Riemannian Stochastic Recursive Momentum Method for non-Convex Optimization

Andi Han , Junbin Gao
Discipline of Business Analytics, The University of Sydney
{andi.han, junbin.gao}@sydney.edu.au

Abstract

We propose a stochastic recursive momentum method for Riemannian non-convex optimization that achieves a nearly-optimal complexity to find epsilon-approximate solution with one sample. The new algorithm requires one-sample gradient evaluations per iteration and does not require restarting with a large batch gradient, which is commonly used to obtain a faster rate. Extensive experiment results demonstrate the superiority of the proposed algorithm. Extensions to nonsmooth and constrained optimization settings are also discussed.

1 Introduction

We consider the problem of expectation (online) minimization over a Riemannian manifold $\mathcal{M}$:

$$\min_{x\in\mathcal{M}} F(x) := \mathbb{E}_{\omega}[f(x,\omega)], \tag{1}$$

where $F\colon \mathcal{M} \to \mathbb{R}$ is a sufficiently smooth and potentially non-convex function. When ω can be finitely sampled from its support Ω, problem (1) reduces to empirical risk (finite-sum) minimization, $\min_{x\in\mathcal{M}} F(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x)$, where n is the number of component functions.

In this paper, we focus on the case where full gradient of $F(x)$ is inaccessible as in online setting or when n is extremely large under finite-sum setting. Riemannian optimization is ubiquitous in a variety of contexts. For example, principal component analysis (PCA) and matrix completion can be formulated on Grassmann manifold [Kasai *et al.*, 2018]. In image processing, the tasks of diffusion tensor imaging segmentation and clustering can be cast on symmetric positive definite (SPD) manifold [Cheng *et al.*, 2012]. Joint diagonalization for independent component analysis (ICA) is a problem over Stiefel manifold, which is useful for signal separation [Theis *et al.*, 2009].

Riemannian geometry provides the minimal ingredients that allow unconstrained optimization methods to be properly defined. A default solver to problem (1) is Riemannian stochastic gradient descent (RSGD) [Bonnabel, 2013], which is a generalization of classic SGD [Robbins and Monro, 1951]. Recently, SGD with coordinate-wise adaptive learning rates has become predominately popular within Deep Learning community, such as RMSProp [Tieleman and Hinton, 2012] and Adam [Kingma and Ba, 2014]. These variants have been proved to improve robustness and escape saddle points faster [Staib *et al.*, 2019]. Under Riemannian optimization, the lack of a canonical coordinate system and the highly nonlinear geometry make it difficult to extend such adaptation effectively. Regardless, Roy *et al.* (2018) proposed constrained SGD with momentum (cSGD-M) and constrained RMSProp (cRMSProp) that adapt learning rates by coordinate-wise operations on matrix manifolds. However unlike Euclidean space, parallelly transporting past gradients will likely distort gradient features, such as sparsity. Also, no convergence guarantee has been provided. Bécigneul and Ganea (2018) generalized Adam-like adaptation and momentum to a product of Riemannian manifolds (referred to as RADAM and RAMSGRAD). Li *et al.* (2020) introduced Cayley-Adam tailored for Stiefel manifold, exploiting its unique geometry. The only work that proves non-convex convergence on matrix manifolds is [Kasai *et al.*, 2019], where they proposed RASA that adapts row and column subspaces of underlying manifold, ensuring a convergence rate of $\widetilde{\mathcal{O}}(1/\sqrt{T})$. This matches that of RSGD up to a logarithmic factor.

Although SGD-based methods enjoy low sampling cost, i.e. $\mathcal{O}(1)$ per iteration (one-sample), the main bottleneck that slows down convergence is the unvanished gradient variance. For this problem, variance reduction (VR) techniques are gaining increasing attention. Many methods, including Riemannian stochastic variance reduced gradient (RSVRG) [Sato *et al.*, 2019], stochastic recursive gradient (RSRG) [Kasai *et al.*, 2018], stochastic path-integrated differential estimator (RSPIDER) [Zhou *et al.*, 2019] are generalized from their Euclidean versions. The main idea is to correct for stochastic gradient deviation by periodically computing a large batch gradient. As a result, gradient variance decreases as training progresses. Besides finite-sum optimization, Riemannian VRs also enjoy favourable complexity under online setting [Zhou *et al.*, 2019; Han and Gao, 2020]. Specifically, RSVRG requires $\mathcal{O}(\epsilon^{-10/3})$ stochastic gradient queries to return an ϵ-approximate solution (see Definition 1), which improves on $\mathcal{O}(\epsilon^{-4})$ of RSGD. RSRG and RSPIDER require an even lower complexity of $\mathcal{O}(\epsilon^{-3})$. This rate has been proved to be optimal for stochastic optimization on Euclidean space under an additional mean-squared smoothness assumption [Arjevani *et al.*, 2019].

Nevertheless, these online VR methods still require com-

	Complexity	Large Batch	Small Batch	Restarting	Manifold types
RSRG/RSPIDER [Han and Gao, 2020; Zhou *et al.*, 2019]	$\mathcal{O}(\epsilon^{-3})$	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-1})$	Yes	General
RSGD [Hosseini and Sra, 2017]	$\mathcal{O}(\epsilon^{-4})$	—	$\mathcal{O}(1)$	No	General
RASA [Kasai *et al.*, 2019]	$\widetilde{\mathcal{O}}(\epsilon^{-4})$	—	$\mathcal{O}(1)$	No	Matrix manifolds
RSRM (this work)	$\widetilde{\mathcal{O}}(\epsilon^{-3})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	No	General

Table 1: Comparison of methods for Riemannian online non-convex optimization

puting a large batch gradient, i.e. $\mathcal{O}(\epsilon^{-2})$, for each epoch and the mini-batch size for each inner iteration should also be at least $\mathcal{O}(\epsilon^{-1})$. Therefore, we introduce a novel online variance reduction method, inspired by a recently proposed recursive momentum estimator [Cutkosky and Orabona, 2019]. Our **contributions** are summarized below.

