loss functions. This paper proposes a Riemannian stochastic quasi-Newton algorithm with variance reduction (R-SQN-VR). We present convergence analyses of the R-SQN-VR on both non-convex and retraction strongly convex functions with retraction and vector transport. The proposed algorithm is tested on the Riemannian centroid computation on the symmetric positive-definite manifold and the low-rank matrix completion on the Grassmann manifold. In all cases, the proposed algorithm outperforms the state-of-the-art Riemannian batch and stochastic gradient algorithms.

1 Introduction

Let f be a smooth real-valued function on a Riemannian manifold $\mathcal{M}$ [1]. The problem considered in this paper is that for a given model variable $w \in \mathcal{M}$ as

$$\min_{w \in \mathcal{M}} \left\{ f(w) := \frac{1}{n} \sum_{i=1}^{n} f_i(w) \right\}, \tag{1}$$

where n is the total number of the elements. This problem has many applications that include principal component analysis (PCA) and the subspace tracking problem [2] on the Grassmann manifold. The low-rank matrix/tensor completion problem is a promising example of the manifold of fixed-rank matrices/tensors [3, 4]. The linear regression problem is also defined on the manifold of fixed-rank matrices [5].

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84.

A popular approach to problem (1) is *Riemannian gradient descent* method, which computes the *Riemannian full gradient*, i.e., $\mathrm{grad} f(w) = \frac{1}{n} \sum_{i=1}^{n} \mathrm{grad} f_i(w)$, for every iteration, where $\mathrm{grad} f_i$ is the *Riemannian stochastic gradient* of f_i on the Riemannian manifold $\mathcal{M}$ for the n-th sample. This estimation is computationally heavy when n is extremely large. A popular alternative is *Riemannian stochastic gradient descent* (R-SGD) method that extends *stochastic gradient descent* (SGD) method in the Euclidean space [6]. Because R-SGD calculates only one $\mathrm{grad} f_i(w)$, its complexity per iteration is independent of n. However, like SGD [7], R-SGD suffers from slow convergence due to a *decaying step size* sequence. *Variance reduction* (VR) methods have recently been proposed to accelerate the convergence of SGD in the Euclidean space [8, 9, 10, 11, 12]. It involves calculating the full gradient estimation periodically and re-using it to reduce the variance of noisy stochastic gradients. However, because all previously described algorithms are *first-order* algorithms, their convergence speeds can be slow because of poor curvature approximations in *ill-conditioned* problems, as seen in Section 5. One promising approach involves *second-order* algorithms, such as stochastic *quasi-Newton* (QN) methods using Hessian evaluations [13, 14, 15, 16]. They achieve faster convergence by exploiting the curvature information of the objective function f. Furthermore, in addressing these two acceleration techniques, [17] and [18] proposed hybrid algorithms of stochastic QN methods accompanied by VR methods.

In examining Riemannian manifolds again, many challenges to the QN method have been addressed in deterministic settings [19, 20, 21]. The VR methods in the Euclidean space have also been extended to Riemannian manifolds, as the so-called R-SVRG [22, 23]. Nevertheless, the second-order stochastic algorithm using the VR method has not been explored thoroughly for problem (1). A similar algorithm to that proposed here was recently developed [24]. However, it is inexact in terms of geometry because it uses tangent vectors belonging to different tangent spaces to calculate

Hessian approximations. It also considers only exponential mapping and parallel translation. To resolve these issues, we propose a Riemannian stochastic QN method based on the limited BFGS (L-BFGS) and VR methods. Our contributions are four-fold:

- We propose a Riemannian limited-memory QN algorithm with a VR method. To the best of our knowledge, this is the first algorithm of its kind in the literature.
- Our convergence analysis deals with both *retraction strongly convex* functions (Definitions 3.7 and 3.8) and *non-convex* functions. To this end, we separately derive different bounds of the inverse Hessian approximation for these two functions.
- The proposed algorithm and its analyses are considered with computationally efficient *retraction* and *vector transport*, instead of the more restrictive *exponential mapping* and *parallel translation*. This gives us a significant advantage in terms in addition to computational efficiency, i.e., wider scope of applicable manifolds. For example, while [23] cannot be applied to the Stiefel and fixed-rank manifolds because they do not have closed-form expressions for parallel translation, our analyses and algorithm can be directly applied to them.

The specific features of the algorithms are two-fold:

- We update the curvature pair of the QN method at every outer loop by exploiting full gradient estimations in the VR method, and thereby capture more precise and stabler curvature-related information. This avoids the need for additional sweeping of samples required in the Euclidean stochastic QN [16], additional gradient estimations required in the Euclidean online BFGS (oBFGS) [14, 13, 25], and additional sub-sampling of the Hessian [16, 17].
- Compared with a simple Riemannian extension of the stochastic QN method, a noteworthy advantage of its combination with the VR method is that the number of computations for transports of curvature information can be drastically reduced. Specifically, the calculations of curvature information and the *second-order modified Riemannian stochastic gradient* are performed uniformly on the tangent space of the outer iterates.

The remainder of this paper is organized as follows: Section 2 presents details of R-SQN-VR. Section 3 presents the preliminaries and Section 4 provides convergence analyses. In Section 5, numerical comparisons on two problems are provided using the R-SGD and R-SVRG, where the results verified the superior performance of the R-SQN-VR. The proposed R-SQN-VR was implemented in the MATLAB toolbox Manopt [26]. Proof sketches and additional experiments are provided as supplementary material.

2 R-SQN-VR

We assume that the manifold $\mathcal{M}$ is endowed with a Riemannian metric structure, i.e., a smooth inner product $\langle \cdot, \cdot \rangle_w$ of tangent vectors is associated with the tangent space $T_w\mathcal{M}$ for all $w \in \mathcal{M}$ [1]. The *norm* $\| \cdot \|_w$ of a tangent vector is the norm associated with the Riemannian metric. The metric structure allows a framework for optimization over manifolds. Conceptually, the constrained optimization problem (1) is translated into an *unconstrained* problem over $\mathcal{M}$.

2.1 R-SGD and R-SVRG

R-SGD: Given a starting point $w_0 \in \mathcal{M}$, R-SGD produces a sequence $\{w_t\}$ in $\mathcal{M}$ that converges to a first-order critical point of problem (1). It updates w as

$$w_{t+1} = R_{w_t}(-\alpha_t \mathrm{grad} f_{i_t}(w_t)),$$

where α_t is a step size and $\mathrm{grad} f_{i_t}(w_t)$ is a Riemannian stochastic gradient for randomly selected i_t-th sample, which is a tangent vector at $w_t \in \mathcal{M}$. $\mathrm{grad} f_{i_t}(w_t)$ represents an unbiased estimator of the Riemannian full gradient $\mathrm{grad} f(w_t)$, and the expectation of $\mathrm{grad} f_{i_t}(w_t)$ over the all samples is $\mathrm{grad} f(w_t)$, i.e., $\mathbb{E}_{i_t}[\mathrm{grad} f_{i_t}(w_t)] = \mathrm{grad} f(w_t)$. The update moves from w_t along the direction $-\mathrm{grad} f_{i_t}(w_t)$ with a step size α_t while remaining on $\mathcal{M}$. This mapping, denoted by $R_w : T_w\mathcal{M} \to \mathcal{M} : \eta \mapsto R_w(\eta)$, is called *retraction* at w, and maps a tangent vector in $T_w\mathcal{M}$ onto $\mathcal{M}$ with a local rigidity condition that preserves the gradients at w. *Exponential mapping*, denoted by Exp, is an instance of the retraction.

