A Note on Stochastic Variance-Reduced Policy Gradient

Hsuan Wang
Department of Computer Science
National Yang Ming Chiao Tung University
sshhaawwnn111.cs08@nycu.edu.tw

1 Introduction

On a very general level, artificial intelligence addresses the problem of an agent that must select the right actions to solve a task.

The approach of Reinforcement Learning is to learn the best actions by direct interaction with the environment and evaluation of the performance in the form of a reward signal. But often times the data available for training is a subset of all the cases of interest, which can be infinite. In this case we need stochastic sampling to approximate the expected performance of the unknown distribution.

However randomness introduces variance that can potentially compromise convergence. So there is a trade-off between per-iteration efficiency and convergence that need to be properly handled with meta-parameters.

Stochastic Variance-Reduced Policy Gradient(SVRPG) tackles the problem of variance by adapting Stochastic variance-reduced gradient(SVRG) in Supervised Learning(SL) to the settings of Reinforcement Learning(RL). However the adaptation is not straightforward and needs to account for I) often non-concave objective function in RL problems;II) approximations in the full gradient computation;III) a non-stationary sampling process.

Compared to prior works such as using baseline for variance reduction, SVRPG has a slight performance advantage over it, and they are orthogonal so they can be used jointly to boost the performance even further at the cost of more computational power.

From the experiment results that they showed at the end, it seems like this method do have some really nice performance, but as we will see in section 4 that some assumptions have to be made for guarantee of convergence.

2 Problem Formulation

2.1 Policy Gradient

A Reinforcement Learning task can be modeled with a *Markov decision process*(MDP) $M = (S, A, P, R, \gamma, \rho)$ where S is a continuous state space; A is a continuous action space; P is a Markovian transition model; R is the reward function; $\gamma \in [0, 1)$ is the discount factor; ρ is the initial state distribution. The agent's behaviour is modeled as a policy π, where $\pi(\cdot|s)$ is the density distribution over A in state s. A trajectory τ is a sequence of states and actions $(s_0, a_0, r_0, s_1, a_1, r_1, \ldots)$ observed by following a stationary policy where $s_o \sim \rho$. We denote the density distribution of all the trajectories induced by policy π as $p(\tau|\pi)$, and with $R(\tau)$ the total discounted reward of a trajectory τ. From the above notation we can rank the policies with there expected total reward:

$$J(\pi) = \mathop{\mathbb{E}}_{\tau \sim p(\cdot|\pi)} [R(\tau)] \tag{1}$$

We denote the parameter of a policy net $\theta \sim \mathbb{R}^d$. The performance of this parameter will be denoted by $J(\theta)$ and the probability of a trajectory $p(\cdot|\theta)$. From here we have the gradient of $J(\theta)$ w.r.t θ as:

$$\nabla J(\pi) = \mathop{\mathbb{E}}_{\tau \sim p(\cdot|\pi)} [R(\tau) \nabla \log p_\theta(\tau)] \tag{2}$$

Because the underlying distribution of trajectories changes overtime, it is necessary to resample at each update or use techniques like importance weight. In the case of resampling, stochastic gradient descent is typically used. At each iteration $k > 0$, a batch $D_N^k = \{\tau_i\}_{i=0}^N$ of $N > 0$ is collected using π_{θ_k}. The policy is then updated as $\theta_{k+1} = \theta_k + \alpha \widehat{\nabla}_N J(\theta_k)$ where α is the step size and $\widehat{\nabla}_N J(\theta)$ is an estimate of Eq.(2) using D_N^k:

$$\widehat{\nabla}_N J(\theta) = \frac{1}{N} \sum_{n=1}^{N} g(\tau_i|\theta), \quad \tau_i \in D_N^k \tag{3}$$

$g(\tau|\theta)$ is an estimator of $R(\tau) \nabla \log p_\theta(\tau)$ using either the REINFORCE or G(PO)MDP definition.

