Decentralized Riemannian Gradient Descent on the Stiefel Manifold

Shixiang Chen [1] **Alfredo Garcia** [1] **Mingyi Hong** [2] **Shahin Shahrampour** [1]

Abstract

We consider distributed non-convex optimization where a network of agents aims at minimizing a global function over the Stiefel manifold. The global function is represented as a finite sum of smooth local functions, where each local function is associated with one agent and agents communicate with each other over an undirected connected graph. The problem is non-convex as local functions are possibly non-convex (but smooth) and the Steifel manifold is a non-convex set. We present a decentralized Riemannian stochastic gradient method (DRSGD) with the convergence rate of $\mathcal{O}(1/\sqrt{K})$ to a stationary point. To have exact convergence with constant stepsize, we also propose a decentralized Riemannian gradient tracking algorithm (DRGTA) with the convergence rate of $\mathcal{O}(1/K)$ to a stationary point. We use multi-step consensus to preserve the iteration in the local consensus region. DRGTA is the first decentralized algorithm with exact convergence for distributed optimization on Stiefel manifold.

1. Introduction

Distributed optimization has received significant attention in the past few years in machine learning, control and signal processing. There are mainly two scenarios where distributed algorithms are necessary: **(i)**. the data is geographically distributed over networks and/or **(ii)**. the computation on a single (centralized) server is too expensive (large-scale data setting). In this paper, we consider the following multi-agent optimization problem

$$\begin{aligned} &\min \frac{1}{n}\sum_{i=1}^{n} f_i(x_i) \\ &\text{s.t.} \quad x_1 = x_2 = \ldots = x_n, \\ &\qquad x_i \in \mathcal{M}, \quad \forall i = 1, \ldots, n, \end{aligned} \tag{1.1}$$

where f_i has $L-$Lipschitz continuous gradient in Euclidean space and $\mathcal{M} := \mathrm{St}(d,r) = \{x \in \mathbb{R}^{d\times r} : x^\top x = I_r\}$ is the Stiefel manifold. Unlike the Euclidean distributed setting, problem (1.1) is defined on the Stiefel manifold, which is a non-convex set. Many important applications can be written in the form (1.1), e.g., decentralized spectral analysis (Kempe & McSherry, 2008; Gang & Bajwa, 2021), dictionary learning (Raja & Bajwa, 2015), eigenvalue estimation of the covariance matrix (Penna & Stańczak, 2014) in wireless sensor networks, and deep neural networks with orthogonal constraint (Arjovsky et al., 2016; Vorontsov et al., 2017; Huang et al., 2018).

Problem (1.1) can generally represent a risk minimization. One approach to solving (1.1) is collecting all variables to a central server and running a centralized algorithm. In this work, however, we consider the *decentralized* setting. Our motivations are two-fold: **(i)**. In some applications, the datasets are collected, stored and manipulated in a distributed manner. Due to privacy concerns and/or inability to gather all data in a central node, centralized methods cannot be implemented. In a decentralized implementation, local parameter vectors (and not data) can be shared amongst neighboring nodes. In this setting, prior to our work, it was not clear how to design a converging decentralized algorithm for problem (1.1), nor was it clear how such an algorithm scales w.r.t the connectivity of the network. **(ii)**. The other popular reason to study decentralized setting is to accelerate the computation for stochastic algorithms in modern computational architectures. (Lian et al., 2017) proved that the decentralized SGD (D-PSGD) algorithm can achieve a linear speedup w.r.t n if the iteration number is large enough, and the convergence rate is the same as centralized SGD (C-PSGD). Due to the communication efficiency, D-PSGD is more efficient than C-PSGD.

[1] The Wm Michael Barnes '64 Department of Industrial and Systems Engineering, Texas A&M University, College Station, TX 77843, USA. [2] The Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA. Correspondence to: Shahin Shahrampour <shahin@tamu.edu>.

Proceedings of the 38th *International Conference on Machine Learning*, PMLR 139, 2021. Copyright 2021 by the author(s).

1.1. Our Contributions

In this paper, we focus on designing efficient decentralized algorithms to solve (1.1) over any connected undirected network. Since Stiefel manifold is nonconvex, the existing decentralized algorithms for Euclidean problems all fail for problem (1.1) (see Section 1.2). We may directly use manifold optimization tools, but the standard techniques from Riemannian optimization use the vector transport for algorithm design, which is unsatisfactory for our problem. We use the fact that Stiefel manifold is embedded in Euclidean space. As such, we combine the strategies of Euclidean algorithms with some Riemannian optimization techniques. Based on the above observations, one key innovation is to project every update direction onto the tangent space, so that we can leverage the retraction property in Lemma 2.3. This step distinguishes our algorithm from Euclidean algorithms. Our contributions are as follows:

1. We show the linear speedup of the decentralized stochastic Riemannian gradient method (Algorithm 1) w.r.t n for solving (1.1). Specifically, the iteration complexity of obtaining an $\epsilon-$stationary point (see Definition 2.2) is $\mathcal{O}(1/\epsilon^2)$ in expectation [1].

2. To achieve exact convergence with constant stepsize, we propose a gradient tracking algorithm (DRGTA) (Algorithm 2) for solving (1.1). For DRGTA, the iteration complexity of obtaining an $\epsilon-$stationary point is $\mathcal{O}(1/\epsilon)$ [1].

3. We develop *new Lipschitz inequalities* for the Riemannian gradient in Lemma 2.4, which will be of independent interest. The benefit of Lemma 2.4 is to provide us with simple analysis.

Importantly, both of the proposed algorithms are retraction-based and DRGTA is vector transport-free. These two features make the algorithms computationally cheap and conceptually simple. DRGTA is the first decentralized algorithm with exact convergence for distributed optimization on the Stiefel manifold.

1.2. Related works

Decentralized optimization has been well-studied in Euclidean space. The decentralized (sub)-gradient methods were studied in (Tsitsiklis et al., 1986; Nedic et al., 2010; Yuan et al., 2016; Chen et al., 2021b) and a distributed dual averaging subgradient method was proposed in (Duchi et al., 2011). However, with a constant stepsize $\beta > 0$, these methods can only converge to a $\mathcal{O}(\frac{\beta}{1-\sigma_2})-$neighborhood of a stationary point, where σ_2 is a network parameter (see Assumption 1). To achieve exact convergence with a fixed stepsize, gradient tracking algorithms were proposed in (Shi et al., 2015; Xu et al., 2015; Di Lorenzo & Scutari, 2016; Qu & Li, 2017; Nedic et al., 2017; Yuan et al., 2018), to name a few. The convergence analysis can be unified via a primal-dual framework (Alghunaim et al., 2020). Another way to use the constant stepsize is decentralized ADMM and its variants (Mota et al., 2013; Chang et al., 2014; Shi et al., 2014; Aybat et al., 2017). Also, decentralized stochastic gradient method for non-convex smooth problems were well-studied in (Lian et al., 2017; Assran et al., 2019; Sun et al., 2020; Xin et al., 2020), etc. We refer to the survey paper (Nedić et al., 2018) for a complete review on the state-of-the-art algorithms and the role of network topology.

The problem (1.1) can be thought as a constrained decentralized problem in Euclidean space, but since the Stiefel manifold constraint is non-convex, none of the above works can solve the problem. On the other hand, we can also treat (1.1) as a smooth problem over the Stiefel manifold. However, the constraint $x_1 = x_2 = \ldots = x_n$ is difficult to handle due to the lack of linearity on $\mathcal{M}$. Since the Stiefel manifold is an embedded submanifold in Euclidean space, our viewpoint is to treat the problem in Euclidean space and develop new tools based on Riemannian manifold optimization (Edelman et al., 1998; Absil et al., 2009; Boumal et al., 2019). For the optimization problem (1.1), a decentralized Riemannian gradient tracking algorithm was presented in (Shah, 2017). The vector transport operation should be used in (Shah, 2017), which yields expensive computation as well as analysis difficulty. Moreover, they need to use asymptotically infinite number of communication steps. A Riemannian gossip algorithm was also proposed for subspace learning on Grassmann manifold (Mishra et al., 2019), but no convergence rate was obtained. Other distributed algorithms were specifically designed either for the PCA problem (Penna & Stańczak, 2014; Raja & Bajwa, 2015; Gang & Bajwa, 2021) or in centralized topology (Fan et al., 2019; Huang & Pan, 2020; Wang et al., 2020). For aforementioned decentralized algorithms, diminishing stepsize or asymptotically infinite number of communication steps should be utilized to get the exact solution. Different from all these works, DRGTA requires a *finite* number of communications using a *constant* step-size in each iteration. After submitting our manuscript, we found that the paper (Ye & Zhang, 2021) proposed a linearly convergent method called Decentralized Exact PCA which can also use finite-step consensus. But it is only designed for the decentralized PCA problem.

