Decentralized Projected Riemannian Stochastic Recursive Momentum Method for Nonconvex Optimization

Kangkang Deng[1], **Jiang Hu**[2*]

[1]Department of Mathematics, National University of Defense Technology, Changsha, 410073, China
[2]Department of Mathematics, University of California, Berkeley, CA 94720, US
freedeng1208@gmail.com, hujiangopt@gmail.com

Abstract

This paper studies decentralized optimization over a compact submanifold within a communication network of n nodes, where each node possesses a smooth non-convex local cost function, and the goal is to jointly minimize the sum of these local costs. We focus particularly on the online setting, where local data is processed in real-time as it streams in, without the need for full data storage. We propose a decentralized projected Riemannian stochastic recursive momentum (DPRSRM) method that employs local hybrid stochastic gradient estimators and uses the network to track the global gradient. DPRSRM achieves an oracle complexity of $\mathcal{O}(\epsilon^{-\frac{3}{2}})$, outperforming existing methods that have at most $\mathcal{O}(\epsilon^{-2})$ complexity. Our method requires only $\mathcal{O}(1)$ gradient evaluations per iteration for each local node and does not require restarting with a large batch gradient. Furthermore, we demonstrate the effectiveness of our proposed methods compared to state-of-the-art ones through numerical experiments on principal component analysis problems and low-rank matrix completion.

Introduction

Decentralized optimization has emerged as a prominent area of research, particularly for its application in large-scale systems, such as sensor networks, distributed computing, and machine learning. In these contexts, data is often partitioned across numerous nodes, rendering centralized optimization approaches impractical due to challenges such as privacy limitations and restricted computational resources. In this work, we are concerned with the distributed smooth optimization on a compact submanifold

$$\begin{aligned} \min \quad & \frac{1}{n}\sum_{i=1}^{n} f_i(x_i), \\ \text{s.t.} \quad & x_1 = \cdots = x_n, \\ & x_i \in \mathcal{M}, \ \ \forall\, i = 1, 2, \ldots, n, \end{aligned} \tag{1}$$

where n is the number of nodes, f_i is the smooth nonconvex local objective at the i-th node, and $\mathcal{M}$ is a (nonconvex) compact smooth submanifold embedded in $\mathbb{R}^{d\times r}$ with the extrinsic dimensions (d, r), e.g., the Stiefel manifold $\mathrm{St}(d,r) := \{x \in \mathbb{R}^{d\times r} : x^\top x = I_r\}$. Problem (1) is prevalent in machine learning, signal processing, and deep learning, see, e.g., the principal component analysis (Ye and Zhang 2021), the low-rank matrix completion (Boumal and Absil 2015; Kasai, Jawanpuria, and Mishra 2019; Deng and Hu 2023), the low-dimensional subspace learning (Ando, Zhang, and Bartlett 2005; Mishra et al. 2019), and the deep neural networks with batch normalization (Cho and Lee 2017; Hu et al. 2024). One challenge in solving (1) comes from the nonconvexity of the manifold constraint, which causes difficulty in achieving the consensus (Chen et al. 2021; Deng and Hu 2023; Hu et al. 2023; Deng, Hu, and Wang 2023).

In this paper, we investigate an online setting where each node i interacts with its local cost function f_i through a stochastic first-order oracle (SFO). This SFO setting is particularly relevant in various online learning and expected risk minimization problems, where the noise introduced by the SFO stems from the variability of sampling over streaming data received at each node. A notable example is online principal component analysis (Cardot and Degras 2018). Our primary focus is on the oracle complexity, defined as the total number of SFO queries required at each node to compute an ϵ-stationary point tuple $\{x_1, \cdots, x_n\}$, as formalized in Definition 1.

Related Works

Decentralized optimization in Euclidean space (i.e., $\mathcal{M} = \mathbb{R}^{d\times r}$) has been extensively studied over the past few decades (see, e.g., (Bianchi and Jakubowicz 2012; Shi et al. 2015; Xu et al. 2015; Qu and Li 2017; Di Lorenzo and Scutari 2016; Tatarenko and Touri 2017; Wai et al. 2017; Yuan et al. 2018; Hong, Hajinezhad, and Zhao 2017; Zeng and Yin 2018; Scutari and Sun 2019; Sun, Lu, and Hong 2020)). However, since problem (1) involves a manifold $\mathcal{M}$, which is often nonlinear and nonconvex, these works may fail when directly applied to solve problem (1).

Perhaps the earliest works for solving (1) are (Mishra et al. 2019; Shah 2017). However, these methods require an asymptotically infinite number of consensus steps for convergence, which limits their practical applicability. For the case where $\mathcal{M}$ is the Stiefel manifold, (Chen et al. 2021) propose a decentralized Riemannian gradient descent method and its gradient-tracking version. To use a single

*Corresponding author.

step of consensus, augmented Lagrangian methods (Wang and Liu 2022, 2023) are also investigated, where a different stationarity is used. (Sun et al. 2024) propose a decentralized retraction-free gradient tracking algorithm, and show that it exhibits ergodic $\mathcal{O}(1/K)$ convergence rate. However, these studies rely on the orthogonal structure of the Stiefel manifold. Recently, (Deng and Hu 2023) used the projection operators instead of retractions and expanded the distributed Riemannian gradient descent algorithm and the gradient tracking version to the compact submanifolds of Euclidean space. Moreover, the integration of decentralized manifold optimization with other algorithms has also been proposed, including the conjugate gradient algorithm (Chen et al. 2024) and the natural gradient method (Hu et al. 2023). Furthermore, (Hu and Deng 2024) achieves single-step consensus for the general compact submanifold by carefully elaborating on the smoothness structure and the asymptotic 1-Lipschitz continuity of the projection operator associated with the submanifold geometry.

Several studies have focused on the finite-sum setting of problem (1), where $f_i = \frac{1}{m}\sum_{r=1}^{m_i} f_{i,r}$. (Chen et al. 2021) propose a decentralized Riemannian stochastic gradient descent method. By combining the variable sample size gradient approximation method with the gradient tracking dynamic, (Zhao, Wang, and Lei 2024) propose a distributed Riemannian stochastic optimization algorithm on the Stiefel manifold. Although both methods can also be used in the online setting, the oracle complexities are $\mathcal{O}(\epsilon^{-2})$, which is not optimal. It is worth noting that the decentralized variance reduced method (Wang and Liu 2023) has been studied. However, they need to periodically calculate the full gradient, which is not suitable for the online setting.

Contribution

In this paper, we propose DPRSRM, a novel online variance-reduced method for decentralized non-convex manifold optimization with stochastic first-order oracles (SFO).

