Sample Complexity of Policy Gradient Finding Second-Order Stationary Points

Long Yang[1] **Qian Zheng**[2] **Gang Pan** [1*]
[1]College of Computer Science and Technology, Zhejiang University, China.
[2]School of Electrical and Electronic Engineering, Nanyang Technological University,Singapore.
[1] {yanglong,gpan}@zju.edu.cn; [2]zhengqian@ntu.edu.sg

Abstract

The policy-based reinforcement learning (RL) can be considered as maximization of its objective. However, due to the inherent non-concavity of its objective, the policy gradient method to a first-order stationary point (FOSP) cannot guarantee a maximal point. A FOSP can be a minimal or even a saddle point, which is undesirable for RL. It has be found that if all the saddle points are *strict*, all the second-order stationary points (SOSP) are exactly equivalent to local maxima. Instead of FOSP, we consider SOSP as the convergence criteria to characterize the sample complexity of policy gradient. Our result shows that policy gradient converges to an $(\epsilon, \sqrt{\epsilon\chi})$-SOSP with probability at least $1 - \widetilde{\mathcal{O}}(\delta)$ after the total cost of $\mathcal{O}\big(\frac{\epsilon^{-\frac{9}{2}}}{(1-\gamma)\sqrt{\chi}}\log\frac{1}{\delta}\big) = \mathcal{O}(\epsilon^{-\frac{9}{2}})$, where $\gamma \in (0,1)$. It significantly improves the state of the art cost $\widetilde{\mathcal{O}}(\epsilon^{-9})$.Our analysis is based on the key idea that decomposes the parameter space $\mathbb{R}^p$ into three non-intersected regions: non-stationary point region, saddle point region, and local optimal region, then making a local improvement of the objective of RL in each region. This technique can be potentially generalized to extensive policy gradient methods. For the complete proof, please refer to https://arxiv.org/pdf/2012.01491.pdf.

Introduction

Policy gradient method (Williams 1992; Sutton et al. 2000) is widely used to search the optimal policy in modern reinforcement learning (RL). Such method (or its variant) searches over a differentiable parameterized class of polices by performing a stochastic gradient on a cumulative expected reward function. Due to its merits such as the simplicity of implementation in the simulated environment; it requires low memory; it can be applied to any differentiable parameterized classes (Agarwal et al. 2020), policy gradient method has achieved significant successes in challenging fields such as robotics (Deisenroth, Neumann, and Peters 2013; Duan et al. 2016), playing Go (Silver et al. 2016, 2017), neural architecture search (Zoph and Le 2017), NLP (Kurita and Søgaard 2019; Whiteson 2019), computer vision (Sarmad, Lee, and Kim 2019), and recommendation system (Pan et al. 2019).

Despite it has tremendous successful applications, suffering from high sample complexity is still a critical challenge for the policy gradient (Haarnoja et al. 2018; Lee, Sungik, and Chung 2019; Xu, Gao, and Gu 2020). Thus, for policy gradient, theory analysis of its sample complexity plays an important role in RL since the sample complexity not only provides an understanding of the policy gradient but also gives insights on how to improve the sample efficiency of the existing RL algorithms.

Investigation of the sample complexity of policy gradient algorithm (or its variant) can be traced back to the pioneer works of (Kearns, Mansour, and Ng 2000; Kakade 2003). Recently, to improve sample efficiency, Yang et al. (2018); Papini et al. (2018); Shen et al. (2019); Xu, Gao, and Gu (2020) introduce stochastic variance reduced gradient techniques (Johnson and Zhang 2013; Nguyen et al. 2017a) to policy optimization, and they have studied the sample complexity of policy gradient methods to achieve a first-order stationary point (FOSP), i.e., θ satisfies the following condition

$$\|\nabla J(\theta)\|_2 \leq \epsilon.$$

However, since the objective of RL is a non-concave function with respect to the standard policy parameterizations (Papini et al. 2018; Agarwal et al. 2020), a FOSP could be a maximal point, a minimal point, and even a saddle point. Both minimal points and saddle points are undesirable for policy gradient since its goal is to search a maximal point, which implies within the numbers of samples provided by Papini et al. (2018); Shen et al. (2019); Xu, Gao, and Gu (2020), we can not guarantee the output of their policy gradient algorithm is a maximal point. This motivates a fundamental question as follows,

Question 1. *How many samples does an agent need to collect to guarantee the policy gradient methods converge to a maximal point certainly?*

Our Work

In this paper, we consider the second-order stationary point (SOSP) to answer Question 1. More specifically, inspired by the previous works from non-convex optimization (Jin et al. 2017; Daneshmand et al. 2018), we investigate the sample complexity of policy gradient methods finding an $(\epsilon, \sqrt{\epsilon\chi})$-

[*]Corresponding author.

SOSP, see Definition 1, i.e., the convergent point θ satisfies

$$\|\nabla J(\theta)\|_2 \le \epsilon, \text{ and } \lambda_{\max}(\nabla^2 J(\theta)) \le \sqrt{\chi\epsilon}.$$

The criterion of $(\epsilon, \sqrt{\epsilon\chi})$-SOSP requires the convergent point with a small gradient and with almost a negative semi-definite Hessian matrix. This criterion not only ensures a convergent point is a FOSP but also rules out both saddle points (whose Hessian are indefinite) and minimal points (whose Hessian are positive definite). Therefore, convergence to a $(\epsilon, \sqrt{\epsilon\chi})$-SOSP guarantees the policy gradient methods converge to a local maximal point clearly. Our result shows that within a cost of

$$\mathcal{O}\big(\frac{\epsilon^{-\frac{9}{2}}}{(1-\gamma)\sqrt{\chi}}\log\frac{1}{\delta}\big) = \widetilde{\mathcal{O}}(\epsilon^{-\frac{9}{2}}),$$

policy gradient converges to an $(\epsilon, \sqrt{\epsilon\chi})$-SOSP with probability at least $1 - \widetilde{\mathcal{O}}(\delta)$. Our result improves the state-of-the-art result of (Zhang et al. 2019) significantly, where they require $\mathcal{O}\big(\frac{\epsilon^{-9}\chi^{\frac{3}{2}}}{\delta}\log\frac{1}{\epsilon\chi}\big) = \widetilde{\mathcal{O}}(\epsilon^{-9})$ samples to achieve an $(\epsilon, \sqrt{\epsilon\chi})$-SOSP.

Notably, we provide a novel analysis that can be potentially generalized to extensive policy gradient methods. Concretely, we decompose the parameter space $\mathbb{R}^p$ into three different regions: non-stationary point region, saddle point region, and local optimal region, then making a local improvement in each region. The main challenge occurs on the saddle point region, where we utilize a technique called correlated negative curvature (CNC) (Daneshmand et al. 2018) to make a local improvement.

Notations

Let $\|\cdot\|_2$ be the Euclidean norm of a vector in $\mathbb{R}^p$. For a symmetric matrix A, we use $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ as its minimum and maximum eigenvalue correspondingly. Let $\|A\|_{op}$ denote the operator norm of the matrix A; furthermore, according to (Van Loan and Golub 1983), if $A \in \mathbb{R}^{p\times p}$ is a symmetric matrix, then $\|A\|_{op} = \max_{1\le i\le p}\{|\lambda_i|\}$, where $\{\lambda_i\}_{i=1}^p$ is the set of the eigenvalues of A. We use $A \succ 0$ to denote a positive definite matrix A. For a function $J(\cdot) : \mathbb{R}^p \to \mathbb{R}$, let ∇J and $\nabla^2 J$ denote its gradient vector and Hessian matrix correspondingly. Let $\mathbb{B}_2(o, r)$ be a p-dimensional ℓ_2 ball with the centre o and radius r, i.e., $\mathbb{B}_2(o, r) = \{x \in \mathbb{R}^p; \|x - o\|_2 \le r\}$. For any real number x, $\lceil x\rceil$and $\lfloor x\rfloor$ denote the nearest integer to x from above and below. We use $\widetilde{\mathcal{O}}$ to hide polylogarithmic factors in the input parameters, i.e., $\widetilde{\mathcal{O}}(f(x)) = \mathcal{O}(f(x)\log(f(x))^{\mathcal{O}(1)})$.

