IEOR 8100: Reinforcement learning

Lecture 4: Policy gradient methods

By Shipra Agrawal

In Q-learning function approximation was used to approximate Q-function, and policy was a greedy policy based on estimated Q-function. In policy gradient methods, we approximate a stochastic policy directly using a parametric function approximator.

More formally, given an MDP (S, A, s_1, R, P), let $\pi_\theta : S \to \Delta^A$ denote a randomized policy parameterized by parameter vector $\theta \in \mathbb{R}^d$, where $\pi(s)$ denotes a probability vector of dimension A with each component being the probability of taking the corresponding action. (Equivalently, we may represent a policy as $\pi_\theta : S \times A \to \mathbb{R}$, with $\pi_\theta(s, a)$ being the probability of taking action a, and $\sum_a \pi_\theta(s, a) = 1$).

For a scalable formulation, we want $d << |S|$. For example, the policy π_θ might be represented by a neural network whose input is a representation of the state, whose output is action selection probabilities, and whose weights form the policy parameters θ. (The architecture for such a [deep] neural network is similar to that for a multi-label classifier, with input being a state, and labels being different actions. And, the network should be trained to predict the probability of different actions given an input state).

For simplicity, assume that π_θ is differentiable with respect to θ, i.e. $\frac{\partial \pi_\theta(s,a)}{\partial \theta}$ exists. This is true for example, if a neural network with differentiable activation functions is used to define π_θ. Let ρ^{π_θ} denote the gain of policy π_θ. This may be defined as long term average reward, long term discounted reward or total reward in an episode or finite horizon. Therefore, solving for optimal policy reduces to the problem of solving

$$\max_\theta \rho^{\pi_\theta}(s_1)$$

In order to use a (stochastic) gradient descent algorithm for finding a stationary point of the above problem, we need to compute (an unbiased) estimate of gradient of $\rho^{\pi_\theta}(s_1)$ with respect to θ.

1 Finite horizon MDP

Here performance measure to optimize is total expected reward over a finite horizon H. For a policy π,

$$\rho^\pi(s_1) = \mathbb{E}[\sum_{t=1}^{H} \gamma^{t-1} r_t | \pi, s_1]$$

Let $\pi(s, a)$ denote the probability of action a in state s for randomized policy π. Let $D^\pi(\tau)$ denote the probability distribution of a trajectory (state-action sequence) $\tau = (s_1, a_1, s_2, \ldots, a_{H-1}, s_H)$ of states on starting from state s_1 and following policy π. That is,

$$D^\pi(\tau) := \prod_{i=1}^{H-1} \pi(s_i, a_i) P(s_i, a_i, s_{i+1})$$

Theorem 1. *For finite horizon MDP (S, A, s_1, P, R, H), let $R(\tau)$ be the total reward for an sample trajectory τ, on following π_θ for H steps, starting from state s_1. Then,*

$$\nabla_\theta \rho^{\pi_\theta}(s_1) = \mathbb{E}_\tau \left[R(\tau) \nabla_\theta \log(D^{\pi_\theta}(\tau)) | s_1 \right] = \mathbb{E}_\tau \left[R(\tau) \sum_{t=1}^{H-1} \nabla_\theta \log(\pi_\theta(s_t, a_t)) | s_1 \right]$$

Proof. Let $R(\tau)$ be expected *total reward* for an entire sample trajectory τ, on following π_θ for H steps, starting from state s_1. That is, given a sample trajectory $\tau = (s_1, a_1, s_2, \ldots, a_{H-1}, s_H)$ from distribution D^{π_θ},

$$R(\tau) := \sum_{t=1}^{H-1} \gamma^{t-1} R(s_t, a_t),$$

Then,

$$\rho^{\pi_\theta}(s) = \mathbb{E}_{\tau \sim D^{\pi_\theta}}[R(\tau)|s_1 = s]$$