- We propose a Riemannian stochastic recursive momentum (RSRM) method that achieves a gradient complexity of $\widetilde{\mathcal{O}}(\epsilon^{-3})$ for online non-convex optimization, matching the lower bound up to a logarithmic factor.
- RSRM requires only $\mathcal{O}(1)$ gradient computations per iteration and does not need restarting with a large batch gradient. Thus, our method preserves the efficiency of SGD while achieving fast convergence as VR methods.
- Our convergence result holds for general manifolds while other online adaptive methods apply to restricted manifold types, such as matrix manifolds [Roy *et al.*, 2018; Kasai *et al.*, 2019], product manifolds [Bécigneul and Ganea, 2018] and Stiefel manifold [Li *et al.*, 2020].
- Our algorithm does not adapt learning rate as in [Cutkosky and Orabona, 2019], therefore resulting in a simplified convergence analysis that does not need gradient Lipschitz assumption. This also benefits practical implementation by reducing the number of hyperparameters. Extensive experiments confirm that our method significantly outperforms other one-sample methods.

The rest of this paper is organized as follows. Section 2 introduces some useful definitions, notations and assumptions used for convergence analysis. Section 3 describes our proposed algorithm and highlights its relations with RSGD, variance reduction and stochastic momentum. Section 4 presents convergence analysis for RSRM and Section 5 evaluates the proposed method on a variety of tasks and manifolds.

2 Preliminaries

Riemannian manifold is a manifold with a smooth inner product $\langle \cdot, \cdot \rangle_x : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}$ defined on tangent space $T_x\mathcal{M}$ for every $x \in \mathcal{M}$. The induced norm on $T_x\mathcal{M}$ is $\|u\|_x := \sqrt{\langle u, u \rangle_x}$. Retraction $R_x : T_x\mathcal{M} \to \mathcal{M}$ maps a tangent vector to manifold surface satisfying $R_x(0) = x$ and $\mathrm{D}R_x(0)[u] = u$. The retraction curve is defined as $c(t) := R_x(t\xi)$ for $\xi \in T_x\mathcal{M}$. Denote $y = R_x(\xi)$. Then vector transport $\mathcal{T}_x^y$ (or equivalently $\mathcal{T}_\xi$) with respect to retraction R maps $u \in T_x\mathcal{M}$ to $\mathcal{T}_x^y u \in T_y\mathcal{M}$ along the defined retraction curve $c(t)$. Exponential map Exp_x is a special instance of retraction by restricting retraction curve to be a geodesic. Similarly, as a special case of vector transport, parallel transport P_x^y transports a tangent vector in 'parallel' while preserving its norm and direction. In this paper, we consider the more general and computationally efficient retraction and vector transport. Therefore our results can be trivially applied to exponential map and parallel transport (See Appendix). Implicitly, we consider only isometric vector transport $\mathcal{T}_x^y$, which satisfies $\langle u, v \rangle_x = \langle \mathcal{T}_x^y u, \mathcal{T}_x^y v \rangle_y$ for all $u, v \in T_x\mathcal{M}$.

Notations. For the discussions that follow, we omit the subscript for norm and inner product, which should be clear from the contexts. We define a sampling set $\mathcal{S} = \{\omega_1, ..., \omega_{|\mathcal{S}|}\}$ with cardinality $|\mathcal{S}|$. Each $\omega_{(\cdot)}$ is sampled independently from Ω. We thus denote the Riemannian stochastic gradient $\mathrm{grad} f_{\mathcal{S}}(x) := \frac{1}{|\mathcal{S}|} \sum_{\omega \in \mathcal{S}} \mathrm{grad} f(x, \omega) \in T_x\mathcal{M}$. We denote $g(t) = \mathcal{O}(h(t))$ if there exists a positive constant M and t_0 such that $g(t) \leq Mh(t)$ for all $t \geq t_0$. We use $\widetilde{\mathcal{O}}(\cdot)$ to further hide poly-logarithmic factors. We refer to $\|\cdot\|$ as the induced norm on tangent space of Riemannian manifold. Now we are ready to make some assumptions as follows.

Assumption 1. *Iterates generated by RSRM stay continuously in a neighbourhood $\mathcal{X} \subseteq \mathcal{M}$ that contains an optimal point x^*. The objective F is continuously differentiable and has bounded suboptimality. That is, for all $x \in \mathcal{X}$, $F(x) - F(x^*) \leq \Delta$, with $\Delta \geq 0$.*

Assumption 2. *Stochastic gradient $\mathrm{grad} f(x, \omega)$ is unbiased and has bounded variance. That is, for all $x \in \mathcal{X}, \omega \in \Omega$, it satisfies that*

$$\mathbb{E}_\omega \mathrm{grad} f(x, \omega) = \mathrm{grad} F(x), \text{ and}$$
$$\mathbb{E}_\omega \|\mathrm{grad} f(x, \omega) - \mathrm{grad} F(x)\|^2 \leq \sigma^2.$$

Assumption 3. *The objective F is retraction L-smooth with respect to retraction R. That is, there exists a positive constant L such that for all $x, y = R_x(\xi) \in \mathcal{X}$, we have*

$$F(y) \leq F(x) + \langle \mathrm{grad} F(x), \xi \rangle + \frac{L}{2}\|\xi\|^2.$$

These three assumptions are standard in Riemannian stochastic gradient methods [Hosseini and Sra, 2017; Kasai *et al.*, 2018]. Note that the assumption of bounded iterates in neighbourhood $\mathcal{X}$ can be made with respect to the entire manifold $\mathcal{M}$, which results in stricter conditions on the retraction and vector transport in the following assumptions. To ensure retraction L-smoothness as in Assumption 3, we require an upper-bounded Hessian property on the pullback function

$F \circ R : T_x\mathcal{M} \to \mathbb{R}$. That is, for all $x \in \mathcal{X}$ and $u \in T_x\mathcal{M}$ with unit norm, $\frac{d^2F(R_x(tu))}{dt^2} \leq L$. In the work of RASA [Kasai *et al.*, 2019], the variance bound in Assumption 2 is replaced by G-gradient Lipschitz, which requires $\|\text{grad}f(x,\omega)\| \leq G$. This, however, amounts to a stronger assumption.