R-SVRG: R-SVRG has double loops, where the k-th outer loop, called *epoch*, has m_k inner iterations. R-SVRG keeps $\tilde{w}^k \in \mathcal{M}$ after m_{k-1} inner iterations of the $(k-1)$-th epoch, and computes the full Riemannian gradient $\mathrm{grad} f(\tilde{w}^k)$ only for this stored $\tilde{w}^k$. It also computes the Riemannian stochastic gradient $\mathrm{grad} f_{i_t}(\tilde{w}^k)$ for the i_t-th sample. Then, by choosing the i_t-th sample for each t-th inner iteration of the k-th epoch at w_t, it calculates ξ_t, i.e., by modifying $\mathrm{grad} f_{i_t}(w_t)$ using both $\mathrm{grad} f(\tilde{w}^k)$ and $\mathrm{grad} f_{i_t}(\tilde{w}^k)$. Here, we omit the superscript k of w_t^k for simpliity. Similarly, in what follows, we use the notation w_t instead of w_t^k if k is clear from the context. Because they belong to different tangent spaces, a simple addition is not well-defined since Riemannian manifolds

Algorithm 1 Riemannian stochastic quasi-Newton with variance reduction (R-SQN-VR).

Require: Update frequency m_k, step size $\alpha_t^k > 0$, memory size L, number of epochs K, and cautious update threshold ϵ.

Initialize $\tilde{w}^0$, and calculate the Riemannian full gradient $\mathrm{grad} f(\tilde{w}^0)$. Initialize index set $\mathcal{J} := \emptyset$.
for $k = 0, 1, \ldots, K-1$ **do**
  Store $w_0 = \tilde{w}^k$.
  for $t = 0, 1, \ldots, m_k - 1$ **do**
    Choose $i_t \in \{1, 2, \ldots, n\}$ uniformly at random.
    Calculate the tangent vector $\tilde{\eta}_t$ from $\tilde{w}^k$ to w_t by $\tilde{\eta}_t = R_{\tilde{w}^k}^{-1}(w_t)$.
    if $k > 1$ **then**
      Transport the stochastic gradient $\mathrm{grad} f_{i_t}(w_t)$ to $T_{\tilde{w}^k}\mathcal{M}$ by $(\mathcal{T}_{\tilde{w}^k}^{w_t})^{-1}\mathrm{grad} f_{i_t}(w_t)$.
      Calculate $\tilde{\xi}_t$ as $\tilde{\xi}_t = (\mathcal{T}_{\tilde{w}^k}^{w_t})^{-1}\mathrm{grad} f_{i_t}(w_t) - (\mathrm{grad} f_{i_t}(\tilde{w}^k) - \mathrm{grad} f(\tilde{w}^k))$.
      Calculate $\tilde{\mathcal{H}}^k\tilde{\xi}_t$, transport $\tilde{\mathcal{H}}^k\tilde{\xi}_t$ back to $T_{w_t}\mathcal{M}$ by $\mathcal{T}_{\tilde{w}^k}^{w_t}$ as $\mathcal{T}_{\tilde{w}^k}^{w_t}\tilde{\mathcal{H}}^k\tilde{\xi}_t$.
      Update w_{t+1} from w_t as $w_{t+1} = R_{w_t}(-\alpha_t^k\mathcal{T}_{\tilde{w}^k}^{w_t}\tilde{\mathcal{H}}^k\tilde{\xi}_t)$.
    else
      Calculate ξ_t as $\xi_t = \mathrm{grad} f_{i_t}(w_t) - \mathcal{T}_{\tilde{w}^k}^{w_t}(\mathrm{grad} f_{i_t}(\tilde{w}^k) - \mathrm{grad} f(\tilde{w}^k))$.
      Update w_{t+1} from w_t as $w_{t+1} = R_{w_t}(-\alpha_t^k\xi_t)$.
    end if
  end for
  Option I-A: $\tilde{w}^{k+1} = g_{m_k}(w_1, w_2, \ldots, w_{m_k})$ (or $\tilde{w}^{k+1} = w_t$ for randomly chosen $t \in \{1, 2, \ldots, m_k\}$).
  Option I-B: $\tilde{w}^{k+1} = w_{m_k}$.
  Calculate the Riemannian full gradient $\mathrm{grad} f(\tilde{w}^{k+1})$.
  Calculate the tangent vector η_k from $\tilde{w}^k$ to $\tilde{w}^{k+1}$ by $\eta_k = R_{\tilde{w}^k}^{-1}(\tilde{w}^{k+1})$.
  Compute $s_k^{k+1} = \mathcal{T}_{\tilde{w}^k}^{\tilde{w}^{k+1}}\eta_k$, and $y_k^{k+1} = \beta_k^{-1}\mathrm{grad} f(\tilde{w}^{k+1}) - \mathcal{T}_{\tilde{w}^k}^{\tilde{w}^{k+1}}\mathrm{grad} f(\tilde{w}^k)$, where $\beta_k = \|\eta_k\|_{\tilde{w}^k} / \|\mathcal{T}_{\eta_k}^R\eta_k\|_{\tilde{w}^k}$.
  if $\langle y_k^{k+1}, s_k^{k+1}\rangle_{\tilde{w}^{k+1}} \geq \epsilon\|s_k^{k+1}\|_{\tilde{w}^{k+1}}^2$ **then**
    Discard pair $(s_{\min\mathcal{J}}^k, y_{\min\mathcal{J}}^k)$ and set $\mathcal{J} := \mathcal{J} - \min\mathcal{J}$ when $|\mathcal{J}| = L$.
    Store pair (s_k^{k+1}, y_k^{k+1}) and set $\mathcal{J} := \mathcal{J} \cup \{k\}$.
  end if
  Transport $\{(s_j^k, y_j^k)\}_{j\in\mathcal{J}} \in T_{\tilde{w}^k}\mathcal{M}$ to $\{(s_j^{k+1}, y_j^{k+1})\}_{j\in\mathcal{J}} \in T_{\tilde{w}^{k+1}}\mathcal{M}$ by $\mathcal{T}_{\eta_k}$.
end for
Option II-A: Output $w_{\mathrm{sol}} = \tilde{w}^K$
Option II-B: Output $w_{\mathrm{sol}} = w_t$ $(= w_t^k)$ for randomly chosen $t \in \{1, 2, \ldots, m_k\}$ and $k \in \{1, 2, \ldots, K\}$.

are not vector spaces. Therefore, once $\mathrm{grad} f_{i_t}(\tilde{w}^k)$ and $\mathrm{grad} f(\tilde{w}^k)$ are transported to $T_{w_t}\mathcal{M}$ by $\mathcal{T}_{\tilde{w}^k}^{w_t}$, the final update is performed as $w_{t+1} = R_{w_t}(-\alpha_t^k\xi_t)$, where ξ_t is set as $\xi_t = \mathrm{grad} f_{i_t}(w_t) - \mathcal{T}_{\tilde{w}^k}^{w_t}(\mathrm{grad} f_{i_t}(\tilde{w}^k) - \mathrm{grad} f(\tilde{w}^k))$, and where $\mathcal{T}_{\tilde{w}^k}^{w_t}$ or $\mathcal{T}_{\tilde{\eta}_t}$ represents *vector transport* from $\tilde{w}^k$ to w_t with $\tilde{\eta}_t \in T_{\tilde{w}^k}\mathcal{M}$ that satisfies $R_{\tilde{w}^k}(\tilde{\eta}_t) = w_t$. The vector transport $\mathcal{T} : T\mathcal{M} \oplus T\mathcal{M} \to T\mathcal{M}, (\eta, \xi) \mapsto \mathcal{T}_\eta\xi$ is associated with R and all $\eta, \xi \in \mathcal{T}_w\mathcal{M}$ [1]. It holds that (i) $\mathcal{T}_\eta\xi \in T_{R_w(\eta)}\mathcal{M}$, (ii) $\mathcal{T}_{0_w}\xi = \xi$, and (iii) $\mathcal{T}_\eta$ is a linear map. *Parallel translation* is a special instance of vector transport. When γ is a curve $\gamma(t) = R_w(t\eta)$ from z to w defined by R with $\eta \in T_w\mathcal{M}$, where $\gamma(0) = w$ and $\gamma(1) = z$, the parallel translation along γ is denoted by $P(\gamma)_z^w$ or $P_{\gamma(0)}^{\gamma(1)}$.