Now, (the calculations below implicitly assume finite state and action space, so that the distribution $D^{\pi_\theta}(\tau)$ has a finite support)

$$\begin{aligned}
\nabla_\theta \rho^{\pi_\theta}(s) &= \nabla_\theta \mathbb{E}_{\tau \sim D^{\pi_\theta}}[R(\tau)|s_1 = s] \\
&= \nabla_\theta \sum_{\tau : D^{\pi_\theta}(\tau) > 0} D^{\pi_\theta}(\tau|s_1 = s) R(\tau) \\
&= \sum_{\tau : D^{\pi_\theta}(\tau) > 0} D^{\pi_\theta}(\tau|s_1 = s) \nabla_\theta \log(D^{\pi_\theta}(\tau|s_1 = s)) R(\tau) \\
&= \mathbb{E}_{\tau \sim D^{\pi_\theta}}[\nabla_\theta \log(D^{\pi_\theta}(\tau)) R(\tau)|s_1 = s]
\end{aligned}$$

Further, for a given sample trajectory τ^i.

$$\begin{aligned}
\nabla_\theta \log(D^{\pi_\theta}(\tau^i)) &= \sum_{t=1}^{H-1} \nabla_\theta \log(\pi_\theta(s_t^i, a_t^i)) + \nabla_\theta \log P(s_t^i, a_t^i, s_{t+1}^i) \\
&= \sum_{t=1}^{H-1} \nabla_\theta \log(\pi_\theta(s_t^i, a_t^i))
\end{aligned}$$

□

The gradient representation given by above theorem is extremely useful, as given a sample trajectory this can be computed only using the policy parameter, and does not require knowledge of the transition model $P(\cdot,\cdot,\cdot)$! This does seem to require knowledge of reward model, but that can be handled by replacing $R(\tau^i)$ by $\hat{R}(\tau^i) = r_1 + \gamma r_2 + \ldots, \gamma^{H-2} r_{H-1}$, the total of sample rewards observed in this trajectory. Since, given a trajectory τ, the quantity $D^{\pi_\theta}(\tau)$ is determined, and $\mathbb{E}[\hat{R}(\tau)|\tau] = R(\tau)$,

$$\begin{aligned}
\nabla_\theta \rho^{\pi_\theta}(s_1) &= \mathbb{E}_\tau [R(\tau) \nabla_\theta \log(D^{\pi_\theta}(\tau))|s_1] \\
&= \mathbb{E}_\tau \left[\hat{R}(\tau) \nabla_\theta \log(D^{\pi_\theta}(\tau))|s_1\right] \\
&= \mathbb{E}_\tau \left[\hat{R}(\tau) \sum_{t=1}^{H-1} \nabla_\theta \log(\pi_\theta(s_t, a_t))|s_1\right]
\end{aligned}$$

Unbiased estimator of gradient from samples. From above, given sample trajectories $\tau^i, i = 1, \ldots, m$, an unbiased estimator for gradient $\nabla_\theta \rho(\pi_\theta)$ is given as:

$$\hat{\mathbf{g}} = \frac{1}{m} \sum_{i=1}^m \hat{R}(\tau^i) \nabla_\theta \log(D^{\pi_\theta}(\tau^i)) = \frac{1}{m} \sum_{i=1}^m \hat{R}(\tau^i) \sum_{t=1}^{H-1} \nabla_\theta \log(\pi_\theta(s_t^i, a_t^i)) \tag{1}$$

Baseline. Note that for any constant b (or b that is conditionally independent of sampling from π_θ given θ), we have:

$$\mathbb{E}_\tau[b \frac{\partial}{\partial \theta_j} \log(D^{\pi_\theta}(\tau))|\theta, s_1] = b \sum_\tau \frac{\partial}{\partial \theta_j} D^{\pi_\theta}(\tau)) = b \frac{\partial}{\partial \theta_j} \sum_\tau D^{\pi_\theta}(\tau) = b \frac{\partial}{\partial \theta_j} 1 = 0$$

