Chapter 19
Recent Advances in Stochastic Riemannian Optimization

Reshad Hosseini and Suvrit Sra

Contents

Abstract Stochastic and finite-sum optimization problems are central to machine learning. Numerous specializations of these problems involve nonlinear constraints where the parameters of interest lie on a manifold. Consequently, stochastic manifold optimization algorithms have recently witnessed rapid growth, also in part due to their computational performance. This chapter outlines numerous stochastic optimization algorithms on manifolds, ranging from the basic stochastic gradient method to more advanced variance reduced stochastic methods. In particular, we present a unified summary of convergence results. Finally, we also provide several basic examples of these methods to machine learning problems, including learning parameters of Gaussians mixtures, principal component analysis, and Wasserstein barycenters.

R. Hosseini (✉)
School of ECE, College of Engineering, University of Tehran, Tehran, Iran

School of Computer Science, Institute of Research in Fundamental Sciences (IPM), Tehran, Iran
e-mail: reshad.hosseini@ut.ac.ir

S. Sra
Massachusetts Institute of Technology, Cambridge, MA, USA
e-mail: suvrit@mit.edu

P. Grohs et al. (eds.), *Handbook of Variational Methods for Nonlinear Geometric Data*, https://doi.org/10.1007/978-3-030-31351-7_19

19.1 Introduction

In this chapter we outline first-order optimization algorithms used for minimizing the expected loss (risk) and its special case, finite-sum optimization (empirical risk). In particular, we focus on the setting where the parameters to be optimized lie on a Riemannian manifold. This setting appears in a variety of problems in machine learning and statistics, including principal components analysis [33], low-rank matrix completion [9, 39], fitting statistical models like Gaussian mixture models [17, 18, 38], Karcher mean computation [22, 33], Wasserstein barycenters [40], dictionary learning [12], low rank multivariate regression [27], subspace learning [28], and structure prediction [34]; see also the textbook [1].

Typical Riemannian manifolds used in applications can be expressed by a set of constraints on Euclidean manifolds. Therefore, one can view a Riemannian optimization problem as a nonlinearly constrained one, for which one could use classical approaches. For instance, if the manifold constitutes a convex set in Euclidean space, one can use gradient projection like methods,[1] or other nonlinear optimization methods [6]. These methods could suffer from high computational costs, or as a more fundamental weakness, they may fail to satisfy the constraints exactly at each iteration of the associated algorithm. Another problem is that the Euclidean gradient does not take into account the geometry of the problem, and even if the projection can be done and the constraints can be satisfied at each iteration, the numerical conditioning may be much worse than a method that respects geometry [1, 42].

Riemannian optimization has shown great success in solving many practical problems because it respects the geometry of the constraint set. The definition of the inner product in Riemannian geometry makes the direction of the gradient to be more meaningful than Euclidean gradients because it considers the geometry imposed by constraints on the parameters of optimization. By defining suitable retractions (geodesic like curves on manifolds), the constraint is always satisfied. Sometimes the inner product is defined to also take into account the curvature information of the cost function. The natural gradient is an important example of the Riemannian gradient shown to be successful for solving many statistical fitting problems [2]. The natural gradient was designed for fitting statistical models and it is a Riemannian gradient on a manifold where the metric is defined by the Fisher information matrix.

[1] Some care must be applied here, because we are dealing with open sets, and thus projection is not well-defined.

Additional Background and Summary

Another key feature of Riemannian optimization is the generalization of the widely important concept of convexity to geodesic convexity. We will see later in this chapter that geodesic convexity help us derive convergence results for accelerated gradient descent methods akin to their famous Euclidean counterpart: Nesterov's accelerated gradient method. Similar to the Euclidean case, there are works that develop results for recognizing geodesic convexity of functions for some special manifolds [37]. Reformulating problems keeping an eye on geodesic convexity also yields powerful optimization algorithms for some practical problems [18].

After summarizing key concepts of Riemannian manifolds, we first sketch the Riemannian analogue of the widely used (Euclidean) stochastic gradient descent method. Though some forms of stochastic gradient descent (SGD) such as natural gradient were developed decades ago, the version of SGD studied here and its analysis has a relatively short history; Bonnabel [8] was the first to give a unifying framework for analyzing Riemannian SGD and provided an asymptotic analysis on its almost sure convergence. We recall his results after explaining SGD on manifolds. We then note how convergence results of [15] for Euclidean non-convex SGD generalize to the Riemannian case under similar conditions [18]. Among recent progress on SGD, a notable direction is that of faster optimization by performing variance reduction of stochastic gradients. We will later outline recent results of accelerating SGD on manifolds and give convergence analysis for geodesically non-convex and convex cases. Finally, we close by summarizing some applications drawn from machine learning that benefit from the stochastic Riemannian algorithms studied herein.

Apart from the algorithms given in this chapter, there exist several other methods that generalize well from the Euclidean to the Riemannian setting. For example in [4] the SAGA algorithm [13] is generalized to Riemannian manifolds along with convergence theory assuming geodesic convexity. In [23] a Riemannian stochastic quasi-Newton method is studied; in [21] an inexact Riemannian trust-region method is developed and applied to finite-sum problems. Adaptive stochastic gradient methods such as ADAM and RMSProp have also been generalized [5, 24, 25]. It was observed however that ADAM works inferior to plain SGD for fitting Gaussian mixture models [16], where momentum and Nesterov SGD offered the best variants that improve on the performance of plain SGD.

The convergence results presented in this chapter are for general Riemannian manifolds and hold for a fairly general class of cost functions. For specific manifolds and functions, one can obtain better convergence results for the algorithms. For example for the case of quadratic optimization with orthogonality constraint, the authors in [26] proved convergence results. The authors in [41] proved convergence for a block eigensolver.

19.2 Key Definitions

We omit standard definitions such as Riemannian manifolds, geodesics, etc.; and defer to a standard textbook such as [20]. Readers familiar with concepts from Riemannian geometry can skip this section and directly move onto Sect. 19.3; however, a quick glance will be useful for getting familiar with our notation.

A retraction is a smooth mapping Ret from the tangent bundle $T\mathcal{M}$ to the manifold $\mathcal{M}$. The restriction of a retraction to $T_x\mathcal{M}$, $\mathrm{Ret}_x : T_x\mathcal{M} \to \mathcal{M}$, is a smooth mapping that satisfies the following:

1. $\mathrm{Ret}_x(0) = x$, where 0 denotes the zero element of $T_x\mathcal{M}$.
2. $D\,\mathrm{Ret}_x(0) = \mathrm{id}_{T_x\mathcal{M}}$, where $D\,\mathrm{Ret}_x$ denotes the derivative of Ret_x and $\mathrm{id}_{T_x\mathcal{M}}$ denotes the identity mapping on $T_x\mathcal{M}$.

One possible candidate for retraction on Riemannian manifolds is the exponential map. The exponential map $\mathrm{Exp}_x : T_x\mathcal{M} \to \mathcal{M}$ is defined as $\mathrm{Exp}_x\, v = \gamma(1)$, where γ is the geodesic satisfying the conditions $\gamma(0) = x$ and $\dot{\gamma}(0) = v$.

A vector transport $\mathcal{T} : \mathcal{M}\times\mathcal{M}\times T\mathcal{M} \to T\mathcal{M}$, $(x, y, \xi) \mapsto \mathcal{T}_{x,y}(\xi)$ is a mapping that satisfies the following properties:

1. There exists an associated retraction Ret and a tangent vector v satisfying $\mathcal{T}_{x,y}(\xi) \in T_{\mathrm{Ret}_x(\xi)}$, for all $\xi \in T_x\mathcal{M}$.
2. $\mathcal{T}_{x,x}\xi = \xi$, for all $\xi \in T_x\mathcal{M}$.
3. The mapping $\mathcal{T}_{x,y}(\cdot)$ is linear.

We use $\mathcal{T}^{\mathrm{Ret}_x}_{x,y}$ to denote the vector transport constructed by the differential of the retraction, i.e., $\mathcal{T}^{\mathrm{Ret}_x}_{x,y}(\xi) = D\,\mathrm{Ret}_x(\eta)[\xi]$, wherein $\mathrm{Ret}_x(\eta) = y$ (in the case of multiple η, we make it clear by writing the value of η), while $\mathcal{P}^{\mathrm{Ret}_x}_{x,y}$ denotes the parallel transport along the retraction curve (again, if there are multiple curves where $\mathrm{Ret}_x(\eta) = y$, we make it clear from context which curve is meant).

The gradient on a Riemannian manifold is defined as the vector $\nabla f(x)$ in tangent space such that

$$Df(x)\xi = \langle \nabla f(x),\, \xi \rangle, \quad \text{for } \xi \in T_x\mathcal{M},$$

where $\langle \cdot,\, \cdot \rangle$ is the inner product in the tangent space $T_x\mathcal{M}$. $Df(x)\xi$ is the directional derivative of f along ξ. Let $\gamma : [-1, 1] \to \mathcal{M}$ be a differentiable curve with $\gamma(0) = x$ and $\dot{\gamma}(0) = \xi$ (for example $\gamma(t) = \mathrm{Exp}(t\xi)$), then the directional derivative can be defined by

$$Df(x)\xi = \frac{d}{d\tau} f(\gamma(\tau))\bigg|_{\tau=0}.$$

Differentials at each point on the manifold forms the cotangent space. The cotangent space on the smooth manifold $\mathcal{M}$ at point x is defined as the dual space of the tangent space. Elements of the cotangent space are linear functionals on the tangent space.

The Hessian of a function is a symmetric bilinear form $D^2 f(x) : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}, (\xi, \eta) \to \langle \nabla_\eta \nabla f(x), \xi \rangle$, where ∇_η is the covariant derivative with respect to η [1]. The Hessian as a operator $\nabla^2 f(x) : T_x\mathcal{M} \to T_x\mathcal{M}$ is a linear operator that maps ν in $T_x\mathcal{M}$ onto the Riesz representation $D^2 f(x)(\nu, .)$. Alternatively, the operator Hessian can be defined by

$$\frac{d}{d\tau}\langle \nabla f(\gamma(\tau)),\ \nabla f(\gamma(\tau))\rangle\Big|_{\tau=0} = 2\langle \nabla f(x),\ (\nabla^2 f)\xi\rangle,$$

where $\gamma : [-1, 1] \to \mathcal{M}$ is a differentiable curve with $\gamma(0) = x$ and $\dot{\gamma}(0) = \xi$.In the following, we give some conditions and definitions needed for the complexity analysis of the algorithms in this book chapter.