Paper Organization

Firstly, we introduce some necessary conceptions of reinforcement learning, policy gradient methods, some standard assumptions for policy optimization. Later, we formally define $(\epsilon, \sqrt{\epsilon\chi})$-SOSP. Our main contribution lies in the "Main Result and Technique Overview" section, where we provide the main result that presents the sample complexity of policy gradient finding an $(\epsilon, \sqrt{\epsilon\chi})$-SOSP, and we provide an overview of the proof technique. Finally, we discuss related works and future works.

Policy Gradient Methods and Some Standard Assumptions

In this section, we introduce some necessary concepts of reinforcement learning, policy gradient and some standard assumptions in policy optimization.

Reinforcement Learning

Reinforcement learning (RL) (Sutton and Barto 2018) is often formulated as *Markov decision processes* (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \rho_0, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space; $P(s^{'}|s, a)$ is the probability of state transition from s to $s^{'}$ under playing the action a; $R(\cdot,\cdot) : \mathcal{S} \times \mathcal{A} \to [R_{\min}, R_{\max}]$ is a bounded reward function, where $R_{\min}, R_{\max}$ two positive scalars. $\rho_0(\cdot) : \mathcal{S} \to [0, 1]$ is the initial state distribution and the discount factor $\gamma \in (0, 1)$.

The parametric *policy* π_θ is a probability distribution over $\mathcal{S} \times \mathcal{A}$ with a parameter $\theta \in \mathbb{R}^p$, and we use $\pi_\theta(a|s)$ to denote the probability of playing a in state s. Let $\tau = \{s_t, a_t, r_{t+1}\}_{t\ge 0} \sim \pi_\theta$ be a trajectory generated by the policy π_θ, where $s_0 \sim \rho_0(\cdot)$, $a_t \sim \pi_\theta(\cdot|s_t)$, $r_{t+1} = R(s_t, a_t)$ and $s_{t+1} \sim P(\cdot|s_t, a_t)$. The *state value function* of π_θ is defined as follows,

$$V^{\pi_\theta}(s) = \mathbb{E}_{\pi_\theta}[\sum_{t=0}^{\infty}\gamma^t r_{t+1}|s_0 = s],$$

where $\mathbb{E}_{\pi_\theta}[\cdot|\cdot]$ denotes a conditional expectation on actions which are selected according to the policy π_θ. The *advantage function* of the policy π_θ is defined as: $A^{\pi_\theta}(s, a) = Q^{\pi_\theta}(s, a) - V^{\pi_\theta}(s)$, where $Q^{\pi_\theta}(s, a)$ is the *state-action value function* that is defined as follows,

$$Q^{\pi_\theta}(s, a) = \mathbb{E}_{\pi_\theta}[\sum_{t=0}^{\infty}\gamma^t r_{t+1}|s_0 = s, a_0 = a].$$

We use $P^{\pi_\theta}(s_t = s|s_0)$ to denote the probability of visiting the state s after t time steps from the initial state s_0 by executing π_θ, and

$$d_{s_0}^{\pi_\theta}(s) = \sum_{t=0}^{\infty}\gamma^t P^{\pi_\theta}(s_t = s|s_0)$$

is the (unnormalized) discounted stationary state distribution of the Markov chain (starting at s_0) induced by π_θ. Furthermore, since $s_0 \sim \rho_0(\cdot)$, we define

$$d_{\rho_0}^{\pi_\theta}(s) = \mathbb{E}_{s_0\sim\rho_0(\cdot)}[d_{s_0}^{\pi_\theta}(s)]$$

as the discounted state visitation distribution over the initial distribution ρ_0.

Recall $\tau = \{s_t, a_t, r_{t+1}\}_{t\ge 0} \sim \pi_\theta$, we define $J(\pi_\theta|s_0)$ as follows,

$$\begin{aligned} J(\pi_\theta|s_0) &= \mathbb{E}_{\tau\sim\pi_\theta, s_0\sim\rho_0(\cdot)}[R(\tau)] \\ &= \mathbb{E}_{s\sim d_{s_0}^{\pi_\theta}(\cdot), a\sim\pi_\theta(\cdot|s)}[R(s, a)], \end{aligned}$$

where $R(\tau) = \sum_{t\ge 0}\gamma^t r_{t+1}$, and $J(\pi_\theta|s_0)$ is "conditional" on s_0 is to emphasize the trajectory τ that starts from

s_0. Furthermore, we define the expected return $J(\theta) =: \mathbb{E}_{s_0\sim\rho_0(\cdot)}[J(\pi_\theta|s_0)]$ as follows,

$$J(\theta) = \mathbb{E}_{s_0\sim\rho_0(\cdot)}[J(\pi_\theta|s_0)] = \mathbb{E}_{s\sim d^{\pi_\theta}_{\rho_0}(\cdot),a\sim\pi_\theta(\cdot|s)}[R(s,a)]$$
$$= \int_{s\in\mathcal{S}} d^{\pi_\theta}_{\rho_0}(s)\int_{a\in\mathcal{A}}\pi_\theta(a|s)R(s,a)\mathrm{d}a\mathrm{d}s. \tag{1}$$

The goal of policy-based reinforcement learning is to solve the following policy optimization problem:

$$\max_{\theta\in\mathbb{R}^p} J(\theta). \tag{2}$$

Policy Gradient Methods

The basic idea of policy gradient (Williams 1992; Sutton et al. 2000) is to update the parameter according to the direction with respect to the gradient of $J(\theta)$, i.e.,

$$\theta_{k+1} = \theta_k + \alpha\widehat{\nabla J(\theta_k)}, \tag{3}$$

where $\alpha > 0$ is step-size, $\widehat{\nabla J(\theta_k)}$ is a stochastic estimator of policy gradient $\nabla J(\theta_k)$. According to (Sutton et al. 2000), we present the *policy gradient theorem* as follows,

$$\nabla J(\theta) = \mathbb{E}_{s\sim d^{\pi_\theta}_{\rho_0}(\cdot),a\sim\pi_\theta(\cdot|s)}[Q^{\pi_\theta}(s,a)\nabla\log\pi_\theta(a|s)]$$
$$= \int_{s\in\mathcal{S}} d^{\pi_\theta}_{\rho_0}(s)\int_{a\in\mathcal{A}} Q^{\pi_\theta}(s,a)\nabla\pi_\theta(s,a)\mathrm{d}a\mathrm{d}s,$$

which provides a possible way to find the estimator of the policy gradient $\nabla J(\theta)$.

One issue that we should address is how to estimate $Q^{\pi_\theta}(s,a)$ appears in the policy gradient theorem. A simple approach is to use a sample return $R(\tau)$ to estimate $Q^{\pi_\theta}(s,a)$, i.e., we calculate the policy gradient estimator as follows,

$$g(\tau|\theta) = \sum_{t\geq 0}\nabla\log\pi_\theta(a_t|s_t)R(\tau). \tag{4}$$

Replace $\widehat{\nabla J(\tau|\theta_k)}$ of (3) with $g(\tau|\theta_k)$, we achieve the update rule of `REINFORCE` (Williams 1992):

$$\theta_{k+1} = \theta_k + \alpha g(\tau|\theta_k). \tag{5}$$

Notably, $R(\tau)$ of (4) could be replace by advantage function $A^{\pi_\theta}(s,a)$, temporal difference error, et al, a recent work (Schulman et al. 2016) summarizes those expressions. In this paper, we mainly consider the policy gradient estimator (4), and the technique that we have proposed can be generalized to other policy gradient estimators.