As a special case of problem (1.1), the Riemannian consensus problem is well-studied; see (Sarlette & Sepulchre, 2009; Tron et al., 2012; Markdahl et al., 2020; Chen et al., 2021a). Recently, it was shown in (Chen et al., 2021a) that the multi-step consensus algorithm (DRCS) converges linearly to the global consensus in a local region.

[1] We have omitted the dependence on network parameters here.

Definition 1.1 (Consensus)**.** *Consensus is the configuration where $x_i = x_j \in \mathcal{M}$ for all $i, j \in [n]$. We define the consensus set as follows*

$$\mathcal{X}^* := \{\mathbf{x} \in \mathcal{M}^n : x_1 = x_2 = \ldots = x_n\}. \tag{1.2}$$

Specifically, DRCS iterates $\{\mathbf{x}_k\}$ have the following convergence property in a neighborhood of $\mathcal{X}^*$

$$\operatorname{dist}(\mathbf{x}_{k+1}, \mathcal{X}^*) \leq \vartheta \cdot \operatorname{dist}(\mathbf{x}_k, \mathcal{X}^*), \quad \vartheta \in (0,1), \tag{1.3}$$

where $\operatorname{dist}^2(\mathbf{x}, \mathcal{X}^*) := \min_{y\in\mathcal{M}} \frac{1}{n}\sum_{i=1}^n \|y - x_i\|_{\mathrm{F}}^2$ and $\mathbf{x}^\top = (x_1^\top \ x_2^\top \ \ldots \ x_n^\top)$. The linear rate of DRCS sheds some lights on designing the decentralized Riemannain gradient method on Stiefel manifold. More details will be provided in Section 3.

2. Preliminaries

Notation: The undirected connected graph $G = (\mathcal{V}, \mathcal{E})$ is composed of $|\mathcal{V}| = n$ nodes representing agents. We use $\mathbf{x}$ to denote the collection of all local variables x_i by stacking them, i.e., $\mathbf{x}^\top = (x_1^\top \ x_2^\top \ \ldots \ x_n^\top)$. The $n-$fold Cartesian product of $\mathcal{M}$ with itself is denoted as $\mathcal{M}^n = \mathcal{M}\times\ldots\times\mathcal{M}$. We use $[n] := \{1, 2, \ldots, n\}$. For $\mathbf{x} \in \mathcal{M}^n$, we denote the $i-$th block by $[\mathbf{x}]_i = x_i$. We denote the tangent space of $\mathcal{M}$ at point x as $\mathrm{T}_x\mathcal{M}$ and the normal space as $N_x\mathcal{M}$. The inner product on $\mathrm{T}_x\mathcal{M}$ is induced by the Euclidean inner product $\langle x, y\rangle = \mathrm{Tr}(x^\top y)$. Denote $\|\cdot\|_{\mathrm{F}}$ as the Frobenius norm and $\|\cdot\|_2$ as the operator norm. The Euclidean gradient of function $g(x)$ is $\nabla g(x)$ and the Riemannian gradient is $\mathrm{grad} g(x)$. Let I_r and 0_r be the $r\times r$ identity matrix and zero matrix, respectively. And let $\mathbf{1}_n$ denote the n dimensional vector of all ones.

The network structure is modeled using a matrix, denoted by W, which satisfies the following assumption.

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

We now introduce some preliminaries of Riemannian manifold and fundamental lemmas.

2.1. Induced Arithmetic Mean

Denote the Euclidean average point of $x_1, \ldots, x_n$ by

$$\hat{x} := \frac{1}{n}\sum_{i=1}^n x_i. \tag{2.1}$$

To measure the degree of consensus, the error $\sum_{i=1}^n \|x_i - \hat{x}\|_{\mathrm{F}}^2$ is typically used in the Euclidean decentralized algorithms. Instead, here we use the induced arithmetic mean (IAM) (Sarlette & Sepulchre, 2009) on $\mathrm{St}(d, r)$, defined as follows

$$\bar{x} := \operatorname*{argmin}_{y\in\mathrm{St}(d,r)} \sum_{i=1}^n \|y - x_i\|_{\mathrm{F}}^2 = \mathcal{P}_{\mathrm{St}}(\hat{x}), \tag{IAM}$$

where $\mathcal{P}_{\mathrm{St}}(\cdot)$ is the orthogonal projection onto $\mathrm{St}(d, r)$. Define

$$\bar{\mathbf{x}} = \mathbf{1}_n \otimes \bar{x}. \tag{2.2}$$

Then the distance between $\mathbf{x}$ and $\mathcal{X}^*$ is given by

$$\operatorname{dist}^2(\mathbf{x}, \mathcal{X}^*) = \min_{y\in\mathrm{St}(d,r)} \frac{1}{n}\sum_{i=1}^n \|y - x_i\|_{\mathrm{F}}^2 = \frac{1}{n}\|\mathbf{x} - \bar{\mathbf{x}}\|_{\mathrm{F}}^2.$$

Furthermore, we define the $l_{F,\infty}$ distance between $\mathbf{x}$ and $\bar{\mathbf{x}}$ as

$$\|\mathbf{x} - \bar{\mathbf{x}}\|_{\mathrm{F},\infty} = \max_{i\in[n]} \|x_i - \bar{x}\|_{\mathrm{F}}. \tag{$l_{F,\infty}$}$$

We will develop the analysis of decentralized Riemannian gradient descent by studying the error distance $\|\mathbf{x} - \bar{\mathbf{x}}\|_{\mathrm{F}}$ and $\|\mathbf{x} - \bar{\mathbf{x}}\|_{\mathrm{F},\infty}$.

2.2. Optimality Condition

Next, we introduce the optimality condition on manifold $\mathcal{M}$. Consider the following centralized optimization problem over a matrix manifold $\mathcal{M}$

$$\min h(x) \quad \text{s.t.} \quad x \in \mathcal{M}. \tag{2.3}$$

Since we use the metric on tangent space $\mathrm{T}_x\mathcal{M}$ induced from the Euclidean inner product $\langle\cdot,\cdot\rangle$, the Riemannian gradient $\mathrm{grad}h(x)$ on $\mathrm{St}(d, r)$ is given by $\mathrm{grad}h(x) = \mathcal{P}_{\mathrm{T}_x\mathcal{M}}\nabla h(x)$, where $\mathcal{P}_{\mathrm{T}_x\mathcal{M}}$ is the orthogonal projection onto $\mathrm{T}_x\mathcal{M}$. More specifically, we have

$$\mathcal{P}_{\mathrm{T}_x\mathcal{M}}y = y - \frac{1}{2}x(x^\top y + y^\top x),$$

for any $y \in \mathbb{R}^{d\times r}$; see (Edelman et al., 1998; Absil et al., 2009). The necessary first-order optimality condition of problem (2.3) is given as follows.

Proposition 2.1. *(Yang et al., 2014; Boumal et al., 2019) Let $x \in \mathcal{M}$ be a local optimum for (2.3). If h is differentiable at x, then $\mathrm{grad}h(x) = 0$.*

Therefore, x is a first-order critical point (or critical point) if $\mathrm{grad}h(x) = 0$. Let $\bar{x}$ be the IAM of $\mathbf{x}$. We define the $\epsilon-$stationary point of problem (1.1) as follows.

Definition 2.2 (ϵ-Stationarity)**.** *We say that $\mathbf{x}^\top = (x_1^\top \ x_2^\top \ \ldots \ x_n^\top)$ is an $\epsilon-$ stationary point of problem (1.1) if the following holds:*

$$\frac{1}{n}\sum_{i=1}^n \|x_i - \bar{x}\|_F^2 \leq \epsilon \quad \forall i, j \in [n]$$

and

$$\|\mathrm{grad} f(\bar{x})\|_F^2 \le \epsilon,$$

where we use the notation $f(\bar{x}) := \frac{1}{n}\sum_{i=1}^n f_i(\bar{x})$.