Definition 19.1 (ρ-Totally Retractive Neighborhood) A neighborhood Ω of a point x is called *ρ-totally retractive* if for all $y \in \Omega$, $\Omega \subset \mathbb{B}(0_y, \rho)$ and $\mathrm{Ret}_y(\cdot)$ is a diffeomorphism on $\mathbb{B}(0_y, \rho)$.

All optimization algorithms given in this book chapter start from an initial point and the point is updated based on a retraction along a direction with a certain step size. The following condition guarantees that all points along retraction in all interactions stay in a set.

Definition 19.2 (Iterations Stay Continuously in $\mathcal{X}$) The iterate $x_{k+1} = \mathrm{Ret}_{x_k}(\alpha_k \xi_k)$ is said to stay continuously in $\mathcal{X}$ if $\mathrm{Ret}_{x_k}(t\xi_k) \in \mathcal{X}$ for all $t \in [0, \alpha_k]$.

Most of the optimization algorithms explained in this chapter need a vector transport. The convergence analysis for many of them is available for the specific case of parallel transport. Some works that go beyond parallel transport still need some extra conditions on the vector transport as explained below. These conditions hold *a forteriori* for parallel transport.

Definition 19.3 (Isometric Vector Transport) The vector transport $\mathcal{T}$ is said to be *isometric* on $\mathcal{M}$ if for any $x, y \in \mathcal{M}$ and $\eta, \xi \in T_x\mathcal{M}$, $\langle \mathcal{T}_{x,y}(\eta),\ \mathcal{T}_{x,y}(\xi)\rangle = \langle \eta,\ \xi\rangle$.

Definition 19.4 (θ-Bounded Vector Transport) The vector transport $\mathcal{T}$ with its associated retraction Ret is said to be *θ-bounded* on $\mathcal{M}$ if for any $x, y = \mathrm{Ret}_x(\xi) \in \mathcal{M}$ and $\xi \in T_x\mathcal{M}$,

$$\|\mathcal{T}_{x,y}\eta - \mathcal{P}^{\mathrm{Ret}_x}_{x,y}\eta\| \le \theta\|\xi\|\|\eta\|, \tag{19.1}$$

where $\mathcal{P}$ is the parallel transport along this associated retraction curve.

Definition 19.5 (θ-Bounded Inverse Vector Transport) The inverse vector transport with its associated retraction Ret is said to be *θ-bounded* on $\mathcal{M}$ if for any $x, y = \mathrm{Ret}_x(\xi) \in \mathcal{M}$ and $\xi \in T_x\mathcal{M}$,

$$\|\left(\mathcal{T}_{x,y}\right)^{-1}\chi - \left(\mathcal{P}^{\mathrm{Ret}_x}_{x,y}\right)^{-1}\chi\| \le \theta\|\chi\|\|\xi\|,$$

where $\mathcal{P}$ is the parallel transport along this associated retraction curve.

The following proposition helps in checking if a vector transport satisfies some of the conditions expressed above.

Proposition 19.6 (Lemma 3.5 in Huang et al. [19]) *Assume that there exists a constant $c_0 > 0$ such that $\mathcal{T}$ satisfies $\|\mathcal{T}_{x,y} - \mathcal{T}_{x,y}^{\mathrm{Ret}_x}\| \leq c_0\|\xi\|$, $\|(\mathcal{T}_{x,y})^{-1} - \left(\mathcal{T}_{x,y}^{\mathrm{Ret}_x}\right)^{-1}\| \leq c_0\|\xi\|$, for any $x, y \in \mathcal{M}$ and the retraction $y = \mathrm{Ret}_x(\xi)$. Then, the vector transport and its inverse are θ-bounded on $\mathcal{M}$, for a constant $\theta > 0$.*

We note that if the vector transport is C^0, then the condition of this proposition holds.

For the convergence analysis of the algorithms in this chapter, the cost function needs to satisfy some of the properties given below.

Definition 19.7 (G-Bounded Gradient) A function $f : \mathcal{X} \to \mathbb{R}$ is said to have a *G-bounded gradient* in $\mathcal{X}$ if $\|\nabla f(x)\| \leq G$, for all $x \in \mathcal{X}$.

Definition 19.8 (H-Bounded Hessian) A function $f : \mathcal{X} \to \mathbb{R}$ is said to have an *H-bounded Hessian* in $\mathcal{X}$ if $\|\nabla^2 f(x)\| \leq H$, for all $x \in \mathcal{X}$.

Definition 19.9 (Retraction L-Smooth) A function $f : \mathcal{X} \to \mathbb{R}$ is said to be *retraction L-smooth* if for any x, $y = \mathrm{Ret}_x(\xi)$ in $\mathcal{X}$, we have

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

If the retraction is the exponential map, then the function is called **geodesically L-smooth**.

Definition 19.10 (Retraction L-Upper-Hessian Bounded) A function $f : \mathcal{X} \to \mathbb{R}$ is said to be upper-Hessian bounded in a subset $\mathcal{U} \subset \mathcal{X}$ if $\mathrm{Ret}_x(t\xi)$ stays in $\mathcal{X}$ for all x, $y = \mathrm{Ret}_x(\xi)$ in $\mathcal{U}$ and $t \in [0, 1]$, and there exists a constant $L > 0$ such that $\frac{d^2 f(\mathrm{Ret}_x(t\xi))}{dt^2} \leq L$.

Definition 19.11 (Retraction μ-Lower-Hessian Bounded) A function $f : \mathcal{X} \to \mathbb{R}$ is said to be lower-Hessian bounded in a subset $\mathcal{U} \subset \mathcal{X}$ if $\mathrm{Ret}_x(t\xi)$ stays in $\mathcal{X}$ for all x, $y = \mathrm{Ret}_x(\xi)$ in $\mathcal{U}$ and $t \in [0, 1]$, and there exists a constant $\mu > 0$ such that $\frac{d^2 f(\mathrm{Ret}_x(t\xi))}{dt^2} \geq \mu$.

Definition 19.12 (Retraction L_l-Lipschitz) A function $f : \mathcal{X} \to \mathbb{R}$ is said to be retraction L_l-Lipschitz in $\mathcal{X}$, if there exists $L_l > 0$ such that for all $x, y \in \mathcal{X}$,

$$\|\mathcal{P}_{x,y}^{\mathrm{Ret}_x} \nabla f(x) - \nabla f(y)\| \leq L_l \|\xi\|, \tag{19.2}$$

where $\mathcal{P}$ is the parallel transport along this associated retraction curve $y = \mathrm{Ret}_x(\xi)$.

If the retraction is the exponential map, then this condition is called **geodesically L_l-Lipschitz**. A function that is geodesically L_l-Lipschitz is also geodesically L-smooth with $L = L_l$ [44].

In the following, we give two propositions and a theorem for checking if a function satisfies some of the conditions explained before. The following proposition is based on a Lemma in [22].

Proposition 19.13 *Suppose that the function $f : \mathcal{X} \to \mathbb{R}$ is retraction L-upper-Hessian bounded in $\mathcal{U} \subset \mathcal{X}$. Then, the function is also retraction L-smooth in $\mathcal{U}$.*

Proposition 19.14 (Lemma 3.8 in Kasai et al. [22]) *Let* Ret *be a retraction on $\mathcal{M}$ and the vector transport associated with the retraction and its inverse be θ-bounded. Assume a function is twice continuously differentiable with H-bounded Hessian. Then the function is retraction L_l-Lipschitz with $L_l = H(1 + \Xi\theta)$ with Ξ being an upper bound for $\|\xi\|$ in (19.2).*

For showing retraction L-smoothness, we can use the following theorem.

Theorem 19.15 (Lemma 2.7 in Boumal et al. [11]) *Let $\mathcal{M}$ be a compact Riemannian submanifold of a Euclidean space. Let* Ret *be a retraction on $\mathcal{M}$. If a function has a Euclidean Lipschitz continuous gradient in the convex hull of $\mathcal{M}$, then the function is retraction L-smooth for some constant L for any retraction.*

The aforementioned conditions of function are quite general. In the following we give some conditions on functions that help to develop stronger convergence results.

Definition 19.16 (g-Convex) A set $\mathcal{X}$ is geodesically convex (g-convex) if for any $x, y \in \mathcal{X}$, there is a geodesic γ with $\gamma(0) = x$, $\gamma(1) = y$ and $\gamma(t) \in \mathcal{X}$ for $t \in [0, 1]$. A function $f : \mathcal{X} \to R$ is called geodesically convex in this set if

$$f(\gamma(t)) \leq (1 - t) f(x) + t f(y).$$

Definition 19.17 (μ-Strongly g-Convex) A function $f : \mathcal{X} \to R$ is called geodesically μ-strongly convex if for any $x, y = \mathrm{Exp}_x(\xi) \in \mathcal{X}$ and g_x subgradient of f at x (gradient if f is smooth), it holds

$$f(y) \geq f(x) + \langle g_x, \xi \rangle + \frac{\mu}{2}\|\xi\|^2.$$

Definition 19.18 (τ-Gradient Dominated) A function $f : \mathcal{X} \to R$ is called τ-gradient dominated if x^* is a global minimizer of f and for every $x \in \mathcal{X}$ we have

$$f(x) - f(x^*) \leq \tau \|\nabla f(x)\|^2. \tag{19.3}$$

The following proposition shows that strongly convex functions are also gradient dominated. Therefore, the convergence analysis developed for gradient dominated functions also holds for strongly convex functions [44].

