Variance Reduction and Deep Policy-Gradient Algorithms

Variance Reduction (Control Variates Method)

We are interested in computing

$$\mathbb{E}[f(x)] \approx \underbrace{\frac{1}{N}\sum_{i=1}^{N} f(x_i)}_{F} \quad x_i \sim P(x)$$

But F (finite-sample average) may have a high variance.

Variance Reduction

Replace F with F' such that

$$\mathbb{E}[F] = \mathbb{E}[F'],\ Var(F') \leq Var(F)$$

Variance Reduction

Consider function $\phi(x)$ such that $\mathbb{E}[\phi(x)] = 0$,

$$\mathbb{E}[F(x) - \phi(x)] = \mathbb{E}[F(x)]$$

$$Var(F(x) - \phi(x)) = Var(F(x)) - 2Cov(F(x), \phi(x)) + Var(\phi(x))$$

The variance can be reduced when $\phi(x)$ is strongly correlated with $F(x)$.

Estimate $\nabla_w V(i)$

$$\nabla_w V(i) = \sum_{k=0}^{T-1} \alpha^k \nabla \log \pi_w(u_k|x_k) \times (\underbrace{r(x_k, u_k) + \alpha V_w(x_{k+1}) - V_w(x_k)}_{\text{TD error}}).$$

Variance Reduction

Variance Reduction Theorem

$$\begin{aligned}\nabla_w V_w(i) =& E_{x_0=i, u_k\sim\pi_w(u_k|x_k)}\left[\sum_{k=0}^{\infty}\alpha^k \nabla_w \log \pi_w(u_k|x_k) Q_w(x_k,u_k)\right] \\ =& E_{x_0=i, u_k\sim\pi_w(u_k|x_k)}\left[\sum_{k=0}^{\infty}\alpha^k \nabla_w \log \pi_w(u_k|x_k) \left(Q_w(x_k,u_k) - G_w(x_k)\right)\right].\end{aligned}$$

To prove the theorem, we will show

$$E\left[\nabla_w \log \pi_w(u_k|x_k) G_w(x_k)\right] = 0.$$

Variance Reduction

Proof

$$
\begin{aligned}
E\left[\nabla_w \log \pi_w(u_k|x_k) G_w(x_k)\right] &= \sum_{x,u} \nabla_w \log \pi_w(u|x) G_w(x) \pi_w(u|x) P(x_k = x|x_0 = i) \\
&= \sum_{x,u} \frac{\nabla_w \pi_w(u|x)}{\pi_w(u|x)} G_w(x) \pi_w(u|x) P(x_k = x|x_0 = i) \\
&= \sum_{x,u} \nabla_w \pi_w(u|x) G_w(x) P(x_k = x|x_0 = i) \\
&= E_{x_k}\left[\sum_u \nabla_w \pi_w(u|x_k) G_w(x_k)\right] \\
&= E_{x_k}\left[G_w(x_k) \nabla_w \left(\sum_u \pi_w(u|x_k)\right)\right]
\end{aligned}
$$

Variance Reduction

Proof

$$
\begin{aligned}
E\left[\nabla_w \log \pi_w(u_k|x_k) G_w(x_k)\right] &= \sum_{x,u} \nabla_w \log \pi_w(u|x) G_w(x) \pi_w(u|x) P(x_k = x|x_0 = i) \\
&= \sum_{x,u} \frac{\nabla_w \pi_w(u|x)}{\pi_w(u|x)} G_w(x) \pi_w(u|x) P(x_k = x|x_0 = i) \\
&= \sum_{x,u} \nabla_w \pi_w(u|x) G_w(x) P(x_k = x|x_0 = i) \\
&= E_{x_k}\left[\sum_u \nabla_w \pi_w(u|x_k) G_w(x_k)\right] \\
&= E_{x_k}\left[G_w(x_k) \nabla_w (1)\right] \\
&= 0
\end{aligned}
$$

Variance Reduction

Estimate $\nabla_w V(i)$

$$\nabla_w V(i) = \sum_{k=0}^{T-1} \alpha^k \nabla \log \pi_w(u_k|x_k) \times (\underbrace{r(x_k, u_k) + \alpha V_w(x_{k+1}) - V_w(x_k)}_{\text{TD error}}).$$

Actor-Critic

- Advantage Actor-Critic:

$$A(s, a) = Q_w(s, a) - V_{\tilde{w}}(s)$$

- TD Actor-Critic:

$$A(s, a) = r(s, a) + \alpha V_w(s') - V_w(s)$$

TD error estimates the advantage function.

