Performance of momentum-based variance reduction method on softmax tabular setting

Mohammadsaeed Masiha (322651), Sadegh Farhadkhani (316853)
EPFL, Switzerland
mohammadsaeed.masiha@epfl.ch, sadegh.farhadkhani@epfl.ch

***Abstract*—We evaluate the performance of momentum-based variance reduction (MVR) technique in three finite state-action reinforcement learning (RL) settings which are equipped with the soft-max tabular policy. We compare MVR with stochastic policy gradient (SPG) and REINFORCE. Our results demonstrate the superiority of MVR to achieve an optimum point in all three tasks.**

I. Introduction

Variance reduction is a common trick recently introduced to improve the convergence rate of stochastic gradient-based optimization algorithms. There exists a large body of work on this technique. In this report, however, we focus on the STORM algorithm proposed by [1] as it does not require large batches. STORM is similar to the conventional momentum technique with an additional correction term that leads to the following update rule

$$\begin{aligned} d_t :=&(1-a)d_{t-1} + a\nabla f(x_t,\xi_t) \\ &+ (1-a)(\nabla f(x_t,\xi_t) - \nabla f(x_{t-1},\xi_t)), \qquad (1)\\ x_{t+1} :=&x_t - \eta_t d_t, \qquad (2)\end{aligned}$$

where η_t is the learning rate, a is a constant in $(0,1]$, and ξ_t is a random variable drawn from some fixed but unknown distribution at iteration t and represents the stochasticity of the algorithm (for instance, it might represent a batch of data points).

We compare STORM with stochastic policy gradient (SPG) and REINFORCE [2] algorithms on the finite state-action environment equipped with softmax tabular policies. In [3], it is shown that the global optimum sample complexity of REINFORCE (regularized with log barrier) is $\mathcal{O}(\epsilon^{-6})$. In [4], for the soft-max policy parameterization, they show that adding momentum terms improves the existing global optimum sample complexity bounds of PG [3] by $\mathcal{O}(\epsilon^{-1.5})$. We are seeking to compare momentum based SPG with REINFORCE without entropy (or log barrier) regularization. Global optimality of such algorithms may be characterized by non-uniform gradient dominance property [5] of value function when we are using softmax tabular policies and we will work on this problem as a future work.

Structure of the report. In Section II, we explain the reinforcement learning setup. In Section III, we explain the details of our experiments. Finally, Section IV, discusses our experimental findings.

II. RL setup

Consider a discrete Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \rho, \gamma)$, where $\mathcal{S}$ is the state space and $\mathcal{A}$ is the action space. $P(s'|s,a)$ denotes the probability of state transition from s to s' after taking action a and $R(\cdot,\cdot) : \mathcal{S}\times\mathcal{A} \to [-R_{\max}, R_{\max}]$ is a bounded reward function, where $R_{\max}$ is a positive scalar. ρ represents the initial distribution on state space $\mathcal{S}$ and $\gamma \in (0,1)$ is the discount factor.

The parametric policy π_θ is a probability distribution over $\mathcal{S}\times\mathcal{A}$ with parameter $\theta \in \mathbb{R}^d$, and $\pi_\theta(a|s)$ denotes the probability of taking action a at a given state s. Let $\tau = \{s_t, a_t\}_{t\geq 0} \sim p(\tau|\pi_\theta)$ be a trajectory generated by the policy π_θ, where $p(\tau|\pi_\theta) := \rho(s_0)\prod_{t=0}^{\infty}\pi_\theta(a_t|s_t)P(s_{t+1}|s_t,a_t)$. The expected return of π is defined as $J(\pi_\theta) := \mathbb{E}_{\tau\sim p(\cdot|\pi_\theta)}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t,a_t)\right]$. In the sequel, we consider a set of parameterized policies $\{\pi_\theta : \theta \in \mathbb{R}^d\}$, with the assumption that π_θ is differentiable with respect to θ. For ease of presentation, we denote $J(\pi_\theta)$ by $J(\theta)$.

The goal of policy-based RL is to find $\theta^* = \arg\max_\theta J(\theta)$. However, in many cases, $J(\theta)$ is a non-concave function which might make the problem NP hard [6]. Therefore, instead, we settle for obtaining an ϵ-FOSP, $\hat{\theta}$, such that $\|\nabla J(\hat{\theta})\| \leq \epsilon$. It can be shown that: $\nabla J(\theta) = \mathbb{E}\left[\sum_{h=0}^{\infty}\left(\sum_{t=h}^{\infty}\gamma^t R(s_t,a_t)\right)\nabla\log\pi_\theta(a_h|s_h)\right]$. In practice, the full gradient cannot be computed due to the infinite horizon length. Instead, it is commonly truncated to a length H horizon as $\nabla J_H(\theta) = \mathbb{E}[\sum_{h=0}^{H-1}\Psi_h(\tau)\nabla\log\pi_\theta(a_h|s_h)]$, where $\Psi_h(\tau) = \sum_{t=h}^{H-1}\gamma^t R(s_t,a_t)$.