According to [Arjevani *et al.*, 2019], under the first three assumptions, SGD is minimax optimal. To obtain faster convergence, one further assumption of mean-squared retraction Lipschitz is required. This assumption is a straightforward generalization of mean-squared Lipschitz on Euclidean space, which is the minimal additional requirement to achieve the complexity lower bound.

Assumption 4. *The objective F is mean-squared retraction $\tilde{L}$ Lipschitz. That is, there exists a positive constant $\tilde{L}$ such that for all $x, y = R_x(\xi) \in \mathcal{X}, \omega \in \Omega$,*

$$\mathbb{E}_\omega \|\text{grad}f(x,\omega) - \mathcal{T}_y^x \text{grad}f(y,\omega)\|^2 \leq \tilde{L}^2\|\xi\|^2$$

holds with vector transport $\mathcal{T}_x^y$ along the retraction curve $c(t) := R_x(t\xi)$.

Note that the standard assumption of retraction Lipschitzness is made with respect to parallel transport and one additional assumption that bounds the difference between vector transport and parallel transport is needed for Assumption 4 to hold [Han and Gao, 2020]. In this work, algorithm complexity is measured by the number of stochastic first order oracles to achieve ϵ-approximate solution, defined as follows.

Definition 1 (ϵ-approximate solution and SFO)**.** *ϵ-approximate solution by a stochastic algorithm is an output x such that $\mathbb{E}\|\text{grad}F(x)\|^2 \leq \epsilon^2$. One stochastic first-order oracle (SFO) outputs a stochastic gradient $\text{grad}f(x,\omega)$ given inputs x and ω drawn from Ω.*

3 Algorithms

3.1 RSGD — Variance Reduction and Momentum

Riemannian stochastic gradient makes the following retraction update: $x_{t+1} = R_{x_t}(-\eta_t \text{grad}f_{\mathcal{S}_t}(x_t))$. This allows updates to follow the negative gradient direction while staying on the manifold. Variance reduction techniques utilize past gradient information to construct a modified gradient estimator with decreasing variance. In particular, the recursive gradient estimator in RSRG/RSPIDER achieves the optimal rate of $\mathcal{O}(\epsilon^{-3})$. That is, for each outer loop, a large batch gradient is computed as $d_0 = \text{grad}f_{\mathcal{S}_0}(x_0)$, where $|\mathcal{S}_0|$ is set to n under finite-sum setting and $\mathcal{O}(\epsilon^{-2})$ under online setting. Within each inner iteration, stochastic gradient is corrected recursively based on its previous iterate:

$$d_t = \text{grad}f_{\mathcal{S}_t}(x_t) - \mathcal{T}_{x_{t-1}}^{x_t}(\text{grad}f_{\mathcal{S}_t}(x_{t-1}) - d_{t-1}), \quad (2)$$

where vector transport $\mathcal{T}_{x_{t-1}}^{x_t}$ is necessary to relate gradients on disjoint tangent spaces. To achieve the optimal complexity, mini-batch size $|\mathcal{S}_t|$ is set to be at least $\mathcal{O}(\epsilon^{-1})$. This choice of batch size can become very large, especially when we desire more accurate solutions.

On the other hand, stochastic gradient with momentum is not new on Euclidean space, while the first paper that presents such an idea on Riemannian manifold is [Roy *et al.*, 2018].

Algorithm 1 Riemannian SRM

1: **Input:** Step size η_t, recursive momentum parameter ρ_t, Initial point x_1.
2: Compute $d_1 = \text{grad}f_{\mathcal{S}_1}(x_1)$.
3: **for** $t = 1, ..., T$ **do**
4: Update $x_{t+1} = R_{x_t}(-\eta_t d_t)$.
5: Compute $d_{t+1} = \text{grad}f_{\mathcal{S}_{t+1}}(x_{t+1}) + (1 - \rho_{t+1})\mathcal{T}_{x_t}^{x_{t+1}}(d_t - \text{grad}f_{\mathcal{S}_{t+1}}(x_t))$.
6: **end for**
7: **Output:** $\tilde{x}$ uniformly chosen at random from $\{x_t\}_{t=1}^T$.

They simply take a combination of current stochastic gradient and transported momentum, given as

$$d_t = \rho_t \mathcal{T}_{x_{t-1}}^{x_t} d_{t-1} + (1-\rho_t)\text{grad}f_{\mathcal{S}_t}(x_t), \quad (3)$$

where ρ_t is commonly set to be 0.9. This idea has then been used in generalizing Adam and AMSGrad to Riemannian optimization [Bécigneul and Ganea, 2018], where they only established convergence on a product of manifolds for geodesically convex functions. Even on Euclidean space, the effectiveness of stochastic momentum over vanilla SGD has remained an open question.

