Tight Convergence Rate of Gradient Descent for Eigenvalue Computation

Qinghua Ding , Kaiwen Zhou , James Cheng
Department of Computer Science and Engineering, The Chineses University of Hong Kong
{qhding, kwzhou, jcheng}@cse.cuhk.edu.hk

Abstract

Riemannian gradient descent (RGD) is a simple, popular and efficient algorithm for leading eigenvector computation [Absil *et al.*, 2009]. However, the existing analysis of RGD for eigenproblem is still not tight, which is $O\big(\frac{1}{\Delta^2}\ln\frac{n}{\epsilon}\big)$ due to [Xu *et al.*, 2018]. In this paper, we show that RGD in fact converges at rate $O\big(\frac{1}{\Delta}\ln\frac{n}{\epsilon}\big)$, and give instances to show the tightness of our result. This improves the best prior analysis by a quadratic factor. Besides, we also give tight convergence analysis of a deterministic variant of Oja's rule due to [Oja, 1982]. We show that it also enjoys fast convergence rate of $O\big(\frac{1}{\Delta}\ln\frac{n}{\epsilon}\big)$. Previous papers only gave asymptotic characterizations [Oja, 1982; Oja, 1989; Yi *et al.*, 2005]. Our tools for proving convergence results include an innovative reduction and chaining technique, and a noisy fixed point iteration argument. Besides, we also give empirical justifications of our convergence rates over synthetic and real data.

1 Introduction

Leading eigenvector computation has been studied extensively during the past few decades. Power method and Lanczos method have been applied to solve this problem, and their convergence rates have been well justified. In comparison, Riemannian gradient descent (RGD) is another efficient algorithm for this problem, but its convergence rate has not been tightly characterized yet [Absil *et al.*, 2009; Xu *et al.*, 2018]. Given a symmetric and semi-positive definite (PSD) matrix A, Riemannian gradient descent uses the following update rule repeatedly to compute the leading eigenvector:

$$\mathbf{x}_{t+1} = \mathcal{R}(\mathbf{x}_t, -\eta_t \cdot \mathrm{grad} f(\mathbf{x}_t)).$$

Here $\mathcal{R}$ is some retraction over the unit hypersphere, and $\mathrm{grad} f(\mathbf{x})$ is the Riemannian gradient of the Riemannian function $f(\mathbf{x})$. (See formal definitions in Section 3.) An important property of RGD is that it embraces a large family of algorithms via different choices of retraction, stepsize scheme and metric. As suggested by the empirical results in [Xu *et al.*, 2018], Riemannian gradient descent with Cayley transformation as the retraction rule is comparable to the Lanczos algorithms in convergence rate but is even more stable.

However, the convergence analysis for Riemannian methods on leading eigenvector computation is not tight. In [Absil *et al.*, 2009], only asymptotic convergence results are proven. And recently, [Xu *et al.*, 2018] proved the convergence rate of $O\big(\frac{1}{\Delta^2}\ln\frac{n}{\epsilon}\big)$ by Lyapunov analysis using a logarithmic potential function. However, we show that the tight convergence rate is $O\big(\frac{1}{\Delta}\ln\frac{n}{\epsilon}\big)$ and the analysis in [Xu *et al.*, 2018] in fact suffers from a quadratic loss. Our method involves an innovative reduction and chaining based argument that enable us to track the progress much better.

Another popular method for leading eigenvector computation is Oja's rule [Oja, 1982]. We remark that the original algorithm in [Oja, 1982] has an online and a deterministic version which are quite different. The online variant has been well studied [Shamir, 2015], while the convergence rate for the deterministic version remains unknown. We focus on the deterministic version throughout the paper.

In terms of convergence analysis of Oja's rule, only asymptotic convergence results are established using the method of invariant set [Oja, 1989; Yi *et al.*, 2005] . However, we prove that the convergence rate of Oja's rule is in fact $O\big(\frac{1}{\Delta}\ln\frac{n}{\epsilon}\big)$. Our method for proving convergence is based on a two-phase argument: in the first phase we obtain an exponential decay of the tail error; and in the second phase we attain an exponential growth of the principal component, by analyzing a noisy fixed point iteration.

An interesting observation is the connection between RGD, Oja's flow and Oja's rule. We show that RGD and Oja's rule are in fact two different discretizations of Oja's flow.

We summarize our main contributions as follows:

- We use innovative reduction and chaining techniques to show that RGD has tight convergence rate of $O\left(\frac{1}{\Delta}\ln\frac{n}{\epsilon}\right)$ for leading eigenvector computation. This improves the prior analysis [Xu *et al.*, 2018] by a quadratic factor.
- We use a two-phase analysis and a detailed study of a noisy fixed point iteration to establish convergence rate of $O\left(\frac{1}{\Delta}\ln\frac{n}{\epsilon}\right)$ for Oja's rule. The prior analysis [Oja, 1982; Oja, 1989; Yi *et al.*, 2005] only gave asymptotic convergence results.
- We validate our convergence results on both synthetic

and real datasets. The empirical iteration complexity matches with our iteration complexity bound, and is much better than that proved in [Xu *et al.*, 2018].

We remark that these techniques could be applied to analyzing other algorithms for eigenproblems as well.

2 Related Works

Riemannian optimization is a meta-algorithm for solving optimization problems using the inherent structure of the feasible space. [Absil *et al.*, 2009] is an excellent survey on this topic. In the past few decades, Riemannian method has been shown to be a powerful tool for solving optimization problems with orthogonality constraints [Wen and Yin, 2013]. It also has a useful Matlab toolbox [Boumal *et al.*, 2014].

The leading eigenvector computation is one of the most widely studied problems using Riemannian optimization method [Absil *et al.*, 2009]. The asymptotic convergence results for RGD on eigenproblems have been established in a number of papers and books, see [Absil *et al.*, 2009] for example. However, the non-asymptotic convergence rate of this method is still undetermined prior to this work, and the tightest analysis before this work is $O\left(\frac{1}{\Delta^2}\ln\frac{n}{\epsilon}\right)$ due to [Xu *et al.*, 2018].