Therefore, choosing any 'baseline' b, following is also an unbiased estimator of the $\nabla_\theta \rho(\pi_\theta)$:

$$\hat{\mathbf{g}} = \frac{1}{m} \sum_{i=1}^m \sum_{t=1}^{H-1} (\hat{R}(\tau^i) - b) \nabla_\theta \log(\pi_\theta(s_t^i, a_t^i))$$

Or, more generally, one could even use a state and time dependent baseline $b_t(s_t^i)$ conditionally is independent of sampling from π_θ given s_t^i, θ, to get estimator:

$$\hat{\mathbf{g}} = \frac{1}{m}\sum_{i=1}^{m}\sum_{t=1}^{H-1}(\hat{R}(\tau^i) - b_t(s_t^i))\nabla_\theta \log(\pi_\theta(s_t^i, a_t^i)) \tag{2}$$

Below we show this is unbiased. The expectations below are over trajectories $(s_1, a_1, \ldots, a_{H-1}, s_H)$, where given state s_t, the action $a_t \sim \pi(s_t, \cdot)$. For any fixed θ, t, the baseline $b_t(s_t)|s_t$ needs to be deterministic or independent of $a_t|s_t$. For simplicity we assume it is determinstic.

$$\begin{aligned}
\mathbb{E}_\tau[\sum_{t=1}^{H-1} b_t(s_t)\frac{\partial}{\partial\theta_j}\log(\pi_\theta(s_t, a_t))|\theta, s_1] &= \mathbb{E}[\sum_{t=1}^{H-1}\mathbb{E}[b_t(s_t)\frac{\partial}{\partial\theta_j}\log(\pi_\theta(s_t, a_t))|s_t]|\theta, s_1] \\
&= \mathbb{E}[\sum_{t=1}^{H-1} b_t(s_t)\mathbb{E}[\frac{\partial}{\partial\theta_j}\log(\pi_\theta(s_t, a_t))|s_t]|\theta, s_1] \\
&= \mathbb{E}[\sum_{t=1}^{H-1} b_t(s_t)\sum_a \pi_\theta(s_t, a)\frac{\partial}{\partial\theta_j}\log(\pi_\theta(s_t, a))|\theta, s_1] \\
&= \mathbb{E}[\sum_{t=1}^{H-1} b_t(s_t)\sum_a \frac{\partial}{\partial\theta_j}\pi_\theta(s_t, a)|\theta, s_1] \\
&= \mathbb{E}[\sum_{t=1}^{H-1} b_t(s_t)\frac{\partial}{\partial\theta_j}\sum_a \pi_\theta(s_t, a)|\theta, s_1] \\
&= \mathbb{E}[\sum_{t=1}^{H-1} b_t(s_t)\frac{\partial}{\partial\theta_j}(1)|\theta, s_1] \\
&= 0
\end{aligned}$$

An example of such state dependent baseline $b_t(s)$, given s and θ, is $V_{H-t}^{\pi_\theta}(s)$, i.e., the value of policy π_θ, starting from state s at time t. We will see later that such a baseline is useful in reducing the variance of gradient estimates.

Vanilla policy gradient algorithm Initialize policy parameter θ, and baseline.
In each iteration,

- Execute current policy π^θ to obtain several sample trajectories τ^i, $i = 1, \ldots, m$.
- Use these sample trajectories and chosen baseline to compute the gradient estimator $\hat{\mathbf{g}}$ as in (2).
- Update $\theta \leftarrow \theta + \alpha\hat{\mathbf{g}}$
- Update baseline as required.

Above is essentially same as the **REINFORCE** algorithm introduced by [Williams, 1988, 1992].