Fisher Information Matrix

For the policy optimization (1), we learn the parameter from the samples that come from an unknown probability distribution. Fisher information matrix (Fisher 1920; Kakade 2002; Ly et al. 2017) provides the information that a sample of the data for the unknown parameter.

Furthermore, according to (Kakade 2002; Bhatnagar et al. 2008), the Fisher information matrix $F(\theta)$ is positive definite, i.e., there exists a constant $\omega > 0$ s.t.,

$$F(\theta) \succ \omega I_p,\ \ \forall\,\theta\in\mathbb{R}^p, \tag{6}$$

where $I_p \in \mathbb{R}^{p\times p}$ is the identity matrix, $F(\theta) = \int_{s\in\mathcal{S}} d^{\pi_\theta}_{\rho_0}(s)\int_{a\in\mathcal{A}}\nabla\log\pi_\theta(a|s)[\nabla\log\pi_\theta(a|s)]^\top\mathrm{d}s\mathrm{d}a$.

Standard Assumptions

Assumption 1. *For each pair $(s,a) \in \mathcal{S}\times\mathcal{A}$, for any $\theta\in\mathbb{R}^p$, and all components i, j, there exists positive two constants $0 \leq G, L, U < \infty$ such that*

$$\left|\nabla_{\theta_i}\log\pi_\theta(a|s)\right| \leq G; \left|\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log\pi_\theta(a|s)\right| \leq L;$$
$$\left|\nabla_{\theta_i}\pi_\theta(a|s)\right| \leq U. \tag{7}$$

Assumptions 1 is a standard condition in policy optimization, and it has be applied to several recent policy gradient literatures (Castro and Meir 2010; Pirotta, Restelli, and Bascetta 2015; Papini et al. 2018; Shen et al. 2019; Xu, Gao, and Gu 2020). Assumption 1 is reasonable a condition since the widely used policy classes such as Gaussian, softmax (Konda and Borkar 1999), and relative entropy policy (Peters, Mülling, and Altun 2010) all satisfy (7). Recently, Zhang et al. (2019); Papini, Pirotta, and Restelli (2019); Wang and Zou (2020) have provided the details to check above policies satisfy Assumptions 1.

According to the Lemma B.2 of (Papini et al. 2018), Assumption 1 implies the expected return $J(\theta)$ is ℓ-Lipschitz smooth, i.e., for any $\theta,\theta^{'}\in\mathbb{R}^p$, we have

$$\|\nabla J(\theta) - \nabla J(\theta^{'})\|_2 \leq \ell\|\theta-\theta^{'}\|_2, \tag{8}$$

where $\ell = \frac{R_{\max}h(hG^2+L)}{(1-\gamma)}$, h is a positive scalar that denotes the horizon of the trajectory τ. The property (8) has been given as the Lipschitz assumption in previous works (Kumar, Koppel, and Ribeiro 2019; Wang et al. 2020), and it has been also verified by lots of recent works with some other regularity conditions (Zhang et al. 2019; Agarwal et al. 2020; Xu, Wang, and Liang 2020).

Furthermore, according to the Lemma 4.1 of (Shen et al. 2019), Assumption 1 implies a property of the policy gradient estimator as follows, for each $\tau\sim\pi_\theta$, we have

$$\|g(\tau|\theta) - \nabla J(\theta)\|_2 \leq \frac{GR_{\max}}{(1-\gamma)^2} =: \sigma. \tag{9}$$

The result of (9) implies the boundedness of the variance of the policy gradient estimator $g(\tau|\theta)$, i.e., $\mathbb{V}\mathrm{ar}(g(\tau|\theta)) = \mathbb{E}[\|g(\tau|\theta) - \nabla J(\theta)\|_2^2] \leq \sigma^2$. The boundedness of $\mathbb{V}\mathrm{ar}(g(\tau|\theta))$ are also proposed as an assumption in the previous works (Papini et al. 2018; Xu, Gao, and Gu 2019, 2020; Wang et al. 2020).

Assumption 2 (Smoothness of Policy Hessian)**.** *The the expected return function $J(\theta)$ is χ-Hessian-Lipschitz, i.e., there exists a constant $0 \leq \chi < \infty$ such that for all $\theta,\theta^{'}\in\mathbb{R}^p$:*

$$\|\nabla^2 J(\theta) - \nabla^2 J(\theta^{'})\|_{op} \leq \chi\|\theta-\theta^{'}\|_2. \tag{10}$$

Assumption 2 requires that for the two near points, the Hessian matrix $\nabla^2 J(\cdot)$ can not change dramatically in the terms of operator norm. For RL, the parameter χ can be deduced by some other regularity conditions, e.g., (Zhang et al. 2019) provides an estimation of χ, see Appendix A.

Second-Order Stationary Point

Due to the non-concavity of $J(\theta)$, finding global maxima is NP-hard in the worst case. The best one can hope is to convergence to stationary points. In this section, we formally define second-order stationary point (SOSP). Furthermore, with the second-order information, we present Assumption 3 to make clear the maximal point that we mainly concern for policy optimization.

Definition 1 (Second-Order Stationary Point (Nesterov and Polyak 2006) [1]). *For the χ-Hessian-Lipschitz function $J(\cdot)$, we say that θ is a second-order stationary point if*

$$\|\nabla J(\theta)\|_2 = 0 \quad \text{and} \quad \lambda_{\max}(\nabla^2 J(\theta)) \leq 0; \tag{11}$$

we say θ is an $(\epsilon, \sqrt{\chi\epsilon})$-second-order stationary point if

$$\|\nabla J(\theta)\|_2 \leq \epsilon \quad \text{and} \quad \lambda_{\max}(\nabla^2 J(\theta)) \leq \sqrt{\chi\epsilon}. \tag{12}$$

The SOSP is a very important concept for the policy optimization (2) because it rules the saddle points (whose Hessian are indefinite) and minimal points (whose Hessian are positive definite), which is usually more desirable than convergence to a first-order stationary point (FOSP). Recently, (Shen et al. 2019; Xu, Gao, and Gu 2020) introduce FOSP to measure the convergence of policy gradient methods. As mentioned in Section , for policy optimization (2), an algorithm converges to a FOSP is not sufficient to ensure that algorithm outputs a maximal point. While SOSP overcomes above shortcomings, which is our main motivation to consider SOSP as a convergence criterion.

Assumption 3 (Structure of $J(\theta)$). *For any $\theta \in \mathbb{R}^p$, at least one of the following holds: (i) $\|\nabla J(\theta)\| \geq \epsilon$; (ii) $\lambda_{\max}(\nabla^2 J(\theta)) \geq \sqrt{\epsilon\chi}$; (iii) θ nears a local maximal point $\theta_\star$: there exists a positive scalar ϱ such that θ falls in to the ball $\mathbb{B}_2(\theta_\star, \varrho)$, and $J(\theta)$ is ζ-strongly concave on $\mathbb{B}_2(\theta_\star, \varrho)$.*

In the standard non-convex literature such as (Ge et al. 2015; Jin et al. 2017), the condition (ii) of Assumption 3 is often called $(\epsilon, \chi, \varrho)$-*strict saddle*. In this case, all the SOSP are local maxima and hence convergence to second-order stationary points is equivalent to convergence to local maxima. In the following three-states MDP (see Figure 1), we verify that the Assumption 3 holds on policy optimization.

