Stochastic Variance Reduction for Policy Gradient Estimation

Tianbing Xu, Qiang Liu, Jian Peng

Baidu Research, CA, UT, Austin, UIUC

November 7, 2018

Reinforcement Learning

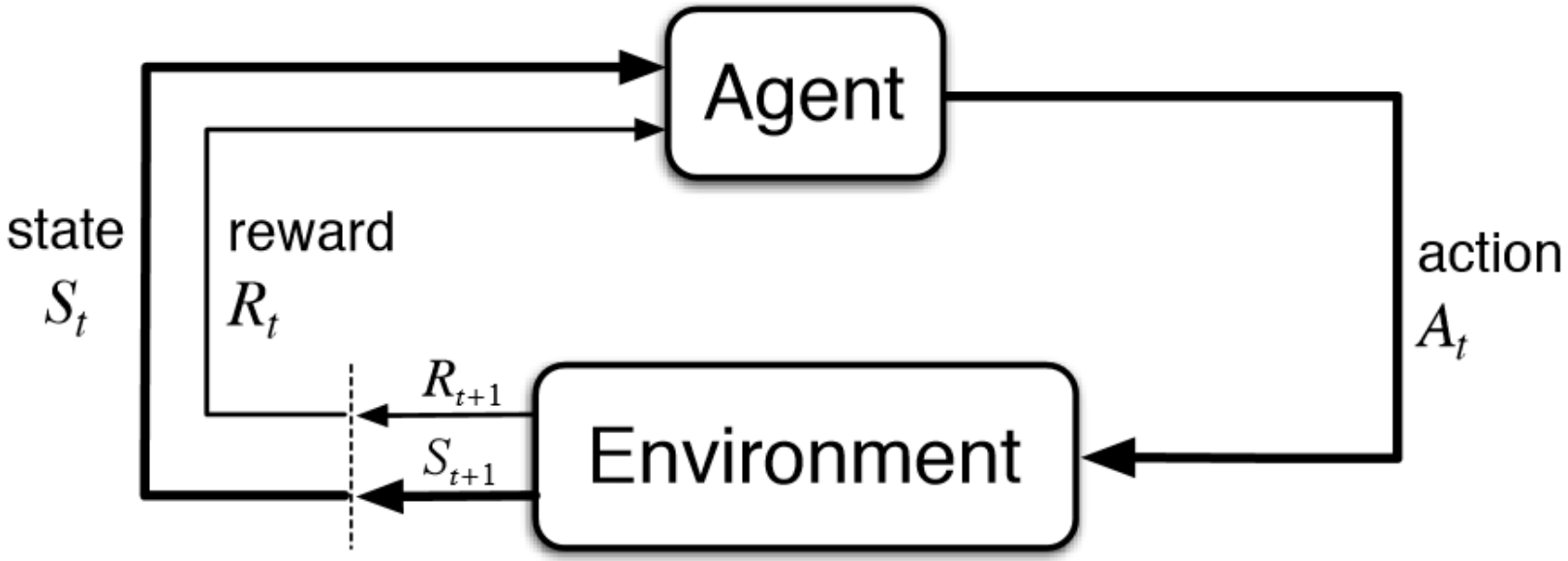


Figure: RL framework

Policy Optimization

Objective:

$$\max_{\theta} E[\sum_{t} r(s_t, a_t)|\pi_\theta]$$

Policy (parameterized as Neural Network):

$$\pi_\theta(a|s)$$

MDP (Markov Decision Process)

$$< \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \gamma >$$

- $\mathcal{S}$: state
- $\mathcal{A}$: action
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to R$
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$

Vanilla Policy Gradient - REINFORCE I

Objective (max the returns):

$$U(\theta) = E_{\pi_\theta}[R(\tau)] = \sum_{\tau} P_\theta(\tau) R(\tau)$$

Given trajectory (transition sequence) generated by exec policy π_θ:

$$\tau = \{s_0, a_0, r_0, ..., s_T, a_T, r_T\}$$

The Reward:

$$R(\tau) = \sum_{t} r(s_t, a_t)$$

Vanilla Policy Gradient - REINFORCE II

Policy Gradient:

$$\nabla_\theta U(\theta) = E_{\tau\sim\pi_\theta}[\nabla_\theta \log P_\theta(\tau) R(\tau)]$$