- To achieve fast and robust performance, the DPRSRM algorithm is built upon gradient tracking (Chen et al. 2021; Deng and Hu 2023) and a stochastic gradient momentum estimator (Cutkosky and Orabona 2019; Han and Gao 2021b), which can be viewed as online variance reduction method. The only existing decentralized stochastic variance-reduced manifold optimization algorithm is the VRSGT proposed by (Wang and Liu 2023). Note that VRSGT is a double-loop algorithm that requires very large minibatch sizes. Conversely, the proposed DPRSRM is a single-loop algorithm with $\mathcal{O}(1)$ oracle queries per update. Numerical experiments demonstrate the effectiveness of the proposed methods compared to state-of-the-art ones through eigenvalue problems and low-rank matrix completion.
- Our algorithm achieves an oracle complexity of $\mathcal{O}(\epsilon^{-\frac{3}{2}})$. A comparison of the oracle complexity of DPRSRM with related algorithms is provided in Table 1, where DPRSRM achieves a lower oracle complexity than the existing decentralized stochastic manifold optimization algorithms (Deng and Hu 2023; Chen et al. 2021; Wang and Liu 2023; Zhao, Wang, and Lei 2024). Moreover, DPRSRM uses a single step of consensus to achieve communication, compared to other project/retraction algorithms (Chen et al. 2021; Deng and Hu 2023; Zhao, Wang, and Lei 2024) that need $\log_{\sigma_2}(\frac{1}{2\sqrt{n}})$ rounds of consensus, where σ_2 is the second largest singular value of the communication graph matrix.

Notation

For the compact submanifold $\mathcal{M}$ of $\mathbb{R}^{d\times r}$, we always take the Euclidean metric $\langle\cdot,\cdot\rangle$ as the Riemannian metric. We use $\|\cdot\|$ to denote the Euclidean norm. We denote the n-fold Cartesian product of $\mathcal{M}$ as $\mathcal{M}^n = \mathcal{M}\times\cdots\times\mathcal{M}$. For any $x\in\mathcal{M}$, the tangent space and normal space of $\mathcal{M}$ at x are denoted by $T_x\mathcal{M}$ and $N_x\mathcal{M}$, respectively. For a differentiable function $h:\mathbb{R}^{d\times r}\to\mathbb{R}$, we denote its Euclidean gradient by $\nabla h(x)$ and its Riemannian gradient by $\mathrm{grad}h(x)$. For a positive integer n, define $[n]=\{1,\dots,n\}$. Let $\mathbf{1}_n\in\mathbb{R}^n$ be a vector where all entries are equal to 1. Define $J := \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$. Unless otherwise explicitly defined, we now provide explanations for all lowercase variables used in this paper. Take x as an example, we denote x_i as a local variable at i-th node; $\hat{x} = \frac{1}{n}\sum_{i=1}^n x_i$ is the Euclidean average. Moreover, we use the bold notations $\mathbf{x} := [x_1^\top,\dots,x_n^\top]^\top\in\mathbb{R}^{nd\times r}$, $\hat{\mathbf{x}} := [\hat{x}^\top,\dots,\hat{x}^\top]^\top\in\mathbb{R}^{nd\times r}$, where $\mathbf{x}$ denotes the collection of all local variables x_i and $\hat{\mathbf{x}}$ is n copies of $\hat{x}$. When applied to the iterative process, in k-th iteration, we use $x_{i,k}$ to denote a local variable at i-th node and $\hat{x}_k = \frac{1}{n}\sum_{i=1}^n x_{k,i}$. Similarly, we also denote $\mathbf{x}_k := [x_{1,k}^\top,\dots,x_{n,k}^\top]^\top\in\mathbb{R}^{nd\times r}$, $\hat{\mathbf{x}}_k = [\hat{x}_k^\top,\dots,\hat{x}_k^\top]^\top\in\mathbb{R}^{nd\times r}$. Other lowercase variables can also be denoted similarly as x. Define the function $f(\mathbf{x}) := \sum_{i=1}^n f_i(x_i)$. Let $\mathbf{W} := W\otimes I_d\in\mathbb{R}^{nd\times nd}$, where $\otimes$ denotes the Kronecker product.

Preliminary

This section introduces the definition of stationary point for problem (1), and gives a key property for compact submanifolds, i.e., proximal smoothness.

Stationary Point

Let $x_1,\cdots,x_n\in\mathcal{M}$ represent the local copies of each node. Let $\mathcal{P}_\mathcal{M}$ be the orthogonal projection onto $\mathcal{M}$. Note that for $\{x_i\}_{i=1}^n\subset\mathcal{M}$,

$$\mathrm{argmin}_{y\in\mathcal{M}}\sum_{i=1}^n\|y-x_i\|^2 = \mathcal{P}_\mathcal{M}(\hat{x}).$$

Any element $\bar{x}$ in $\mathcal{P}_\mathcal{M}(\hat{x})$ is the induced arithmetic mean of $\{x_i\}_{i=1}^n$ on $\mathcal{M}$ (Sarlette and Sepulchre 2009). Let $f(z) := \frac{1}{n}\sum_{i=1}^n f_i(z)$. The ϵ-stationary point of problem (1) is defined as follows.

Definition 1 ((Deng and Hu 2023)). *The set of points $\{x_1,x_2,\cdots,x_n\}\subset\mathcal{M}$ is called an ϵ-stationary point of* (1) *if there exists a $\bar{x}\in\mathcal{P}_\mathcal{M}(\hat{x})$ such that*

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n\|x_i-\bar{x}\|^2\right]\le\epsilon \quad\text{and}\quad \mathbb{E}[\|\mathrm{grad}f(\bar{x})\|^2]\le\epsilon.$$

Algorithm	Manifold types	Communication	Tracking	VR	Oracle
(Chen et al. 2021)	Stiefel manifold	multiple	✗	✗	$\mathcal{O}(\epsilon^{-2})$
(Zhao, Wang, and Lei 2024)	Stiefel manifold	multiple	✗	✗	$\mathcal{O}(\epsilon^{-2})$
This paper	compact submanifold	single	✓	✓	$\mathcal{O}(\epsilon^{-3/2})$

Table 1: Comparison of the oracle complexity results of Riemannian online decentralized methods. "Communication" means rounds of communications per iteration. "Tracking" denotes the gradient tracking, "VR" denotes variance reduction. We do not list the work in (Wang and Liu 2023) since they focus on the finite-sum setting and are not applicable to the online setting.

We refer to these two terms as consensus error and optimality error, respectively.

Proximal Smoothness

Proximal smoothness is an effective tool for addressing the nonconvex nature of manifold constraints within decentralized optimization settings (Deng and Hu 2023). Define the distance from a point $x \in \mathbb{R}^{d\times r}$ to the manifold $\mathcal{M}$ and the nearest-point projection of x onto $\mathcal{M}$ as $\mathrm{dist}(x,\mathcal{M}) := \inf_{y\in\mathcal{M}} \|y-x\|$ and $\mathcal{P}_{\mathcal{M}}(x) := \arg\min_{y\in\mathcal{M}} \|y-x\|$, respectively. For a given number $R>0$, the R-tube around $\mathcal{M}$ is defined as the set $U_{\mathcal{M}}(R) := \{x : \mathrm{dist}(x,\mathcal{M}) < R\}$.

A closed set $\mathcal{M}$ is said to be R-proximally smooth if the projection $\mathcal{P}_{\mathcal{M}}(x)$ is unique whenever $\mathrm{dist}(x,\mathcal{M}) < R$. Following (Clarke, Stern, and Wolenski 1995), an R-proximally smooth set $\mathcal{M}$ satisfies that for any real $\delta \in (0,R)$,

$$\|\mathcal{P}_{\mathcal{M}}(x) - \mathcal{P}_{\mathcal{M}}(y)\| \le \frac{R}{R-\delta}\|x-y\|, \ \ \forall x,y \in \bar{U}_{\mathcal{M}}(\delta), \tag{2}$$

where $\bar{U}_{\mathcal{M}}(\delta) := \{x : \mathrm{dist}(x,\mathcal{M}) \le \delta\}$. For instance, the Stiefel manifold is known to be a 1-proximally smooth set (Balashov and Kamalov 2021), which also provides additional examples of rank manifolds along with their specific proximal smoothness radii.