There are also convergence results for RGD on geodesically convex(g-convex) functions [Zhang and Sra, 2016; Zhang and Sra, 2018]. However, the Rayleigh quotient function is neither g-convex nor g-concave on the hypersphere [Absil *et al.*, 2009]. Thus, the convergence results on g-convex functions is not applicable to it.

The Riemannian method is also called natural gradient descent (NGD) in deep learning society [Amari, 1998], which is further equivalent to mirror descent with certain metric [Raskutti and Mukherjee, 2015]. Recently, Riemannian method is shown to be efficient for training deep neural networks, whose loss surface can be highly non-convex and non-smooth [Li *et al.*, 2018]. Despite the superiority of the Riemannian method, few theoretical justifications have been given so far [Pascanu and Bengio, 2013].

Oja's rule was introduced by Oja for principal component analysis (PCA) in [Oja, 1982]. The deterministic version of Oja's rule, together with its continuous-time analogy called Oja's flow, have been widely studied in the literature since its invention [Chen *et al.*, 1998; Yan *et al.*, 1994; Yi *et al.*, 2005]. [Yan *et al.*, 1994; Chen *et al.*, 1998] proved linear convergence of Oja's flow, and [Yi *et al.*, 2005] proved asymptotic convergence of Oja's rule. However, the non-asymptotic convergence rate of Oja's rule is only known to be at least sublinear prior to this work (see *Table 1* for a summary). A great exposition of this topic can be found in [Helmke and Moore, 2012].

3 Preliminaries and Main Results

We first give the definitions for the terminologies in the Riemannian method. A *manifold* $\mathcal{M}$ is a nonlinear space that is locally isomorphic to the Euclidean space in the neighborhood of each point. For example, the unit hypersphere in the n-dimensional Euclidean space is a submanifold of dimension $n-1$ with the induced metric, and is denoted as $\mathbf{S}^{n-1}$. At each point $\mathbf{x}\in\mathcal{M}$, there is an associated structure, called the *tangent space* $\mathcal{T}_x(\mathcal{M})$, which is also a linear space. For the unit hypersphere, the tangent space is just the tangent plane at some point $\mathbf{x}\in\mathbf{S}^{n-1}$, which can be explicitly expressed as $\mathcal{T}_{\mathbf{x}}(\mathcal{M})=\{\mathbf{y}|\langle\mathbf{x},\mathbf{y}\rangle=1,\mathbf{y}\in\mathbf{R}^n\}$. *Riemannian function* assigns each point on the manifold a real value $f(\mathbf{x})\in\mathbf{R},\forall\mathbf{x}\in\mathcal{M}$.

The leading eigenvector computation considers the problem of finding the leading eigenvector $\mathbf{e}_1$ (or $-\mathbf{e}_1$), given a symmetric and positive definite matrix $A\in\mathbf{R}^{n\times n}$. We denote the eigenpairs of A as $(\lambda_i,\mathbf{e}_i),\forall i\in[n]$, with $\{\lambda_i, i\in[n]\}$ arranged in non-increasing order. We make the standard *eigengap assumption* [Musco and Musco, 2015] that $\lambda_1>\lambda_2$, though it is trivial to generalize it to the case where $\lambda_1=\lambda_2=...=\lambda_p>\lambda_{p+1}$, which is called the general p-th eigengap assumption. The *eigengap* Δ is defined as $\Delta=1-\frac{\lambda_2}{\lambda_1}\in(0,1]$. The leading eigenvector problem is equivalent to the following Riemannian optimization problem [Absil *et al.*, 2009]:

$$\max_{\mathbf{x}\in\mathbf{S}^{n-1}} f(\mathbf{x})=\mathbf{x}^TA\mathbf{x}.$$

Here $f(\mathbf{x})=\mathbf{x}^TA\mathbf{x},\forall\mathbf{x}\in\mathbf{S}^{n-1}$ is a Riemannian function over the unit hypersphere. The *Riemannian gradient* of this function is $\mathbf{g}(\mathbf{x})=(\mathbf{I}-\mathbf{x}\mathbf{x}^T)A\mathbf{x}$, which gives the locally steepest descent direction at $\mathbf{x}$ restricted to the tangent space of $\mathbf{x}$. After updating $\mathbf{x}$ using the gradient by $\mathbf{y}=\mathbf{x}+\eta\mathbf{g}(\mathbf{x})$, we will need a *retraction* that maps the tangent plane to the hypersphere. The most widely used retraction is just the normalization, $\mathcal{R}(\mathbf{x},\eta\mathbf{g}(\mathbf{x}))=\frac{\mathbf{x}+\eta\mathbf{g}(\mathbf{x})}{\|\mathbf{x}+\eta\mathbf{g}(\mathbf{x})\|_2}$. To this end, we have the following RGD update rule for leading eigenvalue computation.

Riemannian gradient descent:

$$\mathbf{g}(\mathbf{x}_t)=(\mathbf{I}-\mathbf{x}_t\mathbf{x}_t^T)A\mathbf{x}_t,\quad \mathbf{x}_{t+1}=\mathcal{R}(\mathbf{x}_t,\eta\mathbf{g}(\mathbf{x}_t)).$$

For leading eigenpair computation, Oja's rule [Yi *et al.*, 2005] is equivalent to Riemannian gradient descent method without retraction. And when $\eta\to0^+$, it evolves to its continuous counterpart, *Oja's flow*, which can be characterized by an ordinary differential equation (ODE) as follows.

$$\begin{aligned}&\textit{Oja's rule}: &&\mathbf{x}_{t+1}=\mathbf{x}_t+\eta\mathbf{g}(\mathbf{x}_t).\\&\textit{Oja's flow}: &&\dot{\mathbf{x}}(t)=\mathbf{g}(\mathbf{x}(t)).\end{aligned}$$

Here $\dot{\mathbf{x}}$ denotes the derivative of $\mathbf{x}$ with respect to t, or equivalently, $\frac{d\mathbf{x}}{dt}$. Through the lens of numerical integration [Hairer *et al.*, 2006], it turns out that Oja's rule and RGD are essentially two different discrete integration algorithm for Oja's flow. Using forward Euler method to discretize Oja's flow gives us Oja's rule, while using the structure-preserving method to discretize it gives us the RGD algorithm (See *Figure 1*).