Assume that we sample m trajectories $\tau^i = \{s_t^i, a_t^i\}_{t\geq 0}$, $1 \leq i \leq m$, and then compute $\hat{\nabla}_m J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\sum_{h=0}^{H-1}\Psi_h(\tau^i)\nabla\log\pi_\theta(a_h^i|s_h^i)$, which is an unbiased estimator for $\nabla J_H(\theta)$. The vanilla SPG method is based on the following update: $\theta \leftarrow \theta + \eta\hat{\nabla}_m J(\theta)$, where η is the learning rate. Momentum-based variance reduction (MVR) is based on the following update:

$$\begin{aligned} d_t :=&(1-a)d_{t-1} + a\hat{\nabla}_m J(\theta_t) \\ &+ (1-a)(\hat{\nabla}_m J(\theta_t) - \bar{\nabla}_m J(\theta_{t-1})), \qquad (3)\\ x_{t+1} :=&x_t - \eta_t d_t, \qquad (4)\end{aligned}$$

where $\bar{\nabla}_m J(\theta_{t-1}) = \frac{1}{m}\sum_{i=1}^{m}\sum_{h=0}^{H-1}\Psi_h(\tau^i)\nabla\log\pi_{\theta_{t-1}}(a_h^i|s_h^i)$ and trajectories $\tau^i = \{s_l^i, a_l^i\}_{l\geq 0}$, $1 \leq i \leq m$ are generated by $\pi_{\theta_t}(a|s)$.