Problem Setup and the Proposed DPRSRM

In this section, we present the problem setup considered in this paper, outlining the assumptions for the objective function and the communication network. Building on this setup, we then develop a decentralized algorithm to solve problem (1) and provide the main convergence rate results.

Problem Setup

Let us start with assumptions on problem (1), based on the fact that any compact C^2-submanifolds in Euclidean space belong to proximally smooth set (Balashov and Kamalov 2021; Clarke, Stern, and Wolenski 1995; Davis, Drusvyatskiy, and Shi 2020).

Assumption 1. *Assume that $\mathcal{M}$ is proximally smooth with radius R. Each objective function f_i is gradient Lipschitz continuous with modulus L_f on the convex hull of $\mathcal{M}$, denoted by $\mathrm{conv}(\mathcal{M})$. Moreover, the objective function $f(\mathbf{x})$ has an optimal value f_* over $\mathcal{M}^n$.*

Under this assumption, the following Riemannian quadratic upper bound for f_i has been estabilshed in (Deng and Hu 2023).

Lemma 1 ((Deng and Hu 2023), Lemma 2)**.** *Under Assumption 1, there exists L_g, for any $x,y\in\mathcal{M}$, the following inequality holds:*

$$f_i(y) - f_i(x) \le \langle \mathrm{grad} f_i(x), y-x\rangle + \frac{L_g}{2}\|y-x\|^2, \ i\in[n]. \tag{3}$$

Moreover, there exists a constant $L_G>0$ such that

$$\|\mathrm{grad} f_i(x) - \mathrm{grad} f_i(y)\| \le L_G\|x-y\|, \ i\in[n]. \tag{4}$$

We now present the assumption for the communication network. Denote by the undirected graph $G := \{\mathcal{V},\mathcal{E}\}$, where $\mathcal{V}=\{1,2,\dots,n\}$ is the set of all nodes and $\mathcal{E}$ is the set of edges. Let W be the mixing matrix associated with G and $W_{ij}\neq 0$ if there is an edge $(i,j)\in\mathcal{E}$ and $W_{ij}=0$ otherwise. We use the following standard assumptions on W, see, e.g., (Chen et al. 2021; Zeng and Yin 2018).

Assumption 2. *We assume that the undirected graph G is connected and W is doubly stochastic, i.e., (i) $W=W^\top$; (ii) $W_{ij}\ge 0$ and $1>W_{ii}>0$ for all i,j; (iii) Eigenvalues of W lie in $(-1,1]$. In addition, the second largest singular value σ_2 of W satisfies in $\sigma_2\in[0,1)$.*

Consider a sufficiently rich probability space $\{\Omega,\mathbb{P},\mathcal{F}\}$. For a given decentralized algorithm, we assume it generates an iterative sequence $\{x_{i,k}\}_{k\ge 0}$, where $x_{i,k}$ denotes the k-th iteration at node i. At each step, node i observes a random vector $\xi_{i,k}$. We then define an increasing sequence of sub-σ-algebras within $\mathcal{F}$, constructed from the random vectors observed in succession by the network nodes: for all $k\ge 1$, $\mathcal{F}_0 := \{\Omega,\emptyset\}$, $\mathcal{F}_k := \sigma\left(\{\xi_{i,0},\xi_{i,1},\dots,\xi_{i,k-1} : i\in\mathcal{V}\}\right)$, where $\emptyset$ represents the empty set. The following assumptions are made regarding the stochastic gradient $\nabla f_i(x,\xi_{i,k})$:

Assumption 3. *For any $\mathcal{F}_k$–measurable variable $x\in\mathcal{M}$ and $k\ge 1$, the algorithm generates a sample $\xi_{i,k}\sim\Omega$ for each node i and returns a stochastic gradient $\mathrm{grad} f_i(x,\xi_{i,k})$, there exists a parameter $\nu = \frac{1}{n}\sum_{i=1}^n \nu_i^2$ such that*

$$\mathbb{E}\left[\mathrm{grad} f_i(x,\xi_{i,k})|\mathcal{F}_k\right] = \mathrm{grad} f_i(x), \tag{5}$$

$$\mathbb{E}\left[\|\mathrm{grad} f_i(x,\xi_{i,k}) - \mathrm{grad} f_i(x)\|^2|\mathcal{F}_k\right] \le \nu_i^2. \tag{6}$$

Moreover, the collection $\{\xi_{i,k} : i\in\mathcal{V}, k\ge 1\}$ consists of independent random variables and $\mathrm{grad} f_i(x,\xi_{i,k})$ is the mean-squared $\bar{L}$-Lipschitz:

$$\mathbb{E}[\|\mathrm{grad} f_i(x,\xi_{i,k}) - \mathrm{grad} f_i(y,\xi_{i,k})\||\mathcal{F}_k] \le \bar{L}\mathbb{E}[\|x-y\|]. \tag{7}$$

To measure the oracle complexity, we give the definition of a stochastic first-order oracle (SFO) for (1).

Definition 2 (**stochastic first-order oracle**). *For the problem* (1)*, a stochastic first-order oracle for each node i is defined as follows: compute the Riemannian gradient* $\mathrm{grad} f_i(x, \xi_i)$ *given a sample* $\xi_i \in \Omega$.

The Algorithm

In this subsection, we introduce the Decentralized Projected Riemannian Stochastic Recursive Momentum (DPRSRM) method for addressing (1), along with its associated convergence results. Inspired by the notable effectiveness of variance reduction and gradient tracking in decentralized frameworks (Wang and Liu 2023; Deng and Hu 2023; Chen et al. 2021), we seek a novel combination of variance reduction and gradient tracking for decentralized online problems on compact submanifolds for improving the oracle complexity. In particular, we focus on the following stochastic gradient estimator using the momentum technique introduced in (Cutkosky and Orabona 2019; Han and Gao 2021b; Deng et al. 2024):

$$\begin{aligned} q_{i,k} =& \mathrm{grad} f_i(x_{i,k}, \xi_{i,k}) \\ &+ (1-\tau)(d_{i,k-1} - \mathrm{grad} f_i(x_{i,k-1}, \xi_{i,k})). \end{aligned} \tag{8}$$

We note that (8) can be rewritten as

$$\begin{aligned} q_{i,k} =& \tau \mathrm{grad} f_i(x_{i,k}, \xi_{i,k}) + (1-\tau)(d_{i,k-1} \\ &+ \mathrm{grad} f_i(x_{i,k}, \xi_{i,k}) - \mathrm{grad} f_i(x_{i,k-1}, \xi_{i,k})), \end{aligned} \tag{9}$$

which hybrids stochastic gradient $\mathrm{grad} f_i(x_{i,k}, \xi_{i,k})$ with the recursive gradient estimator in RSARAH/SRG (Han and Gao 2021a; Zhou, Yuan, and Feng 2019) for $\tau \in [0, 1]$. Since the direction $q_{i,k}$ may be unbounded, we introduce a clipped gradient estimator $d_{i,k}$

$$d_{i,k} = \begin{cases} q_{i,k} & \text{if } \|q_{i,k}\| \le B, \\ B\frac{q_{i,k}}{\|q_{i,k}\|} & \text{otherwise}, \end{cases} \tag{10}$$

where $B > 0$ is a user-defined constant. To further reduce the variance of the gradient estimator in different nodes, we compute the gradient tracking iteration as follows:

$$s_{i,k} = \sum_{j=1}^{n} W_{ij} s_{j,k-1} + d_{i,k} - d_{i,k-1},\ i \in [n]. \tag{11}$$