Before we delve into the technical proofs, we present our main results and briefly discuss the techniques we use. Our first convergence result is the convergence rate of RGD, which is quadratically better than the previous result [Xu *et al.*, 2018].

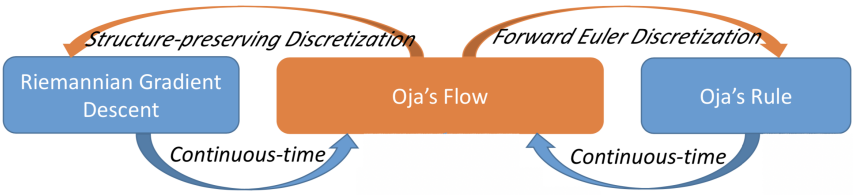

Figure 1: Connections between Riemannian gradient descent, Oja's flow, Oja's rule.

Theorem 1 (Convergence of RGD)**.** *Under random initialization over the unit hypersphere, RGD with stepsize* $\eta = \frac{1}{\lambda_1}$ *converges to* $\boldsymbol{e}_1$ *(or* $-\boldsymbol{e}_1$*), and it achieves precision* $\epsilon \in (0,1)$ *such that* $\|\boldsymbol{x}_t - \boldsymbol{e}_1\|_2 \le \epsilon$ *(or* $\|\boldsymbol{x}_t + \boldsymbol{e}_1\|_2 \le \epsilon$*) in* $O\big(\frac{1}{\Delta}\ln\frac{n}{\epsilon}\big)$ *iterations with high probability.*

Our analysis for RGD uses innovative reduction and chaining techniques. These new techniques enable us to track the convergence procedure step-by-step without much loss and allows us to obtain a tighter bound than [Xu *et al.*, 2018]. We also provide concrete instances to show that our characterization is essentially tight.

Our second convergence result is the convergence rate of Oja's rule as the following. We remark that the previous analysis [Oja, 1982; Oja, 1989; Yi *et al.*, 2005] only gave asymptotic convergence results for this algorithm.

Theorem 2 (Convergence of Oja's rule)**.** *Under random initialization over the unit hypersphere, Oja's rule with stepsize* $\eta = \frac{1}{4\lambda_1}$ *converges to* $\boldsymbol{e}_1$ *(or* $-\boldsymbol{e}_1$*), and it achieves precision* $\epsilon \in (0,1)$ *such that* $\|\boldsymbol{x}_t - \boldsymbol{e}_1\|_2 \le \epsilon$ *(or* $\|\boldsymbol{x}_t + \boldsymbol{e}_1\|_2 \le \epsilon$*) in* $O\big(\frac{1}{\Delta}\ln\frac{n}{\epsilon}\big)$ *iterations with high probability.*

Since no explicit retraction is used in Oja's rule, the convergence in L_2 norm is non-trivial. We prove this result using the invariant set technique [Yi *et al.*, 2005] and a detailed analysis of a noisy fixed-point iteration.

It is important to point out that Oja's flow was shown to converge at rate $O\big(\frac{1}{\lambda_1\Delta}\ln\frac{n}{\epsilon}\big)$[Chen *et al.*, 1998]. However, the analysis heavily depends on the ODE characterization that allows a smooth calculation since it is continuous in time. However, the recurrence in the discrete algorithms can be extremely complex to analyze even just for a few iterations. Thus, we need to develop fundamentally new techniques for proving convergence rates of the discrete algorithms.

The rest of the paper is organized as follows. In Section 4 we present our proof for the convergence rate of RGD (*Theorem 1*). In Section 5, we continue to prove the convergence rate of Oja's rule (*Theorem 2*). We conclude the paper in Section 6.

4 Riemannian Gradient Descent

Our analysis for RGD mainly consists of two phases. First, we reduce the dimension of the convergence procedure from n to 2 for a single step and further for multiple steps, without compromising on the convergence rate (*Lemma 1*). Then, we investigate the basic 2-dimensional case and give the convergence guarantee shown in *Theorem 1*. Here we denote $\angle(\mathbf{a},\mathbf{b})$ as the angle between vectors $\mathbf{a}$ and $\mathbf{b}$.

Lemma 1 (Multiple-step reduction)**.** *If we take* $\eta \le \frac{1}{\lambda_1}$*, then for two sequences* $\{\boldsymbol{x}_t\}$ *and* $\{\boldsymbol{y}_t\}$ *generated by RGD with different initial points,* $\boldsymbol{x}_0 = \sum_{i=1}^n q_i\boldsymbol{e}_i \in \mathbf{S}^{n-1}$*, and* $\boldsymbol{y}_0 = q_1\boldsymbol{e}_1 + \sqrt{1-q_1^2}\boldsymbol{e}_2$*, we have* $\sin^2\angle(\boldsymbol{x}_t,\boldsymbol{e}_1) \le \sin^2\angle(\boldsymbol{y}_t,\boldsymbol{e}_1), \forall t \ge 0$.

Proof of Lemma 1. We first prove the result for a single step. To simplify the notation, we denote $\mathbf{x}' = \mathcal{R}(\mathbf{x}, \eta\mathbf{g}(\mathbf{x}))$ as the one-step update of $\mathbf{x} \in \mathbf{S}^{n-1}$, here $\mathbf{x} = \sum_{i=1}^n q_i\mathbf{e}_i$ is the spectrum decomposition of $\mathbf{x}$. We let $\mathbf{y} = q_1\mathbf{e}_1 + \sqrt{1-q_1^2}\mathbf{e}_2$, which is a counterpart of $\mathbf{x}$ with all the non-principal components concentrated on the second dimension. And we denote $\mathbf{y}' = \mathcal{R}(\mathbf{y}, \eta\mathbf{g}(\mathbf{y}))$ to be the one-step update of $\mathbf{y}$ accordingly.

