Riemannian stochastic variance reduced gradient on Grassmann manifold

Hiroyuki Kasai
The university of Electro-Communications
Chofu-shi, Tokyo, 182-8585, Japan
kasai@is.uec.ac.jp

Hiroyuki Sato
Tokyo University of Science,
Tokyo, Japan
hsato@rs.tus.ac.jp

Bamdev Mishra
Amazon Development Centre India,
Bangalore, India
bamdevm@amazon.com

Abstract

Stochastic variance reduction algorithms have recently become popular for minimizing the average of a large, but finite, number of loss functions. We propose a novel Riemannian extension of the Euclidean stochastic variance reduced gradient algorithm to a compact manifold search space. To this end, we show the developments on the Grassmann manifold. We present a global convergence analysis of the proposed algorithm with a decay step-size and a local convergence rate analysis under a fixed step-size with under some natural assumptions. Numerical comparisons on low-rank matrix completion show that the proposed algorithm outperforms the standard Riemannian stochastic gradient descent algorithm.

1 Introduction

In this paper, we focus on the problem $\min_w f(w)$, where $f(w) := \frac{1}{N}\sum_{n=1}^N f_n(w)$, w is the model variable, N is the number of samples, and $f_n(w)$ is the loss incurred on n-th sample. The *full gradient descent* (GD) algorithm requires evaluations of N derivatives, i.e., $\sum_{n=1}^N \nabla f_n(w)$, per iteration, which is computationally heavy when N is very large. A popular alternative is to use only one derivative $\nabla f_n(w)$ per iteration for n-th sample, which is the basis of the *stochastic gradient descent* (SGD) algorithm. However, SGD suffers from a slower convergence rate. To circumvent this issue, *variance reduction* techniques have been recently proposed to accelerate the convergence of SGD [1, 2, 3, 4, 5, 6, 7]. Particularly, the stochastic variance reduced gradient (SVRG) algorithm is a popular algorithm that enjoys superior convergence properties [1]. For smooth and strongly convex functions, SVRG has convergence rates similar to those of stochastic dual coordinate ascent [5] and stochastic average gradient (SAG) algorithms [3]. The works [8, 9, 10, 11] extend the analysis to particular non-convex unconstrained optimization problems.

In this paper, we deal with problems where the variables have a Riemannian manifold structure. Specifically, the problem of interest is

$$\min_{w \in \mathcal{M}} f(w) := \frac{1}{N}\sum_{n=1}^{N} f_n(w), \tag{1}$$

where $\mathcal{M}$ is a Riemannian manifold. Bonnabel [12] proposes a *Riemannian stochastic gradient descent* algorithm (R-SGD) that extends SGD from the Euclidean space to Riemannian manifolds. The problem (1) is solved as an *unconstrained optimization problem* defined over the Riemannian manifold search space. Building upon this work, we propose a novel extension of SVRG in the Euclidean

9th NIPS Workshop on Optimization for Machine Learning (OPT 2016), Barcelona, Spain.

space to the Riemannian manifold (R-SVRG). This extension is not trivial and requires particular consideration in dealing with averaging, addition, and subtraction of multiple gradients at different points on the manifold $\mathcal{M}$. To this end, this paper specifically focuses on the *Grassmann manifold*. Nonetheless, the proposed algorithm and the analysis presented can be generalized to other compact Riemannian manifolds. The detailed analysis of our proposed algorithm and numerical comparisons are in our extended technical report [13].

It should be mentioned that the recent work [14], which appeared simultaneously with our technical report [13], also proposes R-SVRG on manifolds. The differences of our work with [14] are two fold. First, our convergence analysis deals with global convergence and local rate of convergence analysis separately. This is similar to the typical analysis for batch algorithms on Riemannian manifolds [15]. The second difference is that our assumptions for the local rate of convergence analysis are imposed only in a local neighborhood around a minimum, which are milder and more natural than those in [14] that assumes Lipschitz smoothness in the entire space. Consequently, our analysis should be applicable to wider kinds of manifolds than [14].