The derivation:

$$\begin{aligned}
\nabla_\theta U(\theta) &= \nabla_\theta \sum_\tau P_\theta(\tau) R(\tau) \\
&= \sum_\tau \frac{\nabla_\theta P_\theta(\tau)}{P_\theta(\tau)} P_\theta(\tau) R(\tau) \\
&= E_{\tau\sim\pi_\theta}[\nabla_\theta \log P_\theta(\tau) R(\tau)]
\end{aligned}$$

Problem: High Variance

$$\nabla_\theta U(\theta) = E_{\tau \sim \pi_\theta}[\nabla_\theta \log P_\theta(\tau) R(\tau)]$$

$$\log P_\theta(\tau) = \sum_t \log \pi(a_t | s_t)$$

$$R(\tau) = \sum_t r(s_t, a_t)$$

- Vanilla Policy Gradient Estimation has very high variance
- long horizon, stochastic and noisy environment, stochastic policy
- Sample in-efficiency, need huge number of samples to learn the policy network

Stochastic Variance Reduction (Jonhson and Tong, NIPS 2013, Owen and Zhou, JASA 2000)

To find estimation:

$$\mu = E[f(X)]$$

Monte Carlo Estimation:

$$\hat{\mu} = \frac{1}{m} f(X_i)$$

Control Variate : $h(X)$ is close to $f(X)$

$$\nu = E[h(X] \quad \hat{\nu} = \frac{1}{m} h(X_i)$$

Stochastic Variance Reduction Estimation:

$$\hat{\mu}_{sv} = \nu + (\hat{\mu} - \hat{\nu})$$

Unbiased estimation:

$$E(\hat{\mu}_{sv}) = E[\hat{\mu}] = \mu$$

Policy Gradient Variance Reduction Estimation

Variance Reduction Policy Gradient Estimation:

$$\hat{\nabla}_{sv} U(\theta) = \hat{\nabla} U(\tilde{\theta}) + \left(\hat{\nabla}_m U(\theta) - \hat{\nabla}_m U(\tilde{\theta}) \right)$$

Policy Gradient Estimation:

$$\hat{\nabla}_m U(\theta) = \frac{1}{m} \sum_i U_i(\theta)$$

Control Variate $\tilde{\theta}$ (close to policy network parameter θ) :

$$\hat{\nabla}_m U(\tilde{\theta}) = \frac{1}{m} \sum_i U_i(\tilde{\theta})$$

Unbiased Estimation:

$$E[\hat{\nabla}_{sv}(\theta)] = E[\hat{\nabla}_m U(\theta)] = \nabla U(\theta)$$

Variance Reduction

Variance of Monte Carlo Estimation:

$$Var(\hat{\nabla}_m U(\theta)) = \frac{1}{m} Var(\nabla U(\theta))$$

Stochastic Variance Reduction:

$$Var(\hat{\nabla}_{sv} U(\theta)) \approx \frac{1}{m} Var\left(\nabla U(\theta) - \nabla U(\tilde{\theta})\right)$$

when $\tilde{\theta}$ is close to θ, the Variance is close to 0.
The derivation:

$$\begin{aligned} Var(\hat{\nabla}_{sv} U(\theta)) &= E[\left(\hat{\nabla} U(\tilde{\theta}) + \hat{\nabla}_m U(\theta) - \hat{\nabla}_m U(\tilde{\theta}) - \nabla U(\theta)\right)^2] \\ &\approx E[\left(\left(\hat{\nabla}_m U(\theta) - \hat{\nabla}_m U(\tilde{\theta})\right) - E\left(\hat{\nabla}_m U(\theta) - \hat{\nabla}_m U(\tilde{\theta})\right)\right)^2] \\ &= Var(\hat{\nabla}_m U(\theta) - \hat{\nabla}_m U(\tilde{\theta})) = \frac{1}{m} Var(\nabla U(\theta) - \nabla U(\tilde{\theta})) \end{aligned}$$