Consider one-step update as $\mathbf{x}' = \mathcal{R}(\mathbf{x}, \eta\mathbf{g}(\mathbf{x})) = \sum_{i=1}^n q_i'\mathbf{e}_i$. Then component-wise, we have

$$q_1'^2 = \frac{(1+\eta\lambda_1 - \eta(\sum_{j=1}^n \lambda_j q_j^2))^2 q_1^2}{\sum_{i=1}^n (1+\eta\lambda_i - \eta(\sum_{j=1}^n \lambda_j q_j^2))^2 q_i^2}.$$

It is conceptually easier to proceed the proof by thinking of $(q_1^2, q_2^2, ...q_n^2)$ as a discrete probability distribution. Thus, we denote $p_i = q_i^2, \forall i \in [n]$ and we have $\mathbf{p} = (p_1, ..., p_n) \in \Gamma^{n-1}$, where $\Gamma^{n-1} \triangleq \{\mathbf{p} \ge \mathbf{0} \,|\, \sum_{i=1}^n p_i = 1\}$ is the $(n-1)$-dimensional probability simplex. Our analysis and notations will benefit from this point of view of probability distribution. We further denote $\eta(\lambda_1 - \lambda_i) \triangleq \chi_i$, thus $\chi_1 = 0, \chi_2 = \eta(\lambda_1 - \lambda_2)$, and $\chi_2 \le \chi_i \le \eta\lambda_1, \forall i \ge 2$.

Now we consider the following function $g(\mathbf{p})$, which characterizes the increment of the principal component p_1 by $p_1' = g(\mathbf{p}) \cdot p_1$,

$$g(\mathbf{p}) = \frac{(1+(\sum_{j=1}^n \chi_j p_j))^2}{\sum_{i=1}^n (1-\chi_i + (\sum_{j=1}^n \chi_j p_j))^2 p_i}, \forall \mathbf{p} \in \Gamma^{n-1}.$$

We create a random variable χ that follows the distribution $P(\chi = \chi_i) = p_i$. Then we know $\mathbf{E}(\chi) = \sum_{i=1}^n \chi_i p_i$ and

$$g(\mathbf{p}) = \frac{(1+\mathbf{E}(\chi))^2}{\sum_{i=1}^n (1-\chi_i+\mathbf{E}(\chi))^2 p_i} = \frac{(1+\mathbf{E}(\chi))^2}{1+\mathrm{Var}(\chi)}.$$

Here we used the equation $\mathbf{E}(1-\chi+\mathbf{E}(\chi))^2 = 1+\mathbf{E}(\chi^2)-\mathbf{E}(\chi)^2 = 1+\mathrm{Var}(\chi)$. Then it suffices to study the extremal property of this function, which gives a lower bound on the increment of the principal component and thus a lower bound for the convergence rate.

Note that $\sum_{i=1}^n p_i = 1$, then we can substitute it into $g(\mathbf{p})$ to eliminate p_n. We then consider the following restricted functions over $\bar{\Gamma}^{n-1} = \{\bar{\mathbf{p}} \ge \mathbf{0}, \,|\sum_{i=1}^{n-1} \bar{p}_i \le 1\}$. Note that

$\bar{\Gamma}^{n-1}$ is not a probability simplex, but the projection of Γ^{n-1} onto the first $n-1$ dimensions.

$$\bar{g}(\bar{\mathbf{p}}) = g\Big(\bar{\mathbf{p}}, 1 - \sum_{i=1}^{n-1} p_i\Big);$$
$$\mathbf{E}(\chi) = \sum_{i=1}^{n-1} p_i\chi_i + \Big(1 - \sum_{i=1}^{n-1} p_i\Big)\chi_n;$$
$$\mathbf{E}(\chi^2) = \sum_{i=1}^{n-1} p_i\chi_i^2 + \Big(1 - \sum_{i=1}^{n-1} p_i\Big)\chi_n^2.$$

Also note that $\text{Var}(\chi) = \mathbf{E}(\chi^2) - \mathbf{E}(\chi)^2$. Then we claim that for stepsize $\eta \le \frac{1}{\lambda_1}$, we always have $\frac{\partial}{\partial p_2}\bar{g}(\bar{\mathbf{p}}) \le 0$. To see this, we take the derivative of $\bar{g}(\bar{\mathbf{p}})$ w.r.t. p_2 and obtain

$$\frac{\partial}{\partial p_2}\bar{g}(\bar{\mathbf{p}}) = \frac{(1+\mathbf{E}(\chi))}{(1+\text{Var}(\chi))^2}\big(2(1+\text{Var}(\chi))\cdot\frac{\partial}{\partial p_2}\mathbf{E}(\chi) - (1+\mathbf{E}(\chi))\cdot\frac{\partial}{\partial p_2}\text{Var}(\chi)\big).$$

Taking the derivative of $\mathbf{E}(\chi)$ and $\text{Var}(\chi)$ w.r.t. p_2 gives us $\frac{\partial}{\partial p_2}\mathbf{E}(\chi) = \chi_2 - \chi_n$ and $\frac{\partial}{\partial p_2}\text{Var}(\chi) = (\chi_2^2 - \chi_n^2) - 2(\chi_2 - \chi_n)\mathbf{E}(\chi)$. Substituting these derivatives into that of $\bar{g}(\bar{\mathbf{p}})$ gives

$$\frac{\partial}{\partial p_2}\bar{g}(\bar{\mathbf{p}}) = -\frac{(1+\mathbf{E}(\chi))(\chi_n-\chi_2)}{(1+\text{Var}(\chi))^2}\big(2\mathbf{E}(\chi^2) + (1+\mathbf{E}(\chi))(2-\chi_2-\chi_n)\big).$$

Since $-\frac{(1+\mathbf{E}(\chi))(\chi_n-\chi_2)}{(1+\text{Var}(\chi))^2} \le 0$, we have $\frac{\partial}{\partial p_2}\bar{g}(\bar{\mathbf{p}}) \le 0$ when $2\mathbf{E}(\chi^2) + (1+\mathbf{E}(\chi))(2-\chi_2-\chi_n) \ge 0$. But since $\eta \le \frac{1}{\lambda_1}$, we have $\chi_i \le 1, \forall i \in [n]$ and thus $2-\chi_2-\chi_n \ge 0$. Thus our claim is correct, and hence the minima should satisfy the boundary condition of $\bar{\Gamma}^{n-1}$ as $p_2 = 1 - p_1 - \sum_{i=3}^{n-1} p_i$, or equivalently, $p_n = 0$.