A crucial advantage of gradient tracking-type methods lies in the applicability of the use of a constant step size α. Since $s_{i,k}$ may not remain in the tangent space $T_{x_{i,k}}\mathcal{M}$, we introduce its tangent space projection $v_{i,k} = \mathcal{P}_{T_{x_{i,k}}}(s_{i,k})$ and update the new iterate $x_{i,k+1}$ as follows:

$$x_{i,k+1} = \mathcal{P}_{\mathcal{M}}(\sum_{j=1}^{n} W_{ij} x_{j,k} - \alpha v_{i,k}),\ i \in [n], \tag{12}$$

where the projection-based average (Deng and Hu 2023; Hu and Deng 2024), i.e., $\mathcal{P}_{\mathcal{M}}(\sum_{j=1}^{n} W_{ij} x_{j,k})$, is adopted to ensure the decrease in the consensus error among nodes. For ease of analysis, we stack variables in each node and rewrite (8), (11) and (12) as

$$\begin{cases} \mathbf{q}_k =& \mathrm{grad} f(\mathbf{x}_k, \xi_k) \\ & + (1-\tau)(\mathbf{d}_{k-1} - \mathrm{grad} f(\mathbf{x}_{k-1}, \xi_k)) \\ \mathbf{s}_k =& \mathbf{W}\mathbf{s}_{k-1} + \mathbf{d}_k - \mathbf{d}_{k-1} \\ \mathbf{v}_k =& \mathcal{P}_{T_{\mathbf{x}}\mathcal{M}^n}(\mathbf{s}_k) \\ \mathbf{x}_{k+1} =& \mathcal{P}_{\mathcal{M}^n}(\mathbf{W}\mathbf{x}_k - \alpha \mathbf{v}_k), \end{cases} \tag{13}$$

where $\mathcal{P}_{T_{\mathbf{x}}\mathcal{M}^n} := \mathcal{P}_{T_{x_{1,k}}\mathcal{M}} \times \cdots \times \mathcal{P}_{T_{x_{n,k}}\mathcal{M}}$ and $\xi_k = \{\xi_{i,k}\}_{i \in \mathcal{V}}$. The overall algorithm is given in Algorithm 1.

Algorithm 1: The DPRSRM for solving (1)

Require: Initial point $\mathbf{x}_0 = \bar{\mathbf{x}}_0 \in \mathcal{M}$, $\mathbf{s}_{-1} = \mathbf{d}_{-1} = \mathbf{0}$, α, τ.

Sample $\xi_{i,0}$, let $s_{i,0} = d_{i,0} = \mathrm{grad} f_i(x_{i,0}, \xi_{i,0})$.
$v_{i,0} = \mathcal{P}_{T_{x_{i,0}}\mathcal{M}}(s_{i,0})$.
$x_{i,1} = \mathcal{P}_{\mathcal{M}}(\sum_{j=1}^{n} W_{ij} x_{j,0} - \alpha v_{i,0})$.
for $k = 1, 2, \cdots, K$ **do**
  Update stochastic gradient estimator $q_{i,k}$ via (8)
  Update the clipped gradient estimator $d_{i,k}$ via (10).
  Update Riemannian gradient tracking $s_{i,k}$ via (11).
  Project onto tangent space: $v_{i,k} = \mathcal{P}_{T_{x_{i,k}}\mathcal{M}}(s_{i,k})$.
  Update new iterate $x_{i,k+1}$ via (12).
end for

Main Results

The convergence analysis of Algorithm 1 can be divided into two parts: the consensus error and the optimality error, as defined in Definition 1. Let us first focus on the consensus error. We define a neighborhood around $\mathbf{x} \in \mathcal{M}^n$ as follows:

$$\mathcal{N}(\delta) := \{\mathbf{x} \in \mathcal{M}^n : \|\mathbf{x} - \bar{\mathbf{x}}\| \le \delta\}. \tag{14}$$

Note that if $\mathbf{x} \in \mathcal{N}(\delta)$, then $x_i \in \bar{U}_{\mathcal{M}}(\delta)$ for any $i \in [n]$.

By appropriately selecting the step size α and an integer t, and initializing with $\mathbf{x}_0 \in \mathcal{N}(\delta)$, we can establish an explicit relationship between the consensus error and the step size.

Theorem 1 (Consensus error). *Let $\{\mathbf{x}_k\}_k$ be the sequence generated by Algorithm 1. Suppose that Assumptions 1 and 2 hold. Define $\rho := \frac{R}{R-2\delta}\sigma_2$, $D := \frac{(3L)^2}{(1-\sigma_2)^2}$ and $C := \frac{D}{(1-\rho)^2\sigma_2^2}$. Let α and δ satisfy that*

$$\begin{aligned} &\alpha \le \min\left\{\frac{1}{4L}, \frac{\delta}{\sqrt{nD}}, \frac{(R(1-\sigma_2)-2\delta)\delta}{R\sqrt{nD}}\right\}, \\ &\delta < \frac{1}{2}\min\left\{R, R(1-\sigma_2)\right\}. \end{aligned} \tag{15}$$

If $\mathbf{x}_0 \in \mathcal{N}(\delta)$, it holds that

$$\frac{1}{n}\|\bar{\mathbf{x}}_k - \mathbf{x}_k\|^2 \le C\alpha^2. \tag{16}$$

Now we present the following main theorem of the DPRSRM. For the ease of analysis, we define

$$L := \max\{L_g, L_G, \bar{L}c\}, \tag{17}$$

where L_g, L_G are given in Lemma 1 and $\bar{L}$ occurred in Assumption 3.

Theorem 2 (Optimality error). *Let $\{\mathbf{x}_k\}_k$ be the sequence generated by Algorithm 1 with $B \geq L$. Suppose that Assumptions 1-3 hold. Let α and δ satisfy* (15). *If $\mathbf{x}_0 \in \mathcal{N}(\delta)$, there exists constants M_2, Q, L_2 such that*

$$\begin{aligned}&\min_{1\leq k\leq K}\mathbb{E}[\|\mathrm{grad} f(\bar{x}_k)\|^2]\\ \leq&\frac{4(f(\bar{x}_0)-f_*)}{\alpha K}+4(\mathcal{G}_1\alpha^2+\mathcal{G}_2\alpha^3)+6(\rho_3\alpha^2+\rho_2\nu^2\tau^2)\\ &\frac{48L^2}{(1-\sigma_2^2)K}+\frac{12}{n}(1+\rho_2\tau^2)(\nu^2\tau+\rho_3 n\frac{\alpha^4}{\tau}),\end{aligned}$$

where $\rho_2 := \frac{6(1+\sigma_2^2)}{(1-\sigma_2^2)^2}$, $\rho_3 := \rho_2(8CL^2+4DL^2)$ and $\mathcal{G}_1, \mathcal{G}_2$ are defined as:

$$\begin{aligned}\mathcal{G}_1 :=& \frac{3}{2}CL^2+(\frac{3}{2}M_2^2+\frac{3}{2}(\sqrt{n}L_2+8Q)^2)C^2\\ &+24Q^2D^2+\frac{1}{4}LL_2^2+LD,\\ \mathcal{G}_2 :=& 2(8Q+\sqrt{n}L_2+M_2)^2LC^2+8Q^2D^2L+2M_2C^2L.\end{aligned} \tag{18}$$

The following corollary addresses the finite-time convergence rate of DPRSRM with specific choices of the algorithmic parameters α and τ.