2.2 Stochastic Variance-Reduced Gradient

The optimization problem of maximize or minimize a finite-sum objective function $f(\theta)$ can be written as:

$$\max_\theta \left\{ f(\theta) = \frac{1}{N} \sum_{i=1}^{N} z_i(\theta) \right\} \tag{4}$$

where each z_i may correspond to a data sample x_i from a dataset D_N of size N (i.e, $z_i(\theta) = z(x_i|\theta)$) where x_i is sampled uniformly at random from D_N. A common requirement is that z must be smooth and concave in θ. However, each iteration requires N gradient computations, which can be a heavy loading for large values of N. This is where Stochastic Gradient Descent(SGD) comes into play, but with lower computation cost comes variance that could tamper with the convergence of the algorithm. SVRG deals with this problem by updating with the following rule:

$$\theta_t = \theta_{t-1} - \eta \left(\nabla z_i(\theta_{t-1}) - \nabla z_i(\tilde{\theta}) + \nabla f(\tilde{\theta}) \right) \tag{5}$$

where $\tilde{\theta}$ is a snapshot of the parameter that is updated every m SGD iterations.

If the snapshot $\tilde{\theta}$ is close to optimal(denote with θ^*), $\nabla z_i(\tilde{\theta}) \to \nabla z_i(\theta^*)$:

- Let $\tilde{u} := \nabla f(\tilde{\theta}),\ \tilde{u} - \nabla f(\theta_{t-1}) \approx \nabla z_i(\tilde{\theta}) - \nabla z_i(\theta_{t-1})$
- Updating with $\nabla f(\theta_{t-1}) = \nabla f(\theta_{t-1}) - \tilde{u} + \tilde{u} \approx \nabla z_i(\theta_{t-1}) - \nabla z_i(\tilde{\theta}) + \tilde{u}$.
  Intuitively, this updating rule cancel the randomness induced by random sampling.
- $\tilde{u} \to 0$ when $\tilde{\theta} \to \theta^*,\ \nabla z_i(\theta_{t-1}) - \nabla z_i(\tilde{\theta}) + \tilde{u} \to \nabla z_i(\theta_{t-1}) - \nabla z_i(\theta^*) \to 0$.
  The infinite small gradient allows to use constant learning rate.

2.3 Stochastic Variance-Reduced Gradient

Now, in order to apply this into an RL setting, there are three problems that need to be handled as mentioned in the introduction.

1. Non-concavity: Assumes the objective function $J(\theta)$ is L-smooth. However, the following assumption is sufficient

   Assumption 1. *(On policy derivatives). For each state-action pair (s, a), any value of θ, and all parameter components i, j there exist constants $0 \leq G, F < \infty$ such that:*

$$|\nabla_{\theta_i} \log \pi_\theta(a|s)| \leq G, \qquad |\frac{\partial^2}{\partial \theta_i \partial \theta_j} \log \pi_\theta(a|s)| \leq F.$$

2. Infinite dataset: This mean we can only use an estimate of the full gradient in the algorithm(see Eq. (3)), and others works (e.g., Babanezhad Harikandeh et al. [2015]) analyzed this under the assumption of z being concave, shows that SVRG is robust under the error of estimation. But for the sake of convergence we still need the variance of the estimator to be bounded.

   Assumption 2. *(On the variance of the gradient estimator). There is a constant $V < \infty$ such that, for any policy π_θ:*

$$\mathbb{V}ar\left[g(\cdot|\theta)\right] \leq V.$$

3. Non-stationarity: This is solved using the importance rate, thus eliminating the need for resampling. We need the variance introduced by this technique to be bounded.