Example 1. *In this deterministic MDP, the states s_1 and s_2 equip the action space $\mathcal{A} = \{\texttt{right}, \texttt{left}\}$, while s_0 equips an additional action* up. *The policy $\pi_\theta(\cdot|\cdot)$ with a parameter $\theta = (\theta_1, \theta_2)^\top \in \mathbb{R}^2$, Let $C_o =: [0,1] \times [0,1]$, if $\theta \in C_o$, we define $\pi_\theta(\texttt{right}|s_0) =: p_1 = \frac{1}{\sqrt{2\pi}}(1 - \theta_1^2 + \theta_2^2)$; if $\theta \notin C_o$, we define Gaussian policy $\pi_\theta(\texttt{left}|s_0) =: p_2 = \frac{1}{\sqrt{2\pi}}\exp\big\{-\frac{(2-\|\theta\|_2^2)}{2}\big\}$; otherwise, $\pi_\theta(\texttt{up}|s_0) =: p_3 = 1 - p_1 - p_2$. Then $J(\theta) = p_1 R_1 + p_2 R_2 + p_3 R_3$, i.e.,*

$$J(\theta) = \begin{cases} \frac{1}{\sqrt{2\pi}}(1 - \theta_1^2 + \theta_2^2), & \theta \in C_o \\ \frac{1}{\sqrt{2\pi}}\exp\big\{-\frac{(2-\|\theta\|_2^2)}{2}\big\}, & \theta \notin C_o. \end{cases} \tag{13}$$

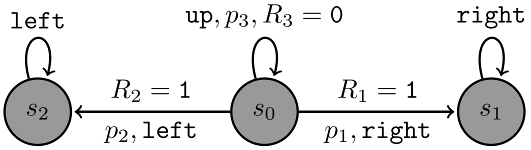

Figure 1: Three-States MDP.

The function $J(\theta)$ (13) satisfies Assumption 3. Since the origin $(0,0) \in C_o$ is a saddle point of $J(\theta)$, and $\lambda_{\max}(\nabla^2 J(\theta)|_{(0,0)}) = 1$, thus the point $(0,0)$ satisfies strict saddle point property. Besides, on the complementary space of C_o, i.e., $\mathbb{R}^2 - C_o$, $J(\theta)$ is a strongly concave function.

Main Result and Technique Overview

Our contribution lies in this section. Theorem 1 presents the sample complexity of policy gradient algorithm (5) finding an $(\epsilon, \sqrt{\chi\epsilon})$-SOSP. Firstly, we provide an overview of the proof technique. Then, we provide all the key steps. Finally, we provide a sketch of the proof of Theorem 1.

Theorem 1. *Under Assumption 1-3, consider $\{\theta_k\}_{k\geq 0}$ generated according to (5), and ι is defined in (21). For a small enough step-size α such that*

$$\alpha \leq \min\Big\{\frac{\epsilon^2}{2\sqrt{\chi\epsilon}R_{\min}^2\omega^2}, \frac{2\epsilon^2}{(\epsilon^2+\sigma^2)\ell}\Big\} = \mathcal{O}(\epsilon^2),$$

the iteration (5) returns an $(\epsilon, \sqrt{\chi\epsilon})$-SOSP with probability at least $1 - \delta - \delta\log\frac{1}{\delta} = 1 - \widetilde{O}(\delta)$ after the times of

$$K = \Big\lceil \frac{6R_{\max}}{\alpha^2(1-\gamma)\iota^2\sqrt{\chi\epsilon}}\log\frac{1}{\delta}\Big\rceil = \mathcal{O}\Big(\frac{\epsilon^{-\frac{9}{2}}}{(1-\gamma)\sqrt{\chi}}\log\frac{1}{\delta}\Big).$$

Remark 1. *Theorem 1 illustrates that policy gradient algorithm (5) needs a cost of $\widetilde{\mathcal{O}}(\epsilon^{-\frac{9}{2}})$ to find an $(\epsilon, \sqrt{\chi\epsilon})$-SOSP. To the best of our knowledge, Zhang et al. (2019) firstly consider to introduce SOSP to measure the sample complexity of policy-based reinforcement learning. Zhang et al. (2019) propose a* modified random-horizon policy gradient *(*MRPG*) algorithm, and they show that* MRPG *needs at least a cost of $\mathcal{O}\big(\epsilon^{-9}\chi^{\frac{3}{2}}\frac{1}{\delta}\log\frac{1}{\epsilon\chi}\big) = \widetilde{\mathcal{O}}(\epsilon^{-9})$ to find an $(\epsilon, \sqrt{\chi\epsilon})$-SOSP. Clearly, result of Theorem 1 improves the sample complexity of (Zhang et al. 2019) significantly from $\widetilde{\mathcal{O}}(\epsilon^{-9})$ to $\widetilde{\mathcal{O}}(\epsilon^{-\frac{9}{2}})$. Additionally, compared to (Zhang et al. 2019), our analysis does* not *invoke a geometric distribution restriction on the horizon. In the real world, the horizon of a trajectory only depends on the simulated environment, it is not necessary to draw a horizon from a geometric distribution, i.e., our result is more practical.*

Technique Overview

Recall that an $(\epsilon, \sqrt{\chi\epsilon})$-SOSP requires a point has a small gradient, and whose Hessian matrix does not have a significantly positive eigenvalue, which inspires us to consider an idea that decomposes the parameter space $\mathbb{R}^p$ into three non-intersected regions, and then analyzing them separately.

[1] Recall problem (2) is a maximization problem, thus this definition of SOSP is slightly different from the minimization problem $\min_x f(x)$, where it requires $\|\nabla f(x)\|_2 \leq \epsilon$ and $\lambda_{\min}(\nabla^2 f(x)) \geq 0$. Similarly, its $(\epsilon, \sqrt{\chi\epsilon})$-SOSP requires $\|\nabla f(x)\|_2 \leq \epsilon$ and $\lambda_{\min}(\nabla^2 f(x)) \geq -\sqrt{\chi\epsilon}$.

❶ **Case I: Non-Stationary Region.** In this case, we consider the region with large gradient, i.e.,

$$\mathcal{L}_1 = \big\{\theta \in \mathbb{R}^p : \|\nabla J(\theta)\|_2 \geq \epsilon\big\}; \tag{14}$$

❷ **Case II: Around Saddle Point.** We consider the region where the norm of the policy gradient is small , while the maximum eigenvalue of the Hessian matrix $\nabla^2 J(\theta)$ is larger than zero:

$$\begin{aligned}\mathcal{L}_2 =& \big\{\theta \in \mathbb{R}^p : \|\nabla J(\theta)\|_2 \leq \epsilon\big\} \\ &\bigcap \big\{\theta \in \mathbb{R}^p : \lambda_{\max}(\nabla^2 J(\theta)) \geq \sqrt{\chi\epsilon}\big\};\end{aligned} \tag{15}$$

❸ **Case III: Local Optimal Region.** In this case, we consider the region $\mathcal{L}_3 = \mathbb{R}^p - (\mathcal{L}_1 \cup \mathcal{L}_2)$:

$$\begin{aligned}\mathcal{L}_3 =& \big\{\theta \in \mathbb{R}^p : \|\nabla J(\theta)\|_2 \leq \epsilon\big\} \\ &\bigcap \big\{\theta \in \mathbb{R}^p : \lambda_{\max}(\nabla^2 J(\theta)) \leq \sqrt{\chi\epsilon}\big\}.\end{aligned} \tag{16}$$

It is noteworthy that the local optimal region, i.e., $\mathcal{L}_3$ is the desirable region where we expect policy gradient algorithm converges to it with high probability. Before we provide the formal proof of Theorem 1, in the next section, we present three separate propositions to make local improvements on above three regions correspondingly. The main challenge occurs on region $\mathcal{L}_2$, where we utilize a technique called correlated negative curvature (CNC) (Daneshmand et al. 2018) to make a local improvement.