Corollary 1. *Let $\{\mathbf{x}_k\}_k$ be the sequence generated by Algorithm 1 with $B \geq L$. Suppose that Assumptions 1-3 hold. Let $\alpha = \frac{1}{K}$, $\tau = \frac{1}{K^{2/3}}$ and $\delta \leq \min\{\frac{R}{2}, \frac{R(1-\sigma_2)}{2}\}$. If $\mathbf{x}_0 \in \mathcal{N}(\delta)$, and $K \geq \max\left\{4L, \sqrt{nD}/\delta, \frac{R\sqrt{nD}}{(R(1-\sigma_2)-2\delta)\delta}\right\}$, it holds that*

$$\begin{aligned}&\mathbb{E}\left[\frac{1}{n}\|\bar{\mathbf{x}}_K-\mathbf{x}_K\|^2\right]\leq\frac{C}{K^{2/3}},\\ &\min_{0\leq k\leq K}\mathbb{E}[\|\mathrm{grad} f(\bar{x}_k)\|^2]\leq\frac{\Gamma_1}{K^{2/3}}+\frac{\Gamma_2}{K}+\frac{6\rho_2\nu^2}{K^{4/3}},\end{aligned} \tag{19}$$

where $\Gamma_1 = 4(f(\bar{x}_0)-f_*)+4\mathcal{G}_1+6\rho_3+\frac{12\sigma}{n}(1+\rho_2)$ and $\Gamma_2 = 4\mathcal{G}_2+\frac{48L^2}{(1-\sigma_2^2)}$. As a consequence, DPRSRM obtains an ϵ-stationary point with at most

$$\mathcal{K} := \mathcal{O}(\max\{\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3\})$$

iterations. Here, $\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3$ are given as follows:

$$\begin{aligned}&\mathcal{K}_1 := (C+\Gamma_1)^{1.5}\epsilon^{-\frac{3}{2}}, \mathcal{K}_2 := \Gamma_2\epsilon^{-1},\\ &\mathcal{K}_3 := (6\rho_2\nu^2)^{3/4}\epsilon^{-\frac{3}{4}}.\end{aligned} \tag{20}$$

According to Corollary 1, DPRSRM achieves an oracle complexity of $\mathcal{O}(\epsilon^{-\frac{3}{2}})$, outperforming existing methods (Chen et al. 2021; Zhao, Wang, and Lei 2024), which have an oracle complexity of at most $\mathcal{O}(\epsilon^{-2})$.

Outline of Convergence Analysis

As shown in the above results, the convergence analysis consists of two critical components: the consensus error and the optimality measure, corresponding to Theorem 1 and Theorem 2, respectively. In this section, we will outline the proof.

Consensus Error

This subsection addresses the consensus error in Theorem 1. In Algorithm 1, the update of the main iterate x_{k+1} involves the projection on $\mathcal{M}$. Due to the nonconvex nature of compact submanifolds, the projection is not always unique. Before proving Theorem 1, we will first demonstrate that the projection operator in Algorithm 1 is well-defined, meaning that the points being projected are always ensured to be within a neighborhood that belongs to $\bar{U}_{\mathcal{M}}(R)$.

We first investigate the uniform boundedness of $\|\mathbf{s}_k\|$ in the following lemma.

Lemma 2. *Let $\{\mathbf{x}_k\}_k$ be the sequence generated by Algorithm 1. Suppose that Assumptions 1 and 2 hold. Then for all k, it holds that*

$$\|\mathbf{s}_k\|^2 \leq nD, \;\; D := \frac{(3L)^2}{(1-\sigma_2)^2}. \tag{21}$$

The following lemma demonstrates that the iterates $\mathbf{x}_k$ will always remain in the neighborhood $\mathcal{N}(\delta)$ under certain conditions.

Lemma 3. *Suppose that Assumptions 1 and 2 hold. Let $x_{i,k}$ be generated by Algorithm 1. Let α and δ satisfy* (15). *If $\mathbf{x}_0 \in \mathcal{N}(\delta)$, then for any $k \geq 1$, it holds that $\mathbf{x}_k \in \mathcal{N}(\delta)$ and*

$$\sum_{j=1}^{n} W_{ij}x_{j,k} - \alpha v_{i,k} \in \bar{U}_{\mathcal{M}}(2\delta),\ i = 1, \cdots, n. \tag{22}$$

Now we give the proof of Theorem 1.

Proof of Theorem 1. Since $\mathbf{x}_0 \in \mathcal{N}(\delta)$, it follows from Lemma 3 that for any $k > 0$, $\mathbf{x}_k \in \mathcal{N}(\delta)$ and

$$\sum_{j=1}^{n} W_{ij}x_{j,k} - \alpha v_{i,k} \in \bar{U}_{\mathcal{M}}(2\delta),\ i = 1, \cdots, n. \tag{23}$$

By the definition of $\bar{\mathbf{x}}_{k+1}$, we have

$$\begin{aligned}\|\mathbf{x}_{k+1}-\bar{\mathbf{x}}_{k+1}\| &\leq \|\mathbf{x}_{k+1}-\bar{\mathbf{x}}_k\|\\ &= \|\mathcal{P}_{\mathcal{M}^n}(\mathbf{W}\mathbf{x}_k-\alpha\mathbf{v}_k)-\mathcal{P}_{\mathcal{M}^n}(\hat{\mathbf{x}}_k)\|\\ &\leq \frac{R}{R-2\delta}\|\mathbf{W}\mathbf{x}_k-\alpha\mathbf{v}_k-\hat{\mathbf{x}}_k\|\\ &\leq \frac{R}{R-2\delta}\sigma_2\|\mathbf{x}_k-\bar{\mathbf{x}}_k\|+\frac{R}{R-2\delta}\sqrt{nD}\alpha\\ &\leq \rho\|\mathbf{x}_k-\bar{\mathbf{x}}_k\|+\frac{\sqrt{nD}\alpha}{\sigma_2},\end{aligned}$$

where the first inequality follows from the optimality of $\bar{\mathbf{x}}_{k+1}$, the second inequality utilizes (23) and the $\frac{R}{R-2\delta}$-Lispchitz continuity of $\mathcal{P}_{\mathcal{M}}$ over $\bar{U}_{\mathcal{M}}(2\delta)$. Then

$$\begin{aligned}\|\bar{\mathbf{x}}_{k+1}-\mathbf{x}_{k+1}\| &\leq \rho\|\bar{\mathbf{x}}_k-\mathbf{x}_k\|+\frac{\sqrt{nD}\alpha}{\sigma_2}\\ &\leq \rho^{k+1}\|\bar{\mathbf{x}}_0-\mathbf{x}_0\|+\frac{\sqrt{nD}}{(1-\rho)\sigma_2}\alpha.\end{aligned} \tag{24}$$

By the initialized strategy of $\mathbf{x}_0$ in Algorithm 1, we have $\mathbf{x}_0 = \bar{\mathbf{x}}_0$. The proof is completed. □

Optimality Error

In this subsection, we outline the proof of Theorem 2. For ease of notation, let us denote

$$\hat{g}_k := \frac{1}{n}\sum_{i=1}^{n} \mathrm{grad} f_i\left(x_{i,k}\right), \quad \hat{\mathbf{g}}_k := \left(\mathbf{1}_n \otimes I_d\right) \hat{g}_k.$$

By applying the Lipschitz-type inequalities on compact submanifolds and combining them with the above lemma, we can demonstrate a sufficient decrease in f.