Our proposed R-SVRG is implemented in the Matlab toolbox Manopt [16]. The implementations are available at `https://bamdevmishra.com/codes/rsvrg/`.

2 Riemannian stochastic variance reduced gradient on Grassmann manifold

After a brief explanation of the variance reduced gradient variants in the Euclidean space, the Riemannian stochastic variance reduced gradient on the Grassmann manifold is proposed.

SVRG in the Euclidean space. The SGD update is $w_{t+1} = w_t - \eta v_t$, where v_t is a randomly selected vector that is called the *stochastic gradient* and η is the step-size. While SGD assumes $\nabla f_n(w_t)$, an *unbiased estimator* of the full gradient, i.e., $\mathbb{E}_n[\nabla f_n(w_t)] = \nabla f(w_t)$, as a stochastic gradient, many recent variants of the variance reduced gradient of SGD attempt to reduce its variance $\mathbb{E}[\|v_t - \nabla f(w_t)\|^2]$ as t increases to achieve better convergence [1, 2, 3, 4, 5, 6, 7]. SVRG introduces an explicit variance reduction strategy with double loops where s-th outer loop, called s-th *epoch*, has m_s inner iterations. SVRG first keeps $\tilde{w} = w^{s-1}_{m_{s-1}}$ or $\tilde{w} = w^{s-1}_t$ for randomly chosen $t \in \{1, \ldots, m_{s-1}\}$ at the end of (s–1)-th epoch, and also sets the initial value of s-th epoch as $w_0^s = \tilde{w}$. It then computes a full gradient $\nabla f(\tilde{w})$. Subsequently, denoting the selected random index $i \in \{1, \ldots, N\}$ by i_t^s, SVRG randomly picks i_t^s-th sample for each $t \geq 1$ at $s \geq 1$ and computes the *modified stochastic gradient* v_t^s as $v_t^s = \nabla f_{i_t^s}(w_{t-1}^s) - \nabla f_{i_t^s}(\tilde{w}^{s-1}) + \nabla f(\tilde{w}^{s-1})$.

The Grassmann manifold. $\mathrm{Gr}(r, d)$ is the set of r-dimensional linear subspaces in $\mathbb{R}^d$. An element on the Grassmann manifold is represented by a $d \times r$ matrix $\mathbf{U}$ with orthonormal columns, i.e., $\mathbf{U}^T\mathbf{U} = \mathbf{I}$. The Grassmann manifold has the structure of a Riemannian manifold [15, Section 3.4]. Notions such as the Riemannian gradient (first order derivative of a cost function), geodesic (shortest curve), exponential mapping (moving in a straight line), and logarithm mapping (difference between elements) have closed-form expressions [15].