Local Improvement on Each Case

Proposition 1 (Local Improvement on $\mathcal{L}_1$)**.** *Under Assumption 1-2. The sequence $\{\theta_k\}_{k\geq 0}$ generated according to (5). If a point $\theta_k \in \mathcal{L}_1$, let*

$$\alpha < \min\Big\{\frac{2\epsilon^2}{(\epsilon^2+\sigma^2)\ell}, \frac{2}{\ell}\Big\} = \mathcal{O}(\epsilon^{-2}), \tag{17}$$

then after one step, we have

$$\begin{aligned}&\mathbb{E}\big[J(\theta_{k+1})\big] - J(\theta_k) \\ &\overset{(\mathbf{a})}{\geq} \big(\alpha - \frac{\ell\alpha^2}{2}\big)\|\nabla J(\theta_k)\|_2^2 - \frac{\ell\alpha^2\sigma^2}{2} \overset{(\mathbf{b})}{\geq} \frac{1}{2}\alpha\epsilon^2.\end{aligned} \tag{18}$$

Proof. See Appendix B. □

Proposition 1 shows that when the gradient is large, the expected return $J(\theta)$ increases in one step. It is noteworthy that the step-size plays an important role in achieving the result of (18). Concretely, for a positive scalar $\alpha - \frac{\ell\alpha^2}{2}$ (i.e., which requires $\alpha < \frac{2}{\ell}$), Eq.(**a**) of (18) guarantees the desired increase whenever the norm of the policy gradient is large enough. At the same time, when considering the lower threshold value ϵ of the norm of the policy gradient in the region $\mathcal{L}_1$, the second term of (18) achieves at least $\alpha\epsilon^2 - \frac{1}{2}\ell\alpha^2(\epsilon^2+\sigma^2)$. Thus, to make a clear improvement, the condition (**b**) requires step-size α should satisfy $\alpha < \frac{2\epsilon^2}{(\epsilon^2+\sigma^2)\ell}$. This implies the step-size condition (17).

Proposition 2 (Local Improvement on $\mathcal{L}_2$)**.** *Under Assumption 1-2, consider the sequence $\{\theta_k\}_{k\geq 0}$ generated by (5). If a point θ_k falls in to $\mathcal{L}_2$, there exists a positive scalar ι (21), and $\widehat{\kappa}_0$ such that*

$$\widehat{\kappa}_0 =: \Big\lfloor \frac{\log\big(1/(1-\sqrt{\alpha}\sigma_{H_0})\big)}{\log(1+\alpha\sqrt{\chi\epsilon})}\Big\rfloor = \mathcal{O}(\epsilon^{-\frac{1}{2}}), \tag{19}$$

where $\sigma_{H_0} = \frac{2p\sqrt{p}hR_{\max}(hG^2+L)}{1-\gamma}$, then after at most $j \leq \widehat{\kappa}_0$ steps, we have

$$\mathbb{E}[J(\theta_{k+j})] - J(\theta_k) \geq \alpha^2\iota^2\sqrt{\chi\epsilon}. \tag{20}$$

Proof. See Appendix D. □

Proposition 2 illustrates that even a point gets stuck in the region thar nears a saddle point, policy gradient method will ensure an increase in the value of $J(\theta)$ within at most $\mathcal{O}(\epsilon^{-\frac{1}{2}})$ steps. We provide proof of Proposition 2 in Appendix C. The proof is very technical, the following correlated negative curvature (CNC) condition (Daneshmand et al. 2018) plays a crucial role in achieving the result of (20). Concretely, let u_p be the unit eigenvector corresponding to the maximum eigenvalue of $\nabla^2 J(\theta)$, CNC ensures the second moment of the projection of policy gradient estimator $g(\tau|\theta)$ along the direction u_p is uniformly bounded away from zero, i.e., there exists a positive scalar ι s.t.,

$$\textbf{CNC}: \ \mathbb{E}[\langle g(\tau|\theta), u_p\rangle^2] \geq \iota^2, \ \ \forall \theta \in \mathbb{R}^p. \tag{21}$$

We provide the derivation of Eq.(21) in the Discussion **??** of Appendix C. CNC shows that the perturbation caused by a stochastic policy gradient estimator $g(\tau|\theta)$ is guaranteed to take an increase in the value of $J(\theta)$.

Proposition 3. *Under Assumption 1-3, consider the sequence $\{\theta_k\}_{k\geq 0}$ generated by (5). For any $\delta \in (0,1)$, $\theta_\star$ satisfies Assumption 3, let the step-size α and the stopping time κ_0 satisfy*

$$\begin{aligned}&\kappa_0 = \Big\lfloor\frac{1}{\alpha^2}\log\frac{1}{\delta}\Big\rfloor, \alpha \leq \min\Big\{\delta, \frac{1}{\zeta}, \frac{\zeta}{\ell^2}\frac{\zeta\varrho^2}{3\sigma^2}\Big\}, \\ &\alpha\log\frac{1}{\alpha} \leq \frac{2\zeta\varrho^4}{27\Big(\frac{G^2R_{\max}^2}{(1-\gamma)^2} + \zeta\varrho^2 + \sigma^2\Big)^2},\end{aligned}$$

and if some iteration θ_k falls into the ball $\mathbb{B}_2(\theta_\star, \frac{\sqrt{3}}{3}\varrho) \subset \mathbb{B}_2(\theta_\star, \varrho)$, i.e., $\|\theta_k - \theta_\star\|_2^2 \leq \frac{1}{3}\varrho^2$. Then, $\forall\, j \in [0, \kappa_0 - 1]$,

$$\mathbb{P}\big(\big\|\theta_{k+j} - \theta_\star\big\|_2 \leq \varrho\big) \geq 1 - \delta\log\frac{1}{\delta}.$$

Proof. See Appendix C. □

Proposition 3 illustrates that once an iteration gets sufficiently close to a local optimum $\theta_\star$, it can get trapped in the neighborhood of $\theta_\star$ for a really long time.

Proof Sketch of Theorem 1

Proof. Our proof contains three steps. *Firstly*, we will prove that within $\lceil\frac{6R_{\max}}{\alpha^2(1-\gamma)\iota^2\sqrt{\chi\epsilon}}\rceil$ steps, with probability at least $\frac{1}{2}$, one iteration falls into $\mathcal{L}_3$. Let above procedure lasts $\lceil\log\frac{1}{\delta}\rceil$ steps, according to the *inclusion-exclusion* formula of probability: after $K_o = \lceil \frac{6R_{\max}}{\alpha^2(1-\gamma)\iota^2\sqrt{\chi\epsilon}} \log\frac{1}{\delta} \rceil$ steps, one of $\{\theta_k\}_{k\geq 0}$ falls into $\mathcal{L}_3$ with probability $1-\delta$. *Secondly*, Proposition 3 shows that once an iteration enters the region $\mathcal{L}_3$, the iteration gets trapped there for at least κ_0 steps with probability $1-\delta\log\dfrac{1}{\delta}$. *Finally*, let $K_o+1<K<K_o+\kappa_0$, combining above two results, the output θ_K falls into the region $\mathcal{L}_3$ with probability at least $1-(\delta+\delta\log\frac{1}{\delta})$, which concludes the result of Theorem 1.

The above discussion implies that we only need to prove: starting from any point, within $\lceil \frac{6R_{\max}}{\alpha^2(1-\gamma)\iota^2\sqrt{\chi\epsilon}} \rceil$ steps, with probability at least $\frac{1}{2}$, one of $\{\theta_k\}_{k\geq 0}$ falls into $\mathcal{L}_3$.