   Assumption 3. *(On the variance of importance weights). There is a constant $W < \infty$ such that, for each pair of policies encountered and for each trajectory:*

$$\mathbb{V}ar\left[w(\tau|\theta_1, \theta_2)\right] \leq W, \qquad \forall \theta_1, \theta_2 \in \mathbb{R}^d, \tau \sim p(\cdot|\theta_1).$$

Finally, we have the SVRG in RL with the updating rule:

$$\theta_t = \theta_{t-1} + \eta \left(\widehat{\nabla}_N J(\tilde{\theta}) + \frac{1}{B} \sum_{i=0}^{B-1} \left[g(\tau_i|\theta_t) - w(\tau_i|\theta_t, \tilde{\theta}) g(\tau_i|\tilde{\theta}) \right] \right) \tag{6}$$

Where $\widehat{\nabla}_N J(\tilde{\theta})$ is the same as Eq. (3). $\tilde{\theta}$ is the snapshot of the parameter that is updated every m SGD. w is the importance weight where $w(\tau_i|\theta_t, \tilde{\theta}) = \frac{p(\tau|\tilde{\theta})}{p(\tau|\theta_t)}$.

As we know the importance sampling is another source of variance (e.g.,Thomas et al. [2015]), to mitigate this, we used a mini-batch $B \ll N$ to average the correction term (e.g., Babanezhad Harikandeh et al. [2015];Avdiukhin and Kasiviswanathan [2021]), we denote this update with:

$$\widetilde{\nabla} J(\theta_t) = \widehat{\nabla}_N J(\tilde{\theta}) + \frac{1}{B} \sum_{i=0}^{B-1} \left[g(\tau_i|\theta_t) - w(\tau_i|\theta_t, \tilde{\theta}) g(\tau_i|\tilde{\theta}) \right]$$

3 Theoretical Analysis

The Full Gradient estimator $\widehat{\nabla}_N J\left(\theta\right) = \frac{1}{N} \sum_{n=1}^{N} g(\tau_i|\theta)$ is an unbiased estimator.

Proof. (This proof is not provided by the paper.)

$$\begin{aligned}
\mathop{\mathbb{E}}_{\tau \sim p(\cdot|\pi)} \left[\widehat{\nabla}_N J\left(\theta\right) \right] &= \mathop{\mathbb{E}}_{\tau \sim p(\cdot|\pi)} \left[\frac{1}{N} \sum_{n=1}^{N} g(\tau_i|\theta) \right] \\
&= \frac{1}{N} \sum_{n=1}^{N} \mathop{\mathbb{E}}_{\tau \sim p(\cdot|\pi)} \left[g(\tau_i|\theta) \right] \\
&= \frac{1}{N} \sum_{n=1}^{N} \nabla J\left(\theta\right) \\
&= \nabla J\left(\theta\right)
\end{aligned}$$

□

The importance sampling term in the SVRPG has a the same expected value as the full gradient:

Proof. (This proof is not provided by the paper.)

$$\begin{aligned}
\mathop{\mathbb{E}}_{\tau_i\sim p(\cdot|\theta_t)}\left[\frac{1}{B}\sum_{i=0}^{B-1}\left[w(\tau_i|\theta_t,\tilde{\theta})g(\tau_i|\tilde{\theta})\right]\right] &= \frac{1}{B}\sum_{i=0}^{B-1}\left[\mathop{\mathbb{E}}_{\tau_i\sim p(\cdot|\theta_t)}\left[w(\tau_i|\theta_t,\tilde{\theta})g(\tau_i|\tilde{\theta})\right]\right] \\
&= \frac{1}{B}\sum_{i=0}^{B-1}\left[\sum_{\tau_i\in\tau}\left[p(\tau_i|\theta_t)w(\tau_i|\theta_t,\tilde{\theta})g(\tau_i|\tilde{\theta})\right]\right] \\
&= \frac{1}{B}\sum_{i=0}^{B-1}\left[\sum_{\tau_i\in\tau}\left[p(\tau_i|\theta_t)\frac{p(\tau_i|\tilde{\theta})}{p(\tau_i|\theta_t)}g(\tau_i|\tilde{\theta})\right]\right] \\
&= \frac{1}{B}\sum_{i=0}^{B-1}\left[\sum_{\tau_i\in\tau}\left[p(\tau_i|\tilde{\theta})g(\tau_i|\tilde{\theta})\right]\right] \\
&= \frac{1}{B}\sum_{i=0}^{B-1}\left[\mathop{\mathbb{E}}_{\tau_i\sim p(\cdot|\tilde{\theta})}\left[p(\tau_i|\tilde{\theta})g(\tau_i|\tilde{\theta})\right]\right] \\
&= \mathop{\mathbb{E}}_{\tau_i\sim p(\cdot|\tilde{\theta})}\left[p(\tau_i|\tilde{\theta})g(\tau_i|\tilde{\theta})\right] = \nabla J(\tilde{\theta})
\end{aligned}$$