R-SVRG on Grassmann. We denote the Riemannian stochastic gradient for i_t^s-th sample as $\mathrm{grad} f_{i_t^s}(\tilde{\mathbf{U}}^{s-1})$ and the *modified Riemannian stochastic gradient* as ξ_t^s instead of v_t^s to show differences with the Euclidean case. R-SVRG keeps a $\tilde{\mathbf{U}}^{s-1} \in \mathrm{Gr}(r, d)$ after m_{s-1} stochastic update steps of (s–1)-th epoch and computes the full Riemannian gradient $\mathrm{grad} f(\tilde{\mathbf{U}}^{s-1})$ only for this stored $\tilde{\mathbf{U}}^{s-1}$. The algorithm also computes the $\mathrm{grad} f_{i_t^s}(\tilde{\mathbf{U}}^{s-1})$ that corresponds to this i_t^s-th sample. Then, picking i_t^s-th sample for each t-th inner iteration of s-th epoch at $\mathbf{U}_{t-1}^s$, we calculate ξ_t^s in the same way as v_t^s in the Euclidean case, i.e., by modifying $\mathrm{grad} f_{i_t^s}(\mathbf{U}_{t-1}^s)$ using both $\mathrm{grad} f(\tilde{\mathbf{U}}^{s-1})$ and $\mathrm{grad} f_{i_t^s}(\tilde{\mathbf{U}}^{s-1})$. Translating the right-hand side of v_t^s to the manifold $\mathcal{M}$ involves the sum of $\mathrm{grad} f_{i_t^s}(\mathbf{U}_{t-1}^s)$, $\mathrm{grad} f_{i_t^s}(\tilde{\mathbf{U}}^{s-1})$, and $\mathrm{grad} f(\tilde{\mathbf{U}}^{s-1})$, which belong to two tangent spaces $T_{\mathbf{U}_{t-1}^s}\mathcal{M}$ and $T_{\tilde{\mathbf{U}}^{s-1}}\mathcal{M}$. This requires particular attention on a manifold and *parallel translation* provides an adequate and flexible solution to handle multiple elements on two separated tangent spaces. More concretely, $\mathrm{grad} f_{i_t^s}(\tilde{\mathbf{U}}^{s-1})$ and $\mathrm{grad} f(\tilde{\mathbf{U}}^{s-1})$ are parallel-transported to $T_{\mathbf{U}_{t-1}^s}\mathcal{M}$ at the current point $\mathbf{U}_{t-1}^s$. Consequently, ξ_t^s at t-th inner iteration of s-th epoch is set as

$$\xi_t^s = \mathrm{grad} f_{i_t^s}(\mathbf{U}_{t-1}^s) - P^{\mathbf{U}_{t-1}^s \leftarrow \tilde{\mathbf{U}}^{s-1}}(\mathrm{grad} f_{i_t^s}(\tilde{\mathbf{U}}^{s-1})) + P^{\mathbf{U}_{t-1}^s \leftarrow \tilde{\mathbf{U}}^{s-1}}(\mathrm{grad} f(\tilde{\mathbf{U}}^{s-1})), \qquad (2)$$

Algorithm 1 R-SVRG with a fixed step-size.

```
Require: Update frequency $m_s > 0$ and step-size $\eta > 0$.
Initialize $\tilde{\mathbf{U}}^0$.
for $s = 1, 2, \ldots$ do
    Calculate the Riemannian full gradient $\mathrm{grad} f(\tilde{\mathbf{U}}^{s-1})$.
    Store $\mathbf{U}_0^s = \tilde{\mathbf{U}}^{s-1}$.
    for $t = 1, 2, \ldots, m_s$ do
        Choose $i_t^s \in \{1, \ldots, N\}$ uniformly at random.
        Calculate the tangent vector $\zeta$ from $\tilde{\mathbf{U}}^{s-1}$ to $\mathbf{U}_{t-1}^s$ by logarithm mapping.
        Calculate the modified Riemannian stochastic gradient $\xi_t^s$ as
        $\xi_t^s = \mathrm{grad} f_{i_t^s}(\mathbf{U}_{t-1}^s) - P^{\mathbf{U}_{t-1}^s \leftarrow \tilde{\mathbf{U}}^{s-1}} \left(\mathrm{grad} f_{i_t^s}(\tilde{\mathbf{U}}^{s-1}) - \mathrm{grad} f(\tilde{\mathbf{U}}^{s-1})\right)$.
        Update $\mathbf{U}_t^s$ from $\mathbf{U}_{t-1}^s$ as $\mathbf{U}_t^s = \mathrm{Exp}_{\mathbf{U}_{t-1}^s}(-\eta \xi_t^s)$ with the exponential mapping.
    end for
    option I: $\tilde{\mathbf{U}}^s = g_{m_s}(\mathbf{U}_1^s, \ldots, \mathbf{U}_{m_s}^s)$ (or $\tilde{\mathbf{U}}^s = \mathbf{U}_t^s$ for randomly chosen $t \in \{1, \ldots, m_s\}$).
    option II: $\tilde{\mathbf{U}}^s = \mathbf{U}_{m_s}^s$.
end for
```

where $P^{\mathbf{U}_{t-1}^s \leftarrow \tilde{\mathbf{U}}^{s-1}}(\cdot)$ represents a parallel-translation operator from $\tilde{\mathbf{U}}^{s-1}$ to $\mathbf{U}_{t-1}^s$ on the Grassmann manifold. Finally, the update rule of R-SVRG is $\mathbf{U}_t^s = \mathrm{Exp}_{\mathbf{U}_{t-1}^s}(-\eta \xi_t^s)$.