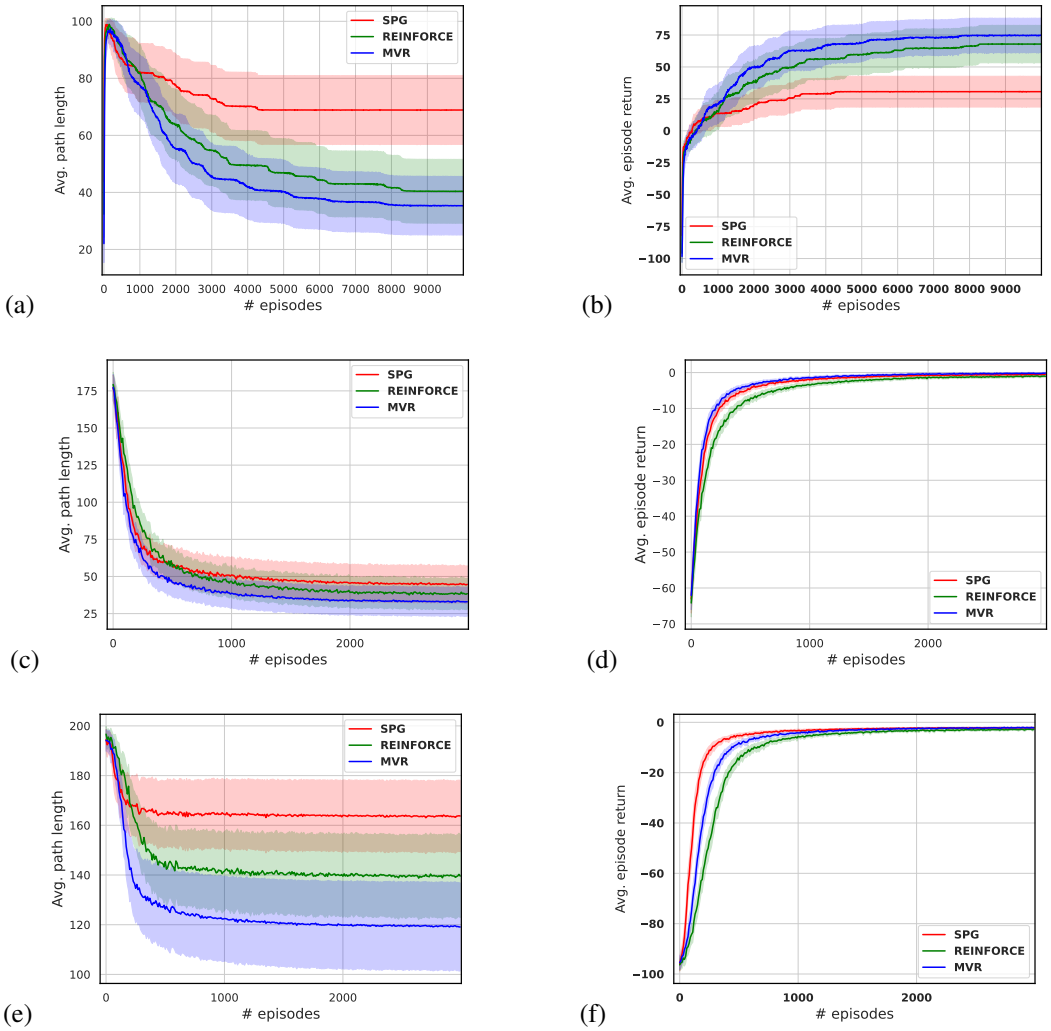

Fig. 1: (a), (b) demonstrate average path length and average episode return of MVR, SPG and REINFORCE in the Cliff environment. (c), (d) show the comparison of the algorithms in the Random shape maze environment. (e), (f) are plots for the random maze environment.

III. Experimental Setup

In this project, we evaluate the performance of MVR in the three following RL settings. We consider grid world environments with finite state and action spaces.

Environments with finite state and action spaces: We considered two grid world environments in our experiments: [7, Example 6.6], and random mazes [8]. In cliff walking, the agent's aim is to reach a goal state from a start state, avoiding a region of cells called *cliff*. The episode is terminated if the agent enters the cliff region or the number of steps exceeds 100 without reaching the goal. The reward is -0.1 in all transitions except those into the cliff where the reward is -100. The reward of reaching the goal is 100. Moreover, we considered a softmax tabular policy for all the experiments of this part.

In random mazes, the size of each maze is 10×10. In the random shape maze, random shape blocks are placed on a grid and the agent tries to reach the goal state finding the shortest path, avoiding blocks. The reward is -0.1 in all transitions except if the agent tries to go to a cell which belongs to a block where the reward is -1. Moreover, the reward of reaching the goal is 1. An episode is terminated if the agent could not reach the goal after 200 steps.

For the environments with finite state and action spaces, we provided an implementation of MVR, SPG, and REINFORCE with NumPy where the gradient of a given trajectory are computed based on their closed forms. In our experiments, we used a Linux server with Intel Xeon Gold 6240 CPU (36 cores) operating at 2.60GHz with 377 GB DDR4 of memory, and Nvidia Titan X GPU.

A. Hyperparameters

We always set the batch size to 1 as we do not study the mini-batch variants of the algorithms. Regarding MVR, SPG, and REINFORCE in grid world environments, we adapted a time-varying learning rate by checking various forms and the

form of $a/(\lfloor t/P \rfloor + b)$ provided the best performance for the first-order methods where a, b, and P are some constants that are needed to be tuned. The parameter P shows the number of episodes that the learning rate remains unchanged. Momentum parameter a in MVR is chosen the best constant ratio in each experiment.

IV. Experimental Results

We compare MVR with two existing first-order methods: vanilla SPG and REINFORCE [2]. For the optimization methods, we use a time-varying learning rate and tune the parameters.

In Fig. 1, the average length of paths traversed by the agent and the average episode return are depicted against the number of episodes for each method. The results are averaged over 64 instances and the shaded region shows the 90% confidence interval. As can be seen in Fig. 1 (a), (b), all the algorithms have a phase at the beginning during which the agent falls off the cliff in most episodes and the average length of paths are small. Then, the agent learns to avoid the cliff but could still not reach the goal in most cases. Finally, it finds a path to the goal and tries to reduce the path length. Note that MVR finds the path quickly and outperforms the other algorithms.

Additionally, we studied the performance of the three aforementioned algorithms on a random maze and a random shape maze [8]. In the random shape maze (See Figures 1 (c), (d)), random shape blocks are placed on a grid and the agent tries to reach the goal state finding the shortest path. As shown in Fig. 1 (c), (d), (e), and (f), MVR again outperforms SPG and REINFORCE in both environments.

References

[1] A. Cutkosky and F. Orabona, "Momentum-based variance reduction in non-convex sgd," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/b8002139cdde66b87638f7f91d169d96-Paper.pdf

[2] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3, pp. 229–256, 1992.

[3] J. Zhang, J. Kim, B. O'Donoghue, and S. Boyd, "Sample efficient reinforcement learning with reinforce," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, 2021, pp. 10 887–10 895.

[4] Y. Ding, J. Zhang, and J. Lavaei, "On the global optimum convergence of momentum-based policy gradient," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 1910–1934.

[5] J. Mei, C. Xiao, C. Szepesvari, and D. Schuurmans, "On the global convergence rates of softmax policy gradient methods," in *International Conference on Machine Learning*. PMLR, 2020, pp. 6820–6829.

[6] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

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

[8] X. Zuo, "mazelab: A customizable framework to create maze and gridworld environments," 2018.