Stochastic Variance Reduction Algorithm

Algorithm 1 Stochastic Variance Reduction for Policy Optimization

for $\ell = 1$ to L **do**
  Initialize the parameter from control variate: $\theta^{\ell} = \tilde{\theta}^{\ell}$.
  Generate rollouts by exec the current policy $\pi_{\theta^{\ell}}$
  for $j = 1$ to J **do**
    Draw an uniformly random mini-batch I_j (with size m)
    Calculate the SVRG Policy Gradient estimation:

$$\hat{\nabla}_{sv} U(\theta^{\ell}) = \frac{1}{N} \sum_{i} \nabla U_i(\tilde{\theta}^{\ell}) + \frac{1}{m} \sum_{i \in I_j} \left(\nabla U_i(\theta^{\ell}) - \nabla U_i(\tilde{\theta}^{\ell}) \right).$$

    Update policy parameter with mini-batch I_j:

$$\theta^{\ell} \leftarrow \theta^{\ell} + \eta \hat{H}^{-1}(\theta^{\ell}) \hat{\nabla}_{sv} U(\theta^{\ell}).$$

  end for
  $\tilde{\theta}^{\ell+1} \leftarrow \theta^{\ell}$
end for

Connection to TRPO and Natural Policy Gradient

TRPO or NPO adopt Fisher information matrix with Monte Carlo Policy Gradient:

$$\theta \leftarrow \theta + \eta \hat{H}^{-1}(\theta)\hat{\nabla} U(\theta).$$

Additionally, we adopt Stochastic Variance Reduction Policy Gradient:

$$\theta \leftarrow \theta + \eta \hat{H}^{-1}(\theta)\hat{\nabla}_{sv} U(\theta).$$

To reduce the variance of policy gradient.

TRPO (Schulman et. al. ICML 2015)

$$\max_{\theta} L(\theta) = E_{\theta_{old}} \left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)} A^{\pi_{\theta\, old}}(a|s) \right]$$
$$s.t. \quad KL(\pi_{\theta_{old}}, \pi_\theta) \leq \delta$$

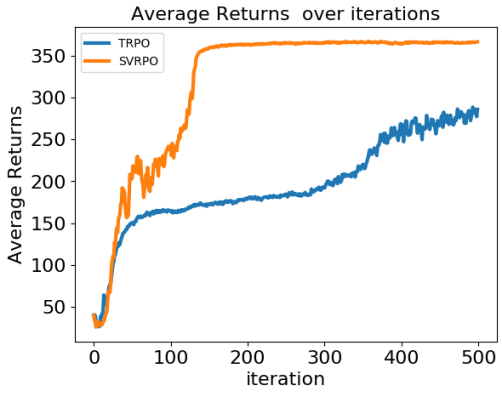
The surrogate objective $L(\theta)$ is the first order approximation to $U(\theta)$,

$$L(\theta) = U(\theta) \quad \nabla_\theta L(\theta) = \nabla_\theta U(\theta)$$

KL divergence is used as the trust region to stabilize the performance. Step size is bounded,

$$\eta \leq \sqrt{\frac{2\delta}{s' H s}}$$

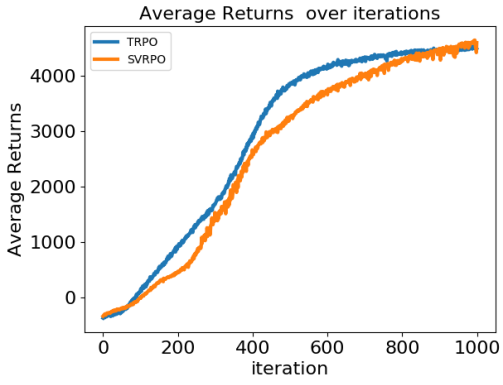
Results: Sample Efficiency and Higher Return



(a) Swimmer.