We define a stochastic process $\{\varsigma_k\}_{k\geq 0}$ ($\varsigma_0=0$) to trace the numbers of samples,

$$\varsigma_{k+1} = \begin{cases} \varsigma_k + 1 & \text{if } \theta_{\varsigma_k} \in \mathcal{L}_1 \cup \mathcal{L}_3 \\ \varsigma_k + \widehat{\kappa}_0 & \text{if } \theta_{\varsigma_k} \in \mathcal{L}_2, \end{cases}$$

where $\widehat{\kappa}_0$ is defined in (19). Let $\beta = \iota^2\sqrt{\chi\epsilon}$, we can rewrite the results of Proposition 1-2 as follows,

$$\mathbb{E}[J(\theta_{\varsigma_{k+1}}) - J(\theta_{\varsigma_k}) | \theta_{\varsigma_k} \in \mathcal{L}_1] \overset{(18)}{\geq} \frac{1}{2}\alpha\epsilon^2, \tag{22}$$

$$\mathbb{E}[J(\theta_{\varsigma_{k+1}}) - J(\theta_{\varsigma_k}) | \theta_{\varsigma_k} \in \mathcal{L}_2] \overset{(20)}{\geq} \alpha^2\beta. \tag{23}$$

Putting the results of (22)-(23) together, let $\alpha^2\beta \leq \frac{1}{2}\alpha\epsilon^2$, i.e., $\alpha \leq \frac{\epsilon^2}{2\beta}$,we have

$$\begin{aligned} &\mathbb{E}[J(\theta_{\varsigma_{k+1}}) - J(\theta_{\varsigma_k}) | \theta_{\varsigma_k} \notin \mathcal{L}_3] \\ \geq &\alpha^2\beta\mathbb{E}[(\varsigma_{k+1} - \varsigma_k) | \theta_{\varsigma_k} \notin \mathcal{L}_3]. \end{aligned} \tag{24}$$

We define the event

$$\mathcal{E}_k =: \bigcap_{j=0}^{k} \{j : \theta_{\varsigma_j} \notin \mathcal{L}_3\}.$$

Let $\mathbf{1}_A$ denote the indicator function, where if event A happens, $\mathbf{1}_A = 1$, otherwise $\mathbf{1}_A = 0$.

$$\begin{aligned} &\mathbb{E}[J(\theta_{\varsigma_{k+1}})\mathbf{1}_{\mathcal{E}_{k+1}} - J(\theta_{\varsigma_k})\mathbf{1}_{\mathcal{E}_k}] \qquad (25) \\ =&\mathbb{E}[J(\theta_{\varsigma_{k+1}})(\mathbf{1}_{\mathcal{E}_{k+1}} - \mathbf{1}_{\mathcal{E}_k})] + \mathbb{E}[(J(\theta_{\varsigma_{k+1}}) - J(\theta_{\varsigma_k}))\mathbf{1}_{\mathcal{E}_k}] \\ \overset{(23)}{\geq}& -\frac{R_{\max}}{1-\gamma}(\mathbb{P}(\mathcal{E}_{k+1} - \mathcal{E}_k)) + \alpha^2\beta\mathbb{E}[\varsigma_{k+1} - \varsigma_k | \mathbf{1}_{\mathcal{E}_k}]\mathbb{P}(\mathcal{E}_k), \end{aligned}$$

where we use the boundedness of $J(\theta) \geq -\frac{R_{\max}}{1-\gamma}$. Summing the above expectation (25) over k, then

$$\begin{aligned} &\mathbb{E}[J(\theta_{\varsigma_{k+1}})\mathbf{1}_{\mathcal{E}_{k+1}}] - J(\theta_0) \\ =& -\frac{R_{\max}}{1-\gamma}(\mathbb{P}(\mathcal{E}_{k+1}) - \mathbb{P}(\mathcal{E}_0)) \\ &\qquad + \alpha^2\beta\sum_{j=0}^{k}\big(\mathbb{E}[\varsigma_{j+1}]\mathbb{P}(\mathcal{E}_j) - \mathbb{E}[\varsigma_j]\mathbb{P}(\mathcal{E}_j)\big) \\ \geq& -\frac{R_{\max}}{1-\gamma} + \alpha^2\beta\sum_{j=0}^{k}\big(\mathbb{E}[\varsigma_{j+1}]\mathbb{P}(\mathcal{E}_{j+1}) - \mathbb{E}[\varsigma_j]\mathbb{P}(\mathcal{E}_j)\big) \end{aligned} \tag{26}$$

$$= -\frac{R_{\max}}{1-\gamma} + \alpha^2\beta\mathbb{E}[\varsigma_{k+1}]\mathbb{P}(\mathcal{E}_{k+1}), \tag{27}$$

where Eq.(26) holds since $\mathbb{P}(\mathcal{E}_{k+1}) - \mathbb{P}(\mathcal{E}_0) \leq 1$; $\mathcal{E}_{j+1} \subset \mathcal{E}_j$ implies $\mathbb{P}(\mathcal{E}_{j+1}) \leq \mathbb{P}(\mathcal{E}_j)$; and Eq.(27) holds since $\varsigma_0 = 0$.

Finally, since

$$\mathbb{E}[J(\theta_{\varsigma_{k+1}})\mathbf{1}_{\mathcal{E}_{k+1}} - J(\theta_{\varsigma_k})\mathbf{1}_{\mathcal{E}_k}] \leq \frac{2R_{\max}}{1-\gamma},$$

and according to the result of (27), if

$$\mathbb{E}[\varsigma_{k+1}] \geq \frac{6R_{\max}}{\alpha^2(1-\gamma)\beta} = \frac{6R_{\max}}{\alpha^2(1-\gamma)\iota^2\sqrt{\chi\epsilon}},$$

we have

$$\mathbb{P}[\mathcal{E}_{k+1}] \leq \frac{1}{2}.$$

This concludes the proof. □

Related Works and Future Works

Compared to the tremendous empirical works, theoretical results of policy gradient methods are relatively scarce. In this section, we compare our result with current works in the following discussion. For clarity, we have presented the complexity comparison to some results in Table 1. Furthermore, we discuss future works to extend our proof technique to other policy gradient methods.

First-Order Measurement

According to (Shen et al. 2019), `REINFORCE` needs $\mathcal{O}(\epsilon^{-4})$ random trajectories to achieve the ϵ-FOSP, and no provable improvement on its complexity has been made so far. Later, Xu, Gao, and Gu (2019) notice the order of sample complexity of `REINFORCE` and `GPOMDP` (Baxter and Bartlett 2001) need $\mathcal{O}(\epsilon^{-4})$ to achieve the ϵ-FOSP. With an additional assumption $\mathbb{V}\mathrm{ar}\Big[\prod_{i\geq 0}\frac{\pi_{\theta_0}(a_i|s_i)}{\pi_{\theta_t}(a_i|s_i)}\Big]$,$\mathbb{V}\mathrm{ar}[g(\tau|\theta)] < +\infty$, Papini et al. (2018) show that the `SVRPG` needs sample complexity of $\mathcal{O}(\epsilon^{-4})$ to achieve the ϵ-FOSP. Under the same assumption as (Papini et al. 2018), Xu, Gao, and Gu (2019) reduce the sample complexity of `SVRPG` to $\mathcal{O}(\epsilon^{-\frac{10}{3}})$. Recently, Shen et al. (2019), Yang et al. (2019a) and Xu, Gao, and Gu (2020) introduce stochastic variance reduced gradient (SVRG) techniques (Johnson and Zhang 2013; Nguyen et al. 2017a; Fang et al. 2018) to policy optimization, their new methods improve sample complexity to $\mathcal{O}(\epsilon^{-3})$ to achieve an ϵ-FOSP. Pham et al. (2020) propose `ProxHSPGA` that is a hybrid stochastic policy gradient estimator by combining existing `REINFORCE` estimator with the adapted `SARAH` (Nguyen et al. 2017a) estimator. Pham et al. (2020) show `ProxHSPGA` also need $\mathcal{O}(\epsilon^{-3})$ trajectories to achieve the ϵ-FOSP. To compare clearly, we summarize more details of the comparison in Table 1.

Second-Order Measurement

As mentioned in the previous section, for reinforcement, an algorithm converges to a FOSP is not sufficient to ensure that algorithm outputs a maximal point, which is our main motivation to consider SOSP to measure the convergence of policy gradient method.