2 Infinite horizon discounted rewards

$$\rho^\pi(s_1) = \lim_{T\to\infty}\mathbb{E}[\sum_{t=1}^{T}\gamma^{t-1}r_t|\pi, s_1] = \sum_s d^\pi(s)\sum_a \pi(s, a)R(s, a)$$

where $d^\pi(s) = \lim_{T\to\infty}\sum_{t=1}^{T}\gamma^{t-1}\Pr(s_t = s|s_1, \pi)$, and the value of a state given a policy π is defined as:

$$V^\pi(s) = \lim_{T\to\infty}\mathbb{E}[\sum_{t=1}^{T}\gamma^{t-1}r_t|s_1 = s, a_t = \pi(s_t), t = 1, \ldots, T] = \sum_a \pi(s, a)Q^\pi(s, a)$$

where

$$Q^{\pi}(s,a) := \lim_{T\to\infty} \mathbb{E}[\sum_{t=1}^{T} \gamma^{t-1} r_t | s_1 = s, a_1 = a, a_t = \pi(s_t), t = 2, \dots, T]$$

Theorem 2. [Policy gradient theorem [Sutton et al., 1999]] *For infinite horizon MDP discounted reward case,*

$$\nabla_\theta \rho^{\pi_\theta}(s_1) = \sum_s d^{\pi_\theta}(s) \sum_a Q^{\pi_\theta}(s,a) \nabla_\theta \pi_\theta(s,a) = \sum_s d^{\pi_\theta}(s) \left(\mathbb{E}_{a\sim\pi(s)}[Q^{\pi_\theta}(s,a) \nabla_\theta \log(\pi_\theta(s,a))] \right)$$

That is gradient of gain with respect to θ can be expressed in terms of gradient of policy function with respect to θ.

Proof. We abbreviate π_θ as π in below. We have:

$$\begin{aligned} Q^{\pi}(s,a) &= R(s,a) + \gamma \sum_{s'} P(s,a,s') V^{\pi}(s') \\ \nabla_\theta Q^{\pi}(s,a) &= \gamma \sum_{s'} P(s,a,s') \nabla_\theta V^{\pi}(s') \end{aligned}$$

$$\begin{aligned} V^{\pi}(s) &= \sum_a \pi(s,a) Q^{\pi}(s,a) \\ \nabla_\theta V^{\pi}(s) &= \sum_a Q^{\pi}(s,a) \nabla_\theta \pi(s,a) + \sum_a \pi(s,a) \nabla_\theta Q^{\pi}(s,a) \end{aligned}$$

Therefore,

$$\begin{aligned} \sum_s d^{\pi}(s) \nabla_\theta V^{\pi}(s) &= \sum_s d^{\pi}(s) \sum_a Q^{\pi}(s,a) \nabla_\theta \pi(s,a) + \sum_s d^{\pi}(s) \sum_a \pi(s,a) \nabla_\theta Q^{\pi}(s,a) \\ &= \sum_s d^{\pi}(s) \sum_a Q^{\pi}(s,a) \nabla_\theta \pi(s,a) + \sum_s d^{\pi}(s) \sum_a \pi(s,a) \left(\gamma \sum_{s'} P(s,a,s') \nabla_\theta V^{\pi}(s') \right) \\ &= \sum_s d^{\pi}(s) \sum_a Q^{\pi}(s,a) \nabla_\theta \pi(s,a) + \sum_{s'} d^{\pi}(s') \nabla_\theta V^{\pi}(s') - \nabla_\theta V^{\pi}(s_1) \qquad (3) \end{aligned}$$

where we obtained the last equation using the following derivation for $d^{\pi}(s')$. Let $\Pr(s \to x, k, \pi)$ is the probability of going from state s to state x in k steps under policy π. Then, for any s',