2.2 Proposed R-SQN-VR

We propose a Riemannian stochastic QN method accompanied by a VR method (R-SQN-VR). A straightforward extension involves updating the modified stochastic gradient ξ_t by premultiplying a linear *inverse Hessian approximation operator* $\mathcal{H}_t^k$ at w_t as

$$w_{t+1} = R_{w_t}(-\alpha_t^k\mathcal{T}_{\tilde{w}^k}^{w_t}\tilde{\mathcal{H}}_t^k\tilde{\xi}_t).$$

This formula is mathematically equivalent to

$$w_{t+1} = R_{w_t}(-\alpha_t^k\mathcal{H}_t^k\xi_t),$$

where $\mathcal{H}_t^k := \mathcal{T}_{\tilde{w}^k}^{w_t} \circ \tilde{\mathcal{H}}^k \circ (\mathcal{T}_{\tilde{w}^k}^{w_t})^{-1}$ and $\xi_t := \mathcal{T}_{\tilde{w}^k}^{w_t}\xi_t$ by denoting the inverse Hessian approximation at $\tilde{w}^k$ simply as $\tilde{\mathcal{H}}^k$. $\mathcal{T}$ is an isometric vector transport explained in Section 4 and $\mathcal{T}^{-1}$ is its inverse. We note that explicit formulas for $\mathcal{T}^{-1}$ are available for some manifolds, e.g., the SPD manifold in Section A.1, even if $\mathcal{T}$ is not the parallel translation. $\mathcal{H}_t^k$ should be positive definite, i.e., $\mathcal{H}_t^k \succ 0$, and is close to the Hessian of f, i.e., $\mathrm{Hess} f(w_t)$. It is noteworthy that $\tilde{\mathcal{H}}^k$ is calculated at only every outer epoch, and remains to be used for $\mathcal{H}_t^k$ throughout the k-th epoch.

Curvature pair (s_k^{k+1}, y_k^{k+1}): This paper addresses the operator $\tilde{\mathcal{H}}^k$ used in L-BFGS intended for large-

scale data. Thus, let s_k^{k+1} and y_k^{k+1} be the variable variation and the gradient variation at $T_{\tilde{w}^{k+1}}\mathcal{M}$, respectively, where the superscript expresses explicitly that they belong to $T_{\tilde{w}^{k+1}}\mathcal{M}$. It should be noted that the curvature pair (s_k^{k+1}, y_k^{k+1}) is calculated at the new $T_{\tilde{w}^{k+1}}\mathcal{M}$ just after the k-th epoch finishes. Furthermore, once the epoch index k is incremented, the curvature pair must be used only at $T_{\tilde{w}^k}\mathcal{M}$ because the calculation of $\tilde{\mathcal{H}}^k$ is performed only at $T_{\tilde{w}^k}\mathcal{M}$.

The variable variation s_k^{k+1} is calculated using the difference between $\tilde{w}^{k+1}$ and $\tilde{w}^k$. This is represented by the tangent vector η_k from $\tilde{w}^k$ to $\tilde{w}^{k+1}$, which is calculated using the inverse of the retraction as $R_{\tilde{w}^k}^{-1}(\tilde{w}^{k+1})$. As η_k belongs to $T_{\tilde{w}^k}\mathcal{M}$, transporting this onto $T_{\tilde{w}^{k+1}}\mathcal{M}$ yields

$$s_k^{k+1} = \mathcal{T}_{\tilde{w}^k}^{\tilde{w}^{k+1}}\eta_k \ \ (= \ \mathcal{T}_{\tilde{w}^k}^{\tilde{w}^{k+1}} R_{\tilde{w}^k}^{-1}(\tilde{w}^{k+1})). \tag{2}$$

The gradient variation y_k^{k+1} is calculated from the difference between the new full gradient $\mathrm{grad}f(\tilde{w}^{k+1}) \in T_{\tilde{w}^{k+1}}\mathcal{M}$ and the previous one, but transported as $\mathcal{T}_{\tilde{w}^k}^{\tilde{w}^{k+1}}\mathrm{grad}f(\tilde{w}^k) \in T_{\tilde{w}^k}\mathcal{M}$ [21], i.e.,

$$y_k^{k+1} = \beta_k^{-1}\mathrm{grad}f(\tilde{w}^{k+1}) - \mathcal{T}_{\tilde{w}^k}^{\tilde{w}^{k+1}}\mathrm{grad}f(\tilde{w}^k), \tag{3}$$

where $\beta_k > 0$ is explained in Section 4.

Inverse Hessian approximation operator $\tilde{\mathcal{H}}^k$: $\tilde{\mathcal{H}}^k$ is calculated using the past curvature pairs as $\tilde{\mathcal{H}}^{k+1} = (\check{\mathcal{V}}_k^{k+1})^\flat\tilde{\mathcal{H}}_k\check{\mathcal{V}}_k^{k+1} + \rho_k s_k^{k+1}(s_k^{k+1})^\flat$, where $\tilde{\mathcal{H}}_k = \mathcal{T}_{\tilde{w}^k}^{\tilde{w}^{k+1}} \circ \tilde{\mathcal{H}}^k \circ (\mathcal{T}_{\tilde{w}^k}^{\tilde{w}^{k+1}})^{-1}$, $\rho_k = 1/\langle y_k^{k+1}, s_k^{k+1}\rangle_{\tilde{w}^{k+1}}$, and $\check{\mathcal{V}}_j^k = \mathrm{id} - \rho_j y_j^k (s_j^k)^\flat$ with identity mapping id [21]. $a^\flat$ denotes the adjoint of $a \in T_w\mathcal{M}$, i.e., $a^\flat : T_w\mathcal{M} \to \mathbb{R} : v \mapsto \langle a, v\rangle_w$. Thus, $\tilde{\mathcal{H}}^k$ depends on $\tilde{\mathcal{H}}^{k-1}$ and (s_{k-1}, y_{k-1}) and, similarly, $\tilde{\mathcal{H}}^{k-1}$ depends on $\tilde{\mathcal{H}}^{k-2}$ and (s_{k-2}, y_{k-2}). Proceeding recursively, $\tilde{\mathcal{H}}^k$ is a function of the initial $\tilde{\mathcal{H}}^0$ and all previous k curvature pairs $\{(s_j, y_j)\}_{j=0}^{k-1}$. L-BFGS restricts use to the most recent L pairs $\{(s_j, y_j)\}_{j=k-L}^{k-1}$ as (s_j, y_j) with $j < k - L$ are likely to have scant curvature information. Based on this idea, L-BFGS performs L updates using the initial $\tilde{\mathcal{H}}^0$. We use the k pairs $\{(s_j, y_j)\}_{j=0}^{k-1}$ when $k < L$.

Now, we consider the final calculation of $\tilde{\mathcal{H}}^k$ used for $\mathcal{H}_t^k$ in the inner iterations of the k-th outer epoch using the L most recent curvature pairs. As this calculation is executed at $T_{\tilde{w}^k}\mathcal{M}$ and a Riemannian manifold is in general not a vector space, all L curvature pairs must be located at $T_{\tilde{w}^k}\mathcal{M}$. To this end, once the curvature pair is calculated in (2) and (3), the past $(L-1)$ pairs of $\{(s_j^k, y_j^k)\}_{j=k-L+1}^{k-1} \in T_{\tilde{w}^k}\mathcal{M}$ are transported into $T_{\tilde{w}^{k+1}}\mathcal{M}$ as $\{(s_j^{k+1}, y_j^{k+1})\}_{j=k-L+1}^{k-1}$ by the same vector transport $\mathcal{T}_{\eta_k}$ used when calculating s_k^{k+1} and y_k^{k+1}. It should be emphasized that this transport is necessary only for every outer epoch instead of every inner loop, and results in a drastic reduction in computational complexity in comparison with the straightforward extension of the Euclidean stochastic L-BFGS [25] into the manifold setting. Consequently, the update is defined as [21]