2.3. Basic Lemmas

Our goal is to develop the decentralized version of centralized Riemannian gradient descent on $\mathrm{St}(d,r)$. The centralized Riemannian gradient descent (Absil et al., 2009; Boumal et al., 2019) iterates as

$$x_{k+1} = \mathcal{R}_{x_k}(-\alpha \mathrm{grad} h(x_k)),$$

i.e., updating along a negative Riemannian gradient direction on the tangent space, and then performing an operation called *retraction* $\mathcal{R}_{x_k}$ to ensure feasibility. We use the definition of retraction in (Boumal et al., 2019, Definition 1). The retraction is the relaxation of exponential mapping, and more importantly, it is computationally cheaper. We also assume the second-order boundedness of retraction. It means that

$$\mathcal{R}_x(\xi) = x + \xi + \mathcal{O}(\|\xi\|_{\mathrm{F}}^2).$$

That is, $\mathcal{R}_x(\xi)$ is locally a good approximation of $x + \xi$. Such approximation is well enough to take the place of exponential map for the first-order algorithms.

Lemma 2.3. *(Boumal et al., 2019; Liu et al., 2019) Let $\mathcal{R}$ be a second-order retraction over* $\mathrm{St}(d,r)$*. We then have*

$$\|\mathcal{R}_x(\xi) - (x+\xi)\|_F \le M\|\xi\|_F^2, \quad \forall x \in \mathrm{St}(d,r), \forall \xi \in \mathrm{T}_x\mathcal{M}. \tag{P1}$$

Moreover, if the retraction is the polar decomposition, for all $x \in \mathrm{St}(d,r)$ and $\xi \in \mathrm{T}_x\mathcal{M}$, the following inequality holds for any $y \in \mathrm{St}(d,r)$ (Li et al., 2019, Lemma 1):

$$\|\mathcal{R}_x(\xi) - y\|_F \le \|x + \xi - y\|_F. \tag{2.4}$$

In the sequel, *retraction* refers to the *polar retraction* to present a simple analysis, unless otherwise noted. More details on the polar retraction is provided in appendix A. Throughout the paper, we assume that every $f_i(x)$ is Lipschitz smooth.

Assumption 2. *Each $f_i(x)$ has $L-$Lipschitz continuous gradient, and let $D := \max_{x\in\mathrm{St}(d,r)} \|\nabla f_i(x)\|_F$. Therefore, $\nabla f(x)$ is also L-Lipschitz continuous and* $D \ge \max_{x\in\mathrm{St}(d,r)} \|\nabla f(x)\|_F$.

We have two similar Lipschitz continuous inequalities on Stiefel manifold as the Euclidean-type ones (Nesterov, 2013). We provide the proof in Appendix.

Lemma 2.4 (Lipschitz-type inequalities)**.** *For any $x, y \in \mathrm{St}(n,d)$ and $\xi \in \mathrm{T}_x\mathcal{M}$, if $f(x)$ is $L-$Lipschitz smooth in Euclidean space, then there exists a constant $L_g = L + L_n$ such that*

$$|f(y) - [f(x) + \langle \mathrm{grad} f(x), y - x\rangle]| \le \frac{L_g}{2}\|y - x\|_F^2, \tag{2.5}$$

where $L_n = \max_{x\in\mathrm{St}(d,r)} \|\nabla f(x)\|_2$*. Moreover, define $L_G := L + 2L_n$. Then, one has*

$$\|\mathrm{grad} f(x) - \mathrm{grad} f(y)\|_F \le L_G\|y - x\|_F. \tag{2.6}$$

The difference between two Riemannian gradients is not well-defined on general manifold. However, since the Stiefel manifold is embedded in Euclidean space, we are free to do so. Another similar inequality as (2.5) is the restricted Lipschitz-type gradient presented in (Boumal et al., 2019, Lemma 4). But they do not provide an inequality as (2.6). One could also consider the following Lipschitz inequality (see (Zhang & Sra, 2016; Absil et al., 2009))

$$\|\mathrm{P}_{x\to y}\mathrm{grad} f(x) - \mathrm{grad} f(y)\|_{\mathrm{F}} \le L_g' d_g(x,y),$$

where $\mathrm{P}_{x\to y} : \mathrm{T}_x\mathcal{M} \to \mathrm{T}_y\mathcal{M}$ is the vector transport and $d_g(x,y)$ is the geodesic distance. Since involving vector transport and geodesic distance brings computational and conceptual difficulties, we choose to use the form of (2.6) for simplicity. In fact, L_g, $\tilde{L}_g$ and L_g' are the same up to a constant. A detailed comparison is provided in appendix C.1.

We will use Lemma 2.3 and Lemma 2.4 to present a parallel analysis to the decentralized Euclidean gradient methods (Nedic et al., 2010; 2017; Lian et al., 2017).

3. Review of consensus on Stiefel manifold

Decentralized gradient-based algorithms (Tsitsiklis et al., 1986; Nedic et al., 2010; Yuan et al., 2016; Shi et al., 2015; Nedic et al., 2017; Lian et al., 2017) rely on the linear convergence of consensus iteration in Euclidean space.

The consensus problem over $\mathrm{St}(d,r)$ is to minimize the quadratic loss function on Stiefel manifold

$$\begin{aligned} \min \varphi^t(\mathbf{x}) &:= \frac{1}{4}\sum_{i=1}^n\sum_{j=1}^n W_{ij}^t\|x_i - x_j\|_{\mathrm{F}}^2 \\ \text{s.t.} \quad & x_i \in \mathcal{M}, \ \forall i \in [n], \end{aligned} \tag{3.1}$$

where the superscript $t \ge 1$ is an integer used to denote the t-th power of the doubly stochastic matrix W. Note that t is introduced to provide flexibility for algorithm design and analysis, and computing W_{ij}^t corresponds to performing t steps of communication on the tangent space. For consensus on the Steifel manifold, the Riemannian gradient method DRCS was proposed in (Chen et al., 2021a), where for any $i \in [n]$,

$$x_{i,k+1} = \mathcal{R}_{x_{i,k}}(\alpha\mathcal{P}_{\mathrm{T}_{x_i}\mathcal{M}}(\sum_{j=1}^n W_{ij}^t x_{j,k})). \tag{3.2}$$

DRCS converges almost surely to consensus when $r \leq \frac{2}{3}d - 1$ with random initialization (Markdahl et al., 2020). However, to study decentralized optimization problem (1.1), the local Q-linear convergence of DRCS is more important. Due to the nonconvexity of $\mathcal{M}$, the Q-linear rate of DRCS holds in a local region defined as follows

$$\mathcal{N} := \mathcal{N}_1 \cap \mathcal{N}_2, \tag{3.3}$$

$$\mathcal{N}_1 := \{\mathbf{x} : \|\mathbf{x} - \bar{\mathbf{x}}\|_{\mathrm{F}}^2 \leq n\delta_1^2\}, \tag{3.4}$$

$$\mathcal{N}_2 := \{\mathbf{x} : \|\mathbf{x} - \bar{\mathbf{x}}\|_{\mathrm{F},\infty} \leq \delta_2\}, \tag{3.5}$$

where δ_1, δ_2 satisfy

$$\delta_1 \leq \frac{1}{5\sqrt{r}}\delta_2 \quad \textit{and} \quad \delta_2 \leq \frac{1}{6}. \tag{3.6}$$

The following convergence result of DRCS can be found in (Chen et al., 2021a, Theorem 2). The formal statement is provided in Fact B.1 in Appendix.

Fact 3.1. ***(Informal)*** *Under Assumption 1, for some* $\bar{\alpha} \in (0, 1]$*, if* $\alpha \leq \bar{\alpha}$ *and* $t \geq \lceil \log_{\sigma_2}(\frac{1}{2\sqrt{n}}) \rceil$*, the sequence* $\{\mathbf{x}_k\}$ *in* (3.2) *achieves consensus linearly if the initialization satisfies* $\mathbf{x}_0 \in \mathcal{N}$ *defined in* (3.3)*. That is, there exists* $\rho_t \in (0, 1)$ *such that* $\mathbf{x}_k \in \mathcal{N}$ *for all* $k \geq 0$ *and*

$$\|\mathbf{x}_{k+1} - \bar{\mathbf{x}}_{k+1}\|_F \leq \rho_t \|\mathbf{x}_k - \bar{\mathbf{x}}_k\|_F. \tag{3.7}$$

4. Decentralized Riemannian gradient descent

Using the results of consensus problem on Stiefel manifold, we can combine the ideas of decentralized gradient method in Euclidean space with the Stiefel manifold optimization. In this section, we propose a distributed Riemannian stochastic gradient method for solving problem (1.1), which is described in Algorithm 1.