Algorithm 1 Riemannian SGD

Given: Smooth manifold $\mathcal{M}$ with retraction Ret; initial value x_0; a differentiable cost function f; number of iterations T.
for $t = 0, 1, \ldots, T-1$ **do**
  Obtain the direction $\xi_t = \nabla f_i(x_t)$, where $\nabla f_i(x_t)$ is the noisy version of the cost gradient
  Use a step-size rule to choose the step-size α_t
  Calculate $x_{t+1} = \mathrm{Ret}_{x_t}(-\alpha_t \xi_t)$
end for
return x_T

Proposition 19.19 *τ-gradient domination is implied by $\frac{1}{2\tau}$-strong convexity as in Euclidean case.*

19.3 Stochastic Gradient Descent on Manifolds

In the most general form, consider the following constrained optimization problem:

$$\min_{x \in \mathcal{M}} f(x). \tag{19.4}$$

We assume $\mathcal{M}$ is a Riemannian manifold and that at each step of SGD we obtain a noisy version of the Riemannian gradient. Riemannian SGD uses the following simple update rule:

$$x_{t+1} = \mathrm{Ret}_{x_t}\big(-\eta_t \nabla f_{i_t}(x_t)\big), \tag{19.5}$$

where ∇f_{i_t} is a noisy version of the Riemannian gradient at time step t and the noise terms at different time steps are assumed to be independent. Note that there is stochasticity in each update. Therefore, the value x_t can be seen as a sample from a distribution depending on the gradient noise until time step t. A sketch of Riemannian SGD is given in Algorithm 1. For providing convergence results for all algorithms, it is assumed the stochastic gradients in all iterations are unbiased, i.e.,

$$\mathbb{E}[\nabla f_{i_t}(x_t) - \nabla f(x_t)] = 0.$$

This unbiasedness condition is assumed in all theorems and we do not state it explicitly in the statements of the theorems.

The cost function used in many practical machine learning problems which is solved by SGD can be defined by

$$f(x) = \mathbb{E}[f(z; x)] = \int f(z; x) dP(z), \tag{19.6}$$

where x denotes the parameters, dP is a probability measure and $f(z; x)$ is the risk function. For this cost function $f_{i_t} = f(z_{i_t}, x_t)$ is the risk evaluated at the current

sample z_{i_t} from the probability law dP. Apparently, the stochastic gradients for this cost function satisfy the condition that stochastic gradients are unbiased. A special case of the above-mentioned cost function is the following finite-sum problem:

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(z_i, x). \tag{19.7}$$

If we assume $z_1, \ldots, z_n$ to be our data, then the empirical distribution over the data $P(Z = z_i) = \frac{1}{n}$ gives rise to the above noted cost function. Therefore, the theoretical analysis for SGD works both for online-algorithms and also finite-sum optimization problems. To further elucidate this consider the following example.

Example: Maximum Likelihood Parameter Estimation
Consider we want to estimate the parameters of a model distribution given by the density $q(z; x)$, where x denotes the parameters. In the online learning framework, we observe a sample z_t from the underlying distribution $p(z)$ at each time step. Observing this new sample, the parameter set is updated by a rule. The update rule should be designed such that in the limit of observing enough samples, the parameters converge to the optimal parameters. The optimal parameters are commonly defined as the parameters that minimize the Kullback-Leibler divergence between the estimated and the true distributions. The following cost function minimizes this divergence:

$$f(x) = \mathbb{E}[-\log q(z; x)] = -\int \log q(z; x) p(z) dz,$$

where q is the density of model distribution and p is the true density. Apparently, this cost function is in the form of cost function defined in (19.6). One of the common update rules for online learning is SGD. For Riemannian SGD, we have $\nabla f(z_t, x_t) = \nabla f_{i_t}(x_t)$ and we use the update rule as in (19.5).

In the finite sample case, consider $z_1, \ldots, z_n$ to be i.i.d. samples from the underlying density $q(z; x)$. A common approach for estimating the parameters is the maximum-likelihood estimate where we are minimizing the following cost function:

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} -\log q(z_i; x).$$

The cost function is a finite-sum cost that can be minimized using SGD. Therefore, it is important to know the conditions under which SGD guarantees convergence.

The following theorem gives the convergence to stationary points of the cost function.

Theorem 19.20 (Theorem 2 in Bonnabel [8]) *Consider the optimization problem in (19.4), where the cost function is the expected risk (19.6). Assume*

- *The manifold $\mathcal{M}$ is a connected Riemannian manifold with injectivity radius uniformly bounded from below by $I > 0$.*
- *The steps stay within a compact set.*
- *The gradients of the f_is are G-bounded.*

Let the step-sizes in Algorithm 1 satisfy the following standard condition

$$\sum \alpha_t^2 < \infty \text{ and } \sum \alpha_t = \infty, \tag{19.8}$$

Then $f(x_t)$ converges a.s. and $\nabla f(x_t) \to 0$ a.s.

Staying within a compact set of the previous theorem is a strong requirement. Under milder conditions, [18] were able to prove the rate of convergence.

Theorem 19.21 (Theorem 5 in Hosseini and Sra [18]) *Assume that the following conditions hold*

- *The functions f_i are retraction L-smooth.*
- *The expected square norm of the gradients of the f_is are G^2-bounded.*

Then for the following constant step-size in Algorithm 1

$$\alpha_t = \frac{c}{\sqrt{T}},$$

we have

$$\min_{0 \le t \le T-1} \mathbb{E}[\|\nabla f(x_t)\|^2] \le \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla f(x_t)\|^2] \le \frac{1}{\sqrt{T}} \left(\frac{f(x_0) - f(x^*)}{c} + \frac{Lc}{2} G^2 \right). \tag{19.9}$$

where $f(x_0)$ is the function value at the initial point and $f(x^*)$ is the minimum function value.

The following theorem shows that it is possible to get a convergence rate without needing bounded gradients with a randomized rule. For this theorem, the stochastic gradients needs to have σ-bounded variance, i.e.,

$$\mathbb{E}[\|\nabla f_{i_t}(x_t) - \nabla f(x_t)\|^2] \le \sigma^2, \qquad 0 \le \sigma < \infty.$$

The conditions and the resulting rate are similar to that of Euclidean case [15], and no further assumptions are necessary.

Theorem 19.22 (Theorem 4 in Hosseini and Sra [18]) *Assume that the following conditions hold.*

- *The functions f_i are retraction L-smooth.*
- *The functions f_i have σ-bounded variance.*

Assume a slightly modified version of SGD that outputs a point x_a by randomly picking one of the iterates, say x_t, with probability $p_t := (2\alpha_t - L\alpha_t^2)/Z_T$, where $Z_T = \sum_{t=1}^{T}(2\alpha_t - L\alpha_t^2)$. Furthermore, choose $\alpha_t = \min\{L^{-1}, c\sigma^{-1}T^{-1/2}\}$ in Algorithm 1 for a suitable constant c. Then, we obtain the following bound on $\mathbb{E}[\|\nabla f(x_a)\|^2]$, which measures the expected gap to stationarity:

$$\mathbb{E}[\|\nabla f(x_a)\|^2] \leq \frac{2L\Delta_1}{T} + \left(c + c^{-1}\Delta_1\right)\frac{L\sigma}{\sqrt{T}} = O\left(\frac{1}{T}\right) + O\left(\frac{1}{\sqrt{T}}\right). \tag{19.10}$$

For Hadamard manifolds (complete, simply-connected Riemannian manifolds with nonpositive sectional curvature), one can prove a.s. convergence under milder conditions. Hadamard manifolds have strong properties, for example the exponential map at any point is globally invertible. Concerning convergence for Hadamard manifolds there is the following result requiring milder assumptions.

Theorem 19.23 (Theorem 3 in Bonnabel [8]) *Consider the optimization problem in (19.4), where the cost function is the expected risk (19.6). Assume*

- *The exponential map is used for the retraction.*
- *The manifold $\mathcal{M}$ is a Hadamard manifold.*
- *There is a lower bound on the sectional curvature denoted by $\kappa < 0$.*
- *There is a point $y \in \mathcal{M}$ such that the negative gradient points towards y when $d(x, y)$ becomes larger than $s > 0$, i.e.,*

$$\inf_{d(x,y)>s} \langle \mathrm{Exp}_x^{-1}(y),\ \nabla f(x)\rangle < 0$$

- *There is a continuous function $g : \mathcal{M} \to \mathbb{R}$ that satisfies*

$$g(x) \geq \max\Big\{1, \mathbb{E}\big[\|\nabla f(z; x)\|^2(1 + \sqrt{\kappa}(d(x, y) + \|\nabla f(z; x)\|))\big],$$
$$\mathbb{E}\big[\big(2\|\nabla f(z; x)\|d(x, y) + \|\nabla f(z; x)\|^2\big)^2\big]\Big\}$$

Then for the step size rule $\alpha_t = -\frac{\beta_t}{g(x_t)}$ in Algorithm 1, wherein β_t satisfying (19.8), $f(x_t)$ converges a.s. and $\nabla f(x_t) \to 0$ a.s.

Algorithm 2 Riemannian SVRG

1: **Given:** Smooth manifold $\mathcal{M}$ with retraction Ret and vector transport $\mathcal{T}$; initial value x_0; a finite-sum cost function f; update frequency m; number of epochs S and K.
2: **for** $k = 0, \ldots,$ K-1 **do**
3: $\quad \tilde{x}^0 = x_k$
4: $\quad$ **for** $s = 0, \ldots,$ S-1 **do**
5: $\quad\quad$ Calculate the full Riemannian gradient $\nabla f(\tilde{x}^s)$
6: $\quad\quad$ Store $x_0^{s+1} = \tilde{x}^s$
7: $\quad\quad$ **for** $t = 0, \ldots, m-1$ **do**
8: $\quad\quad\quad$ Choose $i_t \in \{1, \ldots, n\}$ uniformly at random
9: $\quad\quad\quad$ Calculate $\xi_t^{s+1} = \nabla f_{i_t}(x_t^{s+1}) - \mathcal{T}_{\tilde{x}^s, x_t^{s+1}}(\nabla f_{i_t}(\tilde{x}^s) - \nabla f(\tilde{x}^s))$
10: $\quad\quad\quad$ Use a step-size rule to choose the step-size α_t^{s+1}
11: $\quad\quad\quad$ Calculate $x_{t+1}^{s+1} = \mathrm{Ret}_{x_t^{s+1}}(-\alpha_t^{s+1}\xi_t^{s+1})$
12: $\quad\quad$ **end for**
13: $\quad\quad$ Option I-a: Set $\tilde{x}^{s+1} = x_m^{s+1}$
14: $\quad\quad$ Option II-a: Set $\tilde{x}^{s+1} = x_t^{s+1}$ for randomly chosen $t \in \{0, \ldots, m-1\}$
15: $\quad$ **end for**
16: $\quad$ Option I-b: Set $x_{k+1} = \tilde{x}^S$
17: $\quad$ Option II-b: Set $x_{k+1} = \tilde{x}_t^s$ for randomly chosen $s \in \{0, \ldots, S-1\}$ and $t \in \{0, \ldots, m-1\}$
18: **end for**
19: **return** x_K

19.4 Accelerating Stochastic Gradient Descent

Mainly for finite-sum problems but also for expected risk (19.6) problems, accelerated algorithms have been developed with faster convergence rates than plain SGD. In this section, we review several popular accelerated algorithms that are based on *variance reduction* ideas. Stochastic variance reduced gradient (SVRG) is a popular variance reduction technique that has a superior convergence than plain SGD. A Riemannian version of SVRG (R-SVRG) was proposed in [33] and generalized to use retractions and vector transports in [36]. Variance reduction can be seen in the line 9 of Algorithm 2, where the average gradient is used for adjust the current gradient. Consider a stochastic gradient that has high variance; then subtracting the difference between this gradient and the average gradient at a reference point from this gradient in the current point reduces the effect of high variance. Because we are on a Riemannian manifold, gradients live in different tangent spaces, and a vector transport is needed to make the subtraction meaningful as can be seen in the line 9 of Algorithm 2.