The overall algorithm with a fixed step-size is summarized in Algorithm 1. Additionally, we propose a simple modification of R-SVRG, denoted as R-SVRG+, that uses standard SGD updating only for the first epoch to avoid a bigger overhead at the beginning of the process [17].

For our local convergence rate analysis in **Theorem 3.2** (shown later), we use, as **option I**, the mean value of $\tilde{\mathbf{U}}^s = g_{m_s}(\mathbf{U}_1^s, \ldots \mathbf{U}_{m_s}^s)$ as $\tilde{\mathbf{U}}^s$, where $g_n(\mathbf{U}_1, \ldots, \mathbf{U}_n)$ is the Karcher mean on the Grassmann manifold. Another option is to simply choose $\tilde{\mathbf{U}}^s = \mathbf{U}_t^s$ for $t \in \{1, \ldots, m_s\}$ at random.

3 Main result: convergence analysis

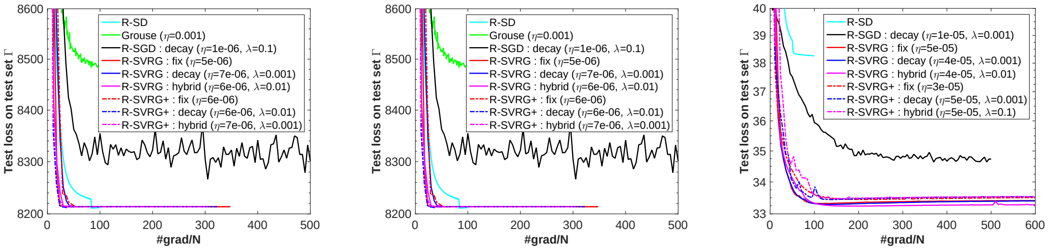
The global convergence analysis (Theorem 3.1) is to guarantee convergence globally to a critical point starting from any initialization point, which is common in a non-convex setting. The local convergence rate analysis (Theorem 3.2), on the other hand, yields a rate in neighborhood of a local minimum. This analysis setting is also very common and standard in manifold optimization. The essential assumptions about Lipschitz smoothness and Hessian are imposed only in this neighborhood. We show that a local linear-convergence rate is achieved under fixed step-size, which is the same as standard SVRG in the Euclidean space for non-convex problems.

We introduce a global convergence result under a *decay step-size* and local convergence rate analysis under a *fixed step-size* setup. Here, we assume that the functions f_n are *β-Lipschitz continuously differentiable* in a local neighborhood in the local convergence analysis.

Theorem 3.1. *Consider* **Algorithm 1** *on a connected Riemannian manifold $\mathcal{M}$ with injectivity radius uniformly bounded from below by $I > 0$. Assume that the sequence of step-sizes $(\eta_t^s)_{m_s \geq t \geq 1, s \geq 1}$ satisfies the condition that $\sum (\eta_t^s)^2 < \infty$ and $\sum \eta_t^s = +\infty$. Suppose there exists a compact set K such that $w_t^s \in K$ for all $t \geq 0$. We also suppose that the gradient is bounded on K, i.e., there exists $A > 0$ such that for all $w_t^s \in K$ and $i_t^s \in Z$ we have $\|\mathrm{grad} f(w_t^s)\| \leq A/3$. Then $f(w_t^s)$ converges a.s. and $\mathrm{grad} f(w_t^s) \to 0$ a.s.*