3.2 Proposed RSRM

Our proposed RSRM is given in Algorithm 1 where we extend the recursive momentum estimator, originally introduced in [Cutkosky and Orabona, 2019; Tran-Dinh *et al.*, 2019]:

$$\begin{aligned} d_t &= \rho_t \text{grad}f_{\mathcal{S}_t}(x_t) + (1-\rho_t)\big(\text{grad}f_{\mathcal{S}_t}(x_t) \\ &\quad - \mathcal{T}_{x_{t-1}}^{x_t}(\text{grad}f_{\mathcal{S}_t}(x_{t-1}) - d_{t-1})\big) \quad (4) \\ &= \text{grad}f_{\mathcal{S}_t}(x_t) + (1-\rho_t)\mathcal{T}_{x_{t-1}}^{x_t}\big(d_{t-1} - \text{grad}f_{\mathcal{S}_t}(x_{t-1})\big), \end{aligned}$$

which hybrids stochastic gradient with the recursive gradient estimator in (2) for $\rho_t \in [0,1]$. This can be also viewed as combining momentum estimator in (3) with a scaled difference of $\text{grad}f_{\mathcal{S}_t}(x_t) - \mathcal{T}_{x_{t-1}}^{x_t}\text{grad}f_{\mathcal{S}_t}(x_{t-1})$. Note that we recover vanilla RSGD when $\rho_t = 1$ and the recursive estimator in (2) when $\rho_t = 0$. As we will demonstrate in Section 4, ρ_t should be decreasing rather than fixed, thereby enabling a smooth transition from RSGD to RSRG. As a result, we do not require restarting the algorithm to achieve the optimal convergence. One remark is that the idea of transition from SGD to VR also appears in many batch size adaptation strategies [Ji *et al.*, 2019; Han and Gao, 2020].

Compared with algorithm designs in Euclidean versions of SRM [Cutkosky and Orabona, 2019; Tran-Dinh *et al.*, 2019], our formulation and parameter settings are largely different. Specifically, Cutkosky and Orabona (2019) further adapt the recursive momentum parameter ρ_t to the learning rate η_t where the latter itself is adapted to the norm of stochastic gradient. This is claimed to relieve the parameter tuning process. However, they reintroduce three parameters, which are even less intuitive to be tuned (even though some are fixed to a default value). As shown in Section 4, we only require tuning the initial step size η_0 and initial momentum parameter ρ_0 (where the latter can be fixed to a good default value). Furthermore, the adaptive step size requires a uniform gradient Lipschitz condition, the same as in [Kasai *et al.*, 2019]

and also a uniform smoothness assumption, which is stronger than mean-squared smoothness in our setting. On the other hand, Tran-Dinh *et al.* (2019) replaces $\text{grad}f_{\mathcal{S}_t}(x_t)$ in (4) with $\text{grad}f_{\mathcal{B}_t}(x_t)$ where $\mathcal{B}_t$ is independent of $\mathcal{S}_t$. This increases sampling complexity per iteration and also complicates its convergence analysis. In addition, they still require a large initial batch size $|\mathcal{S}_0| = \mathcal{O}(\epsilon^{-1})$ while our $|\mathcal{S}_0| = \mathcal{O}(1)$.

4 Convergence Results

In this section, we prove convergence of RSRM. We first present a Lemma that bounds the estimation error of the recursive momentum estimator.

Lemma 1 (Estimation error bound)**.** *Suppose Assumptions 1 to 4 hold and consider Algorithm 1. Then we have*

$$\begin{aligned}&\mathbb{E}\|d_{t+1} - \text{grad}F(x_{t+1})\|^2\\ &\leq (1-\rho_{t+1})^2\big(1+\frac{4\eta_t^2\tilde{L}^2}{|\mathcal{S}_{t+1}|}\big)\mathbb{E}\|d_t - \text{grad}F(x_t)\|^2\\ &+ \frac{4(1-\rho_{t+1})^2\eta_t^2\tilde{L}^2}{|\mathcal{S}_{t+1}|}\mathbb{E}\|\text{grad}F(x_t)\|^2 + \frac{2\rho_{t+1}^2\sigma^2}{|\mathcal{S}_{t+1}|}.\end{aligned}$$

The proof of this Lemma can be found in Appendix where it follows an idea similar to the bound in RSRG/RSPIDER [Han and Gao, 2020]. The key difference is that we further use $\mathbb{E}\|d_t\|^2 \leq 2\mathbb{E}\|d_t - \text{grad}F(x_t)\|^2 + 2\mathbb{E}\|\text{grad}F(x_t)\|^2$ to show dependence on the full gradient. Based on the claims in [Cutkosky and Orabona, 2019], we consider $\rho_t = \mathcal{O}(t^{-2/3})$ and $\eta_t = \mathcal{O}(t^{-1/3})$, so that $\mathbb{E}\|d_{t+1} - \text{grad}F(x_{t+1})\|^2 = \mathcal{O}(t^{-2/3} + \mathbb{E}\|\text{grad}F(x_t)\|^2)$. To see this, denote $s_{t+1} = d_{t+1} - \text{grad}F(x_{t+1})$. Then by noting that $(1-\rho_t)^2 \leq 1-\rho_t \leq 1$ and $\eta_t^2 \leq \eta_t \leq 1$, Lemma 1 suggests that $\mathbb{E}\|s_{t+1}\|^2 \leq \mathcal{O}(1-t^{-2/3})\mathbb{E}\|s_t\|^2 + \mathcal{O}(t^{-2/3})\mathbb{E}\|\text{grad}F(x_t)\|^2 + \mathcal{O}(t^{-4/3})$. Simply setting $\mathbb{E}\|s_{t+1}\|^2 = \mathbb{E}\|s_t\|^2$ yields the result. This implies that $\mathbb{E}\|\text{grad}F(x_t)\|^2 = \mathcal{O}(T^{-2/3})$, which matches the optimal rate of convergence. This claim is stated formally in the following Theorem. For simplicity, we consider $|\mathcal{S}_t| = b$ for all t.