The authors in [33] were able to prove that R-SVRG has the same convergence as in the Euclidean case [32]. Though, the statement on the convergence rate needs additional assumptions and a bound depending on the sectional curvature.

Theorem 19.24 (Theorem 2 in Zhang et al. [33]) *Consider the optimization problem in* (19.4)*, where the cost function is the finite sum* (19.7)*. Consider, we run Riemannian SVRG to solve this problem with* $K = 1$*, Option I-a, Option II-b. Assume*

- *The exponential map is used for the retraction and the parallel transport is used for the vector transport.*
- *The iterations stay in a compact subset $\mathcal{X}$, and the diameter of $\mathcal{X}$ is bounded by D, that is* $\max_{x,y\in\mathcal{X}} d(x, y) \leq D$.
- *The exponential map on $\mathcal{X}$ is invertible.*
- *The sectional curvature is upper-bounded.*
- *There is a lower bound on the sectional curvature denoted by κ.*
- *The functions f_i are geodesically L-smooth.*
- *The function f attains its minimum at $x^* \in \mathcal{X}$.*

Define ζ to be a constant that captures the impact of the manifold curvature.

$$\zeta = \begin{cases} \frac{\sqrt{|\kappa|}D}{\tanh\left(\sqrt{|\kappa|}D\right)}, & \kappa < 0. \\ 1, & \kappa \geq 0. \end{cases} \tag{19.11}$$

Then there exist universal constants $\mu_0 \in (0, 1)$, $\nu > 0$ such that if we set $\alpha_t = \frac{\mu_0}{Ln^{\alpha_1}\zeta^{\alpha_2}}$, $\alpha_1 \in (0, 1]$, $\alpha_2 \in (0, 2]$ and $m = \lfloor \frac{n^{3\alpha_1}}{3\mu_0\zeta^{1-2\alpha_2}} \rfloor$ in Algorithm 2, we have

$$\mathbb{E}[\|\nabla f(x_1)\|^2] \leq \frac{Ln^{\alpha_1}\zeta^{\alpha_2}[f(x_0) - f(x^*)]}{T\nu},$$

where $T = mS$ is the number of iterations.

The abovementioned theorem was stated based on the exponential map and the parallel transport that can be expensive making SVRG impractical for some applications. In [36] the following convergence result is proved when using retractions and vector transports.

Theorem 19.25 (Theorem 4.6 in Sato et al. [36]) *Consider the optimization problem in (19.4), where the cost function is the finite sum (19.7). Consider, we run the Riemannian SVRG algorithm to solve this problem with $K = 1$, Option I-a and Option I-b. Assume*

- *The retraction is of the class C^2.*
- *The iterations stay in a compact subset $\mathcal{X}$.*
- *For each $s \geq 0$, there exists $\eta_t^{s+1} \in T_{\tilde{x}^s}\mathcal{M}$ such that $\mathrm{Ret}_{\tilde{x}^s}(\eta_t^{s+1}) = x_t^{s+1}$.*
- *There exists $I > 0$ such that, for any $x \in \mathcal{X}$, $\mathrm{Ret}_x(.)$ is defined in a ball $\mathbb{B}(0_x, I) \in T_x\mathcal{M}$, which is centered at the origin 0_x in $T_x\mathcal{M}$ with radius I.*
- *The vector transport is continuous and isometric on $\mathcal{M}$.*
- *The functions f_i are twice-differentiable.*

Assume the step-size α_t^s in Algorithm 2 is chosen by the rule (19.8). Then $f(x_t^s)$ converges a.s. and $\nabla f(x_t^s) \to 0$ a.s.

Note that existence of η_t^{s+1} is guaranteed if Ret_x has ρ-totally retractive neighborhood for all $x \in \mathcal{X}$. For the special case of the exponential map and the parallel

transport, many of the conditions of the aforementioned theorem are automatically satisfied or simplified: The parallel transport is an isometry, the exponential map is of class C^2, and the third and fourth conditions can be satisfied by having a connected manifold with the injectivity radius uniformly bounded from below by $I > 0$.

Stochastic recursive gradient (SRG) [29] is another variance reduction algorithm similar to SVRG proposed. It was recently shown that the algorithm achieves the optimal bound for the class of variance reduction methods that only assume the Lipschitz continuous gradients [30]. Recently, the Riemannian counterpart of this algorithm (R-SRG) shown in Algorithm 3 has also been developed [22]. The following theorem gives a convergence result with the minimalistic conditions needed for the proof.

Theorem 19.26 (Theorem 4.5 in Kasai et al. [22]) *Consider the optimization problem in (19.4), where the cost function is the finite sum (19.7). Consider, we run the Riemannian SRG algorithm to solve this problem with $S = 1$. Assume*

- *The iterations stay continuously in a subset $\mathcal{X}$.*
- *The vector transport is θ-bounded.*
- *The vector transport is isometric on $\mathcal{X}$.*
- *The functions f_i are retraction L-smooth.*
- *The functions f_i are retraction L_l-Lipschitz.*
- *The gradients of the f_is are G-bounded.*
- *The function f attains its minimum at $x^* \in \mathcal{X}$.*

Assume a constant step-size $\alpha \leq \frac{2}{L+\sqrt{L^2+8m(L_l^2+G^2\theta^2)}}$ in Algorithm 3. Then, we have

$$\mathbb{E}[\|\nabla f(\tilde{x})\|^2] \leq \frac{2}{\alpha(m+1)}[f(x_0) - f(x^*)].$$

A very similar idea to R-SRG was used in another algorithm called Riemannian SPIDER (R-SPIDER) [45]. The Euclidean counterpart of the R-SPIDER algorithm was shown to have near optimal complexity bound. It can be applied to both the finite-sum and the stochastic optimization problems [14]. The details of the R-SPIDER method are given in Algorithm 4. The algorithm uses retraction and vector transport while the original algorithm and proofs of [45] were for the exponential mapping and the parallel transport. For the analysis of general non-convex functions in this section, we set $T = 1$ meaning that we have a single outer-loop.

Theorem 19.27 (Theorem 1 in Zhou et al. [45]) *Consider the optimization problem in (19.4), where the cost function is the finite sum (19.7). Consider, we run the Riemannian SPIDER algorithm with option I to solve this problem. Assume*

- *The exponential map is used for the retraction and the parallel transport is used for the vector transport.*
- *The functions f_i are geodesically L-Lipschitz.*

Algorithm 3 Riemannian SRG

Given: Smooth manifold $\mathcal{M}$ with retraction Ret and vector transport $\mathcal{T}$; initial value $\tilde{x}^0$; a finite-sum cost function f; update frequency m; number of epochs S.
for $s = 0, \ldots,$ S-1 **do**
  Store $x_0 = \tilde{x}^s$
  Calculate the full Riemannian gradient $\nabla f(x_0)$
  Store $\xi_0 = \nabla f(x_0)$
  Store $x_1 = \mathrm{Ret}_{x_0}(-\alpha_0 \xi_0)$
  for $t = 1, \ldots, m-1$ **do**
    Choose $i_t \in \{1, \ldots, n\}$ uniformly at random
    Calculate $\xi_t = \nabla f_{i_t}(x_t) - \mathcal{T}_{x_{t-1}, x_t}\big(\nabla f_{i_t}(x_{t-1}) - \xi_{t-1}\big)$
    Use a step-size rule to choose the step-size α_t
    Calculate $x_{t+1} = \mathrm{Ret}_{x_t}(-\alpha_t \xi_t)$
  end for
  Set $\tilde{x}^{s+1} = x_t$ for randomly chosen $t \in \{0, \ldots, m\}$
end for
return $\tilde{x}^S$

– *The stochastic gradients have σ-bounded variance.*

Let $T = 1$, $s = \min\big(n, \frac{16\sigma^2}{\epsilon^2}\big)$, $p = n_0 s^{\frac{1}{2}}$, $\alpha_k = \min\big(\frac{\epsilon}{2Ln_0}, \frac{\|\xi_k\|}{4Ln_0}\big)$, $|\mathcal{S}_1| = s$, $|\mathcal{S}_2| = \frac{4s^{\frac{1}{2}}}{n_0}$ *and* $n_0 \in [1, 4s^{\frac{1}{2}}]$ *in Algorithm 4. Then, we achieve* $\mathbb{E}[\|\nabla f(\tilde{x}^1)\|] \leq \epsilon$ *in at most* $K = \frac{14Ln_0\Delta}{\epsilon^2}$ *iterations in expectation, where* $\Delta = f(x_0) - f(x^*)$ *with* $x^* = \arg\min_{x \in \mathcal{M}} f(x)$.

For the online case, the following theorem considers the iteration complexity of the algorithm.

Theorem 19.28 (Theorem 2 in Zhou et al. [45]) *Consider the optimization problem in (19.4), where the cost function is the expected risk (19.6). Assume the same conditions as in Theorem 19.27. Consider we run the Riemannian SPIDER algorithm with option I to solve this problem. Let* $T = 1$, $p = \frac{n_0\sigma}{\epsilon}$, $\alpha_k = \min\big(\frac{\epsilon}{2Ln_0}, \frac{\|\xi_k\|}{4Ln_0}\big)$, $|\mathcal{S}_1| = \frac{64\sigma^2}{\epsilon^2}$, $|\mathcal{S}_2| = \frac{4\sigma}{\epsilon n_0}$ *for* $n_0 \in [1, 4\frac{\sigma}{\epsilon}]$ *in Algorithm 4. Then, we achieve* $\mathbb{E}[\|\nabla f(\tilde{x}^1)\|] \leq \epsilon$ *in at most* $K = \frac{14Ln_0\Delta}{\epsilon^2}$ *iterations in expectation, where* $\Delta = f(x_0) - f(x^*)$ *with* $x^* = \arg\min_{x \in \mathcal{M}} f(x)$.