$$\tilde{\mathcal{H}}^k = ((\check{\mathcal{V}}_{k-1}^k)^\flat \cdots (\check{\mathcal{V}}_{k-L}^k)^\flat)\check{\mathcal{H}}_0^k(\check{\mathcal{V}}_{k-L}^k \cdots \check{\mathcal{V}}_{k-1}^k) \ + \ \cdots \\ + \rho_{k-2}(\check{\mathcal{V}}_{k-1}^k)^\flat s_{k-2}^k (s_{k-2}^k)^\flat (\check{\mathcal{V}}_{k-1}^k) + \rho_{k-1} s_{k-1}^k (s_{k-1}^k)^\flat,$$

where $\check{\mathcal{H}}_0^k$ is the initial inverse Hessian approximation. Because $\check{\mathcal{H}}_0^k$ is not necessarily equal to $\tilde{\mathcal{H}}^{k-L}$, and because it can be any positive definite self-adjoint operator, we use $\check{\mathcal{H}}_0^k = \langle s_{k-1}^k, y_{k-1}^k\rangle_{\tilde{w}^k}/\langle y_{k-1}^k, y_{k-1}^k\rangle_{\tilde{w}^k}\mathrm{id}$ as in the Euclidean case. The practical update of $\tilde{\mathcal{H}}^k$ uses a *two-loop recursion* algorithm [27] in Algorithm A.1 of the supplementary material.

Cautious update: The Euclidean L-BFGS fails on non-convex problems because the Hessian approximation has eigenvalues that are away from zero and not uniformly bounded from above. To circumvent this issue, *cautious update* has been proposed in the Euclidean space [28]. Similarly, we skip the update of the curvature pair when the following condition is not satisfied:

$$\langle y_k^{k+1}, s_k^{k+1}\rangle_{\tilde{w}^{k+1}} \geq \epsilon \|s_k^{k+1}\|_{\tilde{w}^{k+1}}^2, \tag{4}$$

where $\epsilon > 0$ is a predefined constant parameter. According to this update, the positive definiteness of $\tilde{\mathcal{H}}^k$ is guaranteed as long as $\tilde{\mathcal{H}}^{k-1}$ is positive definite.

Second-order modified stochastic gradient $\mathcal{H}_t^k\xi_t$: R-SVRG transports $\mathrm{grad}f(\tilde{w}^k)$ and $\mathrm{grad}f_{i_t}(\tilde{w}^k)$ at $T_{\tilde{w}^k}\mathcal{M}$ into $T_{w_t}\mathcal{M}$ to add them to $\mathrm{grad}f_{i_t}(w_t)$ at $T_{w_t}\mathcal{M}$. If we follow the same strategy, we must also transport L pairs of $\{(s_j^k, y_j^k)\}_{j=k-L}^{k-1} \in T_{\tilde{w}^k}\mathcal{M}$ into the given $T_{w_t^k}\mathcal{M}$ at every inner iteration. Addressing this problem, and given that both the full gradient and the curvature pairs belong to the same tangent space $T_{\tilde{w}^k}\mathcal{M}$, we transport $\mathrm{grad}f_{i_t}(w_t)$ from $T_{w_t^k}\mathcal{M}$ into $T_{\tilde{w}^k}\mathcal{M}$, and complete all calculations on $T_{\tilde{w}^k}\mathcal{M}$. Specifically, after transporting $\mathrm{grad}f_{i_t}(w_t)$ as $(\mathcal{T}_{\tilde{w}^k}^{w_t})^{-1}\mathrm{grad}f_{i_t}(w_t)$ from w_t to $\tilde{w}^k$ using $\tilde{\eta}_t(= R_{\tilde{w}^k}^{-1}(w_t))$, the modified stochastic gradient $\tilde{\xi}_t \in T_{\tilde{w}^k}\mathcal{M}$ is computed as

$$\tilde{\xi}_t = (\mathcal{T}_{\tilde{w}^k}^{w_t})^{-1}\mathrm{grad}f_{i_t}(w_t) - (\mathrm{grad}f_{i_t}(\tilde{w}^k) - \mathrm{grad}f(\tilde{w}^k)).$$

After calculating $\tilde{\mathcal{H}}^k\tilde{\xi}_t \in T_{\tilde{w}^k}\mathcal{M}$ using the two-loop recursion algorithm, we obtain $\mathcal{H}_t^k\xi_t \in T_{w_t}\mathcal{M}$ by transporting $\tilde{\mathcal{H}}^k\tilde{\xi}_t$ to $T_{w_t}\mathcal{M}$ as $\mathcal{T}_{\tilde{w}^k}^{w_t}\tilde{\mathcal{H}}^k\tilde{\xi}_t$. Finally, we update w_{t+1} from w_t as $w_{t+1} = R_{w_t}(-\alpha_t^k\mathcal{H}_t^k\xi_t)$. Although $-\xi_t$ is not generally guaranteed as a descent direction, $\mathbb{E}_{i_t}[-\xi_t] = -\mathrm{grad}f(w_t)$ is a descent direction. Furthermore, the positive definiteness of $\mathcal{H}_t^k$ implies that $-\mathcal{H}_t^k\xi_t$ is an average descent direction due to $\mathbb{E}_{i_t}[-\mathcal{H}_t^k\xi_t] = -\mathcal{H}_t^k\mathrm{grad}f(w_t)$.

3 Preliminaries

We first summarize some definitions from [21].

Definition 3.1 (Upper-Hessian bounded)**.** *f is said to be upper-Hessian bounded in $\mathcal{U} \subset \mathcal{M}$ if there exists a constant $L > 0$ such that $\frac{d^2 f(R_w(t\eta))}{dt^2} \leq L$, for all $w \in \mathcal{U}$ and $\eta \in T_w\mathcal{M}$ with $\|\eta\|_w = 1$, and all t such that $R_w(\tau\eta) \in \mathcal{U}$ for all $\tau \in [0, t]$.*

Definition 3.2 (Lower-Hessian bounded)**.** *f is said to be lower-Hessian bounded in $\mathcal{U} \subset \mathcal{M}$ if there exists a constant $\mu > 0$ such that $\mu \leq \frac{d^2 f(R_w(t\eta))}{dt^2}$, for all $w \in \mathcal{U}$ and $\eta \in T_w\mathcal{M}$ with $\|\eta\|_w = 1$, and all t such that $R_w(\tau\eta) \in \mathcal{U}$ for all $\tau \in [0, t]$.*

Definition 3.3 (ρ-totally retractive neighborhood)**.** *Let Θ_w be a neighborhood of w as a set such that for all $z \in \Theta_w$, $\Theta_w \subset R_z(\mathbb{B}(0_z, \rho))$, and $R_z(\cdot)$ is a diffeomorphism on $\mathbb{B}(0_z, \rho)$, which is the ball in $T_w\mathcal{M}$ with center 0_z and radius ρ, where 0_z is the zero vector in $T_z\mathcal{M}$. Then, Θ_w is said to be a ρ-totally retractive neighborhood of w.*

Now, we summarize some essential lemmas from [21].

Lemma 3.4. *Suppose that f is upper-Hessian bounded. Then, there exists a neighborhood $\mathcal{U}$ of arbitrary $\bar{w}$ and a constant $L > 0$ in Definition 3.1 such that for all $w, z \in \mathcal{U}$*

$$f(z) \leq f(w) + \langle \mathrm{grad} f(w), \xi \rangle_w + \tfrac{1}{2} L \|\xi\|_w^2,$$

where $\xi \in T_w\mathcal{M}$ and $R_w(\xi) = z$.

Lemma 3.5. *Suppose that f is lower-Hessian bounded. Then, there exists a neighborhood $\mathcal{U}$ of arbitrary $\bar{w}$ and a constant $\mu > 0$ in Definition 3.2 such that for all $w, z \in \mathcal{U}$*

$$f(z) \geq f(w) + \langle \mathrm{grad} f(w), \xi \rangle_w + \tfrac{1}{2} \mu \|\xi\|_w^2,$$

where $\xi \in T_w\mathcal{M}$ and $R_w(\xi) = z$.