Theorem 3.2. *Let $\mathcal{M}$ be the Grassmann manifold and $\mathbf{U}^* \in \mathcal{M}$ be a non-degenerate local minimizer of f (i.e., $\mathrm{grad} f(\mathbf{U}^*) = 0$ and the Hessian $\mathrm{Hess} f(\mathbf{U}^*)$ of f at $\mathbf{U}^*$ is positive definite). Assume that there exists a convex neighborhood $\mathcal{U}$ of $\mathbf{U}^* \in \mathcal{M}$ and a positive real number σ such that the smallest eigenvalue of the Hessian of f at each $\mathbf{U} \in \mathcal{U}$ is not less than σ. When each $\mathrm{grad} f_n$ is β-Lipschitz continuously differentiable and $\eta > 0$ is sufficiently small such that $0 < \eta(\sigma - 14\eta\beta^2) < 1$, it then follows that for any sequence $\{\tilde{\mathbf{U}}^s\}$ generated by the algorithm converging to $\mathbf{U}^*$, there exists $K > 0$ such that for all $s > K$,*

$$\mathbb{E}[(\mathrm{dist}(\tilde{\mathbf{U}}^s, \mathbf{U}^*))^2] \quad \leq \quad \frac{4(1 + 8m\eta^2\beta^2)}{\eta m(\sigma - 14\eta\beta^2)} \mathbb{E}[(\mathrm{dist}(\tilde{\mathbf{U}}^{s-1}, \mathbf{U}^*))^2].$$



(a) Test loss (synthetic). (b) Test loss (Jester). (c) Test loss (MovieLens-1M).

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

The local convergence analysis above can be extended to other Riemannian manifolds. In fact, if the curvature is lower bounded and the diameter of domain is upper bounded, the coefficient in Theorem 3.2 is modified. This slight modification yields a quite similar result for general manifolds. In addition, we can also provide a linear convergence rate under the decaying step-sizes although it is worse than the fixed step-size case. This means that we can guarantee global convergence and local linear convergence even if we use decaying step-sizes from beginning to end. Therefore, we could analyze a new step-size switching algorithm between decaying and fixed step-sizes. We empirically evaluate this as *hybrid* step-sizes in Section 4, but its theoretical analysis is a future research work.

4 Numerical comparisons

The matrix completion problem aims at completing an incomplete matrix $\mathbf{X}$ of size $d \times N$ from a small number of entries by assuming a low-rank model for the matrix. If Ω is the set of the known indices in $\mathbf{X}$, then the rank-r matrix completion problem amounts to minimizing $\|\mathcal{P}_\Omega(\mathbf{UA}) - \mathcal{P}_\Omega(\mathbf{X})\|_F^2$, where $\mathbf{U} \in \mathbb{R}^{d\times r}$, $\mathbf{A} \in \mathbb{R}^{r\times N}$, and where the operator $\mathcal{P}_\Omega(\mathbf{X}_{ij}) = \mathbf{X}_{ij}$ if $(i, j) \in \Omega$ and $\mathcal{P}_\Omega(\mathbf{X}_{ij}) = 0$ otherwise. Partitioning $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$, the problem is equivalent to minimizing $\sum_{n=1}^N \|\mathcal{P}_{\Omega_n}(\mathbf{U}\boldsymbol{a}_n) - \mathcal{P}_{\Omega_n}(\boldsymbol{x}_n)\|_2^2/N$, where $\mathbf{U} \in \mathbb{R}^{d\times r}$, $\boldsymbol{a}_n \in \mathbb{R}^r$, and where $\boldsymbol{x}_n \in \mathbb{R}^d$ and the operator $\mathcal{P}_{\Omega_n}$ the sampling operator for the n-th column. Given $\mathbf{U}$, $\boldsymbol{a}_n$ admits a closed-form solution. Finally, the problem only depends on the column space of $\mathbf{U}$, and hence, is on $\mathrm{Gr}(r, d)$ [18].