Theorem 1 (Convergence and complexity of RSRM)**.** *Suppose Assumptions 1 to 4 hold and consider Algorithm 1 with* $\eta_t = c_\eta(t+1)^{-1/3}$, $\rho_t = c_\rho t^{-2/3}$ *where* $c_\eta \leq \frac{1}{L}$ *and* $c_\rho = (\frac{10\tilde{L}^2}{b} + \frac{1}{3})c_\eta^2$. *Then we have*

$$\begin{aligned}\mathbb{E}\|\text{grad}F(\tilde{x})\|^2 &= \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\text{grad}F(x_t)\|^2\\ &\leq \mathcal{O}(\frac{M}{T^{2/3}}) = \widetilde{\mathcal{O}}(\frac{1}{T^{2/3}}),\end{aligned}$$

with $M := (6\Delta + \frac{\sigma^2}{2\tilde{L}^2} + \frac{\sigma^2\ln(T+1)}{\tilde{L}^2})/c_\eta$. *To get ϵ-approximate solution, we require an SFO complexity of* $\widetilde{\mathcal{O}}(\epsilon^{-3})$.

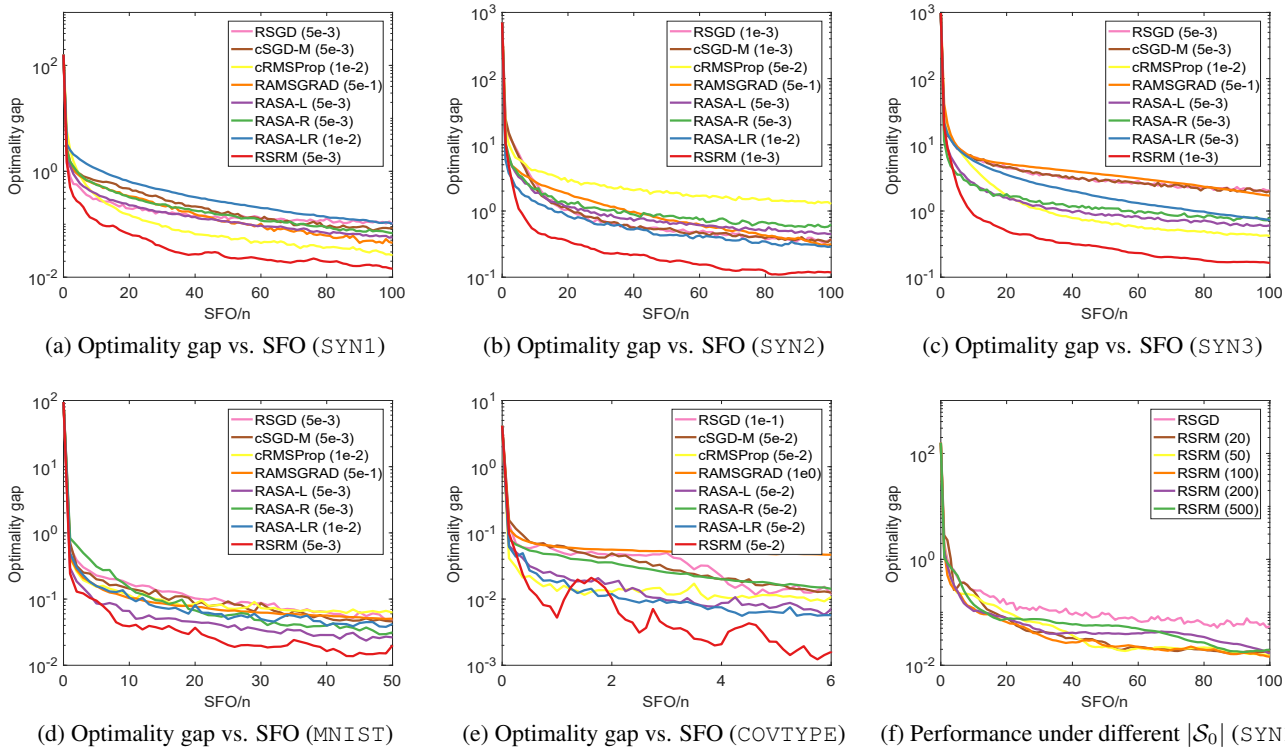
Theorem 1 claims that RSRM achieves a nearly-optimal complexity of $\widetilde{\mathcal{O}}(\epsilon^{-3})$ with one-sample gradient, i.e. $b = \mathcal{O}(1)$. And specifically under noiseless case where $\sigma^2 = 0$, we can further improve this result to the lower bound complexity $\mathcal{O}(\epsilon^{-3})$. One final remark can be made that our step size decays at a rate of $\mathcal{O}(t^{-1/3})$, which is slower compared to the SGD-based rate of $\mathcal{O}(t^{-1/2})$. This step size sequence is crucial for achieving the faster convergence, coupled with gradually reduced variance.

5 Experiments

In this section, we compare our proposed RSRM with other one-sample online methods. The benchmark is the standard RSGD [Bonnabel, 2013]. We also consider cSGD-M and cRMSProp [Roy *et al.*, 2018] where past gradients are transported by a vector transport operator. For cRMSProp, we do not project and vector-transport its adaptation term, which is an element-wise square of stochastic gradient. Instead, we treat it as an element in the ambient Euclidean space and therefore only project the resulting scaled gradient after applying this term. This modification yields the full-matrix version of the vectorized variant of RASA and turns out to significantly outperform its original design. Also we compare with RAMSGRAD [Bécigneul and Ganea, 2018], which is designed for product manifolds. We thus modify the gradient momentum similar as in [Roy *et al.*, 2018] while accumulating square norm of gradient instead of element-wise square. Hence, it only adapts the step size rather than the gradient. Finally, we consider RASA [Kasai *et al.*, 2019] that adapts column and row subspaces of matrix manifolds. We similarly label its variants as RASA-L, RASA-R and RASA-LR to respectively represent adapting row (left) subspace, column (right) subspace and both.