Here, we additionally give some definitions according to the lemmas above.

Definition 3.6 (Retraction L-smooth)**.** *f is said to be retraction L-smooth in $\mathcal{U} \subset \mathcal{M}$ if f satisfies the property in Lemma 3.4.*

Definition 3.7 (Retraction convex)**.** *f is said to be retraction convex in $\mathcal{U} \subset \mathcal{M}$ if, when for all $w \in \mathcal{S}$ and all $\eta \in T_w\mathcal{M}$ with $\|\eta\|_w = 1$, $f(R_w(\tau\eta))$ is convex for all t satisfying $f(R_w(\tau\eta)) \in \mathcal{S}$ for all $\tau \in [0, t]$.*

Definition 3.8 (Retraction μ-strongly convex)**.** *f is said to be retraction μ-strongly convex in $\mathcal{U} \subset \mathcal{M}$ if f satisfies the property in Lemma 3.5.*

We also introduce a lemma to evaluate the difference between parallel translation and vector transport.

Lemma 3.9 (Lemma 3.5 in [21])**.** *Let $\mathcal{T} \in C^0$ be a vector transport associated with the same retraction R as that of the parallel translation $P \in C^\infty$. Under Assumption 1.3, there exists a neighborhood $\mathcal{U}$ of $\bar{w}$ and a constant $\theta > 0$ such that for all $w, z \in \mathcal{U}$,*

$$\|\mathcal{T}_\eta \xi - P_\eta \xi\|_z \leq \theta \|\xi\|_w \|\eta\|_w, \tag{5}$$

where $\xi, \eta \in T_w\mathcal{M}$ and $R_w(\eta) = z$.

The proofs for Lemmas 3.4, 3.5, and 3.9 are in [21]. Furthermore, modifying Lemma 3 in [29] slightly, we obtain the following lemma:

Lemma 3.10. *Let $\mathcal{M}$ be a Riemannian manifold endowed with retraction R and let $\bar{w} \in \mathcal{M}$ be an arbitrary point. Then, there exist $\tau_1 > 0$, $\tau_2 > 0$, and δ_{τ_1,τ_2} such that for all w in a sufficiently small neighborhood of $\bar{w}$ and all $\xi \in T_w\mathcal{M}$ with $\|\xi\|_w \leq \delta_{\tau_1,\tau_2}$, the following inequalities hold:*

$$\tau_1 \mathrm{dist}(w, R_w(\xi)) \;\leq\; \|\xi\|_w \;\leq\; \tau_2 \mathrm{dist}(w, R_w(\xi)). \tag{6}$$

Finally, we introduce the key lemma from Lemma 3.4.

Lemma 3.11. *Suppose that f is upper-Hessian bounded. Then, there exists a neighborhood $\mathcal{U}$ of $\bar{w}$ and a constant $L_l > 0$ such that for all $w, z \in \mathcal{U}$,*

$$\|P(\gamma)_z^w \mathrm{grad} f(z) - \mathrm{grad} f(w)\|_w \leq L_l \|\eta\|_w. \tag{7}$$

Definition 3.12 (Retraction L_l-Lipschitz)**.** *f is said to be retraction L_l-Lipschitz in $\mathcal{U} \subset \mathcal{M}$ if f satisfies the property in Lemma 3.11.*

L and L_l are the counterparts to those of L-smooth and L_l-Lipschitz in the Euclidean case, respectively. However, L and L_l are not identical when γ is a retraction curve, and $L_l = L$ holds when γ is a geodesic.

4 Convergence analysis

We first summarize basic assumptions for all analyses.

Assumption 1. *We assume the following [21]:*

(1.1) The objective function f and its components $f_1, f_2, \ldots, f_n$ are twice-continuously differentiable.

(1.2) For a sequence $\{w_t^k\}$ generated by Algorithm 1, there exists a compact and connected set $\Theta \subset \mathcal{M}$ such that $w_t^k \in \Theta$ for all $k, t \geq 0$.

(1.3) For each $k \geq 1$, there exists a totally retractive neighborhood Θ_k of $\tilde{w}^k$ such that $\{w_t^k\}$ stays in Θ_k for any $t \geq 0$ (Definition 3.3). Furthermore, suppose that there exists $\iota > 0$ such that $\inf_{k \geq 1} \{\sup_{z \in \Theta_k} \|R_{\tilde{w}^k}^{-1}(z)\|_{\tilde{w}^k}\} \geq \iota$.

(1.4) For all $k, t \geq 0$, there exists a ρ-totally retractive neighborhood Θ_* of critical point w^* such that $\{w_t^k\}$ generated by Algorithm 1 continuously remains in Θ_*.

(1.5) $f_1, f_2, \ldots, f_n$ *are retraction L-smooth with respect to retraction* R *in* Θ.

(1.6) The vector transport $\mathcal{T}$ *is isometric on* $\mathcal{M}$*, i.e., it satisfies* $\langle \mathcal{T}_\xi \eta, \mathcal{T}_\xi \zeta \rangle_{R_w(\xi)} = \langle \eta, \zeta \rangle_w$ *for any* $w \in \mathcal{M}$ *and* $\xi, \eta, \zeta \in T_w\mathcal{M}$.

(1.7) There exists a constant c_0 *such that* $\mathcal{T}$ *satisfies* $\|\mathcal{T}_\eta - \mathcal{T}_\eta^R\| \leq c_0\|\eta\|_w, \|\mathcal{T}_\eta^{-1} - (\mathcal{T}_\eta^R)^{-1}\| \leq c_0\|\eta\|_w$ *for all* $w, z \in \mathcal{U}$*, where* $\eta = R_w^{-1}(z)$ *and* $\mathcal{T}^R$ *denotes the differentiated retraction, i.e.,* $\mathcal{T}_\zeta^R \xi = \mathrm{D}R_w(\zeta)[\xi]$ *with* $\xi, \zeta \in T_w\mathcal{M}$.

(1.8) Riemannian stochastic gradient is bounded as $\mathbb{E}_{i_t}[\|\mathrm{grad} f_{i_t}(w_t)\|_{w_t}^2] < C^2$ *[14, 15, 16].*

We now provide the two key facts for bounds of $\mathcal{H}_t^k$; there exist $0 < \gamma_{nc} < \Gamma_{nc}$ such that (8) holds for non-convex functions, and there exist $0 < \gamma_c < \Gamma_c$ such that (9) holds for retraction strongly convex functions.

$$\gamma_{nc}\mathrm{id} \preceq \mathcal{H}_t^k \preceq \Gamma_{nc}\mathrm{id}, \tag{8}$$

$$\gamma_c\mathrm{id} \preceq \mathcal{H}_t^k \preceq \Gamma_c\mathrm{id}, \tag{9}$$

where $\mathbf{A} \preceq \mathbf{B}$ with $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}$ means that $\mathbf{B} - \mathbf{A}$ is positive semi-definite. These are derived in Propositions C.4 and D.3, respectively.

4.1 Global convergence analysis on non-convex functions

We first present a global convergence analysis to a critical point starting from any initialization point, which is common in a non-convex setting with additional but *mild* assumptions.

Assumption 2. *We assume that* f *is bounded below by a scalar* $f_{\inf}$*, and a decaying step size sequence* $\{\alpha_t^k\}$ *satisfies* $\sum \alpha_t^k = \infty$ *and* $\sum(\alpha_t^k)^2 < \infty$*. Furthermore, since* K *is compact, all continuous functions on* Θ *can be bounded. Therefore, there exists* $S > 0$ *such that for all* $w \in \Theta$ *and* $i \in \{1, 2, \ldots, n\}$*, we have* $\|\mathrm{grad} f(w)\|_w \leq S$ and $\|\mathrm{grad} f_i(w)\|_w \leq S$.