Algorithm 1 Decentralized Riemannian Stochastic Gradient Descent (DRSGD) for Solving (1.1)

```
1: **Input:** initial point $\mathbf{x}_0 \in \mathcal{N}$, an integer $t \geq \log_{\sigma_2}(\frac{1}{2\sqrt{n}})$, $0 < \alpha \leq \bar{\alpha}$, where $\bar{\alpha}$ is given in Fact 3.1.
2: **for** $k = 0, \ldots$ {for each node $i \in [n]$, in parallel} **do**
3:     Choose diminishing stepsize $\beta_k = \mathcal{O}(1/\sqrt{k})$
4:     Compute stochastic Riemannian gradient $v_{i,k}$ satisfying $\mathbb{E}v_{i,k} = \mathrm{grad} f_i(x_{i,k})$
5:     Update
       $x_{i,k+1} = \mathcal{R}_{x_{i,k}}(\alpha \mathcal{P}_{\mathrm{T}_{x_{i,k}}\mathcal{M}}(\sum_{j=1}^n W_{ij}^t x_{j,k}) - \beta_k v_{i,k})$
6: **end for**
```

Since we need all the local variables to be equal according to the constraint in (1.1), the initial point $\mathbf{x}_0$ should be in the consensus region $\mathcal{N}$. One can simply initialize all agents from the same point. The line 5 in Algorithm 1 first performs a consensus step and then updates the local variable using Riemannian stochastic gradient direction $v_{i,k}$. The consensus step and computation of Riemannian gradient can be done in parallel[2]. The consensus stepsize α satisfies $\alpha \leq \bar{\alpha}$, which is the same as the consensus algorithm. The constant $\bar{\alpha}$ is given in Fact B.1 in Appendix. Moreover, $\alpha = 1$ works in practice for any W satisfying Assumption 1. If $x_1 = \ldots = x_n = z$, we denote

$$f(z) := \frac{1}{n}\sum_{i=1}^n f_i(z).$$

Moreover, we need the following assumptions on the stochastic Riemannian gradient $v_{i,k}$ and the stepsize β_k.

Assumption 3. *1. The stochastic gradient* $v_{i,k}$ *is unbiased, i.e.,* $\mathbb{E}v_{i,k} = \mathrm{grad} f_i(x_{i,k})$ *for all* $i \in [n], k$ *and* $v_{i,k}$ *is independent of* $v_{j,k}$ *for any* $i \neq j$*. Moreover, the variance is bounded:* $\mathbb{E}\|v_{i,k} - \mathrm{grad} f_i(x_{i,k})\|_F^2 \leq \Xi^2$ *for some* $\Xi > 0$.

2. We assume a uniform upper bound on $\|v_{i,k}\|_F$ *exists, and* $\max_{x \in \mathrm{St}(d,r)} \|v_{i,k}\|_F \leq D$ *for each* $i \in [n]$ *and* k.

The Lipschitz smoothness of $f_i(x)$ in Assumption 2 and unbiased gradients are quite standard in the literature. And Lemma 2.4 suggests that $\mathrm{grad} f_i$ is L_G-Lipschitz continuous. Also, the boundedness of $\|v_{i,k}\|_{\mathrm{F}}$ is a weak assumption given that Stiefel manifold is compact. One common example is the finite-sum form: $f_i = \frac{1}{m_i}\sum_{j=1}^{m_i} f_{ij}$, where f_{ij} is smooth and m_i is the number of functions f_{ij} at local agent i. Then the stochastic gradient $v_{i,k}$ is uniformly sampled from $\mathrm{grad} f_{ij}(x_{i,k}), j \in [m_i]$. We emphasize that the uniform boundedness of gradient is not needed for problems in Euclidean space, but Lipschitz continuity is necessary (Hong et al., 2020).

The step 5 can be seen as applying Riemannian gradient method to solve the following problem

$$\min_{\mathbf{x} \in \mathcal{M}^n} \beta_k f(\mathbf{x}) + \alpha\varphi^t(\mathbf{x}).$$

Similar to the analysis of DGD in Euclidean space, we need to ensure that $\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|_{\mathrm{F}} \to 0$. Hence, the effect of f should be diminishing. The following assumption on the stepsize is also needed to get an $\epsilon-$ solution.

Assumption 4 (Diminishing stepsize)**.** *The stepsize* $\beta_k > 0$

[2] One could also exchange the order of gradient step and communication step, i.e., $x_{i,k+\frac{1}{2}} = \mathcal{R}_{x_{i,k}}(-\beta_k v_{i,k})$, $x_{i,k+1} = \mathcal{R}_{x_{i,k+\frac{1}{2}}}(\alpha\mathcal{P}_{\mathrm{T}_{x_{i,k+\frac{1}{2}}}\mathcal{M}}(\sum_{j=1}^n W_{ij}^t x_{j,k+\frac{1}{2}}))$. Our analysis can also apply to this kind of update if $\mathbf{x}_0 \in \rho_t\mathcal{N}$, where $\rho_t\mathcal{N}$ denotes region $\mathcal{N}$ with the shrunken radius $\rho_t\delta_1, \rho_t\delta_2$. For the Euclidean algorithms, when the graph is complete with $W = \mathbf{1}_n\mathbf{1}_n^\top/n$, the above updates are the same as centralized gradient step. However, they are different on Stiefel manifold.

is non-increasing and

$$\sum_{k=0}^{\infty} \beta_k = \infty, \quad \lim_{k\to\infty} \beta_k = 0, \quad \lim_{k\to\infty} \frac{\beta_{k+1}}{\beta_k} = 1.$$

The assumption $\lim_{k\to\infty} \frac{\beta_{k+1}}{\beta_k} = 1$ is required to show the bound $\frac{1}{n}\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|_{\mathrm{F}}^2 = \mathcal{O}(\frac{\beta_k^2 D^2}{(1-\rho_t)^2})$, see Lemma D.3 in Appendix.

To proceed, we first need to guarantee that $\mathbf{x}_k \in \mathcal{N}$, where $\mathcal{N}$ is the consensus contraction region defined in (3.3). Therefore, uniform bound D and the multi-step consensus requirement $t \geq \lceil \log_{\sigma_2}(\frac{1}{2\sqrt{n}})\rceil$ are necessary in our convergence analysis. With appropriate stepsizes α and β_k, we get the following lemma using the consensus results in Fact 3.1. We provide the proof in Appendix.

Lemma 4.1. *Under Assumptions 1 to 4, let the stepsize α satisfy $0 < \alpha \leq \bar{\alpha}$, β_k satisfy $0 \leq \beta_k \leq \min\{\frac{1-\rho_t}{D}\delta_1, \frac{\alpha\delta_1}{5D}\}, \forall k \geq 0$, and $t \geq \lceil \log_{\sigma_2}(\frac{1}{2\sqrt{n}})\rceil$. If $\mathbf{x}_0 \in \mathcal{N}$, it follows that $\mathbf{x}_k \in \mathcal{N}$ for all $k \geq 0$ generated by Algorithm 1 and*

$$\|\mathbf{x}_{k+1} - \bar{\mathbf{x}}_{k+1}\|_F \leq \rho_t^{k+1}\|\mathbf{x}_0 - \bar{\mathbf{x}}_0\|_F + \sqrt{n}D\sum_{l=0}^{k} \rho_t^{k-l}\beta_l.$$

We have $\beta_k = \mathcal{O}(\frac{1-\rho_t}{D})$ when $\alpha = \mathcal{O}(1)$. Note that $t \geq \lceil \log_{\sigma_2}(\frac{1}{2\sqrt{n}})\rceil$ implies $\rho_t = \mathcal{O}(1)$; see appendix B. When $\beta_k = \beta$ is constant, Lemma 4.1 suggests that $\mathbf{x}_k$ converges linearly to an $\mathcal{O}(\beta)$-neighborhood of $\bar{\mathbf{x}}_k$.

We present the convergence of Algorithm 1. The proof is based on the new Lipschitz inequalities for the Riemannian gradient in Lemma 2.4 and the properties of retraction in Lemma 2.3. We provide it in Appendix.