All methods start with the same initialization and terminate when the maximum iteration number is reached. For competing methods, we consider a square-root decaying step size $\eta_t = \eta_0 t^{-1/2}$, suggested in [Kasai *et al.*, 2019]. We set the parameters of RSRM according to the theory, i.e. $\eta_t = \eta_0 t^{-1/3}$ and $\rho_t = \rho_0 t^{-2/3}$. A default value of $\rho_0 = 0.1$ provides good empirical performance. For all methods, η_0 are selected from $\{1, 0.5, 0.1, ..., 0.005, 0.001\}$. The gradient momentum parameter in cSGD-M and RAMSGRAD is set to be 0.9 and the adaptation momentum parameter in cRMSProp, RAMSGRAD and RASA is set to be 0.999. We choose a mini-batch size of 5 for RSRM and 10 for all other algorithms to ensure an identical per-iteration cost of gradient evaluation. The initial batch size for RSRM is fixed to be 100 (except for the problem of ICA where it is set to be 200). All algorithms are coded in Matlab and experiments are conducted on a laptop with a i5-8600 3.1GHz CPU processor.

We consider principal component analysis (PCA) on Grassmann manifold, joint diagonalization of independent component analysis (ICA) on Stiefel manifold and computing Riemannian centroid (RC) on SPD manifold. Stiefel manifold $\text{St}(r,d) = \{\mathbf{X} \in \mathbb{R}^{d\times r} : \mathbf{X}^T\mathbf{X} = \mathbf{I}_r\}$ is defined as the set of $d \times r$ column orthonormal matrices, which is a natural embedded submanifold of $\mathbb{R}^{d\times r}$. Grassmann manifold $\mathcal{G}(r,d)$ is the set of r-dimensional subspaces in $\mathbb{R}^d$. One representation of Grassmann manifold is by a Stiefel matrix $\mathbf{X} \in \mathbb{R}^{d\times r}$ with orthonormal columns that span the subspace. This representation is not unique. Indeed, any $\mathbf{XR}$ for $\mathbf{R} \in O(r)$ is equivalent to $\mathbf{X}$, where $O(r)$ is the orthogonal group of dimension r. Hence, Grassmann manifold can be viewed as a quotient of Stiefel manifold, written as



(a) Optimality gap vs. SFO (SYN1) (b) Optimality gap vs. SFO (SYN2) (c) Optimality gap vs. SFO (SYN3)

(d) Optimality gap vs. SFO (MNIST) (e) Optimality gap vs. SFO (COVTYPE) (f) Performance under different $|\mathcal{S}_0|$ (SYN1)

Figure 1: PCA problems on Grassmann manifold

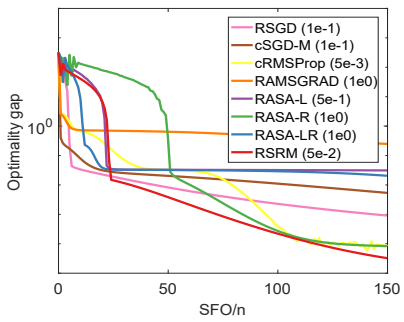
$\mathrm{St}(r,d)/O(r)$. Finally, SPD manifold $\mathcal{S}_{++}^d$ is the set of $d \times d$ symmetric positive definite matrices, which forms the interior of a convex cone embedded in $\mathbb{R}^{d(d+1)/2}$. Manifold retractions and vector transports used in the following experiments are discussed in Appendix.

5.1 PCA on Grassmann Manifold

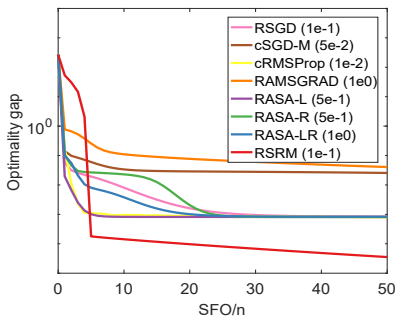
Consider n samples, represented by $\mathbf{x}_i \in \mathbb{R}^d, i = 1, ..., n$. PCA aims to find a subspace where projection onto this subspace minimizes reconstruction error. This defines a problem on Grassmann manifold, written as $\min_{\mathbf{U}\in\mathcal{G}(r,d)} \frac{1}{n}\sum_{i=1}^n \|\mathbf{x}_i - \mathbf{U}\mathbf{U}^T\mathbf{x}_i\|^2$. We first test RSRM on a baseline synthetic dataset (SYN1) with $(n,d,r) = (10^4, 10^2, 10)$. Then we increase dimension to 500 (SYN2) and consider a higher rank case with $r = 20$ (SYN3). In addition, we consider two empirical datasets, MNIST [LeCun *et al.*, 1998] with $(n,d,r) = (60000, 784, 10)$ and COVTYPE from LibSVM [Chang and Lin, 2011] with $(n,d,r) = (581012, 54, 10)$. We measure performance in terms of the difference between current function value to the minimum, pre-calculated using Matlab function PCA. Convergence results and the best-tuned η_0 are shown in Figure 1. We find that RSRM consistently outperforms others on every dataset (Figure 1(a) to 1(e)). It is also observed that on 'easy' datasets, such as SYN1 and SYN2, adaptive methods perform similarly compared with well-tuned SGD whereas RSRM demonstrates its clear advantage. Figure 1(f) shows that RSRM seems to be insensitive to the initial batch size and surprisingly, larger batch size provides no benefit for this problem.