Actor-Critic with Neural Networks

Critic

double-Q, target-Q, and clipped-Q

Actor-Critic with Neural Networks

Critic

double-Q, target-Q, and clipped-Q

Actor

Sample a mini-batch sampled from replay buffer. Train the policy network with a weighted cross-entropy loss:

$$L(\theta) = -\sum_{(s,a)} A(s,a) \log \pi_\theta(a|s)$$

where

$$A(s,a) = r(s,a) + \alpha V_w(s') - V_w(s).$$

Policy Gradient

$$w \leftarrow w + \beta \nabla_w V_w(x_0)$$

Policy Gradient: An Optimization Perspective

Solve the following optimization problem

$$\max_{\tilde{w}} V_w(x_0) + \nabla_w V_w(x_0)(\tilde{w} - w) - \frac{1}{2\beta}(\tilde{w} - w)^\top \mathbb{I}(\tilde{w} - w).$$

Note that $V_w(x_0) + \nabla_w V_w(x_0)(\tilde{w} - w)$ is the linear approximation of function $V_{\tilde{w}}(x_0)$ and

$$(\tilde{w} - w)^\top \mathbb{I}(\tilde{w} - w) = \|\tilde{w} - w\|_2^2$$

is called the proximal term that prevents the algorithm moving too far away from the current point w. It can be easily verified that the optimal solution is

$$\tilde{w} = w + \beta \nabla_w V_w(x_0).$$

Optimization over the Local Policy Space

The proximal term is in terms of the Euclidean distance of weight vector w. While the weights w define a policy, the change of w does not directly correlate with the change of the performance of the policy.
Instead, we consider

$$\max_{\tilde{w}} V_w(x_0) + \nabla_w V_w(x_0)(\tilde{w} - w) - \frac{1}{\beta}\mathbb{E}_{x\sim\rho_w}\left[\mathrm{KL}(\pi_{\tilde{w}}(\cdot|x)\|\pi_w(\cdot|x))\right].$$

where

$$\mathrm{KL}(p\|q) = \int_{-\infty}^{\infty} p(u)\log\left(\frac{p(u)}{q(u)}\right)\,du$$

is the Kullback-Leibler divergence of distributions p and q.

Natural Policy Gradient (NPG) (Kakade 2001)

To solve the optimization problem above, we consider the second-order Taylor approximation

$$\begin{aligned}
&\mathrm{KL}(\pi_{\tilde{w}}(\cdot|x)\|\pi_w(\cdot|x)) \\
=&\mathrm{KL}(\pi_w(\cdot|x)\|\pi_w(\cdot|x)) + \left(\nabla_{\tilde{w}}\mathrm{KL}(\pi_{\tilde{w}}(\cdot|x)\|\pi_w(\cdot|x))|_{\tilde{w}=w}\right)^\top (\tilde{w}-w) \\
&+ \frac{1}{2}(\tilde{w}-w)^\top \left(\nabla^2_{\tilde{w}}\mathrm{KL}(\pi_{\tilde{w}}(\cdot|x)\|\pi_w(\cdot|x))|_{\tilde{w}=w}\right)(\tilde{w}-w) + o(\|\tilde{w}-w\|^3).
\end{aligned}$$

It is easy to verify that

$$\mathrm{KL}(\pi_w(\cdot|x)\|\pi_w(\cdot|x)) = 0$$

and

$$\nabla_{\tilde{w}}\mathrm{KL}(\pi_{\tilde{w}}(\cdot|x)\|\pi_w(\cdot|x))|_{\tilde{w}=w} = 0.$$