$$\begin{aligned}
d^\pi(s') &= \sum_{t=1}^{\infty} \gamma^{t-1} \Pr(s_t = s'|s_1, \pi) \\
&= \sum_{t=1}^{\infty} \gamma^{t-1} \Pr(s_1 \to s', t-1, \pi) \\
&= \sum_{t=2}^{\infty} \gamma^{t-1} \Pr(s_1 \to s', t-1) + \mathbf{1}(s' = s_1) \\
&= \sum_{t=2}^{\infty} \gamma^{t-1} \left(\sum_{s,a} \Pr(s_1 \to s, t-2, \pi) \pi(s,a) P(s,a,s') \right) + \mathbf{1}(s' = s_1) \\
&= \sum_{t=1}^{\infty} \gamma^{t} \left(\sum_{s,a} \Pr(s_1 \to s, t-1, \pi) \pi(s,a) P(s,a,s') \right) + \mathbf{1}(s' = s_1) \\
&= \gamma \sum_{s,a} \left(\sum_{t=1}^{\infty} \gamma^{t-1} \Pr(s_t = s|s_1, \pi) \right) \pi(s,a) P(s,a,s') + \mathbf{1}(s' = s_1) \\
&= \gamma \sum_{s,a} d^\pi(s) \pi(s,a) P(s,a,s') + \mathbf{1}(s' = s_1)
\end{aligned}$$

Moving the terms around in (3):

$$\nabla_\theta V^\pi(s_1) = \sum_s d^\pi(s) \sum_a Q^\pi(s,a) \nabla_\theta \pi(s,a)$$

That is,

$$\nabla_\theta \rho^\pi(s_1) = \sum_s d^\pi(s) \sum_a Q^\pi(s,a) \nabla_\theta \pi(s,a)$$

□

Remarks on Theorem 4. The key aspect of the expression for the gradient is that there are no terms of the form $\frac{\partial d^{\pi_\theta}(s)}{\partial \theta}$: the effect of policy changes on the distribution of states does not appear. This is convenient for approximating the gradient by sampling. For example, if s was sampled from the distribution obtained by following π_θ, then $\sum_a Q^{\pi_\theta}(s,a) \nabla_\theta \pi_\theta(s,a)$ would be an unbiased estimate of $\nabla_\theta \rho(\pi_\theta)$. Of course, $Q^{\pi_\theta}(s,a)$ is also not normally known and must be estimated (in an unbiased way).

2.1 Vanilla Policy Gradient Algorithm

Estimation using samples. Suppose we run policy π several times starting from s_1 to observe sample trajectories $\{\tau^i\}$. Then, at each time step t in a trajectory τ, set

$$\hat{Q}_t := \sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'}$$

where $r_{t'}$ is the observed reward at time t'. Then, $\hat{Q}_t$ is an unbiased estimate of $Q(s_t, a_t)$ at time t (almost, assuming large enough T), i.e,

$$\mathbb{E}[\hat{Q}_t | s_t, a_t] \quad \approx \quad Q(s_t, a_t)$$

Let

$$F_t := \hat{Q}_t \nabla_\theta \log \pi_\theta(s_t, a_t)$$

Then,

$$\mathbb{E}[F_t|s_t] = \mathbb{E}[\mathbb{E}[\hat{Q}_t|s_t, a_t]\nabla_\theta \log(\pi_\theta(s_t, a_t))|s_t] = \mathbb{E}[Q(s_t, a_t)\nabla_\theta \log(\pi_\theta(s_t, a_t))|s_t] = \sum_a Q(s_t, a)\nabla_\theta \pi_\theta(s_t, a)$$

And,

$$\begin{aligned}
\mathbb{E}[\sum_{t=1}^{\infty} \gamma^{t-1} F_t|s_1] &= \sum_{t=1}^{\infty} \gamma^{t-1} \sum_{s,a} \mathbb{E}[F_t|s_t = s, a_t = a] \Pr(a_t = a|s_t = s, s_1) \Pr(s_t = s|s_1) \\
&\approx \sum_{t=1}^{\infty} \gamma^{t-1} \sum_s \sum_a \left(Q(s,a)\nabla_\theta \log(\pi_\theta(s,a))\right) \pi_\theta(s,a) \Pr(s_t = s|s_1) \\
&= \sum_{t=1}^{\infty} \gamma^{t-1} \sum_s \left(\sum_a Q(s,a)\nabla_\theta \pi_\theta(s,a)\right) \Pr(s_t = s|s_1) \\
&= \sum_s \left(\sum_a Q(s,a)\nabla_\theta \pi_\theta(s,a)\right) \sum_{t=1}^{\infty} \gamma^{t-1} \Pr(s_t = s|s_1) \\
&= \sum_s \left(\sum_a Q(s,a)\nabla_\theta \pi_\theta(s,a)\right) d^\pi(s) \\
&= \nabla_\theta \rho^{\pi_\theta}(s_1)
\end{aligned}$$