In other words, we have proved that when $\eta \le \frac{1}{\lambda_1}$, the increment of principal component satisfies $g(\mathbf{p}) \ge g(p_1, (p_2 + p_n), p_3, ..., p_{n-1}, 0)$. Using similar arguments over p_{n-1}, ..., p_3, we can show that $g(\mathbf{p}) \ge g(p_1, \sum_{i=2}^{n} p_i, 0, ..., 0) = g(p_1, 1-p_1, 0, ..., 0)$. This means that the increment of the principal component attains minima when all the weights $p_2, p_3, ..., p_n$ are concentrated on the second leading component. Thus when $\eta \le \frac{1}{\lambda_1}$, we have

$$g(\mathbf{p}) \ge \frac{(1+\chi_2(1-p_1))^2}{1+(1-p_1)\chi_2^2 - (1-p_1)^2\chi_2^2}.$$

This together with the fact that $\sin^2\angle(\mathbf{x}', \mathbf{e}_1) = 1 - g(\mathbf{p})p_1$ gives

$$\sin^2\angle(\mathbf{x}', \mathbf{e}_1) \le \frac{(1-\chi_2 p_1)^2}{1+p_1(1-p_1)\chi_2^2}\cdot\sin^2\angle(\mathbf{x}, \mathbf{e}_1).$$

Note that the right hand side is in fact equivalent to $\sin^2\angle(\mathbf{y}', \mathbf{e}_1)$, then we get $\sin^2\angle(\mathbf{x}', \mathbf{e}_1) \le \sin^2\angle(\mathbf{y}', \mathbf{e}_1)$. This proves the lemma for a single step. Now we critically use a chaining argument to show that this is also true for multiple steps.

With some abuse on notations, suppose we have an array $\{\mathbf{p}(t) = (p_1(t), ..., p_n(t))\}$ produced by the RGD algorithm above. Then we have

$$p_1(t+1) \ge g(p_1(t), 1-p_1(t), 0, ..., 0)\cdot p_1(t).$$

We further construct another *virtual sequence* $\{\mathbf{r}(t) = (r_1(t), ..., r_n(t))\}$, which is also generated by the gradient process but with a different starting point $r_1(0) = p_1(0)$, $r_2(0) = 1 - p_1(0)$, $r_i(0) = 0, \forall 3 \le i \le n$. Then $\mathbf{r}(t), \forall t \ge 0$ will only have nonzero values for its first two components. And we have the recurrence for $\mathbf{r}(t)$ as

$$r_1(t+1) = g(r_1(t), 1-r_1(t), 0, ..., 0)\cdot r_1(t).$$

To establish our reduction for multiple steps, it suffices to prove that $p_1(t) \ge r_1(t), \forall t \ge 0$. Note that we have validated this inequality for $t = 0$ by *Lemma 1*, since $p_1(1) = g(\mathbf{p}(0))\cdot p_1(0) \ge g(p_1(0), 1-p_1(0), 0, ..., 0)\cdot p_1(0) = r_1(1)$.

In the following, we will prove that $p_1(t) \ge r_1(t), \forall t \ge 0$ by induction. The induction base is that $\forall t \le k$, we have $p_1(t) \ge r_1(t)$. In fact, if the function $h(p) = p\cdot g(p, 1-p, 0, ..., 0), \forall p \in [0,1]$ is monotonically increasing, then by induction base we have

$$\begin{aligned} p_1(t+1) &\ge g(p_1(t), 1-p_1(t), 0, ..., 0)\cdot p_1(t) \\ &\ge g(r_1(t), 1-r_1(t), 0, ..., 0)\cdot r_1(t) = r_1(t+1), \end{aligned}$$

and this concludes our proof. The fact that $h(p)$ is a monotonic function can be validated via its derivative:

$$\frac{\mathrm{d}}{\mathrm{d}p}h(p) = \frac{(1+\chi_2(1-p))}{(1+p(1-p)\chi_2^2)^2}\big((1-\chi_2 p)^2 + \chi_2(1-p)(1-\chi_2^2 p^2)\big).$$

Note that for $\chi_2, p \in [0,1]$, we have $\frac{\mathrm{d}}{\mathrm{d}p}h(p) \ge 0$. Thus, $h(p)$ is indeed monotone, which establishes our previous statements. In other words, if $\mathbf{x}_0 = \sum_{i=1}^{n} q_i\mathbf{e}_i$, and $\mathbf{y}_0 = q_1\mathbf{e}_1 + \sqrt{1-q_1^2}\mathbf{e}_2$, then after t iterations, we will have $\sin^2\angle(\mathbf{x}_t, \mathbf{e}_1) = 1 - p_1(t) \le 1 - r_1(t) = \sin^2\angle(\mathbf{y}_t, \mathbf{e}_1)$. This proves the lemma. □

Since the worst-case convergence rate occurs when the initial vector lies in the span of the first and second principal dimensions due to *Lemma 1*, it suffices to lower bound the rate for the two-dimensional case. By putting all these arguments together, it is straight-forward to establish *Theorem 1* as follows.

Proof of Theorem 1. By *Lemma 1*, we have $\sin^2\angle(\mathbf{x}_t, \mathbf{e}_1) \le \sin^2\angle(\mathbf{y}_t, \mathbf{e}_1), \forall t \ge 0$. Since the series of $\mathbf{y}_t$ only remains in the two-dimensional space of $span\{\mathbf{e}_1, \mathbf{e}_2\}$, in essence its convergence follows the case where $d = 2$.