Theorem 4.2. *Under Assumptions 1 to 4, suppose $\mathbf{x}_0 \in \mathcal{N}$, $t \geq \lceil \log_{\sigma_2}(\frac{1}{2\sqrt{n}})\rceil$, $0 < \alpha \leq \bar{\alpha}$. If*

$$\beta_k = \frac{1}{\sqrt{k+1}} \cdot \min\{\frac{1}{5L_g}, \frac{\alpha\delta_1}{5D}, \frac{1-\rho_t}{D}\delta_1\}, \tag{4.1}$$

it follows that

$$\min_{k\leq K} \mathbb{E}\|\mathrm{grad} f(\bar{x}_k)\|_F^2 \leq \frac{4(f(\bar{x}_0) - f^*) + \frac{6L_g\Xi^2}{n}\sum_{k=0}^K \beta_k^2}{\sum_{k=0}^K \beta_k} \tag{4.2}$$

$$+ \frac{(2CD^2L_G^2 + 4\mathcal{T}_1 D^4)\sum_{k=0}^K \beta_k^3 + 4\mathcal{T}_2 L_g D^4 \sum_{k=0}^K \beta_k^4}{\sum_{k=0}^K \beta_k},$$

where $C = \mathcal{O}(\frac{1}{(1-\rho_t)^2})$ is given in Lemma D.3 in Appendix. And $\mathcal{T}_1 = 2(4\sqrt{r} + 6\alpha)^2C^2 + 8M^2$ and $\mathcal{T}_2 = 201\alpha^2C^2 + 9M^2$. Therefore, with stepsize $\beta_k = \mathcal{O}(1/\sqrt{k})$, we have

$$\min_{k\leq K} \mathbb{E}\|\mathrm{grad} f(\bar{x}_k)\|_F^2 = \mathcal{O}\left(\frac{f(\bar{x}_0) - f^*}{\tilde{\beta}\sqrt{K+1}} + \frac{\Xi^2\ln(K+1)}{n\sqrt{K+1}}\right)$$
$$+ \mathcal{O}\left(\frac{\max\{D^2, L_G^2\}\cdot(C + \mathcal{T}_1 + \mathcal{T}_2)}{\sqrt{K+1}}\right),$$

where $\tilde{\beta} = \min\{1/L_g, (1-\rho_t)/D\}$.

Theorem 4.2 together with Lemma D.3 implies that the iteration complexity of obtaining an $\epsilon-$stationary point defined in Definition 2.2 is $\mathcal{O}(1/\epsilon^2)$ in expectation. The communication round per iteration is $t \geq \lceil \log_{\sigma_2}(\frac{1}{2\sqrt{n}})\rceil$ since we need to ensure $\mathbf{x}_k \in \mathcal{N}$. For sparse networks, t can be $\mathcal{O}(n^2\log n)$ (Chen et al., 2021a).

Note that in Assumption 3, the uniform bound D is required for every $v_{i,k}$. This is used in the proof of Lemma 4.1. We can prove a weaker version of Theorem 4.2 without assuming bounded variance Ξ in Assumption 3. However, we hope to provide a parallel analysis as the D-PSGD (Lian et al., 2017) to show the linear speedup of DRSGD can be achieved w.r.t the network size n. With the bounded variance assumption, we have an $\mathcal{O}(\frac{\Xi^2}{n})$ term in (4.2). This reveals the role of the batch size n in DRSGD. Following (Lian et al., 2017), if we use the constant stepsize $\beta_k = \frac{1}{2L_G + \sqrt{(K+1)/n}}$ where K is sufficiently large, we can obtain the following result

$$\min_{k=0,\ldots,K} \mathbb{E}\|\mathrm{grad} f(\bar{x}_k)\|_{\mathrm{F}}^2$$
$$\leq \frac{8L_G(f(\bar{x}_0) - f^*)}{K+1} + \frac{8(f(\bar{x}_0) - f^* + \frac{3L_G}{2})\Xi}{\sqrt{n(K+1)}}.$$

More details are provided in Corollary D.5 in Appendix. Therefore, if K is sufficiently large, the convergence rate is $\mathcal{O}(1/\sqrt{nK})$. To obtain an $\epsilon-$stationary point, the computational complexity of single node is $\mathcal{O}(\frac{1}{n\epsilon^2})$. Moreover, K should be proportional to $\mathcal{O}(1/\Xi)$ in Corollary D.5. This also means that in deterministic setting ($\Xi = 0$), the linear speedup cannot be obtained.

One remaining issue is that the communication round $t \geq \lceil \log_{\sigma_2}(\frac{1}{2\sqrt{n}})\rceil$ is too large. In practice, we find that $t = 1$ performs well as shown in the experiments in Section 6. We conjecture that when the stepsize is small enough, DRSGD will not deviate from the consensus algorithm DRCS too much. We will theoretically study this in the future.

5. Gradient tracking on Stiefel manifold

In this section, we study the decentralized gradient tracking method, which is based on the DIGing algorithm (Qu & Li, 2017; Nedic et al., 2017) for solving Euclidean problems. With an auxiliary gradient tracking sequence to estimate the full gradient, the constant stepsize can be used and faster convergence rate can be shown for the Euclidean algorithms (Nedic et al., 2017; Shi et al., 2015). Our method, termed Decentralized Riemannian Gradient Tracking Algorithm (DRGTA), is described in Algorithm 2.

In Algorithm 2, the step 4 is to project the direction $y_{i,k}$ onto the tangent space $\mathrm{T}_{x_{i,k}}\mathcal{M}$, which follows a retraction update. The sequence $\{y_{i,k}\}$ is to approximate the

Algorithm 2 Decentralized Riemannian Gradient Tracking over Stiefel manifold (DRGTA) for Solving (1.1)

```
1: Input: initial point $\mathbf{x}_0 \in \mathcal{N}$, an integer $t \geq \log_{\sigma_2}(\frac{1}{2\sqrt{n}})$,
   $0 < \alpha \leq \bar{\alpha}$ and stepsize $\beta$ according to (5.2).
2: Let $y_{i,0} = \mathrm{grad} f_i(x_{i,0})$ on each node $i \in [n]$.
3: for $k = 0, \ldots$ {for each node $i \in [n]$, in parallel} do
4:     Projection onto tangent space: $v_{i,k} = \mathcal{P}_{\mathrm{T}_{x_{i,k}}\mathcal{M}} y_{i,k}$.
5:     Update
       $x_{i,k+1} = \mathcal{R}_{x_{i,k}}(\alpha \mathcal{P}_{\mathrm{T}_{x_{i,k}}\mathcal{M}}(\sum_{j=1}^n W_{ij}^t x_{j,k}) - \beta v_{i,k})$.
6:     Riemannian gradient tracking:
       $$y_{i,k+1} = \sum_{j=1}^n W_{ij}^t y_{j,k} + \mathrm{grad} f_i(x_{i,k+1}) - \mathrm{grad} f_i(x_{i,k}).$$
7: end for
```

Riemannian gradient $\mathrm{grad} f_i(x_{i,k})$. More specifically, the sequence $\{y_k\}$ tracks the average Riemannian gradient $\frac{1}{n}\sum_{i=1}^n \mathrm{grad} f_i(x_{i,k})$. Although it is not mathematically sound to perform addition operation between tangent spaces in differential geometry, we can view $\mathrm{grad} f_i(x_{i,k})$ as the projected Euclidean gradient. Note that $y_{i,k}$ is not necessarily on the tangent space $\mathrm{T}_{x_{i,k}}\mathcal{M}$. Therefore, it is important to define $v_{i,k} = \mathcal{P}_{\mathrm{T}_{x_{i,k}}\mathcal{M}} y_{i,k}$ so that we can use the properties of retraction in Lemma 2.3. Such a projection onto tangent space, followed by the retraction operation, distinguishes the algorithm from the Euclidean space gradient tracking algorithms. Multi-step consensus of gradient is also required in step 5 and step 6. The consensus stepsize α satisfies the same condition as that of Algorithm 1.