5.2 ICA on Stiefel Manifold

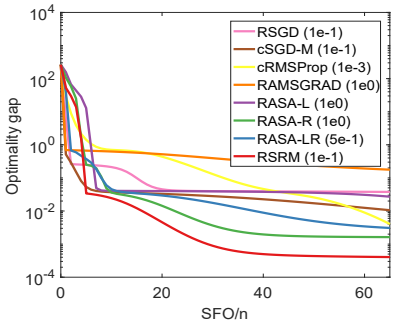
ICA (or blind source separation) aims to recover underlying components of observed multivariate data by assuming mutual independence of source signals. Joint diagonalization is a useful pre-processing step that searches for a pseudo-orthogonal matrix (i.e. Stiefel matrix) [Theis *et al.*, 2009] by solving $\min_{\mathbf{U}\in\mathrm{St}(r,d)} -\frac{1}{n}\sum_{i=1}^n \|\mathrm{diag}(\mathbf{U}^T\mathbf{X}_i\mathbf{U})\|_F^2$ with $\mathrm{diag}(\mathbf{A})$ returning diagonal elements of matrix $\mathbf{A}$. The symmetric matrices $\mathbf{X}_i \in \mathbb{R}^{d\times d}$ can be time-lagged covariance matrices or cumulant matrices constructed from the observed signals. We consider three image datasets described as follows. YALEB [Wright *et al.*, 2008] collects $n = 2414$ face images taken from various lighting environments. CIFAR100 [Krizhevsky *et al.*, 2009] contains $n = 60000$ images of 100 objects and COIL100 [Nene *et al.*, 1996] is made up of $n = 7200$ images from 100 classes. To construct covariance representations from these datasets, we first downsize each image to 32×32 before applying Gabor-based kernel to extract Gabor features. Then the feature information is fused in a region covariance descriptors of size 43×43. We choose $r = d = 43$ for all problems and the results are presented in Figure 2. Optimal solution is obtained by running RSRM for sufficiently long. Similarly, we find that RSRM, although showing slow progress at the initial epochs, quickly converges to a lower value compared with others. This is mainly attributed to its variance reduction nature.



(a) Optimality gap vs. SFO (YALEB)

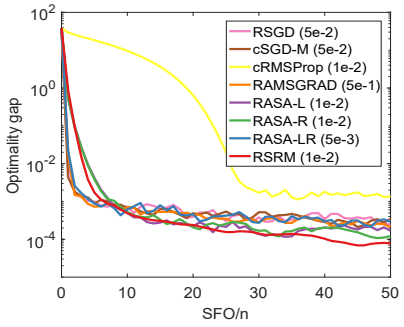


(b) Optimality gap vs. SFO (CIFAR100)

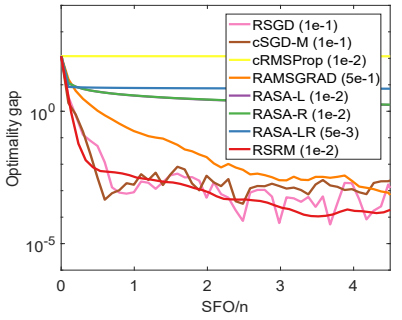


(c) Optimality gap vs. SFO (COIL100)

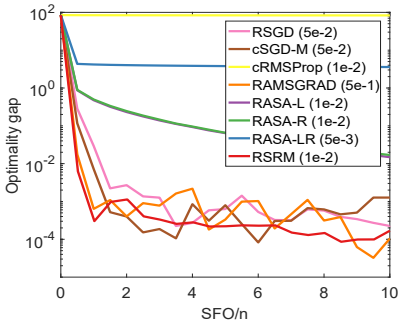
Figure 2: ICA problems on Stiefel manifold



(a) Optimality gap vs. SFO (SYN1)



(b) Optimality gap vs. SFO (YABLEB)



(c) Optimality gap vs. SFO (KYLBERG)

Figure 3: RC problems on SPD manifold

5.3 RC on SPD Manifold

Computing Riemannian centroid on SPD manifold $\mathcal{S}_{++}^d$ are fundamental in many computer vision tasks [Cheng *et al.*, 2012]. The problem concerns finding a mean representation of a set of SPD matrices, $\mathbf{X}_i$. The geodesic distance on $\mathcal{S}_{++}^d$ induced by Affine Invariant Riemannian Metric (AIRM) is $d^2(\mathbf{X}_1, \mathbf{X}_2) = \|\log(\mathbf{X}_1^{-1/2} X_2 \mathbf{X}_1^{-1/2})\|_F^2$ where $\log(\cdot)$ is the principal matrix logarithm. Riemannian centroid with respect to this distance is obtained by solving $\min_{\mathbf{C}\in\mathcal{S}_{++}^d} \frac{1}{n}\sum_{i=1}^n d^2(\mathbf{C}, \mathbf{X}_i)$. We first consider a simulated dataset consisting of $n = 5000$ SPD matrices in $\mathbb{R}^{10\times 10}$, each with a condition number of 20. Then we test our methods on YALEB face dataset [Wright *et al.*, 2008] and KYLBERG [Kylberg, 2014] dataset that consists of $n = 4480$ texture images of 28 classes. For each pixel, we generate a 5-dimensional feature vector ($d = 5$), including pixel intensity, first-order and second-order gradients. Subsequently, the covariance representation is similarly constructed for each image. Convergence results are shown in Figure 3 where the optimal solutions are calculated by Riemannian Barzilai-Borwein algorithm [Iannazzo and Porcelli, 2018]. By examining the figures, we also verify superiority of RSRM where it enjoys a more stable convergence due to variance reduction and sometimes converges to a lower objective value, as shown in Figure 3(a) and (b). For the two real datasets, cRMSProp and RASA fails to perform comparably.

5.4 Additional Experiment Results

More experiment results are included in Appendix. To examine sensitivity of algorithm performance to different initializations, we include three additional independent runs for each problem instance. We observe that proposed RSRM is superior regardless of initializations and yields more robust performance against its alternatives. We also find RSRG to be time-efficient from the convergence plots against runtime.