Here we consider the convergence property of the basic 2-dimensional case. Denote $\theta_t \triangleq \angle(\mathbf{x}_t, \mathbf{e}_1)$, then we have $q_1(t) = |\cos\theta_t|, q_2(t) = |\sin\theta_t|$. And we have

$$\tan\theta_{t+1} = \left(1 - \frac{\chi_2}{1+\chi_2\frac{\tan^2\theta_t}{1+\tan\theta_t^2}}\right)\cdot\tan\theta_t.$$

Since we have $\frac{\tan^2\theta_t}{1+\tan^2\theta_t} \in [0,1)$, then $1+\chi_2\frac{\tan^2\theta_t}{1+\tan^2\theta_t} < 1+\chi_2$. Thus

$$|\tan\theta_t| \le \frac{|\tan\theta_t|}{1+\chi_2} \Leftrightarrow \tan^2\theta_t \le \frac{\tan^2\theta_t}{(1+\chi_2)^2}.$$

By simple chaining, we have $\tan^2\theta_t \le \frac{\tan^2\theta_0}{(1+\chi_2)^{2t}} \to 0$. Note that $\mathrm{sgn}(\tan(\theta_t)) = \mathrm{sgn}(\tan(\theta_0))$. Thus, we conclude that either $\theta_t \to 0$ or $\theta_t \to \pi$, which corresponds to the cases where $\mathbf{x}_t$ converges to $\mathbf{e}_1$ and $-\mathbf{e}_1$. Without loss of generality, we consider the case where $\theta_t \to 0$, then we can bound $\sin^2\theta_t$ as

$$\sin^2\theta_t \le \tan^2\theta_t \le \frac{\tan^2\theta_0}{(1+\chi_2)^{2t}}.$$

Thus when $\eta \le \frac{1}{\lambda_1}$, with high probability over random initialization over the hypersphere, we have $\tan^2\theta_0 = \Theta(n)$, and hence

$$\|\mathbf{x}_t - \mathbf{e}_1\|_2^2 = \sin^2\theta_t \le O(n(1+\chi_2)^{-2t}) \le O(ne^{-2\chi_2 t}).$$

Taking $\eta = \frac{1}{\lambda_1}$, we have $\chi_2 = \Delta$ and for $t = O\Big(\frac{1}{\Delta}\ln\frac{n}{\epsilon}\Big)$, the required precision is attained. □

Remark 1 (Tightness of the analysis)**.** *We show that our analysis is essentially tight by considering the concrete instance where $\{\boldsymbol{x}_t\}$ is defined by the RGD process with starting point $\boldsymbol{x}_0 = \frac{1}{\sqrt{2}}(\boldsymbol{e}_1+\boldsymbol{e}_2)$, then we have $\tan\theta_t \ge (1-\Delta)^t\tan\theta_0 = \Omega(ne^{-2\Delta t})$ with high probability. Thus it requires $\Omega\big(\frac{1}{\Delta}\ln\big(\frac{n}{\epsilon}\big)\big)$ iterations to converge in the least.*

5 Oja's Rule

The difference between Oja's rule and RGD lies in the retraction step. Since no explicit retraction is enforced during the execution of Oja's rule, the convergence behavior is very different from that of RGD, and the analysis also needs different techniques.

To give our non-asymptotic analysis for L_2 convergence, we first review the following lemma previously established in [Yi *et al.*, 2005], using the invariant set technique.

Lemma 2 (Invariant set property [Yi *et al.*, 2005])**.** *If we let $\eta \le \frac{1}{2\lambda_1}$ in Oja's rule, then we always have $\boldsymbol{x}_t A\boldsymbol{x}_t \le \frac{1}{\eta}$ throughout the process, and $\tan^2\theta_t$ decreases as follows.*

$$\tan^2\theta_t \le \tan^2\theta_0\left(\frac{1+\eta\lambda_2}{1+\eta\lambda_1}\right)^{2t}.$$

We will also use the following version of fixed point iteration theorem.

Proposition 1 (Fixed point convergence theorem)**.** *Let g and g' be continuous on $[a,b]$ and suppose that if $a \le x \le b$ then $a \le g(x) \le b$. Also suppose that $\lambda = \max_{a\le x\le b}|g'(x)| < 1$. Then:*

1. *There exists a unique solution $\alpha \in [a,b]$ to the equation $x = g(x)$.*
2. *For any initial estimation $x_0 \in [a,b]$, we have $|x_n - \alpha| \le \lambda|x_{n-1}-\alpha|$, and $\lim_{n\to\infty} x_n = \alpha$.*

Our proof for *Theorem 2* mainly studies the convergence property of a noisy fixed point iteration.

Proof of Theorem 2. Due to *Lemma 2* and $\tan^2\theta_t = \frac{\sum_{i=2}^n q_i^2(t)}{q_1^2(t)}$, we have

$$\sum_{i=2}^n q_i^2(t) \le q_1^2(t)\tan^2\theta_0\Big(1-\frac{\eta(\lambda_1-\lambda_2)}{1+\eta\lambda_1}\Big)^{2t}.$$

And note that $\lambda_1 q_1^2(t) \le \mathbf{x}_t^T A\mathbf{x}_t \le \frac{1}{\eta}$, thus $q_1^2(t) \le \frac{1}{\eta\lambda_1}$. Taking $\eta\lambda_1 = \frac{1}{4}$, then we have

$$\sum_{i=2}^n q_i^2(t) \le 4\tan^2\theta_0\Big(1-\frac{\Delta}{5}\Big)^{2t} \le 4\tan^2\theta_0\exp\Big(-\frac{2\Delta t}{5}\Big).$$

Thus, to achieve tail error $\sum_{i=2}^n q_i^2(t) \le \delta$, it suffices to assure

$$4\tan^2\theta_0\exp\Big(-\frac{2\Delta t}{5}\Big) \le \delta \Leftrightarrow t \ge \frac{5}{2\Delta}\ln\frac{4\tan^2\theta_0}{\delta}.$$

And after $t_1 = O\Big(\frac{1}{\Delta}\ln\frac{\tan\theta_0}{\delta}\Big)$ iterations, we always have tail error bounded by δ. Suppose that we have already ran t_1 iterations to reduce the tail error to some $\delta \le \frac{1}{4}$, then we have $\sum_{i=2}^n \lambda_i q_i^2(t) \le \lambda_1\sum_{i=2}^n q_i^2(t) \le \lambda_1\delta$. It is not hard to show that Oja's rule updates $q_1(t)$ as follows,

$$q_1(t+1) = \Big(\frac{5}{4} - \eta\sum_{i=1}^n \lambda_i q_i^2(t)\Big)q_1(t).$$

Note that $q_1(t)$ preserves the sign throughout the process. Thus, w.l.o.g. we assume $q_1(t) \ge 0, \forall t \ge 0$. And thus

$$\frac{1}{4}\big(5-q_1^2(t)-\delta\big)q_1(t) \le q_1(t+1) \le \frac{1}{4}\big(5-q_1^2(t)\big)q_1(t).$$

Then we study the convergence of the fixed point iterations $g_\zeta(x) = \frac{1}{4}(5-\zeta-x^2)x$, where $\zeta \in [0,\delta]$. Since $q_1^2 \le \frac{1}{\eta\lambda_1} = 4$, it suffices to study this fixed point iteration over $(0,2]$. We heve the first-order and second-order derivatives of $g_\zeta(x)$ as $g'_\zeta(x) = \frac{1}{4}(5-\zeta-3x^2)$ and $g''_\zeta(x) = -\frac{3}{2}x$.