The authors of [44] give the following convergence theorem for the same algorithm. The following theorems are for finite-sum and online settings.

Theorem 19.29 (Theorem 2 in Zhang et al. [44]) *Consider the same problem and assume the same conditions as in Theorem 19.27. Consider, we run the Riemannian SPIDER algorithm with option II to solve this problem. Let* $T = 1$, $p = \lceil n^{1/2} \rceil$, $\alpha_k = \min\{\frac{1}{2L}, \frac{\epsilon}{\|\xi_k\| L}\}$, $|\mathcal{S}_1| = n$, *and* $|\mathcal{S}_2| = \lceil n^{1/2} \rceil$ *for each iteration in Algorithm 4. Then, we achieve* $\mathbb{E}[\|\nabla f(\tilde{x}^1)\|^2] \leq 10\epsilon^2$ *in at most* $K = \frac{4L\Delta}{\epsilon^2}$ *iterations, where* $\Delta = f(x_0) - f(x^*)$ *with* $x^* = \arg\min_{x \in \mathcal{M}} f(x)$.

Algorithm 4 Riemannian SPIDER

Given: Smooth manifold $\mathcal{M}$ with retraction Ret and vector transport $\mathcal{T}$; initial value $\tilde{x}^0$; noisy version of the cost function f_i; iteration interval p^t, mini-batch sizes $|\mathcal{S}_1^t|$ and $|\mathcal{S}_{2,k}^t|$; number of epochs T and K^t.
for $t = 0, \ldots, T-1$ **do**
  $x_0 = \tilde{x}^t$
  for $k = 0, \ldots, K^t - 1$ **do**
    if $\mathrm{mod}(k, p^t) = 0$ **then**
      Draw minibatch size $|\mathcal{S}_1^t|$ and compute $\xi_k = \nabla f_{\mathcal{S}_1^t}(x_k)$
    else
      Draw minibatch size $|\mathcal{S}_2^t|$ and compute $\nabla f_{\mathcal{S}_2^t}(x_k)$
      Compute $\xi_k = \nabla f_{\mathcal{S}_2^t}(x_k) - \mathcal{T}_{x_{k-1},x_k}\big(\nabla f_{\mathcal{S}_2^t}(x_{k-1}) - \xi_{k-1}\big)$
    end if
    if $\xi_k \leq 2\epsilon_k$ **then**
      Option II: $\tilde{x}^{t+1} = x_k$, break
    end if
    Use a step-size rule to choose the step-size α_k^t
    Calculate $x_{k+1} = \mathrm{Ret}_{x_k}(-\alpha_k^t \xi_k)$
  end for
  Option I: Output $\tilde{x}^{t+1} = x_k$ for randomly chosen $k \in \{0, \ldots, K-1\}$
end for
return $\tilde{x}^T$

Theorem 19.30 (Theorem 1 in Zhang et al. [44]) *Consider the same problem and assume the same conditions as in Theorem 19.28. Consider, we run the Riemannian SPIDER algorithm with option II to solve this problem. Let* $T = 1$, $p = \frac{1}{\epsilon}$, $\alpha_k = \min\{\frac{1}{2L}, \frac{\epsilon}{\|\xi_k\| L}\}$, $|\mathcal{S}_1| = \frac{2\sigma^2}{\epsilon^2}$, *and* $|\mathcal{S}_2| = \frac{2}{\epsilon}$ *for each iteration in Algorithm 4. Then, we achieve* $\mathbb{E}[\|\nabla f(\tilde{x}^1)\|^2] \leq 10\epsilon^2$ *in at most* $K = \frac{4L\Delta}{\epsilon^2}$ *iterations, where* $\Delta = f(x_0) - f(x^*)$ *with* $x^* = \arg\min_{x \in \mathcal{M}} f(x)$.

Among the convergence results presented in this section, R-SPIDER is the only algorithm that has strong convergence without the need for the strong condition that the iterates stay in a compact set. This condition is hard to ensure even for simple problems. Another important point to mention is that the step-sizes suggested by the theorems are very small, and in practice much larger step-sizes with some decaying rules are usually used.

19.5 Analysis for G-Convex and Gradient Dominated Functions

For g-convex or gradient dominated functions, we obtain faster convergence rates for the algorithms explained in the previous sections. For plain SGD, [43] proved faster convergence for g-convex functions as stated in the following theorem.

Theorem 19.31 (Theorem 14 in Zhang et al. [43]) *Consider the R-SGD Algorithm for solving the optimization problem in (19.4), where the cost function is the expected risk (19.6). Assume*

- *The function f is g-convex.*
- *The exponential map is used for the retraction.*
- *The iterations stay in a compact subset $\mathcal{X}$, and the diameter of $\mathcal{X}$ is bounded by D, that is $\max_{x,y\in\mathcal{X}} d(x, y) \leq D$.*
- *There is a lower bound on the sectional curvature denoted by κ.*
- *The functions f_i are geodesically L-smooth.*
- *The function f attains its minimum at $x^* \in \mathcal{X}$.*
- *The functions f_i have σ-bounded variance.*
- *The manifold is Hadamard (Riemannian manifolds with global non-positive curvature).*

Define ζ to be a constant that captures the impact of manifold curvature defined by

$$\zeta = \frac{\sqrt{|\kappa|}D}{\tanh\left(\sqrt{|\kappa|}D\right)}. \tag{19.12}$$

Then the R-SGD algorithm with $\alpha_t = \frac{1}{L+\frac{\sigma}{D}\sqrt{(t+1)\zeta}}$ in Algorithm 1 satisfies

$$\mathbb{E}[f(\bar{x}_T) - f(x^*)] \leq \frac{\zeta L D^2 + 2D\sigma\sqrt{\zeta T}}{2(\zeta + T - 1)},$$

where $\bar{x}_1 = x_1$, $\bar{x}_{t+1} = \mathrm{Exp}_{\bar{x}_t}(\frac{1}{t+1}\mathrm{Exp}^{-1}_{\bar{x}_t}(x_{t+1}))$, for $1 \leq t \leq T-1$ and $\bar{x}_T = \mathrm{Exp}_{\bar{x}_{T-1}}(\frac{\zeta}{\zeta+T-1}\mathrm{Exp}^{-1}_{\bar{x}_{T-1}}(x_T))$.

The aforementioned theorem shows that we need a decaying step size for obtaining faster convergence for the R-SGD algorithm, while Theorem 19.22 needed constant step size for getting a convergence rate for general non-convex functions. Decaying step-size is usually used in practice and the above theorem can be a motivation, because near local minima the function can be assumed to be g-convex. For the case of strongly g-convex functions, the authors of [43] proved a stronger convergence result stated below.

Theorem 19.32 (Theorem 12 in Zhang et al. [43]) *Consider the R-SGD Algorithm for solving the optimization problem in (19.4), where the cost function is the expected risk (19.6). Assume*

- *The function f is μ-strongly g-convex.*
- *The exponential map is used for the retraction.*
- *The iterations stay in a compact subset $\mathcal{X}$, and the diameter of $\mathcal{X}$ is bounded by D, that is $\max_{x,y\in\mathcal{X}} d(x, y) \leq D$.*
- *There is a lower bound on the sectional curvature denoted by κ.*
- *The function f attains its minimum at $x^* \in \mathcal{X}$.*

– *The expected square norm of the gradients of the f_is are G^2-bounded.*
– *The manifold is Hadamard (Riemannian manifolds with global non-positive curvature).*

Then the R-SGD algorithm with $\alpha_t = \frac{2}{\mu(t+2)}$ in Algorithm 1 satisfies

$$\mathbb{E}[f(\bar{x}_T) - f(x^*)] \leq \frac{2\zeta G}{(T+2)},$$

where $\bar{x}_0 = x_0$, $\bar{x}_{t+1} = \operatorname{Exp}_{\bar{x}_t}(\frac{2}{t+2}\operatorname{Exp}^{-1}_{\bar{x}_t}(x_{t+1}))$ and ζ is a constant given in (19.12).

For strongly g-convex functions, [33] proved a linear convergence rate for the R-SVRG algorithm given in the following theorem.

Theorem 19.33 (Theorem 1 in Zhang et al. [33]) *Consider the optimization problem in (19.4), where the cost function is the finite sum (19.7). Consider, we run the Riemannian SVRG algorithm to solve this problem with $K = 1$, Option I-a and Option I-b. Assume the same conditions as in Theorem 19.24. Furthermore, assume that the function f is μ-strongly g-convex. If we use an update frequency and a constant step size in Algorithm 2 such that the following holds*

$$a = \frac{3\zeta\alpha L^2}{\mu - 2\zeta\alpha L^2} + \frac{(1 + 4\zeta\alpha^2 - 2\alpha\mu)^m(\mu - 5\zeta\alpha L^2)}{\mu - 2\zeta\alpha L^2} < 1,$$

then the iterations satisfy

$$\mathbb{E}[f(\tilde{x}^S) - f(x^*)] \leq \frac{L}{2}\mathbb{E}[d^2(\tilde{x}^S, x^*)] \leq \frac{L}{2}a^S d^2(x^0, x^*).$$

For a class of functions more general than strongly g-convex functions, that is gradient-dominated functions, it is also possible to prove that R-SVRG has a strong linear convergence rate.

Theorem 19.34 (Theorem 3 in Zhang et al. [33]) *Consider the optimization problem in (19.4), where the cost function is the finite sum (19.7). Consider, we run the Riemannian SVRG algorithm to solve this problem with Option II-a, Option I-b. Assume the same conditions as in Theorem 19.24. Furthermore, assume that the function f is τ-gradient dominated. If we use the parameters $\alpha = \frac{\mu_0}{Ln^{2/3}\zeta^{1/3}}$, $m = \lfloor \frac{n}{3\mu_0} \rfloor$, $S = \lceil (6 + \frac{18\mu_0}{n-3})\frac{L\tau\zeta^{1/2}\mu_0}{\nu n^{1/3}} \rceil$ for some universal constants $\mu_0 \in (0, 1)$ and $\nu > 0$ in Algorithm 2, then we have*

$$\mathbb{E}[\|\nabla f(x^K)\|^2] \leq 2^{-K}\|\nabla f(x^0)\|^2,$$
$$\mathbb{E}[f(x^K) - f(x^*)] \leq 2^{-K}[\|f(x^0) - f(x^*)\|].$$

The aforementioned strong convergence results for R-SVRG are valid when using the exponential map and the parallel transport. For the general retraction and the vector transport there is not any global rate of convergence result yet. However, the authors in [36, Theorem 5.14] proved a local linear convergence result for the R-SVRG algorithm.