5.1. Convergence of Riemannian gradient tracking

We first briefly revisit the idea of gradient tracking (GT) algorithm DIGing in Euclidean space. Note that if we consider the decentralized optimization problem (1.1) without the Stiefel manifold constraint, then Algorithm 2 is exactly the same as the DIGing. Since the Riemannian gradient $\mathrm{grad} f_i$ becomes simply the Euclidean gradient ∇f_i and projection onto the tangent space and retraction are not needed. The main advantage of Euclidean gradient tracking algorithm is that one can use constant stepsize $\beta > 0$, which is due to following observation: for all $k \geq 0$, it follows that

$$\frac{1}{n}\sum_{i=1}^n y_{i,k} = \frac{1}{n}\sum_{i=1}^n \nabla f_i(x_{i,k}).$$

That is, the average of sequence $y_{i,k}$ is the same as that of $\nabla f_i(x_{i,k})$. It can be shown that the following inexact gradient sequence converges to a stationary point (Nedic et al., 2017)

$$x_{i,k+1} = \sum_{i=1}^n W_{ij} x_{j,k} - \beta \frac{1}{n}\sum_{i=1}^n \nabla f_i(x_{i,k}).$$

However, the average of gradient information is unavailable in the decentralized setting. Therefore, GT uses $\frac{1}{n}\sum_{i=1}^n y_{i,k}$ to approximate $\frac{1}{n}\sum_{i=1}^n \nabla f_i(x_{i,k})$. Inspired by this, $y_{i,k}$ is used to approximate the Riemannian gradient, i.e., if

$$y_{i,k+1} = \sum_{j=1}^n W_{ij}^t y_{j,k} + \mathrm{grad} f_i(x_{i,k+1}) - \mathrm{grad} f_i(x_{i,k}),$$

then it follows that

$$\frac{1}{n}\sum_{i=1}^n y_{i,k} = \frac{1}{n}\sum_{i=1}^n \mathrm{grad} f_i(x_{i,k}) \quad \text{i.e.} \quad \hat{y}_k = \hat{g}_k.$$

Therefore, $\{\mathbf{y}_k\}$ tracks the average of Riemannian gradient, and if $\|\hat{g}_k\|_{\mathrm{F}} \to 0$ and the sequence $\{\mathbf{x}_k\}$ achieves consensus, then $\mathbf{x}_k$ also converges to a critical point. This is because

$$\begin{aligned} &\|\mathrm{grad} f(\bar{x}_k)\|_{\mathrm{F}}^2 \leq 2\|\hat{g}_k\|_{\mathrm{F}}^2 + 2\|\mathrm{grad} f(\bar{x}_k) - \hat{g}_k\|_{\mathrm{F}}^2 \\ \overset{(2.6)}{\leq}\ & 2\|\hat{g}_k\|_{\mathrm{F}}^2 + \frac{2L_G^2}{n}\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|_{\mathrm{F}}^2. \end{aligned}$$

The above observations will play important roles in the convergence analysis. To achieve consensus, we still need multi-step consensus in DRGTA. The multi-step consensus also helps us to show the uniform boundedness of $y_{i,k}$ and $v_{i,k}$, $i \in [n]$ for all $k \geq 0$, which is important to guarantee $\mathbf{x}_k \in \mathcal{N}$. We get that the sequence stays in consensus region $\mathcal{N}$ in Lemma 5.1. We provide the proof in Appendix.

Lemma 5.1 (Uniform bound of y_i and stay in $\mathcal{N}$)**.** *Under Assumptions 1 and 2, let $\mathbf{x}_0 \in \mathcal{N}$, $t \geq \log_{\sigma_2}(\frac{1}{2\sqrt{n}})$, α satisfy $0 < \alpha \leq \bar{\alpha}$, β satisfy $0 \leq \beta \leq \bar{\beta} := \min\{\frac{1-\rho_t}{L_G+2D}\delta_1, \frac{\alpha\delta_1}{5(L_G+2D)}\}$, then $\|y_{i,k}\|_F \leq L_G + 2D$ for all $i \in [n]$ and $\mathbf{x}_k \in \mathcal{N}$ for all $k \geq 0$. Moreover, we have*

$$\frac{1}{n}\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|_F^2 \leq C_1(L_G + 2D)^2\beta^2, \tag{5.1}$$

for some $C_1 = \mathcal{O}(\frac{1}{(1-\rho_t)^2})$, and C_1 is independent of L_G and D.

We present the $\mathcal{O}(1/\epsilon)$ iteration complexity to obtain the $\epsilon-$stationary point of (1.1) as follows. The proof of DIGing can be unified by the primal-dual framework (Alghunaim et al., 2020). However, DRGTA cannot be rewritten in the primal-dual form. The proof is mainly established with the help of Lemma 2.4 and the properties of IAM. We provide it in Appendix.

Theorem 5.2. *Under Assumptions 1 and 2, let $\mathbf{x}_0 \in \mathcal{N}$, $t \geq \lceil\log_{\sigma_2}(\frac{1}{2\sqrt{n}})\rceil$, $0 < \alpha \leq \bar{\alpha}$, and*

$$0 < \beta \leq \min\{\bar{\beta}, \frac{1}{8L_G}, \frac{1}{4L_G(2\mathcal{G}_3 + (8\tilde{C}_0 + \frac{1}{2}\tilde{C}_2)\alpha\delta_1)}\}, \tag{5.2}$$

where $\bar{\beta}$ is given in Lemma 5.1. Then it follows that for the sequences generated by Algorithm 2

$$\min_{k=0,\ldots,K} \frac{1}{n}\|\mathbf{y}_k\|_F^2 \leq \frac{8(f(\bar{x}_0) - f^* + \tilde{C}_4 + \mathcal{G}_4 L_G)}{\beta \cdot K}, \tag{5.3}$$

$$\begin{aligned} &\min_{k\leq K} \frac{1}{n}\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|_F^2 \\ &\leq \frac{8\beta(f(\bar{x}_0) - f^* + \tilde{C}_4 + \mathcal{G}_4 L_G)\tilde{C}_0 + \tilde{C}_1}{K}, \end{aligned} \tag{5.4}$$

$$\begin{aligned} &\min_{k\leq K} \|\mathrm{grad} f(\bar{x}_k)\|_F^2 \\ &\leq \frac{(16 + \alpha^2\delta_1^2\tilde{C}_0)(f(\bar{x}_0) - f^* + \tilde{C}_4 + \mathcal{G}_4 L_G) + \tilde{C}_1 L_G}{\beta \cdot K}, \end{aligned} \tag{5.5}$$

where the constants above are given by

$$\begin{aligned} &\mathcal{G}_3 = \mathcal{G}_1\tilde{C}_0 + \mathcal{G}_0\tilde{C}_0 + \mathcal{G}_2, \\ &\mathcal{G}_4 = \frac{\mathcal{G}_0\tilde{C}_0\delta_1^2\alpha^2}{25} + \tilde{C}_1(\mathcal{G}_1 + 4rC_1), \\ &\tilde{C}_0 = \frac{2}{(1-\rho_t)^2}, \quad \tilde{C}_1 = \frac{2}{1-\rho_t^2} \cdot \frac{1}{n}\|\mathbf{x}_0 - \bar{\mathbf{x}}_0\|_F^2, \\ &\tilde{C}_2 = \frac{2}{(1-\sigma_2^t)^2}, \quad \tilde{C}_3 = \frac{2}{1-\sigma_2^{2t}} \cdot \frac{1}{n}\|\mathbf{y}_0 - \hat{\mathbf{G}}_0\|_F^2, \\ &\tilde{C}_4 = (8\alpha^2\tilde{C}_1\tilde{C}_2 L_G^2 + \tilde{C}_3) \cdot \frac{\beta}{2} = \mathcal{O}(\frac{L_G}{(1-\sigma_2^t)^2}). \end{aligned}$$

The constants $\mathcal{G}_0 = \mathcal{O}(r^2C_1)$, $\mathcal{G}_1 = \mathcal{O}(r^2C_1)$ and $\mathcal{G}_2 = \mathcal{O}(M)$ are given in Lemma E.2 in the appendix. We have $\mathcal{G}_3 = \mathcal{O}(\frac{r^2C_1}{(1-\rho_t)^2} + M)$ and $\mathcal{G}_4 = \mathcal{O}(\frac{r^2C_1\delta_1^2}{1-\rho_t^2})$. Recall that $\beta \leq \bar{\beta}$ is required to guarantee that the sequence $\{\mathbf{x}_k\}$ always stays in the consensus region $\mathcal{N}$. And note that ρ_t is the linear rate of Riemannian consensus, which is greater than σ_2^t. The stepsize β follows

$$\beta = \mathcal{O}(\min\{\frac{1-\rho_t}{L_G + 2D}, \frac{(1-\rho_t)^2}{L_G}\cdot\frac{1}{r^2C_1 + M(1-\rho_t)^2}\}).$$

This matches the bound of DIGing (Qu & Li, 2017; Nedic et al., 2017). Then Theorem 5.2 suggests that the consensus error rate is $\mathcal{O}(\frac{1}{(r^2C_1+M)L_G} \cdot \frac{f(\bar{x}_0)-f^*}{K} + \frac{\|\mathbf{x}_0-\bar{\mathbf{x}}_0\|_\mathrm{F}^2}{n(1-\rho_t^2)K})$ and the convergence rate of $\min_{k=0,\ldots,K} \|\mathrm{grad} f(\bar{x}_k)\|_\mathrm{F}^2$ is given by $\mathcal{O}(\frac{(r^2C_1+M)(L_G+2D)(f(\bar{x}_0)-f^*))}{K(1-\rho_t)^2} + \frac{\|\mathbf{x}_0-\bar{\mathbf{x}}_0\|_\mathrm{F}^2}{n(1-\rho_t)^4T} + \frac{r^2C_1\delta_1^2L_G}{K(1-\rho_t)^6})$. Moreover, if the initial points satisfy $x_{1,0} = x_{2,0} = \ldots = x_{n,0}$, we have $\tilde{C}_1 = \tilde{C}_3 = \tilde{C}_4 = 0$.