Lemma 4. *Let $\{\mathbf{x}_k\}_k$ be the sequence generated by Algorithm 1. Suppose that Assumptions 1 and 2 hold. Let α and δ satisfy* (15)*. If $\mathbf{x}_0 \in \mathcal{N}(\delta)$, it follows that*

$$f(\bar{x}_{k+1}) \leq f(\bar{x}_k) - \frac{\alpha}{4}\|\mathrm{grad} f(\bar{x}_k)\| + \mathcal{G}_1\alpha^3 + \mathcal{G}_2\alpha^4 + \frac{3\alpha}{2n}(\|\mathrm{grad} f(\mathbf{x}_k) - \mathbf{d}_k\|^2 + \|\hat{\mathbf{d}}_k - \mathbf{s}_k\|^2),$$

where $\mathcal{G}_1$ and $\mathcal{G}_2$ are constant defined by (18)*.*

Let us build the following lemma on the relationship between $\mathbf{s}_k$ and $\hat{\mathbf{d}}_k$.

Lemma 5. *Let $\{\mathbf{x}_k\}_k$ be the sequence generated by Algorithm 1. Suppose that Assumptions 1-3 hold. It holds that for any k,*

$$\sum_{k=0}^{K}\mathbb{E}[\|\hat{\mathbf{d}}_k - \mathbf{s}_k\|^2] \leq \rho_2 \sum_{k=0}^{K-1}\tau^2\mathbb{E}[\|\mathbf{d}_k - \mathrm{grad} f(\mathbf{x}_k)\|^2] + \frac{8nL^2}{1-\sigma_2^2} + \rho_3 n\alpha^2 K + \rho_2\nu^2 n\tau^2 K. \tag{25}$$

We also have the gradient estimation error bound.

Lemma 6. *Let $\{\mathbf{x}_k\}_k$ be the sequence generated by Algorithm 1 with $B \geq L$. Suppose that Assumptions 1-3 hold. Then the expected estimation error of the estimator is bounded by*

$$\sum_{k=0}^{K}\mathbb{E}[\|\mathbf{d}_k - \mathrm{grad} f(\mathbf{x}_k)\|^2] \leq \frac{n\nu^2}{\tau} + 2\nu^2\tau K + 2\rho_3 n\frac{\alpha^4}{\tau}K, \tag{26}$$

where ρ_3 is defined in Lemma 5.

With these preparations, we give the proof of Theorem 2.

Proof of Theorem 2. Combining Lemma 4 and Lemma 5 yields that

$$\begin{aligned}
&\sum_{k=0}^{K}\frac{\alpha}{4}\mathbb{E}[\|\mathrm{grad} f(\bar{x}_k)\|^2] \leq f(\bar{x}_0) + (\mathcal{G}_1\alpha^3 + \mathcal{G}_2\alpha^4)K \\
&+ \frac{3\alpha}{2n}\sum_{k=0}^{K}\mathbb{E}[\|\mathrm{grad} f(\mathbf{x}_k) - \mathbf{d}_k\|^2 + \|\hat{\mathbf{d}}_k - \mathbf{s}_k\|^2] \\
\leq& f(\bar{x}_1) + (\mathcal{G}_1\alpha^3 + \mathcal{G}_2\alpha^4)K + \frac{3\alpha}{2}(\rho_3\alpha^2 + \rho_2\nu^2\tau^2)K \\
&+ \frac{12L^2}{1-\sigma_2^2}\alpha + \frac{3\alpha}{2n}(1+\rho_2\tau^2)\sum_{k=0}^{K}\mathbb{E}[\|\mathrm{grad} f(\mathbf{x}_k) - \mathbf{d}_k\|^2].
\end{aligned} \tag{27}$$

Incorporating Lemma 6 into (27) completes the proof. □

Proof of Corollary 1. Given the choice of α and $K \geq \max\left\{4L, \sqrt{nD}/\delta, \frac{R\sqrt{nD}}{(R(1-\sigma_2)-2\delta)\delta}\right\}$, we can infer that α satisfies the condition (15). Therefore, it follows from Theorem 2 that

$$\begin{aligned}
&\min_{1\leq k\leq K}\mathbb{E}[\|\mathrm{grad} f(\bar{x}_k)\|^2] \\
\leq& \frac{4(f(\bar{x}_0) - f_*)}{\alpha K} + 4(\mathcal{G}_1\alpha^2 + \mathcal{G}_2\alpha^3) + 6(\rho_3\alpha^2 + \rho_2\nu^2\tau^2) \\
&\frac{48L^2}{(1-\sigma_2^2)K} + \frac{12}{n}(1+\rho_2\tau^2)(\nu^2\tau + \rho_3 n\frac{\alpha^4}{\tau}) \\
\leq& \frac{4(f(\bar{x}_0) - f_*) + 4\mathcal{G}_1 + 6\rho_3 + \frac{12\sigma}{n}(1+\rho_2)}{K^{2/3}} \\
&+ (4\mathcal{G}_2 + \frac{48L^2}{(1-\sigma_2^2)})\frac{1}{K} + \frac{6\rho_2\nu^2}{K^{4/3}}.
\end{aligned} \tag{28}$$

The proof is completed. □

Numerical Experiments

In this section, we compare our proposed DPRSRM with DRSGD in (Chen et al. 2021) and DRPGD in (Deng and Hu 2023) on decentralized principal component analysis. It is important to note that the original DRPGD is a deterministic algorithm that utilizes the local full gradient. To adapt it to the online setting, we replace the full gradient with a stochastic gradient for the local updates. The numerical results on decentralized low-rank matrix completion are provided in the supplementary material.