6 Perspectives

Since RSRM only requires $\mathcal{O}(1)$ gradient queries, it is readily implementable for deep learning tasks with manifold constraints. For example, the use of orthonormality constraint in neural network requires optimizing over Stiefel manifold [Li *et al.*, 2020]. For this purpose, our algorithm can be implemented in some manifold deep learning libraries, such as McTorch [Meghwanshi *et al.*, 2018].

More promisingly, we extend RSRM for nonsmooth and constrained optimization on Riemannian manifolds in Appendix. We also include a complete proof roadmap for convergence analysis along with necessary assumptions. With an improved convergence guarantee, we believe RSRM can outperform the default Riemannian stochastic proximal gradient and Riemannian stochastic Frank-Wolfe solvers. It is worth mentioning that the formulations we propose are different to standard references, which should be of interest in itself.

References

[Arjevani *et al.*, 2019] Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *arXiv:1912.02365*, 2019.

[Bécigneul and Ganea, 2018] Gary Bécigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. *arXiv:1810.00760*, 2018.

[Bonnabel, 2013] Silvere Bonnabel. Stochastic gradient descent on riemannian manifolds. *Transactions on Automatic Control*, 58(9):2217–2229, 2013.

[Chang and Lin, 2011] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27, 2011.

[Cheng *et al.*, 2012] Guang Cheng, Hesamoddin Salehian, and Baba C Vemuri. Efficient recursive algorithms for computing the mean diffusion tensor and applications to dti segmentation. In *European Conference on Computer Vision*, pages 390–401. Springer, 2012.

[Cutkosky and Orabona, 2019] Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. In *Advances in Neural Information Processing Systems*, pages 15236–15245, 2019.

[Han and Gao, 2020] Andi Han and Junbin Gao. Variance reduction for riemannian non-convex optimization with batch size adaptation. *arXiv:2007.01494*, 2020.

[Hosseini and Sra, 2017] Reshad Hosseini and Suvrit Sra. An alternative to em for Gaussian mixture models: Batch and stochastic Riemannian optimization. *arXiv:1706.03267*, 2017.

[Iannazzo and Porcelli, 2018] Bruno Iannazzo and Margherita Porcelli. The Riemannian Barzilai–Borwein method with nonmonotone line search and the matrix geometric mean computation. *IMA Journal of Numerical Analysis*, 38(1):495–517, 2018.

[Ji *et al.*, 2019] Kaiyi Ji, Zhe Wang, Bowen Weng, Yi Zhou, Wei Zhang, and Yingbin Liang. History-gradient aided batch size adaptation for variance reduced algorithms. *arXiv:1910.09670*, 2019.

[Kasai *et al.*, 2018] Hiroyuki Kasai, Hiroyuki Sato, and Bamdev Mishra. Riemannian stochastic recursive gradient algorithm. In *International Conference on Machine Learning*, pages 2516–2524, 2018.

[Kasai *et al.*, 2019] Hiroyuki Kasai, Pratik Jawanpuria, and Bamdev Mishra. Riemannian adaptive stochastic gradient algorithms on matrix manifolds. *arXiv:1902.01144*, 2019.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[Kylberg, 2014] G Kylberg. The kylberg texture dataset v. 1.0. external report (blue series) 35, centre for image analysis, swedish university of agricultural sciences and uppsala university, uppsala, sweden (2011). 2014.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Li *et al.*, 2020] Jun Li, Li Fuxin, and Sinisa Todorovic. Efficient Riemannian optimization on the Stiefel manifold via the Cayley transform. *arXiv:2002.01113*, 2020.

[Meghwanshi *et al.*, 2018] Mayank Meghwanshi, Pratik Jawanpuria, Anoop Kunchukuttan, Hiroyuki Kasai, and Bamdev Mishra. Mctorch, a manifold optimization library for deep learning. *arXiv:1810.01811*, 2018.

[Nene *et al.*, 1996] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-100). 1996.

[Robbins and Monro, 1951] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

[Roy *et al.*, 2018] Soumava Kumar Roy, Zakaria Mhammedi, and Mehrtash Harandi. Geometry aware constrained optimization techniques for deep learning. In *Conference on Computer Vision and Pattern Recognition*, pages 4460–4469, 2018.

[Sato *et al.*, 2019] Hiroyuki Sato, Hiroyuki Kasai, and Bamdev Mishra. Riemannian stochastic variance reduced gradient algorithm with retraction and vector transport. *SIAM Journal on Optimization*, 29(2):1444–1472, 2019.

[Staib *et al.*, 2019] Matthew Staib, Sashank J Reddi, Satyen Kale, Sanjiv Kumar, and Suvrit Sra. Escaping saddle points with adaptive gradient methods. *arXiv:1901.09149*, 2019.

[Theis *et al.*, 2009] Fabian J Theis, Thomas P Cason, and P-A Absil. Soft dimension reduction for ica by joint diagonalization on the stiefel manifold. In *International Conference on Independent Component Analysis and Signal Separation*, pages 354–361. Springer, 2009.

[Tieleman and Hinton, 2012] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2):26–31, 2012.

[Tran-Dinh *et al.*, 2019] Quoc Tran-Dinh, Nhan H Pham, Dzung T Phan, and Lam M Nguyen. Hybrid stochastic gradient descent algorithms for stochastic nonconvex optimization. *arXiv:1905.05920*, 2019.

[Wright *et al.*, 2008] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2008.

[Zhou *et al.*, 2019] Pan Zhou, Xiaotong Yuan, Shuicheng Yan, and Jiashi Feng. Faster first-order methods for stochastic non-convex optimization on Riemannian manifolds. *arXiv:1811.08109*, 2019.