Thus $g_\zeta(x)$ is concave in $(0,+\infty)$. And since the zero of $g'_\zeta(x)$ on $(0,+\infty)$ is $s_\zeta = \sqrt{\frac{5-\zeta}{3}}$, we have $g_\zeta(x)$ increasing with ζ in $(0,s_\zeta]$, and decreasing in $(s_\zeta,+\infty)$. To guarantee that $g_\zeta(x)$ is positive in interval $(0,2]$, it suffices to let $g_\zeta(2) = \frac{1}{2}(1-\zeta) > 0$, and this is equivalent to $\zeta < 1$.

Denote $t_\zeta = g_\zeta(s_\zeta) = \frac{5-\zeta}{6}\cdot s_\zeta$, thus for any $x \in (0,2]$, a single update is guaranteed to bring it to $(0,t_\zeta]$ and get trapped within it. But $t_\zeta \le t_0 = \frac{5}{6}\sqrt{\frac{5}{3}} \simeq 1.076 < s_\zeta$. Thus, w.l.o.g., we may only consider the interval $(0,t_0]$. And note the fixed point of $g_\zeta(x)$ in $(0,t_0]$ is $r_\zeta = \sqrt{1-\zeta}$.

By *Proposition 1*, since $g'_\zeta(0) = \frac{1}{4}(5-\zeta) > 1$, thus $x = 0$ cannot be a stable fixed point; and since $g'_\zeta(r_\zeta) = \frac{1}{2}(1+\zeta) < 1$, r_ζ is a stable fixed point. Summarizing our analysis above, over the interval $(0,t_0]$, the function family

Scenarios	Oja's flow	Riemannian GD	Oja's rule
$d = 1$	$O\big(\frac{1}{\lambda_1 \Delta} \ln \frac{n}{\epsilon}\big)$ (*Folklore*)	$O\left(\frac{1}{\Delta} \ln \frac{n}{\epsilon}\right)$ (*Thm 1*)	$O\left(\frac{1}{\Delta} \ln \frac{n}{\epsilon}\right)$ (*Thm 2*)
$1 < d < n$	$O\big(\frac{1}{\lambda_d \Delta_d} \ln \frac{1}{\epsilon}\big)$[Chen *et al.*, 1998]	$O(\frac{1}{\epsilon})$(*Folklore*)	$O(\frac{1}{\epsilon})$(*Folklore*)

Table 1: Convergence results.

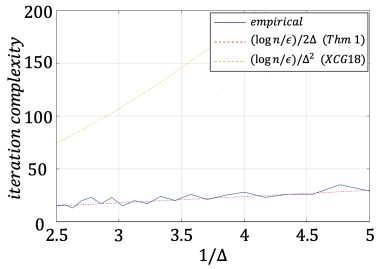

Figure 2: Synthetic data with $\Delta \geq 0.2$.

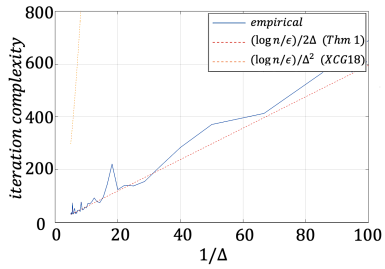

Figure 3: Synthetic data with $\Delta \leq 0.2$.

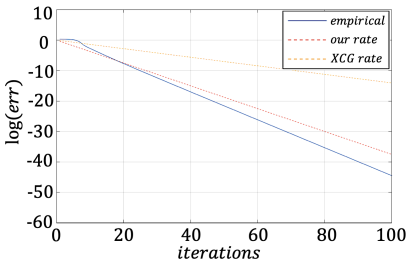

Figure 4: On **Schenk** dataset.

$\{g_\zeta(x), \forall \zeta \in [0, \delta]\}$ is concave, increasing and positive, and each function have fixed point r_ζ. For $x \in (0, r_\zeta]$, we have $g_\zeta(x) \geq x$; and for $x \in (r_\zeta, t_0]$, we have $g_\zeta(x) \leq x$.

After studying the property of $g_\zeta(x)$, we now turn to bound the convergence of a noisy fixed point iteration, where in each iteration t, we are given a different $\zeta(t) \in [0, \delta]$, and update according to $x_t = g_{\zeta(t)}(x_{t-1})$. Specially, setting $\zeta(t) = \sum_{i=2}^n \lambda_i q_i^2(t)/\lambda_1$ recovers the original Oja's rule.

By *Proposition 1* and detailed case analysis[1], we claim that if we set $\delta = \epsilon \leq \frac{1}{4}$, then the worst case iteration complexity is $O\big(\max\{\ln \frac{1}{x_0}, \ln \frac{1}{\epsilon}\}\big)$. Putting together the tail error reduction and the noisy fixed point iteration gives us the final convergence rate as

$$O\left(\max\left\{\frac{1}{\Delta} \ln \frac{\tan\theta_0}{\epsilon},\ \ln \frac{1}{q_1},\ \ln \frac{1}{\epsilon}\right\}\right).$$

Then with high probability, we have $\tan\theta_0 = \Theta(\sqrt{n})$ and $q_1 = \Theta(1/n)$ under random initialization over the hypersphere. This proves the theorem. □

We summarize the convergence properties for RGD and Oja's rule in *Table 1* for reference. We remark that the sublinear rate $O\big(\frac{1}{\epsilon}\big)$ when $1 < d < n$ is not supposed to be tight in general.