For the R-SRG algorithm, [22] gives a convergence result for the g-convex case as stated in the following.

Theorem 19.35 (Theorem 4.1 in Kasai et al. [22]) *Consider the optimization problem in (19.4), where the cost function is the finite sum (19.7). Assume the same conditions as in Theorem 19.26 hold, and furthermore assume that*

$$\|\mathcal{P}_{x,y}^{\mathrm{Ret}_x} \nabla f_i(x) - \nabla f_i(y)\|^2 \leq L \langle \mathcal{P}_{x,y}^{\mathrm{Ret}_x} \nabla f_i(x) - \nabla f_i(y),\ \mathrm{Exp}_y^{-1} x \rangle,$$

where L is the constant for the retraction smooth function f. For the Euclidean case, this condition is equal to have a convex and L-smooth function. Consider, we run the Riemannian SRG algorithm to solve this problem using the parameters α and m in Algorithm 3 such that $\alpha < 2/L$ and $(\beta - L^2)\alpha^2 + 3L\alpha - 2 \leq 0$, where

$$\beta := 2\big((2L_l + 2\theta G + L)\theta G + \nu L\big)m. \tag{19.13}$$

Then for $s > 0$,

$$\mathbb{E}[\|\nabla f(\tilde{x}^s)\|^2] \leq \frac{2}{\alpha(m+1)} \mathbb{E}[\|f(\tilde{x}^{s-1}) - f(x^*)\|] + \frac{\alpha L}{2 - \alpha L} \mathbb{E}[\|\nabla f(\tilde{x}^{s-1})\|^2].$$

For μ-strongly g-convex functions, the authors of [22] proved linear convergence as stated below. The nice feature of the R-SRG algorithm is that it is the only method that achieves linear convergence without needing the exponential map and the parallel transport.

Theorem 19.36 (Theorem 4.3 in Kasai et al. [22]) *Consider the optimization problem in (19.4), where the cost function is the finite sum (19.7). Assume the same conditions as in Theorem 19.35 and furthermore assume that the function f is μ-strongly convex. Consider, we run the Riemannian SRG algorithm to solve this problem using the parameters α and m in Algorithm 3 such that such that $\alpha_m := \frac{1}{\mu\alpha(m+1)} + \frac{\alpha L}{2-\alpha L} < 1$. Then,*

$$\mathbb{E}[\|\nabla f(\tilde{x}^s)\|^2] \leq \sigma_m^s \mathbb{E}[\|\nabla f(\tilde{x}^0)\|^2].$$

Similarly for τ gradient dominated functions, the authors of [22] obtained linear convergence.

Theorem 19.37 (Theorem 4.6 in Kasai et al. [22]) *Consider the optimization problem in (19.4), where the cost function is the finite sum (19.7). Assume the same conditions as in Theorem 19.26 hold and furthermore assume that the function is*

τ-gradient dominated. Consider, we run Riemannian SRG algorithm to solve this problem with the same α in Algorithm 3 as that of Theorem 19.26 and assume $\bar{\sigma}_m := \frac{2\tau}{\alpha(m+1)} < 1$. Then for $s > 0$,

$$\mathbb{E}[\|\nabla f(\tilde{x}^s)\|^2] \leq \bar{\sigma}_m^s \mathbb{E}[\|\nabla f(\tilde{x}^0)\|^2].$$

For τ-gradient dominated functions, [45] was able to prove stronger convergence results for the R-SPIDER algorithm. The following two theorems are convergence results for the finite-sum and online cases. Unlike the analysis for the general non-convex case, here the authors use a fixed step-size and adaptive batch sizes.

Theorem 19.38 (Theorem 3 in Zhou et al. [45]) *Consider the finite sum problem (19.7) solved using the R-SPIDER algorithm with option I. Assume the same conditions as in Theorem 19.27, and furthermore assume that the function f is τ-gradient dominated. At iteration t of Algorithm 4, set $\epsilon_0 = \frac{\sqrt{\Delta}}{2\sqrt{\tau}}$, $\epsilon_t = \frac{\epsilon_0}{2^t}$, $s_t = \min(n, \frac{32\sigma^2}{\epsilon_{t-1}^2})$, $p^t = n_0^t s_t^{\frac{1}{2}}$,$\alpha_k = \frac{\|\xi_k\|}{2Ln_0}$,$|\mathcal{S}_1^t| = s_t$, $|\mathcal{S}_{2,k}^t| = \min\left(\frac{8p^t\|\xi_{k-1}\|^2}{(n_0^t)^2\epsilon_{t-1}^2}, n\right)$ and $K^t = \frac{64Ln_o^t\Delta^t}{\epsilon_{t-1}^2}$ where $n_0^t \in [1, \frac{8\sqrt{s}\|\xi_{k-1}\|^2}{\epsilon_{t-1}^2}]$ and $\Delta = f(x_0) - f(x^*)$ with $x^* = \arg\min_{x\in\mathcal{M}} f(x)$. Then the sequence $\tilde{x}^t$ satisfies*

$$\mathbb{E}[\|\nabla f(\tilde{x}^t)\|^2] \leq \frac{\Delta}{4^t\tau}.$$

Theorem 19.39 (Theorem 4 in Zhou et al. [45]) *Consider the optimization problem in (19.4) solved using the R-SPIDER algorithm with option I. Assume the same conditions as in Theorem 19.28, and furthermore assume that the function f is τ-gradient dominated. At iteration t of Algorithm 4, set $\epsilon_0 = \frac{\sqrt{\Delta}}{2\sqrt{\tau}}$, $\epsilon_t = \frac{\epsilon_0}{2^t}$, $p^t = \frac{\sigma n_0^t}{\epsilon_{t-1}}$, $\alpha_k^t = \frac{\|\xi_k\|}{2L_l n_0^t}$, $|\mathcal{S}_1^t| = \frac{32\sigma^2}{\epsilon_{t-1}^2}$, $|\mathcal{S}_{2,k}^t| = \frac{8\sigma\|\xi_{k-1}\|^2}{n_0^t\epsilon_{t-1}^3}$ and $K^t = \frac{64Ln_o^t\Delta^t}{\epsilon_{t-1}^2}$ where $n_0 \in [1, \frac{8\sigma\|\xi_{k-1}\|^2}{\epsilon_{t-1}^3}]$ and $\Delta = f(x_0) - f(x^*)$ with $x^* = \arg\min_{x\in\mathcal{M}} f(x)$. Then the sequence $\tilde{x}^t$ satisties,*

$$\mathbb{E}[\|\nabla f(\tilde{x}^t)\|^2] \leq \frac{\Delta}{4^t\tau}.$$

The authors of [44] give the following analysis of the R-SPIDER algorithm for τ-gradient dominated functions.

Theorem 19.40 (Theorem 3 in Zhang et al. [44]) *Consider the same problem and assume the same conditions as in Theorem 19.28. Consider, we run the Riemannian SPIDER algorithm with option II to solve this problem. Let $p = \lceil n^{1/2}\rceil$,$\epsilon_t = \sqrt{\frac{M_0}{10\tau 2^t}}$,$\alpha_t = \frac{\epsilon_t}{L}$,$|\mathcal{S}_1| = n$, and $|\mathcal{S}_2| = \lceil n^{1/2}\rceil$ in each iteration of Algorithm 4, where $M_0 > f(x_0) - f(x^*)$ with $x^* = \arg\min_{x\in\mathcal{M}} f(x)$. Then the algorithm returns $\tilde{x}^T$ that satisfies*

$$\mathbb{E}[f(\tilde{x}^T) - f(x^*)] \leq \frac{M_0}{2^T}.$$

The authors of [44] also give another proof for the R-SPIDER algorithm with different parameters that give better iteration complexity for τ-gradient dominated functions with respect to n.

Theorem 19.41 (Theorem 4 in Zhang et al. [44]) *Consider the same problem and assume the same conditions as in Theorem 19.28. Consider, we run the Riemannian SPIDER algorithm with option II to solve this problem. In Algorithm 4, let* $T = 1$, $p = \lceil 4L\tau \log(4) \rceil$, $\alpha = \frac{1}{2L}$, $|\mathcal{S}_1| = n$, *and* $|\mathcal{S}_{2,k}| = \lceil \min\left\{n, \frac{4\tau p L^2 \| \operatorname{Exp}_{x_{k-1}}^{-1}(x_k)\|^2 2^{\lceil k/p \rceil}}{M_0}\right\} \rceil$, *where* $M_0 > f(x_0) - f(x^*)$ *with* $x^* = \arg\min_{x \in \mathcal{M}} f(x)$. *Then, the algorithm returns* $\tilde{x}^K$ *after* $K = pS$ *iterations that satisfies*

$$\mathbb{E}[f(\tilde{x}^K) - f(x^*)] \leq \frac{M_0}{2^S}.$$

The theorems of the algorithms in the previous sections showing convergence speed of different algorithms are summarized in Tables 19.1 and 19.2. The incremental first order oracle (IFO) complexity for different algorithms are calculated by counting the number of evaluations needed to reach the ϵ accuracy of gradient ($\mathbb{E}[\|\nabla f(x)\|^2] \leq \epsilon$) or function ($\mathbb{E}[f(x) - f(x^*)] \leq \epsilon$) in the theorems given in the previous sections.