To the best of our knowledge, Zhang et al. (2019) firstly introduce SOSP to RL to measure the sample complexity of

Algorithm	Conditions	Measurement	Complexity
`REINFORCE` (Williams 1992)	Assumption 1	First-Order	$\mathcal{O}(\epsilon^{-4})$
`GPOMDP` (Baxter and Bartlett 2001)	Assumption 1	First-Order	$\mathcal{O}(\epsilon^{-4})$
`SVRPG` (Papini et al. 2018)	Assumption 1; $\mathbb{V}\text{ar}\left[\prod_{i\geq 0}\frac{\pi_{\theta_0}(a_i\|s_i)}{\pi_{\theta_t}(a_i\|s_i)}\right] < +\infty$	First-Order	$\mathcal{O}(\epsilon^{-4})$
`SVRPG` (Xu, Gao, and Gu 2019)	Assumption 1; $\mathbb{V}\text{ar}\left[\prod_{i\geq 0}\frac{\pi_{\theta_0}(a_i\|s_i)}{\pi_{\theta_t}(a_i\|s_i)}\right] < +\infty$	First-Order	$\mathcal{O}(\epsilon^{-\frac{10}{3}})$
`HAPG` (Shen et al. 2019)	Assumption 1	First-Order	$\mathcal{O}(\epsilon^{-3})$
`VRMPO` (Yang et al. 2019a)	Assumption 1	First-Order	$\mathcal{O}(\epsilon^{-3})$
`SRVR-PG` (Xu, Gao, and Gu 2020)	Assumption 1; $\mathbb{V}\text{ar}\left[\prod_{i\geq 0}\frac{\pi_{\theta_0}(a_i\|s_i)}{\pi_{\theta_t}(a_i\|s_i)}\right] < +\infty$	First-Order	$\mathcal{O}(\epsilon^{-3})$
`ProxHSPGA` (Pham et al. 2020)	Assumption 1; $\mathbb{V}\text{ar}\left[\prod_{i\geq 0}\frac{\pi_{\theta_0}(a_i\|s_i)}{\pi_{\theta_t}(a_i\|s_i)}\right] < +\infty$	First-Order	$\mathcal{O}(\epsilon^{-3})$
`MRPG` (Zhang et al. 2019)	Assumption 1 and Eq.(6)	Second-Order	$\widetilde{\mathcal{O}}(\epsilon^{-9})$
Our work	Assumption 1-3	Second-Order	$\widetilde{\mathcal{O}}(\epsilon^{-\frac{9}{2}})$

Table 1: Complexity comparison, where the result of first-order measurement requires $\|\nabla J(\theta)\|_2 \leq \epsilon$, section-order measurement requires an additional condition $\lambda_{\max}(\nabla^2 J(\theta)) \leq \sqrt{\chi\epsilon}$.

policy gradient methods. Zhang et al. (2019) propose `MRPG` that needs at least $\widetilde{\mathcal{O}}(\epsilon^{-9})$ samples, which is worse than our result $\widetilde{\mathcal{O}}(\epsilon^{-\frac{9}{2}})$. We have discussed this comparison in the previous Remark 1.

Additionally, it is noteworthy that although we are all adopting the CNC technique to ensure the local improvement on saddle point region, our technique is different from Zhang et al. (2019) at least from two aspects: Firstly, our CNC condition is more general since we consider the fundamental policy gradient estimator (5) and our analysis can be extended to generalized to extensive policy optimization algorithms; while the CNC result of Zhang et al. (2019) is limited in their proposed algorithm `MRPG`; Secondly, on the region $\mathcal{L}_2$, our result shows that within at most $\mathcal{O}(\epsilon^{-\frac{1}{2}})$ steps, policy gradient ensures an increase in the value of $J(\theta)$. While, Zhang et al. (2019) require $\Omega(\epsilon^{-5}\log\frac{1}{\epsilon})$, which is the main reason why our analysis to achieve a better sample complexity than Zhang et al. (2019).

Future Works

In this paper, we mainly consider Monte Carlo gradient estimator (4), the technique of proof can be generalized to extensive policy gradient methods such as replacing $R(\tau)$ with state-action value function $Q^{\pi}(s_t, a_t)$, advantage function $A^{\pi}(s_t, a_t)$, baseline function $R(\tau) - V^{\pi}(s_t, a_t)$, and temporal difference (TD) learning error $r_{t+1} + \gamma V^{\pi}(s_{t+1}, a_{t+1}) - V^{\pi}(s_t, a_t)$. Our result of $\widetilde{\mathcal{O}}(\epsilon^{-\frac{9}{2}})$ to achieve $(\epsilon, \sqrt{\epsilon\chi})$-SOSP is still far from the best-known ϵ-FOSP result $\mathcal{O}(\epsilon^{-3})$. In theory, Allen-Zhu and Li (2018) and Xu, Jin, and Yang (2018) independently show that finding a SOSP is not much harder than FOSP. Recently, in non-convex optimization, Ge et al. (2019) show that with a simple variant of SVRG, we can find a SOSP that almost matches the known the first-order stationary points. This provides a motivation that we can introduce some latest developments such as (Daneshmand et al. 2018; Jin, Netrapalli, and Jordan 2018; Zhou, Xu, and Gu 2018a,b; Ge et al. 2019; Fang, Lin, and Zhang 2019) to give some fresh understanding to RL algorithms. Besides, it will be also interesting to rethink the sample complexity of $(\epsilon, \sqrt{\epsilon\chi})$-SOSP of the works in reinforcement learning (Papini et al. 2018; Shen et al. 2019; Yang et al. 2019a; Pham et al. 2020), where they have proposed SVRG version of policy gradient methods. It is noteworthy that we don't consider the actor-critic type algorithms. Recently Yang et al. (2019b); Kumar, Koppel, and Ribeiro (2019); Agarwal et al. (2020); Xu, Wang, and Liang (2020); Wang et al. (2020) have analyzed the complexity of actor-critic or natural actor-critic algorithms. It will be very interesting to rethink the sample complexity of the $(\epsilon, \sqrt{\epsilon\chi})$-SOSP of actor-critic or natural actor-critic algorithms.

Conclusion

In this paper, we provide the sample complexity of the policy gradient method finding second-order stationary points. Our result shows that policy gradient methods converge to an $(\epsilon, \sqrt{\epsilon\chi})$-SOSP at a cost of $\widetilde{\mathcal{O}}(\epsilon^{-\frac{9}{2}})$, which improves the the best-known result of by a factor of $\widetilde{\mathcal{O}}(\epsilon^{-\frac{9}{2}})$. Besides, we think the technique of proof can be potentially generalized to extensive policy optimization algorithms, and give some fresh understanding to the existing algorithms.

Acknowledgements

Natural Science Foundation of China (No. 61925603) Zhejiang Lab (No. 2019KC0AD02).

References

Agarwal, A.; Kakade, S. M.; Lee, J. D.; and Mahajan, G. 2020. Optimality and Approximation with Policy Gradient Methods in Markov Decision Processes. *Conferecnce on Learning Theory* .

Allen-Zhu, Z.; and Li, Y. 2018. Neon2: Finding local minima via first-order oracles. In *Advances in Neural Information Processing Systems*, 3716–3726.

Baxter, J.; and Bartlett, P. L. 2001. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research* 15: 319–350.

Bhatnagar, S.; Ghavamzadeh, M.; Lee, M.; and Sutton, R. S. 2008. Incremental natural actor-critic algorithms. In *Advances in neural information processing systems*, 105–112.

Castro, D. D.; and Meir, R. 2010. A convergent online single time scale actor critic algorithm. *Journal of Machine Learning Research* 11(Jan): 367–410.

Daneshmand, H.; Kohler, J.; Lucchi, A.; and Hofmann, T. 2018. Escaping saddles with stochastic gradients. *International Conference on Machine Learning* .

Deisenroth, M. P.; Neumann, G.; and Peters, J. 2013. A survey on policy search for robotics. *Foundations and Trends® in Machine Learning* .

Duan, Y.; Chen, X.; Houthooft, R.; Schulman, J.; and Abbeel, P. 2016. Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning*, 1329–1338.