Theorem 4.1. *Let* $\mathcal{M}$ *be a Riemannian manifold and* $w^* \in \mathcal{M}$ *be a non-degenerate local minimizer of* f*. Consider Algorithm 1 and suppose that Assumptions 1 and 2 hold, and that the largest eigenvlue of the Riemannian Hessian of the mapping* $w \mapsto \|\mathrm{grad} f(w)\|_w^2$ *for all* $w \in \mathcal{M}$ *is upper-bounded by a uniform positive real number. Then,* $\lim_{k\to\infty} \mathbb{E}[\|\mathrm{grad} f(w_t)\|_{w_t}^2] = 0$.

4.2 Global convergence rate analysis on non-convex functions

We now present a global convergence rate analysis. This requires a *strict* selection of a fixed step size satisfying the condition below.

Theorem 4.2. *Let* $\mathcal{M}$ *be a Riemannian manifold and* $w^* \in \mathcal{M}$ *be a non-degenerate local minimizer of* f*. Consider Algorithm 1 with Options I-B and II-B, and suppose Assumption 1. Let the constants* θ *be in (5),* τ_1 *and* τ_2 *be in (6), and* L_l *be in (7).* L *is the constant in Lemma 3.4.* γ_{nc} *and* Γ_{nc} *are the constants in (8), respectively. Set* $\nu = \frac{\sqrt{L_l^2+\tau_2^2C^2\theta^2}\zeta^{1-a_2}\Gamma_{nc}}{n^{a_1/2}}$ *and* $\alpha_t^k = \alpha = \frac{\mu_0}{\sqrt{L_l^2+\tau_2^2C^2\theta^2}n^{a_1}\zeta^{a_2}\Gamma_{nc}}$*, where* $0 < a_1 < 1$*, and* $0 < a_2 < 2$*. Given sufficiently small* $\mu_0 \in (0, 1)$*, suppose that* $\varrho > 0$ *is chosen such that* $\frac{\sqrt{L_l^2+\tau_2^2C^2\theta^2}}{L\Gamma_{nc}}\gamma_{nc}\left(1 - \frac{\varrho\Gamma_{nc}}{\mu_0\gamma_{nc}}\right) \geq \frac{2\mu_0(e-1)}{\zeta^{2-a_2}\tau_1^2} + \frac{\mu_0}{n^{a_1}\zeta^{a_2}} + \frac{4\mu_0^2(e-1)}{n^{3a_1/2}\zeta^{a_2}\tau_1^2}$ *holds. Set* $m = \lfloor \frac{n^{3a_1/2}}{5\mu_0\zeta^{1-2a_2}} \rfloor$ *and* $T = mK$*. Then,*

$$\mathbb{E}[\|\mathrm{grad} f(w_{\mathrm{sol}})\|_{w_{\mathrm{sol}}}^2] \leq \frac{\sqrt{L_l^2+\tau_2^2C^2\theta^2}n^{a_1}\zeta^{a_2}[f(w^0)-f(w^*)]}{T\varrho}.$$

4.3 Local convergence rate analysis on retraction strongly convex functions

Finally, we present a local convergence rate in a neighborhood of a local minimum by introducing a *local* assumption for retraction strong convexity. This is also standard in manifold optimization. It should be noted that if we extend this local assumption to the entire manifold, as in R-SVRG [23], our rate directly results in a *global* rate. However, such a global assumption is fairly restrictive in terms of the cost functions and manifolds that can be considered and, hence, the standard literature mostly focuses on local rate analysis.

Assumption 3. *We assume that the objective function* $f_1, f_2, \ldots, f_n$ *are retraction strongly convex with respect to* R *in* Θ *(Definition 3.8). Moreover,* $\mathcal{T}$ *satisfies the locking condition [21] defined as* $\mathcal{T}_\eta \xi = \beta \mathcal{T}_\eta^R \xi$*, where* $\beta = \frac{\|\xi\|_w}{\|\mathcal{T}_\eta^R \xi\|_{R_w(\eta)}}$*, for all* $\eta, \xi \in T_w\mathcal{M}$ *and all* $w \in \mathcal{M}$.

Theorem 4.3. *Let* $\mathcal{M}$ *be a Riemannian manifold and* $w^* \in \mathcal{M}$ *be a non-degenerate local minimizer of* f*. Suppose Assumption 1 and 3 hold. Let the constants* θ *be in (5),* τ_1 *and* τ_2 *be in (6), and* L_l *be in (7).* L *and* μ *are the constants in Lemmas 3.4 and 3.5, respectively.* γ_c *and* Γ_c *are the constants in (9). Let* α *be a positive number satisfying* $\gamma_c\mu^2\tau_1^2 > 14\alpha L(L_l^2 + \tau_2^2C^2\theta^2)\Gamma_c^2$*. It then follows that for any sequence* $\{\tilde{w}^k\}$ *generated by Algorithm 1 with Option I-A under a fixed step size* $\alpha_t^k := \alpha$ *and* $m_k := m$ *converging to* w^**, there exists* $0 < K_{th} < K$ *such that for all* $k > K_{th}$*,*

$$\mathbb{E}[f(\tilde{w}^{k+1}) - f(w^*)] \leq \frac{\mu\tau_1^2+16m\alpha^2L(L_l^2+\tau_2^2C^2\theta^2)\Gamma_c^2}{2m\alpha(\gamma_c\mu^2\tau_1^2-14\alpha L\Gamma_c^2(L_l^2+\tau_2^2C^2\theta^2))}\mathbb{E}[f(\tilde{w}^k) - f(w^*)].$$

The proof structure is different from that of R-SVRG

[22, 23] due to the bound of $\mathbb{E}[\|\xi_t\|_{w_t}^2]$ and the existence of $\mathcal{H}_t^k$. We note the rate degradation from [22, 23]. Although the adaptive sampling in SQN-VR improves this degradation [30], it is still worse than that of SVRG. To the best of our knowledge, no theoretical rate result better than or equal to that of SVRG [8] has been given in the Euclidean SQN-VR [17]. This issue is a common subject of research in both the Euclidean and the Riemannian settings to improve the theoretical rate. However, it should be emphasized that R-SQN-VR shows empirically much better performances than R-SVRG, especially on an ill-conditioned problem, as shown in Figure 1.

5 Numerical comparisons

This section details a comparison of R-SQN-VR with R-SGD using a decaying step size sequence, and with R-SVRG on a fixed step size. The decaying step size sequence was $\alpha_k = \alpha(1 + \alpha\varsigma(k-1))^{-1}$. We also compared them with two Riemannian batch methods, i.e., R-SD, which is the steepest descent algorithm on Riemannian manifolds with backtracking line search [1], and R-L-BFGS, which is the Riemannian L-BFGS with the strong Wolfe condition [20, 31]. All experiments were executed in MATLAB on a 4.0 GHz Intel Core i7 PC with 16 GB RAM, and were stopped when the gradient norm was below 10^{-8} or when they reached a predefined maximum number of iterations. All results except for those of R-SD and R-L-BFGS were the *best tuned* results from multiple choices of step sizes α and a fixed $\varsigma = 10^{-3}$. This paper addresses the Riemannian centroid computation problem of the symmetric positive-definite (SPD) manifold and the low-rank matrix completion problem on the Grassmann manifold.