The decentralized principal component analysis (PCA) problem can be expressed mathematically as follows:

$$\min_{\mathbf{x}\in\mathcal{M}^n} -\frac{1}{2n}\sum_{i=1}^{n}\mathrm{tr}(x_i^\top A_i^\top A_i x_i), \quad \text{s.t.} \quad x_1 = \ldots = x_n, \tag{29}$$

where $\mathcal{M}$ is the Stiefel manifold $\mathrm{St}(d,r)$, $A_i \in \mathbb{R}^{m_i\times d}$ is the data matrix corresponding to the i-th node, and m_i denotes the number of samples. It is worth noting that if x^* is a solution to this problem, then any transformation of x^* by an orthogonal matrix $Q \in \mathbb{R}^{r\times r}$ is also a valid solution. The distance between two points x and x^* is then calculated as:

$$d_s(x, x^*) := \min_{Q\in\mathbb{R}^{r\times r},\, Q^\top Q = QQ^\top = I_r} \|xQ - x^*\|.$$

Synthetic Dataset

In our study, we set the parameters as follows: $m_1 = \ldots = m_n = 1000$, $d = 10$, and $r = 5$. A matrix $B \in \mathbb{R}^{1000n\times d}$ is generated, and its singular value decomposition (SVD) is performed, yielding $B = U\Sigma V^\top$, where $U \in \mathbb{R}^{1000n\times d}$ and $V \in \mathbb{R}^{d\times d}$ are orthogonal matrices, and $\Sigma \in \mathbb{R}^{d\times d}$ is a diagonal matrix. To control the distribution of singular values, we define $\tilde{\Sigma} = \mathrm{diag}(\gamma^j)$ with γ chosen from the interval $(0,1)$. The matrix A is then formed as $A = U\tilde{\Sigma}V^\top \in \mathbb{R}^{1000n\times d}$. The matrices A_i are derived by partitioning the rows of A into n equally sized subsets. It can be shown that the first r columns of V represent the solution to (29). In

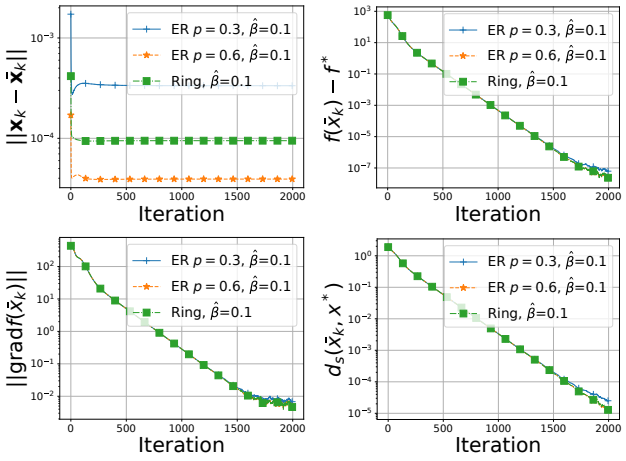

Figure 1: Numerical results on the synthetic dataset with different network graphs.

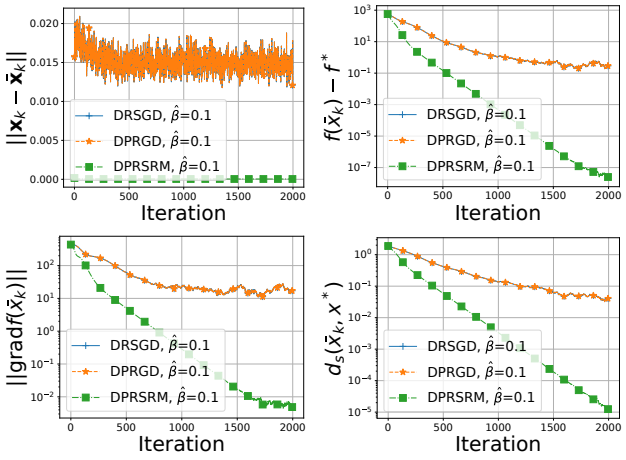

Figure 2: Results on the synthetic dataset with ER $p = 0.6$.

our experiments, the parameters γ and n are set to 0.8 and 8, respectively.

We employ fixed step sizes for all algorithms. The step size is set to $\alpha = \frac{\hat{\beta}}{\sqrt{K}}$ with K being the maximal number of iterations. The grid search is utilized to find the best $\hat{\beta}$ for each algorithm. The momentum parameter is chosen as $\tau = 0.999$. The batch size in each node is set as 10 and the maximum iteration is set $K = 2000$. The clipping constant is set as $B = 10^8$. We choose the polar decomposition as the retraction operator for DRSGD. We test several graph matrices to model the topology across the nodes, namely, the Erdos-Renyi (ER) network with probability $p = 0.3, 0.6$, and the Ring network. Throughout this section, we select the mixing matrix W to be the Metropolis constant edge weight matrix (Shi et al. 2015).

Firstly, we test all algorithms with different network graphs, namely, Ring, ER $p = 0.3$, and ER $p = 0.6$. The results are shown in Figure 1. We see that there is not much difference among different graphs except for the consensus

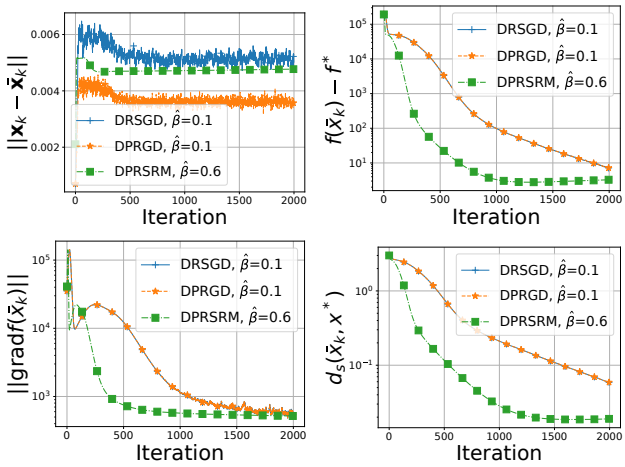

Figure 3: Results on Mnist dataset with ER $p = 0.3$.

error. This is consistent with the existing results for DRSGD (Chen et al. 2021) and DPRGD (Deng and Hu 2023). Secondly, Figure 2 presents a comparison among the three algorithms. Our DPRSRM outperforms the other two, with DRSGD and DPRGD showing comparable performance.

Mnist Dataset

We also conduct numerical tests on the Mnist dataset (LeCun 1998). The training images consist of 60000 handwritten images of size 32×32 and are used to generate A_i's. We first normalize the data matrix by dividing 255 and randomly split the data into $n = 8$ nodes with equal cardinality. Then, each node holds a local matrix A_i of dimension $\frac{60000}{n} \times 784$. We compute the first 5 principal components, i.e., $d = 784, r = 5$.

For all algorithms, we use the fixed step sizes $\alpha = \frac{\hat{\beta}}{60000}$ with a best-chosen $\hat{\beta}$, batch size 1500 and momemtum parameter $\tau = 0.999$. Similar to the synthetic setting, our DPRSRM algorithm demonstrates superior performance compared to other algorithms in terms of objective function values, gradient norms, and distances to the optimal solution. It is important to note that due to the use of stochastic gradients, consensus among the algorithms may be affected by noise, which can be reduced by using a smaller step size.

Conclusions and Limitations

This work develops a decentralized projection Riemannian stochastic recursive momentum method by assuming that each node has access to a stochastic first-order oracle. Our algorithm leverages local hybrid variance reduction and gradient tracking to achieve a lower oracle complexity compared with the existing online methods. It requires only $\mathcal{O}(1)$ gradient evaluations per iteration for each local node and does not require restarting with a large batch gradient.