Natural Policy Gradient (NPG) (Kakade 2001)

Furthermore,

$$\nabla^2_{\tilde{w}} \mathrm{KL}(\pi_{\tilde{w}}(\cdot|x) \| \pi_w(\cdot|x))|_{\tilde{w}=w} = \mathbb{E}_{\pi_w}\left[(\nabla_w \log \pi_w(u|x))\, \nabla \log \pi_w(u|x)^\top\right] := \mathrm{FI}_w(x)$$

is the Fisher information matrix. Now define the average Fisher information matrix to be

$$\mathrm{FI}_w = \mathbb{E}_{\rho_w}\left[\mathrm{FI}_w(x)\right].$$

Natural Policy Gradient (NPG) (Kakade 2001)

Then the optimization problem associated with the proximal gradient ascent becomes

$$\max_{\tilde{w}} V_w(x_0) + \nabla_w V_w(x_0)(\tilde{w} - w) - \frac{1}{2\beta}(\tilde{w} - w)^\top \mathrm{FI}_w(\tilde{w} - w).$$

In other words, we use the average Fisher information matrix in NPG:

$$w \leftarrow w + \beta \mathrm{FI}_w^\dagger \nabla_w V_w(x_0),$$

where $\dagger$ is the Moore-Penrose pseudo-inverse.

Optimality and Convergence of PG and NPG (Tabular)

Optimality Theorem (Mei et al 2020)

Choosing learning rate $\beta = (1-\alpha)^3/8$, we have

$$V^*(d) - V_{w_t}(d) \leq \frac{16S}{(1-\alpha)^5} \frac{||\rho_*/\rho_d||_\infty^2}{c^2} \frac{1}{t},$$

where S is the number of states, ρ is the discounted state distribution, c is a constant and d is distribution of the initial condition.

Optimality Theorem (Agarwal 2020)

Choosing $\beta = (1-\alpha)^2 \log|\mathcal{A}|$, after t iterations of NPG,

$$V^*(x) - V_{w_t}(x) \leq \frac{2}{(1-\alpha)^2 t} \quad \forall x \in \mathcal{S}.$$

NPG and Soft Policy Iteration

Softmax Policy (Tabular Case)

$$\pi_w(u|x) = \frac{\exp(w_{x,u})}{\sum_{u \in A} \exp(w_{x,u})}.$$

Soft Policy Iteration (Tabular Case)

$$\pi_{t+1}(u|x) \propto \frac{\exp(\beta Q_{\pi_t}(x,u))}{\sum_v \exp(\beta Q_{\pi_t}(x,v))}$$

NPG and Soft Policy Iteration

NPG + Entropy Regularization

$$w \leftarrow w + \beta F_w^{\dagger} \nabla_w V_{w,\tau}(x_0),$$

where

$$V_{w,\tau}(x_0) = V_{w,\tau}(x_0) + \tau E_{\rho_w}\left[\sum_u \pi(u|x) \log \frac{1}{\pi(u|x)}\right]$$

NPG + Entropy Regularization = Soft Policy Iteration (Tabular Case)

$$\pi_{t+1}(u|x) \propto (\pi_t(u|x))^{1-\frac{\beta\tau}{1-\alpha}} \frac{\exp(\frac{\beta}{1-\alpha} Q_{\pi_t}(x,u))}{\sum_v \exp(\beta Q_{\pi_t}(x,v))}.$$

It becomes soft policy iteration when $\beta = \frac{1-\alpha}{\tau}$.

Soft Policy Iteration becomes Policy Iteration when $\tau \to 0$.