where τ is the set of all possible trajectories. □

Lemma 1. *With the unbiased estimator $\widehat{\nabla}_N J(\theta)$, the SVRPG update is an unbiased update:*

$$\mathbb{E}\left[\widehat{\nabla}_N J(\tilde{\theta}) + \frac{1}{B}\sum_{i=0}^{B-1}\left[g(\tau_i|\theta_t) - w(\tau_i|\theta_t,\tilde{\theta})g(\tau_i|\tilde{\theta})\right]\right] = \nabla J(\tilde{\theta}).$$

Proof. (This proof is not provided by the paper.)

$$\mathbb{E}\left[\widetilde{\nabla} J(\theta_t)\right] = \mathbb{E}\left[\widehat{\nabla}_N J(\tilde{\theta}) + \frac{1}{B}\sum_{i=0}^{B-1}\left[g(\tau_i|\theta_t) - w(\tau_i|\theta_t,\tilde{\theta})g(\tau_i|\tilde{\theta})\right]\right]$$

$$\begin{aligned}
&= \mathbb{E}\left[\widehat{\nabla}_N J(\tilde{\theta})\right] + \mathbb{E}\left[\frac{1}{B}\sum_{i=0}^{B-1}\left[g(\tau_i|\theta_t)\right]\right] - \mathbb{E}\left[\frac{1}{B}\sum_{i=0}^{B-1}\left[w(\tau_i|\theta_t,\tilde{\theta})g(\tau_i|\tilde{\theta})\right]\right] \\
&= \nabla J(\tilde{\theta}) + \nabla J(\theta_t) - \nabla J(\tilde{\theta}) \\
&= \nabla J(\theta_t)
\end{aligned}$$

□

Theorem 1. *(Convergence of the SVRPG algorithm). Under Assumptions 1, 2, 3, the parameter θ returned by the algorithm has, for some positive constants ψ, ζ, ξ, and for proper choice of the step size α and the epoch size m, the following property:*

$$\mathbb{E}\left[||\nabla J(\theta_A)||_2^2\right] < \frac{J(\theta^*) - J(\theta_0)}{\psi T} + \frac{\zeta}{N} + \frac{\xi}{B},$$

where θ^* is a global optimum and $T = m \times S$, ψ, ζ, ξ depend only on G, F, V, W, α and m.

From this result we can see that the expected value of the gradient has an upper-bound $\frac{J(\theta^*)-J(\theta_0)}{\psi T} + \frac{\zeta}{N} + \frac{\xi}{B}$ that must converge to 0. From observation we can see that there are 3 big O terms $O(\frac{1}{T}), O(\frac{1}{N}), O(\frac{1}{B})$.

I) $O(\frac{1}{T})$ is intuitive in a sense that the more iteration you compute the closer it gets to 0, the better the result. It is also coherent with prior works on convergence of non-concave SVRG (e.g., Reddi et al. [2016]). II) $O(\frac{1}{N})$ is due to the N sampled trajectory to do full gradient approximation at the start of every epoch, intuitively the more trajectories you sample the less variance in the approximation. III) $O(\frac{1}{B})$ is due to the importance sampling. Recall that we used the average of a mini-batch in every

iteration to mitigate the effect of variance introduced by importance sampling, similarly the bigger the mini-batch the less variance.