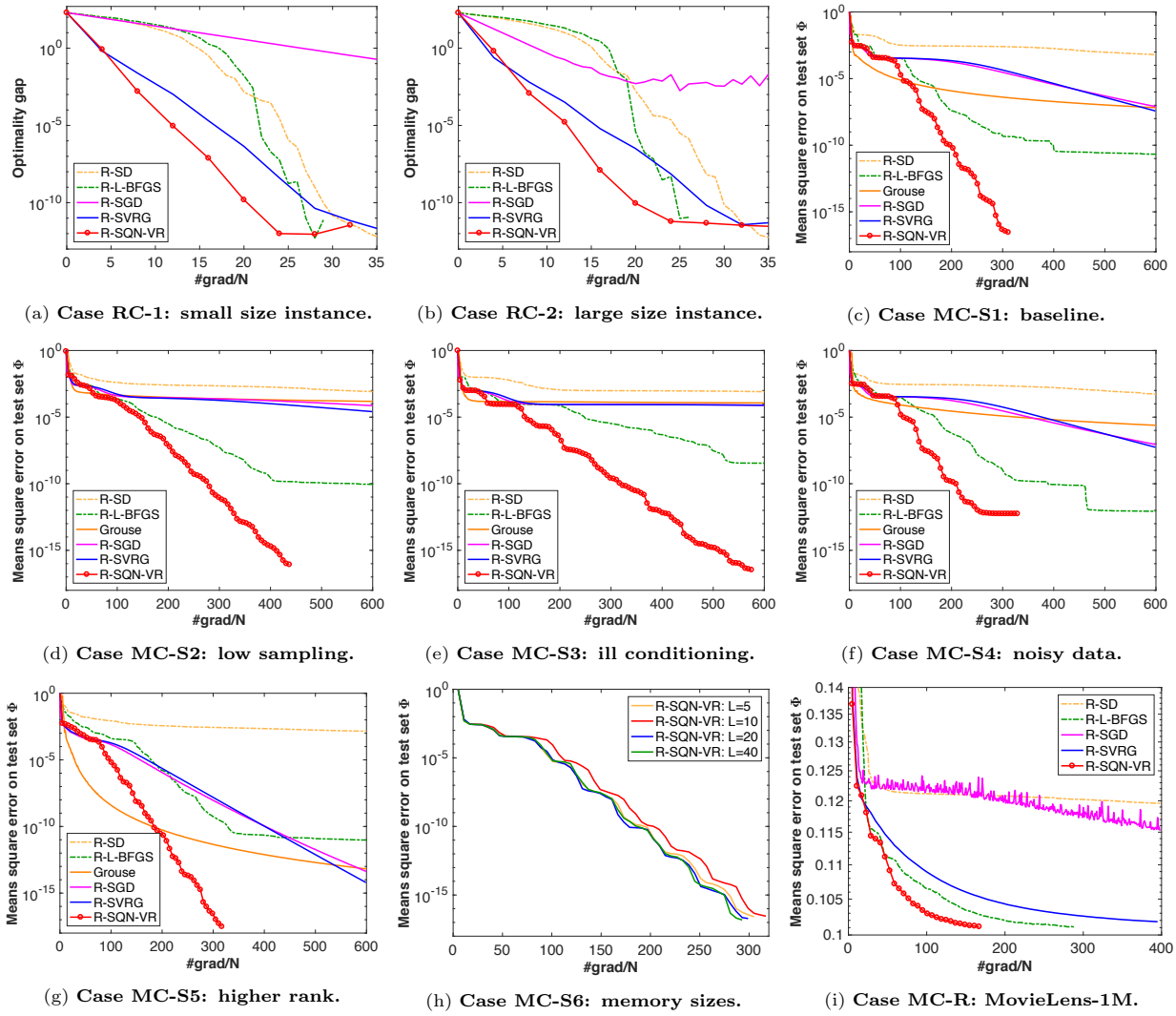
5.1 Riemannian centroid problem

The Riemannian centroid was introduced as the notion of *mean* on Riemannian manifolds by Karcher [32]. It generalizes the notion of an "average" on a manifold. Given n points on $\mathcal{S}_{++}^d$ with matrix representations $\mathbf{Q}_1, \ldots, \mathbf{Q}_n$, the Riemannian centroid is defined as the solution to the problem $\min_{\mathbf{X} \in \mathcal{S}_{++}^d} \frac{1}{2n} \sum_{i=1}^n (\mathrm{dist}(\mathbf{X}, \mathbf{Q}_i))^2$, where $\mathrm{dist}(p, q) = \| \log(p^{-1/2} q p^{-1/2}) \|_F$ represents the distance along the corresponding geodesic between elements on $\mathcal{S}_{++}^d$ with respect to the affine-invariant metric. The gradient of the loss function is computed as $\frac{1}{n} \sum_{i=1}^n -\log(\mathbf{Q}_i \mathbf{X}^{-1}) \mathbf{X}$. The first comparison is the Riemannian centroid problem on SPD matrices [31]. All experiments used a batch size fixed at 1 and $L = 4$, were initialized randomly, and were stopped when the number of iterations reached 10 for R-SVRG, and R-SQN-VR, and 60 for others. α was tuned from $\{10^{-5}, 10^{-4}, \ldots, 10^{-1}\}$. m_k and the batch size were $3n$ and 1, respectively. Figures 1(a) and (b) show the results in terms of the optimality gap when $n = 500$ with $d = 3$ (**Case RC-1**) and with $n = 1500$ (**Case RC-2**), respectively. These results reveal that R-SQN-VR yielded the best performance.

5.2 Matrix completion problem

The matrix completion problem amounts to completing an incomplete matrix $\mathbf{X}$, say of size $d \times n$, from a small number of entries by assuming a low-rank structure. If Ω is the set of known indices in $\mathbf{X}$, the rank-r matrix completion problem amounts to solving $\min_{\mathbf{U}, \mathbf{A}} \|\mathcal{P}_\Omega(\mathbf{UA}) - \mathcal{P}_\Omega(\mathbf{X})\|_F^2$, where $\mathbf{U} \in \mathbb{R}^{d \times r}, \mathbf{A} \in \mathbb{R}^{r \times n}$, and the operator $\mathcal{P}_\Omega$ acts as $\mathcal{P}_\Omega(\mathbf{X}_{pq}) = \mathbf{X}_{pq}$ if $(p, q) \in \Omega$, and $\mathcal{P}_\Omega(\mathbf{X}_{pq}) = 0$ otherwise. Partitioning $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$, the previous problem is equivalent to $\min_{\mathbf{U} \in \mathbb{R}^{d \times r}, \, \boldsymbol{a}_i \in \mathbb{R}^r} \frac{1}{n} \sum_{i=1}^n \|\mathcal{P}_{\Omega_i}(\mathbf{U}\boldsymbol{a}_i) - \mathcal{P}_{\Omega_i}(\boldsymbol{x}_i)\|_2^2$, where $\boldsymbol{x}_i \in \mathbb{R}^d$, and the operator $\mathcal{P}_{\Omega_i}$ is the sampling operator for the i-th column. Given $\mathbf{U}$, $\boldsymbol{a}_i$ admits a closed-form solution. Consequently, the problem depends only on the column space of $\mathbf{U}$, and is on the Grassmann manifold [33].

We first considered a synthetic dataset. The proposed algorithm was also compared with Grouse [2], a state-of-the-art stochastic gradient algorithm on the Grassmann manifold. The algorithms were initialized randomly as in [34]. α was tuned from $\{10^{-3}, 5 \times 10^{-3}, 10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$ for R-SGD, R-SVRG and R-SQN-VR, and $\{1, 10, 100\}$ for Grouse. We set explicitly the condition number, denoted by CN, of the matrix, representing the ratio of the maximal to the minimal singular values of the matrix. We also set the over-sampling ratio (OS) for the number of known entries. Gaussian noise was added at noise level σ as suggested in [34]. m_k and the batch size were set to $5n$ and 50, respectively. The maximum number of the outer iterations to stop was 100 for R-SVRG and R-SQN-VR, and $100(m_k + 1)$ for the others. It should be noted that this experiment evaluated the projection-based vector transport and the QR-decomposition-based retraction, which do not satisfy the locking condition, but are computationally efficient. The motivation is to show that our algorithm also empirically performs well without using the specific vector transport. The baseline problem (**MC-S1**) is a case where $n = 5000$, $d = 200$, rank $r = 5$, $L = 10$, OS $= 8$, $\sigma = 10^{-10}$, and CN $= 50$. Moreover, changing some parameters from those in **MC-S1**, we evaluated the case of lower sampling with OS $= 4$ (**MC-S2**), the ill-conditioned case with CN $= 100$ (**MC-S3**), the case involving higher noise with $\sigma = 10^{-6}$ (**MC-S4**), and that involving higher rank with $r = 10$ (**MC-S5**). The re-