Trust Region Policy Optimization (TRPO)

Performance Difference Lemma (Kakade and Langford (2002))

$$V_{\tilde{w}}(x_0) = V_w(x_0) + \frac{1}{1-\alpha} E_{\rho_{\tilde{w}}} \left[\sum_u \pi_{\tilde{w}}(u|x) \left(Q_w(x,u) - V_w(x) \right) \right],$$

Proof

According to definition of $V_{\tilde{w}}(x_0)$, we have

$$\begin{aligned} V_{\tilde{w}}(x_0) &= E_{x_k \sim \mathbb{P}_{\tilde{w}}(x_k|x_0), u_k \sim \pi_{\tilde{w}}(x_k)} \left[\sum_{k=0}^{\infty} \alpha^k r(x_k, u_k) \right] \\ &= E_{x_k \sim \mathbb{P}_{\tilde{w}}(x_k|x_0), u_k \sim \pi_{\tilde{w}}(x_k)} \left[\sum_{k=0}^{\infty} \alpha^k \left(r(x_k, u_k) + V_w(x_k) - V_w(x_k) \right) \right]. \end{aligned}$$

Trust Region Policy Optimization (TRPO)

Proof

Since

$$\sum_{k=0}^{\infty} \alpha^k V_w(x_k) = \sum_{k=0}^{\infty} \alpha^{k+1} V_w(x_{k+1}) + V_w(x_0),$$

the equation above is equivalent to

$$\begin{aligned}
&V_{\tilde{w}}(x_0) \\
=&E\left[\sum_{k=0}^{\infty} \alpha^k \left(r(x_k, u_k) + \alpha V_w(x_{k+1}) - V_w(x_k)\right)\right] + V_w(x_0) \\
=&E_{x_k \sim \mathbb{P}_{\tilde{w}}(x_k|x_0), u_k \sim \pi_{\tilde{w}}(x_k)}\left[\sum_{k=0}^{\infty} \alpha^k \left(Q(x_k, u_k) - V_w(x_k)\right)\right] + V_w(x_0).
\end{aligned}$$

Trust Region Policy Optimization (TRPO)

Proof

$$
\begin{aligned}
&\mathbb{E}_{x_k\sim\mathbb{P}_{\tilde{w}}(x_k|x_0),u_k\sim\pi_{\tilde{w}}(x_k)}\left[\sum_{k=0}^{\infty}\alpha^k\left(Q(x_k,u_k)-V_w(x_k)\right)\right]\\
=&\sum_{k=0}^{\infty}\sum_{x,u}\alpha^k\left(Q(x_k,u_k)-V_w(x_k)\right)\Pr(x_k=x|x_0)\pi_{\tilde{w}}(u|x)\\
=&\sum_{x,u}\pi_{\tilde{w}}(u|x)\left(Q(x_k,u_k)-V_w(x_k)\right)\sum_{k=0}^{\infty}\alpha^k\Pr(x_k=x|x_0)\\
=&\frac{1}{1-\alpha}\sum_{u,x}\pi_{\tilde{w}}(u|x)\left(Q(x_k,u_k)-V_w(x_k)\right)\left((1-\alpha)\sum_{k=0}^{\infty}\alpha^k\Pr\left(x_k=x|x_0\right)\right)\\
=&\frac{1}{1-\alpha}\mathbb{E}_{x\sim\rho_{\tilde{w}}}\left[\sum_{u}\pi_{\tilde{w}}(u|x)\left(Q_w(x,u)-V_w(x)\right)\right].
\end{aligned}
$$

Trust Region Policy Optimization (TRPO)

Local Approximation

$$V_{\tilde{w}}(x_0) \approx V_w(x_0) + E_{\rho_w}\left[\sum_u \pi_{\tilde{w}}(u|x) A_w(x,u)\right]$$

Trust Region Policy Optimization (TRPO)

TRPO (Schulman at al 2015)

$$\max_{\tilde{w}} E_{\rho_w}\left[\sum_u \pi_{\tilde{w}}(u|x) A_w(x,u)\right]$$

$$\text{subject to:} \quad E_{\rho_w}\left[\mathrm{KL}(\pi_{\tilde{w}} \| \pi_w)\right] \leq \epsilon$$

TRPO =NPG with a different learning rate

$$w \leftarrow w + \tilde{\beta} \mathrm{FI}_w^{\dagger} \nabla_w V_w(x_0),$$

$\tilde{\beta}$ is chosen with a backtracking line search to satisfy the constraint.