As mentioned above, at the very start of each epoch is a full gradient approximation, since $\theta_{t=0} = \tilde{\theta}$, the B trajectories sampled in the first iteration of every epoch seems like a waste, so here is the updated, more practical version of Eq .(6):

$$\theta_1 = \tilde{\theta} + \eta \widehat{\nabla}_N J(\tilde{\theta})$$

$$\theta_t = \theta_{t-1} + \eta \left(\widehat{\nabla}_N J(\tilde{\theta}) + \frac{1}{B} \sum_{i=0}^{B-1} \left[g(\tau_i|\theta_t) - w(\tau_i|\theta_t, \tilde{\theta}) g(\tau_i|\tilde{\theta}) \right] \right) \quad for\ t = 2, \ldots, m,$$

Combining everything together, we have the SVRPG algorithm:

Algorithm 1: Stochastic Variance-Reduced Policy Gradient

Input: number of epochs S, epoch size m, step size α, batch size N, mini-batch size B, gradient estimator g, initial parameter $\theta_m^0 := \tilde{\theta}^0$

for $s = 0$ *to* $S - 1$ *do* **do**
  $\tilde{\theta}^s := \theta_m^s$
  $\theta_0^{s+1} := \tilde{\theta}^s$
  Sample N trajectories τ_i from $p(\cdot|\tilde{\theta}^s)$
  $\tilde{u} = \widehat{\nabla}_N J(\tilde{\theta}^s)$ see Eq. (3)
  for $t = 1$ *to m do* **do**
    if $t = 1$ **then**
      $\theta_t = \tilde{\theta} + \eta \widehat{\nabla}_N J(\tilde{\theta})$
    else
      $\theta_t = \theta_{t-1} + \eta \left(\widehat{\nabla}_N J(\tilde{\theta}) + \frac{1}{B} \sum_{i=0}^{B-1} \left[g(\tau_i|\theta_t) - w(\tau_i|\theta_t, \tilde{\theta}) g(\tau_i|\tilde{\theta}) \right] \right)$
    end
  end
end

4 Conclusion

SVRPG provides a way of reducing variance in RL problem, where interacting with the environment and collecting data is very costly, accomplishing a better performance under limited resource, moreover, it can be combined with the traditional baseline variance reduction method. Although there is quite a bit of assumptions made in order to achieve convergence guarantee, they are reasonable assumptions and can be met by most RL problems in my opinion. Finally, experiment empirically shows that SVRPG does have a noticeable advantage over traditional actor-only methods.

There is still things that can improve SVRPG even further that I have not discuss in this paper such as adaptive step size and adaptive epoch length. Or lower the variance by normalizing the importance weight (e.g., Tirinzoni et al. [2019]) at the cost of some bias. Future direction may consider applying this to an actor-critic framework, replacing $g(\tau|\theta)$ with an approximation by the critic.

References

Reza Babanezhad Harikandeh, Mohamed Osama Ahmed, Alim Virani, Mark Schmidt, Jakub Konečnỳ, and Scott Sallinen. Stopwasting my gradients: Practical svrg. *Advances in Neural Information Processing Systems*, 28, 2015.

Philip Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High confidence policy improvement. In *International Conference on Machine Learning*, pages 2380–2388. PMLR, 2015.

Dmitrii Avdiukhin and Shiva Kasiviswanathan. Federated learning under arbitrary communication patterns. In *International Conference on Machine Learning*, pages 425–435. PMLR, 2021.

Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, pages 314–323. PMLR, 2016.

Andrea Tirinzoni, Mattia Salvini, and Marcello Restelli. Transfer of samples in policy search via multiple importance sampling. In *International Conference on Machine Learning*, pages 6264–6274. PMLR, 2019.