Table 19.1 Comparison of the IFO complexity for different Riemannian stochastic optimization algorithms under finite-sum and online settings

	Method	general non-convex	g-convex	Theorem
Finite-sum	R-SGD* [18]	$O\left(\frac{L}{\epsilon} + \frac{L^2\sigma^2}{\epsilon^2}\right)$	–	19.22
	R-SRG [22]	$O\left(n + \frac{L^2}{\epsilon^2}\right)$	$O\left(\left(n + \frac{1}{\epsilon}\right)\log\left(\frac{1}{\epsilon}\right)\right)$	19.26, 19.35
	R-SRG* [22]	$O\left(n + \frac{L^2\rho_l^2 + \theta^2}{\epsilon^2}\right)$	$O\left(\frac{\left(n+\frac{1}{\epsilon}\right)\log\left(\frac{1}{\epsilon}\right)}{\log(c(1-\beta/L^2))}\right)$	19.26, 19.35
	R-SVRG [33]	$O\left(n + \frac{\zeta^{1/2} n^{2/3}}{\epsilon}\right)$	–	19.24
	R-SPIDER [45]	$O\left(\min\left(n + \frac{L\sqrt{n}}{\epsilon}, \frac{L\sigma}{\epsilon^{3/2}}\right)\right)$	–	19.27, 19.38
	R-SPIDER [44]	$O\left(n + \frac{L\sqrt{n}}{\epsilon}\right)$	–	19.29

(continued)

Table 19.1 (continued)

Online	R-SGD* [18]	$O\left(\frac{L}{\epsilon}+\frac{L^2\sigma^2}{\epsilon^2}\right)$	–	19.22
	R-SPIDER [45]	$O\left(\frac{L\sigma}{\epsilon^{3/2}}\right)$	–	19.28
	R-SPIDER [44]	$O\left(\frac{L\sigma^2}{\epsilon^{3/2}}\right)$	–	19.30

The ϵ-accuracies of gradients are reported for general non-convex and g-convex functions. Star in front of the method names means using the general retraction and the parallel transport, and no star means using the exponential map and the parallel transport in the method. The parameter ζ (19.11) is determined by manifold curvature and diameter, σ is the standard deviation of stochastic gradients, θ is the constant in θ-bounded vector transport, $\rho_l = L_l/L$ for retraction L-smooth and retraction L_l-Lipschitz function, the parameter β is defined in (19.13) and $c > 1$ is a constant. Apparently for the parallel transport $\theta = 0$ and $\rho_l = 1$

Table 19.2 Comparison of the IFO complexity for different Riemannian stochastic optimization algorithms under finite-sum and online settings

	Method	τ-gradient dominated	μ-strongly g-convex	Theorem
Finite-sum	R-SGD [44]	–	$\frac{\zeta G}{\epsilon}$	19.32
	R-SRG [22]	$O\left(\left(n+L_\tau^2\right)\log\left(\frac{1}{\epsilon}\right)\right)$	$O\left(\left(n+L_\mu\right)\log\left(\frac{1}{\epsilon}\right)\right)$	19.36, 19.37
	R-SRG* [22]	$O\left(\left(n+\tau^2(L^2\rho_l^2+\theta^2)\right)\log\left(\frac{1}{\epsilon}\right)\right)$	$O\left(\frac{(n+L_\mu)\log\left(\frac{1}{\epsilon}\right)}{\log(c(1-\beta/L^2))}\right)$	19.36, 19.37
	R-SVRG [33]	$O\left(\left(n+L_\tau\zeta^{1/2}n^{2/3}\right)\log\left(\frac{1}{\epsilon}\right)\right)$	$O\left(\left(n+\zeta L_\mu^2\right)\log\left(\frac{1}{\epsilon}\right)\right)$	19.33, 19.34
	R-SPIDER [45]	$O\left(\min\left(\left(n+L_\tau\sqrt{n}\right)\log\left(\frac{1}{\epsilon}\right),\frac{L_\tau\sigma}{\epsilon^{1/2}}\right)\right)$	←	19.38
	R-SPIDER [44]	$O\left(\left(n+\min\left(L_\tau\sqrt{n},L_\tau^2\right)\right)\log\left(\frac{1}{\epsilon}\right)\right)$	←	19.40, 19.41
Online	R-SGD [44]	–	$\frac{\zeta G}{\epsilon}$	19.32
	R-SPIDER [45]	$O\left(\frac{L_\tau\sigma}{\epsilon^{1/2}}\right)$	←	19.39

The ϵ-accuracies of functions are reported for μ-strongly g-convex and τ-gradient dominated functions. The results of Theorems 19.34, 19.36, 19.37, 19.38 are originally given for the ϵ-accuracy of gradient, and they also hold for the ϵ-accuracy of function because of (19.3). The parameters $L_\tau = 2\tau L$ and $L_\mu = \frac{L}{\mu}$ are condition numbers, G is the bound for the norm of the stochastic gradients, and other parameters are the same as those given in Table 19.1. From Proposition 19.19, it is clear that the complexity results for τ-gradient dominated functions also hold for μ-strongly g-convex functions, and to obtain complexity results it is enough to change L_τ to L_μ in the equations

19.6 Example Applications

We list below a few finite-sum optimization problems drawn from a variety of applications. Riemannian stochastic methods turn out to be particularly effective for solving these problems. We only include the formulation, and refer the reader to the cited works for details about implementation and empirical performance. The manifolds occurring in the examples below are standard, and the reader can find explicit implementations of retractions, vector transport, etc., within the MANOPT software [10], for instance.

Stochastic PCA
Suppose we have observations $z_1, \ldots, z_n \in \mathbb{R}^d$. The stochastic PCA problem is to compute the top eigenvector of the matrix $\sum_{i=1}^n z_i z_i^T$. This problem can be written as a finite-sum optimization problem on the sphere $\mathbb{S}^{d-1}$ as follows

$$\min_{x^T x=1} \quad -x^T \left(\sum\nolimits_{i=1}^n z_i z_i^T \right) x = -\sum\nolimits_{i=1}^n (z_i^T x)^2. \tag{19.14}$$

Viewing (21.100) as a Riemannian optimization problem was proposed in [33], who solved it using R-SVRG, in particular, by proving that the cost function satisfies a Riemannian gradient-dominated condition (probabilistically). One can extend this problem to solve for the top-k eigenvectors by considering is as an optimization problem on the Stiefel manifold.

A challenge for the methods discussed in the present paper, except R-SGD and R-SPIDER explained in Sect. 19.4 is the requirements for the iterates to remain within a predefined compact set. While the whole manifold is compact, for obtaining a precise theoretical characterization of the computational complexity of the algorithms involved, the requirement to remain within a compact set is important.

GMM
Let $z_1, \ldots, z_n$ be observations in $\mathbb{R}^d$ that we wish to model using a Gaussian mixture model (GMM). Consider the mixture density

$$p(z; \{\mu_j, \Sigma_j\}_{j=1}^k) := \sum\nolimits_{j=1}^k \pi_j \mathcal{N}(z; \mu_j, \Sigma_j),$$

where $\mathcal{N}(z; \mu, \Sigma)$ denotes the Gaussian density evaluated at z and parameterized by μ and Σ. This leads to the following maximum likelihood problem:

(continued)

$$\max_{\{\pi_j,\mu_j,\Sigma_j\}_{j=1}^k} \sum_{i=1}^n \log p(z_i; \{\mu_j, \Sigma_j\}_{j=1}^k). \tag{19.15}$$

In [18], the authors reformulate (21.52) to cast it as a problem well-suited for solving using R-SGD. They consider the reformulated problem

$$\max_{\{\omega_j,S_j\succ 0\}_{j=1}^k} \sum_{i=1}^n \log\Big(\sum_{j=1}^k \frac{\exp(\omega_j)}{\sum_{k=1}^k \exp(\omega_k)} q(y_i; S_j)\Big), \tag{19.16}$$

where $y_i = [z_i; 1]$, and $q(y; S_j)$ is the centered normal distribution parameterized by $S_j = \begin{bmatrix} \Sigma_j + \mu_j\mu_j^T & \mu_j \\ \mu_j^T & 1 \end{bmatrix}$. With these definitions, problem (21.61) can be viewed as an optimization problem on the product manifold $\left(\prod_{j=1}^k \mathbb{P}^{d+1}\right) \times \mathbb{R}^{k-1}$.

Importantly, in [18] it was shown that SGD generates iterates that remain bounded, which is crucial, and permits one to invoke the convergence analysis without resorting to projection onto a compact set.

Karcher Mean

Let $A_1, \ldots, A_n$ be hermitian (strictly) positive definite (hpd) matrices. This set is a manifold, commonly endowed with the Riemmanian metric $\langle \eta, \xi \rangle = \operatorname{tr}(\eta X^{-1} \xi X^{-1})$. This metric leads to the distance $d(X, Y) := \|\log(X^{-1/2} Y X^{-1/2})\|_F$ between hpd matrices X and Y. The Riemannian centroid (also called the "Karcher mean") is defined as the solution to the following finite-sum optimization problem:

$$\min_{X\succ 0} \quad \sum_{i=1}^n w_i d^2(X, A_i), \tag{19.17}$$

where the weights $w_i \geq 0$ and $\sum_{i=1}^n w_i = 1$. This problem is often used as a defacto benchmark problem for testing Riemannian optimization problems (see e.g., [33]). The objective function in (19.14) is both geodesically L-smooth as well as strongly convex, both properties can be exploited to obtain faster convergence [22, 33].

It is important to note that this problem is over the manifold of hpd matrices, which is a noncompact manifold. Hence, to truly invoke the convergence theorems (except for R-SGD and R-SPIDER explained in Sect. 19.4), we need to ensure lower bounds on the curvature as well as ensure that iterates remain within a compact set. Lower bounds on the curvature can be obtained in terms

(continued)

of $\min_{1\le i\le n} \lambda_{\min}(A_i)$; ensuring that the iterates remain within a compact set can be ensured via projection. Fortunately, for (19.17), a simple compact set containing the solution is known, since we know that (see e.g., [7]) its solution X^* satisfies $\textsc{Hm}(A_1, \ldots, A_n) \preceq X^* \preceq \textsc{Am}(A_1, \ldots, A_n)$, where HM and AM denote the Harmonic and Arithmetic Means, respectively. A caveat, however, is that R-SVRG and related methods do not permit a projection operation and assume their iterates to remain in a compact set by fiat; R-SGD, however, allows metric projection and can be applied. Nevertheless, in practice, one can invoke any of the methods discussed in this chapter.

We note in passing here that the reader may also be interested in considering the somewhat simpler "Karcher mean" problems that arise when learning hyperbolic embeddings [35], as well as Fréchet-means on other manifolds [3, 31].