Trust Region Policy Optimization (TRPO)

Monotonic Improvement Theorem

With carefully chosen learning rates, TRPO guarantees that under general function approximations:

$$V_{w_{k+1}}(x_0) \geq V_{w_k}(x_0).$$

Proximal Policy Optimization (PPO)

TRPO (Schulman at al 2015)

$$\max_{\tilde{w}} \mathbb{E}_{\rho_w}\left[\sum_u \pi_{\tilde{w}}(u|x) A_w(x,u)\right] = \max_{\tilde{w}} \mathbb{E}_{\rho_w, u\sim\pi_w}\left[\frac{\pi_{\tilde{w}}(u|x)}{\pi_w(u|x)} A_w(x,u)\right]$$

subject to: $E_{\rho_w}\left[\mathrm{KL}(\pi_{\tilde{w}} \| \pi_w)\right] \leq \epsilon$

Drawback

Both NPG and TRPO are second-order methods, requiring the inverse of the Fisher information matrix.

PPO

Similar to TRPO but is a first-order method.

Proximal Policy Optimization (PPO)

PPO-Penalty

- Collect multiple epochs and use minibatch SGD to optimize

$$\max_{\tilde{w}} \mathbb{E}_{\rho_w, u\sim\pi_w}\left[\frac{\pi_{\tilde{w}}(u|x)}{\pi_w(u|x)} A_w(x,u) - \beta \mathrm{KL}(\pi_{\tilde{w}} \| \pi_w)\right]$$

- Compute $d = \mathbb{E}_{\rho_w}\left[\mathrm{KL}(\pi_{\tilde{w}} \| \pi_w)\right]$. If $d \leq \frac{\epsilon}{1.5}$, $\beta \leftarrow \beta/2$; and If $d \geq 1.5\epsilon$, $\beta \leftarrow 2\beta$.

Proximal Policy Optimization (PPO)

PPO-Clip

Consider a modified objective

$$\max_{\tilde{w}} \mathbb{E}_{\rho_w, u\sim\pi_w}\left[\min\left\{\frac{\pi_{\tilde{w}}(u|x)}{\pi_w(u|x)}A_w(x,u), \left(\frac{\pi_{\tilde{w}}(u|x)}{\pi_w(u|x)}\right)_{1-\epsilon}^{1+\epsilon} A_w(x,u)\right\}\right]$$

Intuition

- Clip is to make sure the new policy does not deviate too much from the old policy.
- The min is to have a conservative estimate of the objective. Ignore the clipping if it improves the objective and include it when it makes the objective worse (to discourage the change).

References

- R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, *Policy gradient methods for reinforcement learning with function approximation.* NeurIPS, 1999.
- S. Kakade, *A Natural Policy Gradient.* NeurIPS, 2001.
- S. Kakade and J. Langford. *Approximately optimal approximate reinforcement learning.* In ICML, volume 2, pp. 267–274, 2002.
- J. Schulman, S. Levine, P. Abbeel, M. Jordan, P. Moritz. *Trust region policy optimization.* ICML, 2015.
- Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. *Proximal policy optimization algorithms.* arXiv preprint arXiv:1707.06347, 2017.
- Alekh Agarwal, Nan Jiang, Sham Kakade, Wen Sun, *Reinforcement Learning: Theory and Algorithms.* Available at `https://rltheorybook.github.io/`
- J. Mei, C. Xiao, C. Szepesvari, and D. Schuurmans, *On the Global Convergence Rates of Softmax Policy Gradient Methods.* ICML, 2020.
- A. Agarwal, S. Kakade, J. D. Lee, and G. Mahajan. *Optimality and Approximation with Policy Gradient Methods in Markov Decision Processes*, COLT, 2020.