6. Numerical experiment

We consider the following decentralized eigenvector problem:

$$\min_{\mathbf{x}\in\mathcal{M}^n} -\frac{1}{2n}\sum_{i=1}^n x_i^\top A_i^\top A_i x_i, \quad \text{s.t.} \quad x_1 = \ldots = x_n, \tag{6.1}$$

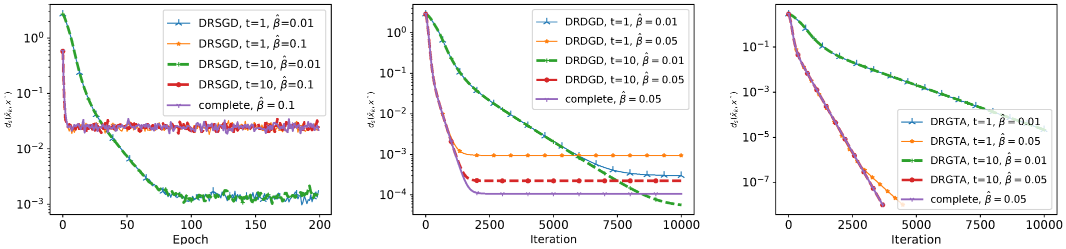
where $A_i \in \mathbb{R}^{m_i\times d}, i \in [n]$ is the local data matrix for agent i and m_i is the sample size. Denote the global data matrix by $A := [A_1^\top\ A_2^\top \ldots A_n^\top]^\top$. It is known that the global minimizer of (6.1) is given by the first r leading eigenvectors of $A^\top A = \sum_{i=1}^n A_i^\top A_i$, denoted by x^*. DRSGD and DRGTA are proved to only converge to the critical points, but we find that they always converge to x^* in our experiments. Denote the column space of a matrix x by $[x]$. To measure the quality of the solution, the distance between column space $[x]$ and $[y]$ can be defined via the canonical correlations between $x \in \mathbb{R}^{d\times r}$ and $y \in \mathbb{R}^{d\times r}$(Golub & Zha, 1995). One can define it by

$$d_s(x,y) := \min_{Q\in\mathrm{O}(r)} \|uQ - v\|_\mathrm{F},$$

where $\mathrm{O}(r)$ is the orthogonal group, u and v are the orthogonal basis of $[x]$ and $[y]$, respectively. In the sequel, we fix $\alpha = 1$ and generate the initial points uniformly randomly satisfying $x_{1,0} = \ldots = x_{n,0} \in \mathcal{M}$. If full batch gradient is used in Algorithm 1, we call it DRDGD, otherwise one stochastic gradient is uniformly sampled without replacement in DRSGD. In DRSGD, one epoach represents the number of complete passes through the dataset, while one iteration is used in the deterministic algorithms. For DRSGD, we set the maximum epoch to 200 and early stop it if $d_s(\bar{x}_k, x^*) \leq 10^{-5}$. For DRGTA and DRDGD, we set the maximum iteration number to 10^4 and the termination condition is $d_s(\bar{x}_k, x^*) \leq 10^{-8}$ or $\|\mathrm{grad} f(\bar{x}_k)\|_\mathrm{F} \leq 10^{-8}$. We set $\beta_k = \frac{\hat{\beta}}{\frac{1}{n}\sum_{i=1}^n m_i}$ for DRGTA and DRDGD where $\hat{\beta}$ will be specified later. For DRSGD, we set $\beta = \frac{\hat{\beta}}{\sqrt{200}}$. Due to the space limit, we show the results on the ring graph, and select the weight matrix W to be the Metroplis constant weight (Shi et al., 2015). More comparisons on different networks and dataset are provided in Appendix. For reproducibility of results, our code is made available at `https://github.com/chenshixiang/Decentralized_Riemannian_gradient_descent_on_Stiefel_manifold`.

6.1. Synthetic data

We report the convergence results of DRSGD, DRDGD and DRGTA with different t and $\hat{\beta}$ on synthetic data. We fix $m_1 = \ldots = m_n = 1000$, $d = 100$ and $r = 5$ and generate $m_1 \times n$ i.i.d samples following standard multi-variate Gaussian distribution to obtain A. Let $A = USV^\top$



(a) DRSGD (b) DRDGD (c) DRGTA

Figure 1. Synthetic data, agents number $n = 32$, eigengap $\Delta = 0.8$.

be the truncated SVD. Given an eigengap $\Delta \in (0, 1)$, we modify the singular values of A to be a geometric sequence, i.e. $S_{i,i} = S_{0,0} \times \Delta^{i/2}, i \in [d]$. Typically, larger Δ results in more difficult problem.

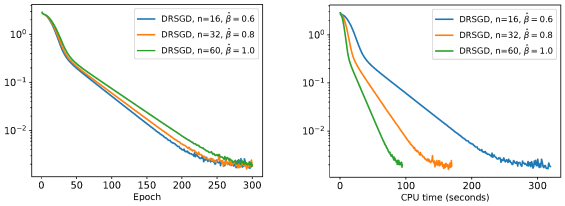
In Figure 1, we show the results of DRSGD, DRDGD and DRGTA on the data with $n = 32$ and $\Delta = 0.8$. The y-axis is the log-scale distance. The first four lines in each testing case are for the ring graph, and the last one is on a complete graph with equally weighted matrix, which aims to show the case of $t \to \infty$. In Figure 1(a), when fixing $\hat{\beta}$, it is shown that that smaller $\hat{\beta}$ produces higher accuracy, which verifies Theorem 4.2. We see DRSGD performs almost the same with different $t \in \{1, 10, \infty\}$. For the two deterministic algorithms DRDGD and DRGTA, we see that DRDGD can use larger $\hat{\beta}$ if more communication rounds t is used in Figure 1(b),(c). DRDGD cannot achieve exact convergence with constant stepsize, while DRGTA successfully solves the problem using $t \in \{10, \infty\}, \hat{\beta} = 0.05$.

6.2. Real-world data

We provide some numerical results on the MNIST dataset (LeCun). For simplicity, we fix $t = 1$ and $r = 5$ in this section. The graph is still the ring and W is the Metropolis constant weight matrix. For MNIST, there are 60000 samples and the dimension is given by $d = 784$. We normalize the data matrix such that the elements are in $[0, 1]$. The data set is evenly partitioned into n subsets. The stepsizes of DRDGD and DRGTA are set to $\beta = \frac{\hat{\beta}}{60000}$.

We demonstrate the linear speedup of DRSGD for different n. The experiments are evaluated in a HPC cluster, where each computation node is an Intel Xeon 6248R CPU. The computation nodes are connected by Mellanox HDR 100 InfiniBand. We use 2 CPU cores each computation node in the HPC cluster. And we treat one CPU core as one network node in our problem. The codes are implemented in python with mpi4py (Dalcín et al., 2005).

We set the maximum epoch as 300 in all experiments. The stepsize is set to $\beta = \frac{\sqrt{n}}{10000\sqrt{300}}\hat{\beta}$, where $\hat{\beta}$ is tuned for the best performance. The results in Figure 2 are $\log d_s(\bar{x}_k, x^*)$ v.s. epoch and $\log d_s(\bar{x}_k, x^*)$ v.s. CPU time, respectively. As we see in Figure 2(a), the solutions accuracy of $n = 16, 32, 60$ are almost the same, while the CPU time in Figure 2(b) can be accelerated by nearly linear ratio.



(a) epoch (b) time

Figure 2. Comparison results of different number of nodes on MNIST. Ring graph associated with Metropolis constant weight matrix, $t = 1$, $\beta = \frac{\sqrt{n}}{10000\sqrt{300}}\hat{\beta}$.