We compare R-SVRG(+) with R-SGD, R-SD (Riemannian steepest descent algorithm), and Grouse [18]. We consider both *fixed* step-size and *decay* step-size sequence $\eta_k = \eta_0(1 + \eta_0\lambda\lfloor k/m_s\rfloor)^{-1}$ where k is the number of iterations used and η_0 is set through cross validation. We also consider a *hybrid* step-size sequence based on our analyses that follows the decay step-size till $s_{TH} = 5$ epoch, and subsequently switches to a fixed step-size. $m_s = 5N$ is also fixed by following [1], and batch-size is fixed to 10. In all the figures, the x-axis is the computational cost measured by the number of gradient computations divided by N. Algorithms are initialized randomly and are stopped when either the stochastic gradient norm is below 10^{-8} or the number of iterations exceeds 100. We first consider a synthetic dataset with $N = 1000$, $d = 500$, and $r = 5$. Algorithms are initialized randomly as suggested in [18]. This instance considers the loss on a test set Γ, which is different from the training set Ω. The over-sampling ratio (OS) is 5, where the OS determines the number of entries that are known. Figure 1(a) shows the superior performance of our proposed algorithms on Γ. Next, we consider the Jester dataset 1 [19] consisting of ratings of 100 jokes by 24983 users for $r = 5$. Figure 1(b) shows the superior performance of R-SVRG(+) on the test set. The final test compares the algorithms on the MovieLens-1M dataset with a million ratings of 6040 users and 3952 movies. Figure 1(c) shows the results on the test set for all algorithms except Grouse, which faces issues with convergence. Overall, R-SVRG(+) shows much faster convergence than others.

5 Conclusion

We have proposed a Riemannian stochastic variance reduced gradient algorithm (R-SVRG) for problems on the Grassmann manifold. We proved that R-SVRG generates globally convergent sequences with a decay step-size condition and is locally linearly convergent with a fixed step-size under some natural assumptions. Numerical comparisons on the matrix completion problem suggested the superior performance of R-SVRG on various different benchmarks.

References

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

[2] Julien Mairal. Incremental majorization-minimization optimization with application to largescale machine learning. *SIAM J. Optim.*, 25(2):829–855, 2015.

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

[4] S. Shalev-Shwartz and T. Zhang. Proximal stochastic dual coordinate ascent. Technical report, arXiv preprint arXiv:1211.2717, 2012.

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

[6] A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS*, pages 1646–1654, 2014.

[7] Y. Zhang and L Xiao. Stochastic primal-dual coordinate method for regularized empirical risk minimization. *SIAM J. Optim.*, 24(4):2057–2075, 2014.

[8] D. Garber and E. Hazan. Fast and simple PCA via convex optimization. Technical report, arXiv preprint arXiv:1509.05647, 2015.

[9] S. Shalev-Shwartz. SDCA without duality. Technical report, arXiv preprint arXiv:1502.06177, 2015.

[10] Z. Allen-Zhu and Y. Yan. Improved SVRG for non-strongly-convex or sum-of-non-convex objectives. Technical report, arXiv preprint arXiv:1506.01972, 2015.

[11] Z. Allen-Zhu and E. Hazan. Variance reduction for faster non-convex optimization. Technical report, arXiv preprint arXiv:1603.05643, 2016.

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

[13] H. Kasai, H. Sato, and B. Mishra. Riemannian stochastic variance reduced gradient on grassmann manifold. *arXiv preprint: arXiv:1605.07367*, 2016.

[14] H. Zhang, S. J. Reddi, and S. Sra. Fast stochastic optimization on Riemannian manifolds. In *Accepted for publication in NIPS*, 2016.

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

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

[17] J. Konečný and P. Richtárik. Semi-stochastic gradient descent methods. Technical report, arXiv preprint arXiv:1312.1666, 2013.

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

[19] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: a constant time collaborative filtering algorithm. *Inform. Retrieval*, 4(2):133–151, 2001.