(b) Walker

(c) Hopper



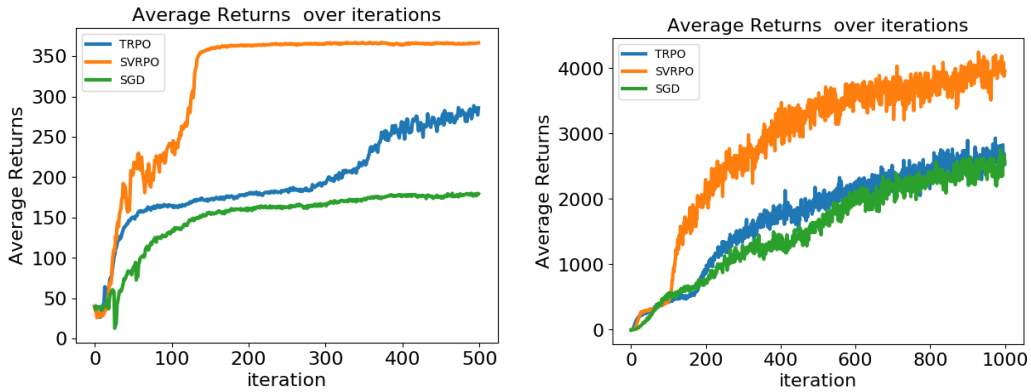
(d) Half-Cheetah

(e) Ant

(f) Humanoid

Figure: Performance Comparison of Variance Reduction and TRPO for six Mujoco Control Tasks.

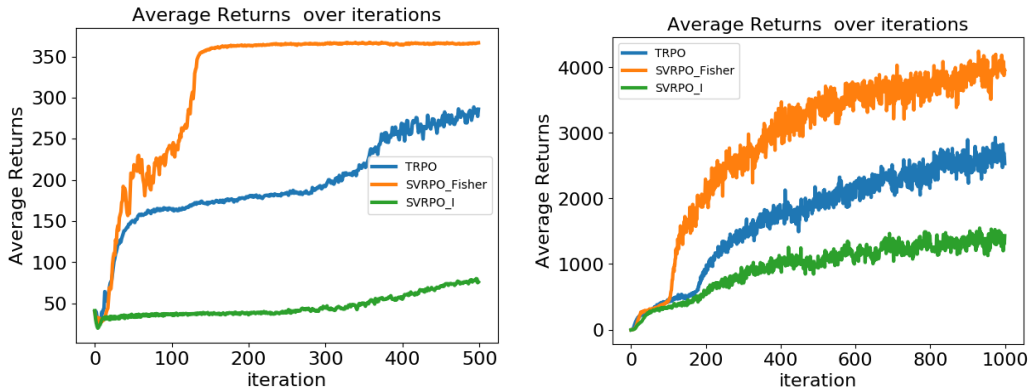
Results: Ablation Study (SVRG vs SGD)



(a) Swimmer. (b) Walker

Figure: Performance Advantage of Variance Reduction Policy gradient (SVRG vs vanilla SGD in our algorithm) with baseline TRPO for Swimmer and Walker.

Results: Ablation Study (Fisher Matrix)



(a) Swimmer. (b) Walker.

Figure: Fisher information matrix's accelerated convergence rates on our algorithm with baseline TRPO for Swimmer and Walker.

Running Video

- Swimmer
- Walker
- Hopper
- Half-Cheetah
- Ant

For Further Reading

Rie Johnson and Tong Zhang

Accelerating stochastic gradient descent using predictive variance reduction

NIPS 2013

Simon S. Du, Jianshu Chen, Lihong Li, Lin Xiao, Dengyong Zhou

Stochastic Variance Reduction Methods for Policy Evaluation

ICML 2017

Art Owen and Yi Zhou

Safe and Effective Importance Sampling

Journal of the American Statistical Association, 2000

For Further Reading

John Schulman and Sergey Levine and Philipp Moritz and Michael Jordan and Michael Jordan
Trust Region Policy Optimization
ICML 2015

Ronald J. Williams
simple statistical gradient following algorithms for connectionist reinforcement learning
Machine Learning 1992