Fang, C.; Li, C. J.; Lin, Z.; and Zhang, T. 2018. SPIDER: Near-Optimal Non-Convex Optimization via Stochastic Path-Integrated Differential Estimator. In *Advances in Neural Information Processing Systems*, 686–696.

Fang, C.; Lin, Z.; and Zhang, T. 2019. Sharp analysis for nonconvex sgd escaping from saddle points. In *Conference on Learning Theory*.

Fisher, Aylmer, R. 1920. A Mathematical Examination of the Methods of De- termining the Accuracy of an Observation by the Mean Error. *Monthly Notices of the Royal Astronomical Society* (80): 758–770.

Ge, R.; Huang, F.; Jin, C.; and Yuan, Y. 2015. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, 797–842.

Ge, R.; Li, Z.; Wang, W.; and Wang, X. 2019. Stabilized SVRG: Simple variance reduction for nonconvex optimization. In *Conference on Learning Theory*, 1–55.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*.

Jin, C.; Ge, R.; Netrapalli, P.; Kakade, S. M.; and Jordan, M. I. 2017. How to escape saddle points efficiently. In *International Conference on Machine Learning*, 1724–1732.

Jin, C.; Netrapalli, P.; and Jordan, M. I. 2018. Accelerated gradient descent escapes saddle points faster than gradient descent. *Conference on Learning Theory* .

Johnson, R.; and Zhang, T. 2013. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, 315–323.

Kakade, S. M. 2002. A natural policy gradient. In *Advances in Neural Information Processing Systems*, 1531–1538.

Kakade, S. M. 2003. *On the sample complexity of reinforcement learning*. Ph.D. thesis, University of London London, England.

Kearns, M. J.; Mansour, Y.; and Ng, A. Y. 2000. Approximate planning in large POMDPs via reusable trajectories. In *Advances in Neural Information Processing Systems*, 1001–1007.

Konda, V. R.; and Borkar, V. S. 1999. Actor-Critic–Type Learning Algorithms for Markov Decision Processes. *SIAM Journal on control and Optimization* 38(1): 94–123.

Kumar, H.; Koppel, A.; and Ribeiro, A. 2019. On the Sample Complexity of Actor-Critic Method for Reinforcement Learning with Function Approximation. *In Optimization Foundations for Reinforcement Learning Workshop at Advances in Neural Information Processing Systems* .

Kurita, S.; and Søgaard, A. 2019. Multi-Task Semantic Dependency Parsing with Policy Gradient for Learning Easy-First Strategies. *Association for Computational Linguistics (ACL)* .

Lee, S. Y.; Sungik, C.; and Chung, S.-Y. 2019. Sample-efficient deep reinforcement learning via episodic backward update. In *Advances in Neural Information Processing Systems*, 2110–2119.

Ly, A.; Marsman, M.; Verhagen, J.; Grasman, R. P.; and Wagenmakers, E.-J. 2017. A tutorial on Fisher information. *Journal of Mathematical Psychology* 80: 40–55.

Nesterov, Y.; and Polyak, B. T. 2006. Cubic regularization of Newton method and its global performance. *Mathematical Programming* 108(1): 177–205.

Nguyen, L. M.; Liu, J.; Scheinberg, K.; and Takáč, M. 2017a. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *International Conference on Machine Learning*.

Pan, F.; Cai, Q.; Tang, P.; Zhuang, F.; and He, Q. 2019. Policy gradients for contextual recommendations. In *The World Wide Web Conference*, 1421–1431.

Papini, M.; Binaghi, D.; Canonaco, G.; Pirotta, M.; and Restelli, M. 2018. Stochastic Variance-Reduced Policy Gradient. In *International Conference on Machine Learning*.

Papini, M.; Pirotta, M.; and Restelli, M. 2019. Smoothing policies and safe policy gradients. *arXiv preprint arXiv:1905.03231* .

Peters, J.; Mülling, K.; and Altun, Y. 2010. Relative Entropy Policy Search. In *AAAI*, 1607–1612.

Pham, N. H.; Nguyen, L. M.; Phan, D. T.; Nguyen, P. H.; van Dijk, M.; and Tran-Dinh, Q. 2020. A Hybrid Stochastic Policy Gradient Algorithm for Reinforcement Learning. *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)* .

Pirotta, M.; Restelli, M.; and Bascetta, L. 2015. Policy gradient in lipschitz markov decision processes. *Machine Learning* 100(2-3): 255–283.

Sarmad, M.; Lee, H. J.; and Kim, Y. M. 2019. RL-GAN-Net: A Reinforcement Learning Agent Controlled GAN Network for Real-Time Point Cloud Shape Completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5898–5907.

Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; and Abbeel, P. 2016. High-dimensional continuous control using generalized advantage estimation. *International Conference on Learning Representations* .

Shen, Z.; Ribeiro, A.; Hassani, H.; Qian, H.; and Mi, C. 2019. Hessian Aided Policy Gradient. In *International Conference on Machine Learning*, 5729–5738.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature* 529(7587): 484.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. 2017. Mastering the game of Go without human knowledge. *Nature* 550(7676): 354.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Sutton, R. S.; McAllester, D. A.; Singh, S. P.; and Mansour, Y. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, 1057–1063.

Van Loan, C. F.; and Golub, G. H. 1983. *Matrix computations*. Johns Hopkins University Press.

Wang, L.; Cai, Q.; Yang, Z.; and Wang, Z. 2020. Neural policy gradient methods: Global optimality and rates of convergence. *International Conference on Learning Representations* .

Wang, Y.; and Zou, S. 2020. Finite-sample Analysis of Greedy-GQ with Linear Function Approximation under Markovian Noise. *Conference on Uncertainty in Artificial Intelligence (UAI)* .

Whiteson, S. 2019. A Survey of Reinforcement Learning Informed by Natural Language. International Joint Conferences on Artificial Intelligence.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4): 229–256.

Xu, P.; Gao, F.; and Gu, Q. 2019. An Improved Convergence Analysis of Stochastic Variance-Reduced Policy Gradient. *Conference on Uncertainty in Artificial Intelligence* .

Xu, P.; Gao, F.; and Gu, Q. 2020. Sample Efficient Policy Gradient Methods with Recursive Variance Reduction. *International Conference on Learning Representations* .

Xu, T.; Wang, Z.; and Liang, Y. 2020. Improving sample complexity bounds for actor-critic algorithms. *arXiv preprint arXiv:2004.12956* .

Xu, Y.; Jin, R.; and Yang, T. 2018. First-order stochastic algorithms for escaping from saddle points in almost linear time. In *Advances in Neural Information Processing Systems*, 5530–5540.

Yang, L.; Shi, M.; Zheng, Q.; Meng, W.; and Pan, G. 2018. A unified approach for multi-step temporal-difference learning with eligibility traces in reinforcement learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2984–2990.

Yang, L.; Zheng, G.; Zhang, H.; Zhang, Y.; Zheng, Q.; and Pan, G. 2019a. Policy optimization with stochastic mirror descent. *arXiv preprint arXiv:1906.10462* .

Yang, Z.; Chen, Y.; Hong, M.; and Wang, Z. 2019b. Provably Global Convergence of Actor-Critic: A Case for Linear Quadratic Regulator with Ergodic Cost. In *Advances in Neural Information Processing Systems*, 8351–8363.

Zhang, K.; Koppel, A.; Zhu, H.; and Başar, T. 2019. Global convergence of policy gradient methods to (almost) locally optimal policies. *arXiv preprint arXiv:1906.08383* .

Zhou, D.; Xu, P.; and Gu, Q. 2018a. Finding local minima via stochastic nested variance reduction. *arXiv preprint arXiv:1806.08782* .

Zhou, D.; Xu, P.; and Gu, Q. 2018b. Stochastic nested variance reduction for nonconvex optimization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 3925–3936.

Zoph, B.; and Le, Q. V. 2017. Neural architecture search with reinforcement learning. *International Conference on Learning Representation* .