Acknowledgements

This effort was supported by the SciAI Center, and funded by the Office of Naval Research (ONR), under Grant

Number N00014-23-1-2729 (J.H.), as well as by the National Natural Science Foundation of China (NSFC) grants 12401419 (K.D.).

References

Ando, R. K.; Zhang, T.; and Bartlett, P. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6(11): 1871–1853.

Balashov, M.; and Kamalov, R. 2021. The Gradient Projection Method with Armijo's Step Size on Manifolds. *Computational Mathematics and Mathematical Physics*, 61: 1776–1786.

Bianchi, P.; and Jakubowicz, J. 2012. Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization. *IEEE Transactions on Automatic Control*, 58(2): 391–405.

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

Cardot, H.; and Degras, D. 2018. Online principal component analysis in high dimension: Which algorithm to choose? *International Statistical Review*, 86(1): 29–50.

Chen, J.; Ye, H.; Wang, M.; Huang, T.; Dai, G.; Tsang, I.; and Liu, Y. 2024. Decentralized Riemannian Conjugate Gradient Method on the Stiefel Manifold. In *The Twelfth International Conference on Learning Representations*.

Chen, S.; Garcia, A.; Hong, M.; and Shahrampour, S. 2021. Decentralized Riemannian gradient descent on the Stiefel manifold. In *International Conference on Machine Learning*, 1594–1605. PMLR.

Cho, M.; and Lee, J. 2017. Riemannian approach to batch normalization. *Advances in Neural Information Processing Systems*, 30.

Clarke, F. H.; Stern, R. J.; and Wolenski, P. R. 1995. Proximal smoothness and the lower-C2 property. *Journal of Convex Analysis*, 2(1-2): 117–144.

Cutkosky, A.; and Orabona, F. 2019. Momentum-based variance reduction in non-convex sgd. *Advances in neural information processing systems*, 32.

Davis, D.; Drusvyatskiy, D.; and Shi, Z. 2020. Stochastic optimization over proximally smooth sets. *arXiv:2002.06309*.

Deng, K.; and Hu, J. 2023. Decentralized projected Riemannian gradient method for smooth optimization on compact submanifolds. *arXiv preprint arXiv:2304.08241*.

Deng, K.; Hu, J.; and Wang, H. 2023. Decentralized Douglas-Rachford splitting methods for smooth optimization over compact submanifolds. *arXiv preprint arXiv:2311.16399*.

Deng, K.; Hu, J.; Wu, J.; and Wen, Z. 2024. Oracle complexity of augmented Lagrangian methods for nonsmooth manifold optimization. *arXiv preprint arXiv:2404.05121*.

Di Lorenzo, P.; and Scutari, G. 2016. NEXT: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2): 120–136.

Han, A.; and Gao, J. 2021a. Improved variance reduction methods for Riemannian non-convex optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11): 7610–7623.

Han, A.; and Gao, J. 2021b. Riemannian Stochastic Recursive Momentum Method for non-Convex Optimization. In Zhou, Z.-H., ed., *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 2505–2511.

Hong, M.; Hajinezhad, D.; and Zhao, M.-M. 2017. Prox-PDA: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks. In *International Conference on Machine Learning*, 1529–1538. PMLR.

Hu, J.; Ao, R.; So, A. M.-C.; Yang, M.; and Wen, Z. 2024. Riemannian natural gradient methods. *SIAM Journal on Scientific Computing*, 46(1): A204–A231.

Hu, J.; and Deng, K. 2024. Improving the communication in decentralized manifold optimization through single-step consensus and compression. *arXiv preprint arXiv:2407.08904*.

Hu, J.; Deng, K.; Li, N.; and Li, Q. 2023. Decentralized Riemannian natural gradient methods with Kronecker-product approximations. *arXiv:2303.09611*.

Kasai, H.; Jawanpuria, P.; and Mishra, B. 2019. Riemannian adaptive stochastic gradient algorithms on matrix manifolds. In *International Conference on Machine Learning*, 3262–3271. PMLR.

LeCun, Y. 1998. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

Mishra, B.; Kasai, H.; Jawanpuria, P.; and Saroop, A. 2019. A Riemannian gossip approach to subspace learning on Grassmann manifold. *Machine Learning*, 108(10): 1783–1803.

Qu, G.; and Li, N. 2017. Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, 5(3): 1245–1260.

Sarlette, A.; and Sepulchre, R. 2009. Consensus optimization on manifolds. *SIAM Journal on Control and Optimization*, 48(1): 56–76.

Scutari, G.; and Sun, Y. 2019. Distributed nonconvex constrained optimization over time-varying digraphs. *Mathematical Programming*, 176(1-2): 497–544.

Shah, S. M. 2017. Distributed optimization on Riemannian manifolds for multi-agent networks. *arXiv:1711.11196*.

Shi, W.; Ling, Q.; Wu, G.; and Yin, W. 2015. EXTRA: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2): 944–966.

Sun, H.; Lu, S.; and Hong, M. 2020. Improving the sample and communication complexity for decentralized non-convex optimization: Joint gradient estimation and tracking. In *International Conference on Machine Learning*, 9217–9228. PMLR.

Sun, Y.; Chen, S.; Garcia, A.; and Shahrampour, S. 2024. Global Convergence of Decentralized Retraction-Free Optimization on the Stiefel Manifold. *arXiv preprint arXiv:2405.11590*.

Tatarenko, T.; and Touri, B. 2017. Non-convex distributed optimization. *IEEE Transactions on Automatic Control*, 62(8): 3744–3757.

Wai, H.-T.; Lafond, J.; Scaglione, A.; and Moulines, E. 2017. Decentralized Frank–Wolfe algorithm for convex and nonconvex problems. *IEEE Transactions on Automatic Control*, 62(11): 5522–5537.

Wang, L.; and Liu, X. 2022. Decentralized Optimization Over the Stiefel Manifold by an Approximate Augmented Lagrangian Function. *IEEE Transactions on Signal Processing*, 70: 3029–3041.

Wang, L.; and Liu, X. 2023. A variance-reduced stochastic gradient tracking algorithm for decentralized optimization with orthogonality constraints. *Journal of Industrial and Management Optimization*, 19(10): 7753–7776.

Xu, J.; Zhu, S.; Soh, Y. C.; and Xie, L. 2015. Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes. In *IEEE Conference on Decision and Control*, 2055–2060.

Ye, H.; and Zhang, T. 2021. DeEPCA: Decentralized exact PCA with linear convergence rate. *The Journal of Machine Learning Research*, 22(1): 10777–10803.

Yuan, K.; Ying, B.; Zhao, X.; and Sayed, A. H. 2018. Exact diffusion for distributed optimization and learning Part II: Convergence analysis. *IEEE Transactions on Signal Processing*, 67(3): 724–739.

Zeng, J.; and Yin, W. 2018. On nonconvex decentralized gradient descent. *IEEE Transactions on Signal Processing*, 66(11): 2834–2848.

Zhao, J.; Wang, X.; and Lei, J. 2024. Distributed Riemannian Stochastic Gradient Tracking Algorithm on the Stiefel Manifold. *arXiv preprint arXiv:2405.16900*.

Zhou, P.; Yuan, X.-T.; and Feng, J. 2019. Faster first-order methods for stochastic non-convex optimization on Riemannian manifolds. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 138–147. PMLR.