7. Conclusions

We proposed two decentralized Riemannian gradient methods and established their convergence rates. Future studies include several directions. Firstly, for the eigenvector problem (6.1), it will be interesting to establish the linear convergence of DRGTA. Secondly, our algorithms rely on the Q-linear rate of the Riemannian consensus algorithm in (Chen et al., 2021a). In the future work, we will explore other manifolds, especially those embedded in Euclidean space.

Acknowledgements

We gratefully acknowledge the support of National Science Foundation, Grant # ECCS-1933878 as well as Air Force Office of Scientific Research, Grant # 19RT0424. Portions of this research were conducted with high performance research computing resources provided by Texas A&M University *(https://hprc.tamu.edu)*.

References

Absil, P.-A., Mahony, R., and Sepulchre, R. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

Absil, P.-A., Mahony, R., and Trumpf, J. An extrinsic look at the riemannian hessian. In *International Conference on Geometric Science of Information*, pp. 361–368. Springer, 2013.

Alghunaim, S. A., Ryu, E., Yuan, K., and Sayed, A. H. Decentralized proximal gradient algorithms with linear convergence rates. *IEEE Transactions on Automatic Control*, 2020.

Arjovsky, M., Shah, A., and Bengio, Y. Unitary evolution recurrent neural networks. In *International Conference on Machine Learning*, pp. 1120–1128. PMLR, 2016.

Assran, M., Loizou, N., Ballas, N., and Rabbat, M. Stochastic gradient push for distributed deep learning. In *International Conference on Machine Learning*, pp. 344–353. PMLR, 2019.

Aybat, N. S., Wang, Z., Lin, T., and Ma, S. Distributed linearized alternating direction method of multipliers for composite convex consensus optimization. *IEEE Transactions on Automatic Control*, 63(1):5–20, 2017.

Boumal, N., Absil, P.-A., and Cartis, C. Global rates of convergence for nonconvex optimization on manifolds. *IMA Journal of Numerical Analysis*, 39(1):1–33, 2019.

Chang, T.-H., Hong, M., and Wang, X. Multi-agent distributed optimization via inexact consensus admm. *IEEE Transactions on Signal Processing*, 63(2):482–497, 2014.

Chen, S., Garcia, A., Hong, M., and Shahrampour, S. On the local linear rate of consensus on the stiefel manifold. *arXiv preprint arXiv:2101.09346*, 2021a.

Chen, S., Garcia, A., and Shahrampour, S. On distributed non-convex optimization: Projected subgradient method for weakly convex problems in networks. *IEEE Transactions on Automatic Control*, 2021b.

Dalcín, L., Paz, R., and Storti, M. Mpi for python. *Journal of Parallel and Distributed Computing*, 65(9):1108–1115, 2005.

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

Duchi, J. C., Agarwal, A., and Wainwright, M. J. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic control*, 57(3):592–606, 2011.

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

Fan, J., Wang, D., Wang, K., and Zhu, Z. Distributed estimation of principal eigenspaces. *Annals of statistics*, 47(6):3009, 2019.

Gang, A. and Bajwa, W. U. A linearly convergent algorithm for distributed principal component analysis. *arXiv preprint arXiv:2101.01300*, 2021.

Golub, G. H. and Zha, H. The canonical correlations of matrix pairs and their numerical computation. In *Linear algebra for signal processing*, pp. 27–49. Springer, 1995.

Hong, M., Zeng, S., Zhang, J., and Sun, H. On the divergence of decentralized non-convex optimization. *arXiv preprint arXiv:2006.11662*, 2020.

Huang, L., Liu, X., Lang, B., Yu, A., Wang, Y., and Li, B. Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Huang, L.-K. and Pan, S. Communication-efficient distributed pca by riemannian optimization. In *International Conference on Machine Learning*, pp. 4465–4474. PMLR, 2020.

Kempe, D. and McSherry, F. A decentralized algorithm for spectral analysis. *Journal of Computer and System Sciences*, 74(1):70–83, 2008.

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

Li, X., Chen, S., Deng, Z., Qu, Q., Zhu, Z., and So, A. M. C. Nonsmooth optimization over stiefel manifold: Riemannian subgradient methods. *arXiv preprint arXiv:1911.05047*, 2019.

Lian, X., Zhang, C., Zhang, H., Hsieh, C.-J., Zhang, W., and Liu, J. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pp. 5330–5340, 2017.

Liu, H., So, A. M.-C., and Wu, W. Quadratic optimization with orthogonality constraint: Explicit Łojasiewicz exponent and linear convergence of retraction-based line-search and stochastic variance-reduced gradient methods. *Mathematical Programming Series A*, 178(1-2):215–262, 2019.

Liu, S., Qiu, Z., and Xie, L. Convergence rate analysis of distributed optimization with projected subgradient algorithm. *Automatica*, 83:162–169, 2017.

Markdahl, J., Thunberg, J., and Goncalves, J. High-dimensional kuramoto models on stiefel manifolds synchronize complex networks almost globally. *Automatica*, 113:108736, 2020.

Mishra, B., Kasai, Hiroyuki, P. J., and Saroop, A. A riemannian gossip approach to subspace learning on grassmann manifold. *Machine Learning*, 108:1783–1803, 2019.

Mota, J. F., Xavier, J. M., Aguiar, P. M., and Püschel, M. D-admm: A communication-efficient distributed algorithm for separable optimization. *IEEE Transactions on Signal Processing*, 61(10):2718–2723, 2013.

Nedic, A., Ozdaglar, A., and Parrilo, P. A. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4):922–938, 2010.

Nedic, A., Olshevsky, A., and Shi, W. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4): 2597–2633, 2017.

Nedić, A., Olshevsky, A., and Rabbat, M. G. Network topology and communication-computation tradeoffs in decentralized optimization. *Proceedings of the IEEE*, 106 (5):953–976, 2018.

Nesterov, Y. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Penna, F. and Stańczak, S. Decentralized eigenvalue algorithms for distributed signal detection in wireless networks. *IEEE Transactions on Signal Processing*, 63(2): 427–440, 2014.

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

Raja, H. and Bajwa, W. U. Cloud k-svd: A collaborative dictionary learning algorithm for big, distributed data. *IEEE Transactions on Signal Processing*, 64(1):173–188, 2015.

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

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

Shi, W., Ling, Q., Yuan, K., Wu, G., and Yin, W. On the linear convergence of the admm in decentralized consensus optimization. *IEEE Transactions on Signal Processing*, 62(7):1750–1761, 2014.

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

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

Tron, R., Afsari, B., and Vidal, R. Riemannian consensus for manifolds with bounded curvature. *IEEE Transactions on Automatic Control*, 58(4):921–934, 2012.

Tsitsiklis, J., Bertsekas, D., and Athans, M. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE transactions on automatic control*, 31(9):803–812, 1986.

Vorontsov, E., Trabelsi, C., Kadoury, S., and Pal, C. On orthogonality and learning recurrent networks with long term dependencies. In *International Conference on Machine Learning*, pp. 3570–3578. PMLR, 2017.

Wang, L., Liu, X., and Zhang, Y. A distributed and secure algorithm for computing dominant svd based on projection splitting. *arXiv preprint arXiv:2012.03461*, 2020.

Xin, R., Khan, U. A., and Kar, S. A near-optimal stochastic gradient method for decentralized non-convex finite-sum optimization. *arXiv preprint arXiv:2008.07428*, 2020.

Xu, J., Zhu, S., Soh, Y. C., and Xie, L. Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 2055–2060. IEEE, 2015.

Yang, W. H., Zhang, L.-H., and Song, R. Optimality conditions for the nonlinear programming problems on Riemannian manifolds. *Pacific J. Optimization*, 10(2):415–434, 2014.

Ye, H. and Zhang, T. Deepca: Decentralized exact pca with linear convergence rate. *arXiv preprint arXiv:2102.03990*, 2021.

Yuan, K., Ling, Q., and Yin, W. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, 26(3):1835–1854, 2016.

Yuan, K., Ying, B., Zhao, X., and Sayed, A. H. Exact diffusion for distributed optimization and learning—part i: Algorithm development. *IEEE Transactions on Signal Processing*, 67(3):708–723, 2018.

Zhang, H. and Sra, S. First-order methods for geodesically convex optimization. In *Conference on Learning Theory*, pp. 1617–1638, 2016.