(a) **Case RC-1: small size instance.** (b) **Case RC-2: large size instance.** (c) **Case MC-S1: baseline.**

(d) **Case MC-S2: low sampling.** (e) **Case MC-S3: ill conditioning.** (f) **Case MC-S4: noisy data.**

(g) **Case MC-S5: higher rank.** (h) **Case MC-S6: memory sizes.** (i) **Case MC-R: MovieLens-1M.**

Figure 1: Performance evaluations on Riemannian centroid problem and low-rank matrix completion problem.

sults of the MSE on test set Φ, which was different from the training set Ω, are shown in Figures 1(c)–(h). This gives the prediction accuracy of the missing elements. From the figures, we confirm the superior performance of R-SQN-VR. **MC-S6** for different memory sizes L reveals that a larger size does not always lead to better results, which has also been noted in [15]. Finally, we experimented with a real-world dataset, the MovieLens-1M (`http://grouplens.org/datasets/movielens/`), which contains one million ratings for $n = 3952$ movies of $d = 6040$ users. We randomly split this set into 80/10/10 percent data as train/validation/test partitions. α was chosen from $\{10^{-5}, 5\times10^{-5}, \ldots, 10^{-2}, 5\times10^{-2}\}$, the batch size was 50, $r = 10$, and $L = 10$. The algorithms were also terminated when the MSE on the validation set started to increase or the number of the outer iterations reached 100. Figure 1(i) shows the results excluding Grouse, which encountered issues with convergence on this set (**MC-R**). R-SQN-VR showed faster convergence than all the other methods.

6 Conclusions

In this paper, we proposed a Riemannian stochastic quasi-Newton algorithm with variance reduction (R-SQN-VR) that is well suited to finite-sum minimization problems. We presented a rigorous convergence analysis for taking the Hessian approximation into a variance reduction stochastic setting on a manifold. Our proposed algorithm makes the explicit use of retraction and vector transport on manifolds, which makes it appealing for larger numbers of manifolds. Numerical comparisons showed the benefits of our proposed algorithm on a number of applications.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

[2] L. Balzano, R. Nowak, and B. Recht. Online identification and tracking of subspaces from highly incomplete information. In *Allerton*, pages 704–711, 2010.

[3] B. Mishra and R. Sepulchre. R3MC: A Riemannian three-factor algorithm for low-rank matrix completion. In *IEEE CDC*, pages 1137–1142, 2014.

[4] H. Kasai and B. Mishra. Low-rank tensor completion: a Riemannian manifold preconditioning approach. In *ICML*, 2016.

[5] G. Meyer, S. Bonnabel, and R. Sepulchre. Linear regression under fixed-rank constraints: A Riemannian approach. In *ICML*, 2011.

[6] S. Bonnabel. Stochastic gradient descent on Riemannian manifolds. *IEEE Trans. on Automatic Control*, 58(9):2217–2229, 2013.

[7] H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Statistics*, pages 400–407, 1951.

[8] R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS*, pages 315–323, 2013.

[9] N. L. Roux, M. Schmidt, and F. R. Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *NIPS*, pages 2663–2671, 2012.

[10] S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *JMLR*, 14:567–599, 2013.

[11] A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS*, 2014.

[12] S. J. Reddi, A. Hefny, S. Sra, B. Poczos, and A. Smola. Stochastic variance reduction for nonconvex optimization. In *ICML*, 2016.

[13] N. N. Schraudolph, J. Yu, and S. Gunter. A stochastic quasi-Newton method for online convex optimization. In *AISTATS*, 2007.

[14] A. Mokhtari and A. Ribeiro. RES: Regularized stochastic BFGS algorithm. *IEEE Trans. on Signal Process.*, 62(23):6089–6104, 2014.

[15] X. Wang, S. Ma, D. Goldfarb, and W. Liu. Stochastic quasi-Newton methods for nonconvex stochastic optimization. *SIAM J. Optim.*, 27(2), 2016.

[16] R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer. A stochastic quasi-Newton method for large-scale optimization. *SIAM J. Optim.*, 26(2), 2016.

[17] P. Moritz, R. Nishihara, and M. I. Jordan. A linearly-convergent stochastic L-BFGS algorithm. In *AISTATS*, pages 249–258, 2016.

[18] R. Kolte, M. Erdogdu, and A. Ozgur. Accelerating SVRG via second-order information. In *OPT*, 2015.

[19] D. Gabay. Minimizing a differentiable function over a differential manifold. *Journal of Optimization Theory and Applications*, 37(2):177–219, 1982.

[20] W. Ring and B. Wirth. Optimization methods on Riemannian manifolds and their application to shape space. *SIAM J. Optim.*, 22(2):596–627, 2012.

[21] W. Huang, K. A. Gallivan, and P.-A. Absil. A Broyden class of quasi-Newton methods for Riemannian optimization. *SIAM J. Optim.*, 25(3):1660–1685, 2015.

[22] H. Sato, H. Kasai, and B. Mishra. Riemannian stochastic variance reduced gradient. *arXiv preprint: arXiv:1702.05594*, 2017.

[23] H. Zhang, S. J. Reddi, and S. Sra. Fast stochastic optimization on Riemannian manifolds. In *NIPS*, 2016.

[24] A. Roychowdhury. Accelerated stochastic quasi-Newton optimization on Riemann manifolds. *arXiv preprint arXiv:1704.01700*, 2017.

[25] A. Mokhtari and A. Ribeiro. Global convergence of online limited memory BFGS. *JMLR*, 16:3151–3181, 2015.

[26] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt: a Matlab toolbox for optimization on manifolds. *JMLR*, 15(1):1455–1459, 2014.

[27] J. Nocedal and Wright S.J. *Numerical Optimization*. Springer, New York, USA, 2006.

[28] D. Li and M. Fukushima. On the global convergence of BFGS method for nonconvex unconstrained optimization. *SIAM J. Optim.*, 11(4):1054–1064, 2011.

[29] W. Huang, P.-A. Absil, and K. A. Gallivan. A Riemannian symmetric rank-one trust-region method. *Math. Program., Ser. A*, 150:179–216, 2015.

[30] R. Zhao, W. B. Haskell, and V. Y. F. Tan. Stochastic L-BFGS: Improved convergence rates

and practical acceleration strategies. *IEEE Trans. on Signal Process.*, 66(5):1155–1169, 2017.

[31] X. Yuana, P.-A. Huang, W. Absil, and K. A. Gallivan. A Riemannian limited-memory BFGS algorithm for computing the matrix geometric mean. In *ICCS*, 2016.

[32] H Karcher. Riemannian center of mass and mollifier smoothing. *Comm. Pure Appl. Math.*, 30(5):509–541, 1977.

[33] N. Boumal and P.-A. Absil. Low-rank matrix completion via preconditioned optimization on the Grassmann manifold. *Linear Algebra and its Applications*, 475:200–239, 2015.

[34] D. Kressner, M. Steinlechner, and B. Vandereycken. Low-rank tensor completion by Riemannian optimization. *BIT Numer. Math.*, 54(2):447–468, 2014.

[35] R. Bhatia. *Positive definite matrices*. Princeton series in applied mathematics. Princeton University Press, 2007.

[36] X. Pennec, P. Fillard, and N. Ayache. A Riemannian framework for tensor computing. *Int. Jornal of Computer Vision*, 66(1):41–66, 2006.

[37] B. Jeuris, R. Vandebril, and B. Vandereycken. A survey and comparison of contemporary algorithms for computing the matrix geometric mean. *ETNA*, 2012.

[38] A. Edelman, T.A. Arias, and S.T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20(2):303–353, 1998.

[39] L. Bottou, F. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *arXiv preprint: arXiv:1606.04838*, 2016.

[40] H. Zhang and S. Sra. First-order methods for geodesically convex optimization. In *COLT*, 2016.