Wasserstein Barycenters
Consider two centered multivariate Gaussian distributions with covariance matrices Σ_1 and Σ_2. The Wasserstein W_2 optimal transport distance between them is given by

$$d_W^2(\Sigma_1, \Sigma_2) := \operatorname{tr}(\Sigma_1 + \Sigma_2) - 2\operatorname{tr}[(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2})^{1/2}]. \tag{19.18}$$

The Wasserstein barycenter of n different centered Gaussians is then given by the solution to the optimization problem

$$\min_{X\succ 0} \quad \sum\nolimits_{i=1}^{n} w_i d_W^2(X, \Sigma_i). \tag{19.19}$$

While (21.83) is a (Euclidean) convex optimization problem, it lends itself to more efficient solution by viewing it as a Riemannian convex optimization problem [40]. A discussion about compact sets similar to the Karcher mean example above applies here too.

Riemannian Dictionary Learning
Dictionary learning problems seek to encode input observations using a sparse combination of an "overcomplete basis". The authors of [12] study a Riemannian version of dictionary learning, where input hpd matrices must be encoded as sparse combinations of a set of hpd "dictionary atoms." This problem may be cast as the finite-sum minimization problem

(continued)

$$\min_{B,\alpha_1,\ldots,\alpha_n} \quad \sum_{i=1}^{n} d^2\left(X_i, \sum_{j=1}^{m} \alpha_{ij} B_j\right) + R(B, \alpha_1, \ldots, \alpha_n). \tag{19.20}$$

In other words, we seek to approximate each input matrix $X_i \approx \sum_{j=1}^{m} \alpha_{ij} B_j$, using $B_j \succ 0$ and nonegative coefficients α_{ij}. The function $R(\cdot)$ is a suitable regularizer on the tensor B and the coefficient matrix α, and $d(\cdot, \cdot)$ denotes the Riemannian distance.

For this particular problem, we can invoke any of the discussed stochastic methods in practice; though previously, results only for SGD have been presented [12]. By assuming a suitable regularizer $R(\cdot, \cdot)$ we can ensure that the problem has a solution, and that the iterates generated by the various methods remain bounded.

Acknowledgements SS acknowledges partial support from an NSF-CAREER grant, the DARPA-Lagrange program, and Amazon Research.

References

1. Absil, P.A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton (2009)
2. Amari, S.I.: Natural gradient works efficiently in learning. Neural Comput. **10**(2), 251–276 (1998)
3. Arnaudon, M., Barbaresco, F., Yang, L.: Medians and means in Riemannian geometry: existence, uniqueness and computation. In: Matrix Information Geometry, pp. 169–197. Springer, Berlin (2013)
4. Babanezhad, R., Laradji, I.H., Shafaei, A., Schmidt, M.: Masaga: a linearly-convergent stochastic first-order method for optimization on manifolds. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 344–359. Springer, Berlin (2018)
5. Bécigneul, G., Ganea, O.E.: Riemannian adaptive optimization methods (2018). Preprint. arXiv:1810.00760
6. Bertsekas, D.P.: Nonlinear Programming, 2nd edn. Athena Scientific, Nashua (1999)
7. Bhatia, R.: Positive Definite Matrices. Princeton University Press, Princeton (2007)
8. Bonnabel, S.: Stochastic gradient descent on Riemannian manifolds. IEEE Trans. Autom. Control **58**(9), 2217–2229 (2013)
9. Boumal, N., Absil, P.A.: RTRMC: a Riemannian trust-region method for low-rank matrix completion. In: Advances in Neural Information Processing Systems, pp. 406–414 (2011)
10. Boumal, N., Mishra, B., Absil, P.A., Sepulchre, R.: Manopt, a matlab toolbox for optimization on manifolds. J. Mach. Learn. Res. **15**(1), 1455–1459 (2014)
11. Boumal, N., Absil, P.A., Cartis, C.: Global rates of convergence for nonconvex optimization on manifolds. IMA J. Numer. Anal. **39**(1), 1–33 (2019)
12. Cherian, A., Sra, S.: Riemannian dictionary learning and sparse coding for positive definite matrices. IEEE Trans. Neur. Net. Lear. Syst. **28**(12), 2859–2871 (2017)
13. Defazio, A., Bach, F., Lacoste-Julien, S.: SAGA: a fast incremental gradient method with support for non-strongly convex composite objectives. In: Advances in Neural Information Processing Systems, pp. 1646–1654 (2014)

14. Fang, C., Li, C.J., Lin, Z., Zhang, T.: Spider: near-optimal non-convex optimization via stochastic path-integrated differential estimator. In: Advances in Neural Information Processing Systems, pp. 687–697 (2018)
15. Ghadimi, S., Lan, G.: Stochastic first- and zeroth-order methods for nonconvex stochastic programming. SIAM J. Optim. **23**(4), 2341–2368 (2013)
16. Guadarrama: Fitting large-scale gaussian mixtures with accelerated gradient descent. Master's Thesis, University of Edinburgh (2018)
17. Hosseini, R., Sra, S.: Matrix manifold optimization for Gaussian mixtures. In: Advances in Neural Information Processing Systems, pp. 910–918 (2015)
18. Hosseini, R., Sra, S.: An alternative to EM for Gaussian mixture models: batch and stochastic Riemannian optimization. Math. Program. (2019)
19. Huang, W., Gallivan, K.A., Absil, P.A.: A broyden class of quasi-Newton methods for riemannian optimization. SIAM J. Optim. **25**(3), 1660–1685 (2015)
20. Jost, J.: Riemannian Geometry and Geometric Analysis. Springer, Berlin (2011)
21. Kasai, H., Mishra, B.: Inexact trust-region algorithms on Riemannian manifolds. In: Advances in Neural Information Processing Systems, pp. 4249–4260 (2018)
22. Kasai, H., Sato, H., Mishra, B.: Riemannian stochastic recursive gradient algorithm. In: International Conference on Machine Learning, pp. 2516–2524 (2018)
23. Kasai, H., Sato, H., Mishra, B.: Riemannian stochastic quasi-Newton algorithm with variance reduction and its convergence analysis. In: Twenty-First International Conference on Artificial Intelligence and Statistics, vol. 84, pp. 269–278 (2018)
24. Kasai, H., Jawanpuria, P., Mishra, B.: Riemannian adaptive stochastic gradient algorithms on matrix manifolds. In: International Conference on Machine Learning, pp. 3262–3271 (2019)
25. Kumar Roy, S., Mhammedi, Z., Harandi, M.: Geometry aware constrained optimization techniques for deep learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4460–4469 (2018)
26. Liu, H., So, A.M.C., Wu, W.: Quadratic optimization with orthogonality constraint: explicit łojasiewicz exponent and linear convergence of retraction-based line-search and stochastic variance-reduced gradient methods. Math. Program. 1–48 (2018)
27. Meyer, G., Bonnabel, S., Sepulchre, R.: Linear regression under fixed-rank constraints: a Riemannian approach. In: International Conference on Machine Learning (2011)
28. Mishra, B., Kasai, H., Jawanpuria, P., Saroop, A.: A Riemannian gossip approach to subspace learning on Grassmann manifold. Mach. Learn. **108**(10), 1783–1803 (2019)
29. Nguyen, L.M., Liu, J., Scheinberg, K., Takáč, M.: SARAH: a novel method for machine learning problems using stochastic recursive gradient. In: International Conference on Machine Learning, pp. 2613–2621 (2017)
30. Nguyen, L.M., van Dijk, M., Phan, D.T., Nguyen, P.H., Weng, T.W., Kalagnanam, J.R.: Optimal finite-sum smooth non-convex optimization with SARAH. arXiv preprint arXiv: 1901.07648 (2019)
31. Nielsen, F., Bhatia, R.: Matrix Information Geometry. Springer, Berlin (2013)
32. Reddi, S.J., Hefny, A., Sra, S., Poczos, B., Smola, A.: Stochastic variance reduction for nonconvex optimization. In: International Conference on Machine Learning, pp. 314–323 (2016)
33. Zhang, H., Reddi, S., Sra, S.: Riemannian SVRG: Fast stochastic optimization on Riemannian manifolds. In: Advances in Neural Information Processing Systems, pp. 4592–4600 (2016)
34. Rudi, A., Ciliberto, C., Marconi, G., Rosasco, L.: Manifold structured prediction. In: Advances in Neural Information Processing Systems, pp. 5610–5621 (2018)
35. Sala, F., De Sa, C., Gu, A., Re, C.: Representation tradeoffs for hyperbolic embeddings. In: International Conference on Machine Learning, vol. 80, pp. 4460–4469 (2018)
36. Sato, H., Kasai, H., Mishra, B.: Riemannian stochastic variance reduced gradient algorithm with retraction and vector transport. SIAM J. Optim. **29**(2), 1444–1472 (2019)
37. Sra, S., Hosseini, R.: Conic geometric optimization on the manifold of positive definite matrices. SIAM J. Optim. **25**(1), 713–739 (2015)

38. Sra, S., Hosseini, R., Theis, L., Bethge, M.: Data modeling with the elliptical gamma distribution. In: Artificial Intelligence and Statistics, pp. 903–911 (2015)
39. Vandereycken, B.: Low-rank matrix completion by Riemannian optimization. SIAM J. Optim. **23**(2), 1214–1236 (2013)
40. Weber, M., Sra, S.: Riemannian Frank-Wolfe methods with application to the Karcher and Wasserstein means. arXiv: 1710.10770 (2018)
41. Xu, Z., Gao, X.: On truly block eigensolvers via Riemannian optimization. In: International Conference on Artificial Intelligence and Statistics, pp. 168–177 (2018)
42. Yuan, X., Huang, W., Absil, P.A., Gallivan, K.A.: A Riemannian quasi-Newton method for computing the karcher mean of symmetric positive definite matrices. Technical Reporet FSU17-02, Florida State University (2017)
43. Zhang, H., Sra, S.: First-order methods for geodesically convex optimization. In: Conference on Learning Theory, pp. 1617–1638 (2016)
44. Zhang, J., Zhang, H., Sra, S.: R-SPIDER: A fast Riemannian stochastic optimization algorithm with curvature independent rate. arXiv: 1811.04194 (2018)
45. Zhou, P., Yuan, X.T., Feng, J.: Faster first-order methods for stochastic non-convex optimization on Riemannian manifolds. In: The 22nd International Conference on Artificial Intelligence and Statistics, pp. 138–147 (2019)