6 Experiments

We re-implement experiments from [Xu *et al.*, 2018] for validation of our results. Our setting is exactly the same as [Xu *et al.*, 2018], with $n = 1000$ and $A = U\Sigma U^T$. Here U is a random orthonormal matrix and $\Sigma = [\Sigma_1\ \Sigma_2]$. Further, we have $\Sigma_2 = [\frac{|g_1|}{n}, ..., \frac{|g_{n-6}|}{n}]$ for $g_i \sim \mathcal{N}(0, 1)$; and $\Sigma_1 = [1, 1-\Delta, 1-1.1\Delta, ..., 1-1.4\Delta]$ for certain Δ. We refer readers to [Xu *et al.*, 2018] for detailed parameter settings. Here we let $\epsilon = e^{-5} \simeq 0.0067$ be the desired precision, and compare the empirical iteration complexity to attain this precision compared with the iteration complexity bound of ours, $m_1 = \frac{1}{2\Delta} \ln \frac{n}{\epsilon} \simeq 5.954 \cdot \frac{1}{\Delta}$, and that of [Xu *et al.*, 2018], $m_2 = \frac{1}{\Delta^2} \ln \frac{n}{\epsilon} \simeq 11.908 \cdot \big(\frac{1}{\Delta}\big)^2$.

As shown by the experiment results, the empirical iteration complexity linearly depends on $\frac{1}{\Delta}$, which is well characterized by our iteration complexity bound. However, the quadratic bound of [Xu *et al.*, 2018] is already much higher than the empirical complexity when $\Delta \geq 0.2$ (*Figure 2*); and this gap becomes huge when $\Delta \leq 0.2$ (*Figure 3*).

Moreover, we also validate our convergence results on the **Schenk**[2] dataset used by [Xu *et al.*, 2018] for fair comparision. **Schenk** dataset provides a 10,728 × 10,728 PSD sparse matrix with 85,000 non-zeros in it. The eigengap of this matrix is roughly $\Delta = 0.39$. And our convergence bound in *Theorem 1* gives $\epsilon \leq O(ne^{-2\Delta t})$, thus $\ln(\epsilon) \propto -\Delta t$; while [Xu *et al.*, 2018] gives $\ln(\epsilon) \propto -\Delta^2 t$. As shown in *Figure 4*, our convergence bound is much closer to empirical rate than theirs. This also justify the tightness of our analysis.

7 Conclusions

In this paper, we proved tight convergence rate of $O(\frac{1}{\Delta} \ln \frac{n}{\epsilon})$ for RGD and Oja's rule, for leading eigenvector computation. Our methods for proving convergence mainly involves a special reduction and chaining technique, together with a noisy fixed point iteration argument. We believe these methods will also be useful when analyzing other algorithms for eigenproblems.

Acknowledgments

We thank the reviewers for their valuable comments. Qinghua Ding would like to thank Prof. Lifeng Sun for giving nice suggestions on this work. This work was partially supported GRF 14208318 from the RGC and ITF 6904945 from the ITC of HKSAR, and the National Natural Science Foundation of China (NSFC) (Grant No. 61672552).

[1] Not presented here due to limited space. Please refer to the full version of this paper.

[2] https://sparse.tamu.edu/

References

[Absil *et al.*, 2009] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[Amari, 1998] Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

[Boumal *et al.*, 2014] Nicolas Boumal, Bamdev Mishra, P-A Absil, and Rodolphe Sepulchre. Manopt, a matlab toolbox for optimization on manifolds. *The Journal of Machine Learning Research*, 15(1):1455–1459, 2014.

[Chen *et al.*, 1998] Tianping Chen, Yingbo Hua, and Wei-Yong Yan. Global convergence of oja's subspace algorithm for principal component extraction. *IEEE Transactions on Neural Networks*, 9(1):58–67, 1998.

[Hairer *et al.*, 2006] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, volume 31. Springer Science & Business Media, 2006.

[Helmke and Moore, 2012] Uwe Helmke and John B Moore. *Optimization and dynamical systems*. Springer Science & Business Media, 2012.

[Li *et al.*, 2018] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, pages 6389–6399, 2018.

[Musco and Musco, 2015] Cameron Musco and Christopher Musco. Randomized block krylov methods for stronger and faster approximate singular value decomposition. In *Advances in Neural Information Processing Systems*, pages 1396–1404, 2015.

[Oja, 1982] Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3):267–273, 1982.

[Oja, 1989] Erkki Oja. Neural networks, principal components, and subspaces. *International journal of neural systems*, 1(01):61–68, 1989.

[Pascanu and Bengio, 2013] Razvan Pascanu and Yoshua Bengio. Revisiting natural gradient for deep networks. *arXiv preprint arXiv:1301.3584*, 2013.

[Raskutti and Mukherjee, 2015] Garvesh Raskutti and Sayan Mukherjee. The information geometry of mirror descent. *IEEE Transactions on Information Theory*, 61(3):1451–1457, 2015.

[Shamir, 2015] Ohad Shamir. A stochastic pca and svd algorithm with an exponential convergence rate. In *International Conference on Machine Learning*, pages 144–152, 2015.

[Wen and Yin, 2013] Zaiwen Wen and Wotao Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142(1-2):397–434, 2013.

[Xu *et al.*, 2018] Zhiqiang Xu, Xin Cao, and Xin Gao. Convergence analysis of gradient descent for eigenvector computation. International Joint Conferences on Artificial Intelligence Organization, 2018.

[Yan *et al.*, 1994] Wei-Yong Yan, Uwe Helmke, and John B Moore. Global analysis of oja's flow for neural networks. *IEEE Transactions on Neural Networks*, 5(5):674–683, 1994.

[Yi *et al.*, 2005] Zhang Yi, Mao Ye, Jian Cheng Lv, and Kok Kiong Tan. Convergence analysis of a deterministic discrete time system of oja's pca learning algorithm. *IEEE Transactions on Neural Networks*, 16(6):1318–1328, 2005.

[Zhang and Sra, 2016] Hongyi Zhang and Suvrit Sra. First-order methods for geodesically convex optimization. In *Conference on Learning Theory*, pages 1617–1638, 2016.

[Zhang and Sra, 2018] Hongyi Zhang and Suvrit Sra. An estimate sequence for geodesically convex optimization. In *Conference On Learning Theory*, pages 1703–1723, 2018.