where the last step follows from the policy gradient theorem. Therefore, following is an unbiased (almost, for large T) estimate of gradient of $\nabla_\theta \rho(\pi, s_1)$ of policy π at $\hat{\theta}$, starting in state s_1:

$$\hat{\mathbf{g}} = \sum_{t=1}^{T} \gamma^{t-1} F_t = \sum_{t=1}^{T} \gamma^{t-1} \hat{Q}_t \nabla_\theta \log(\pi_\theta(s_t, a_t))$$

From N sample trajectories $\{\tau^i, i = 1, \ldots, N\}$ we can develop sample average estimate

$$\hat{\mathbf{g}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \gamma^{t-1} F_t^i = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \gamma^{t-1} \hat{Q}_t^i \ \nabla_\theta \log(\pi_\theta(s_t^i, a_t^i)) \tag{4}$$

Baseline. We can obtain another unbiased estimate by replacing F_t by

$$F_t' = (\hat{Q}_t - b_t(s_t))\nabla_\theta \log(\pi_\theta(s_t, a_t))$$

for an arbitrary baseline function $b_t(s)$. Then,

$$\mathbb{E}[\sum_t \gamma^{t-1} F_t'] = \mathbb{E}[\sum_t \gamma^{t-1} F_t] - \mathbb{E}[\sum_t \gamma^{t-1} b_t(s_t)\nabla_\theta \log(\pi_\theta(s_t, a_t)) = \mathbb{E}[\sum_t \gamma^{t-1} F_t]$$

The last step follows because:

$$\mathbb{E}[b_t(s_t)\nabla_\theta \log(\pi_\theta(s_t, a_t))|s_t] = b_t(s_t) \sum_a \nabla_\theta \pi_\theta(s_t, a) = b_t(s_t)\nabla_\theta(1) = 0$$

That is, following is also an unbiased gradient estimate

$$\hat{\mathbf{g}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \gamma^{t-1} (\hat{Q}_t^i - b_t(s_t^i)) \ \nabla_\theta \log(\pi(s_t^i, a_t^i)) \tag{5}$$

Vanilla Policy Gradient Algorithm. Initialize policy parameter θ, and baseline function $b_t(s), \forall s$. In each iteration,

1. Execute current policy π^θ to obtain several sample trajectories τ^i, $i = 1, \ldots, m$.
2. For any given sample trajectory i, use observed rewards $r_1, r_2, \ldots$, to compute $\hat{Q}_t^i := \sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'}$.
3. Use $\hat{Q}_t^i$ and baseline function $b_t(s)$ to compute the gradient estimator $\hat{\mathbf{g}}$ as in (5).
4. Update $\theta \leftarrow \theta + \alpha \hat{\mathbf{g}}$.
5. Re-optimize baseline.

2.2 Variance reduction using baseline [Greensmith et al., 2002, 2004]

Lemma 3. *For any two random variables A, B, such that for some filtration $\mathcal{F}$, $\mathbb{E}[B|\mathcal{F}] = B, \mathbb{E}[A|\mathcal{F}] = B$, almost surely,*

$$Var(A) = \mathbb{E}[(A-B)^2] + Var(B)$$

Proof. For such r.v.,

$$\mathbb{E}[A|B] = \mathbb{E}[\mathbb{E}[A|\mathcal{F}, B]|B] = \mathbb{E}[B|B] = B$$

Therefore, firstly, $\mathbb{E}[A - B] = \mathbb{E}[\mathbb{E}[A - B|B]] = \mathbb{E}[B - B] = 0$, so that

$$\mathbb{E}[A] = \mathbb{E}[B].$$

And,

$$\mathbb{E}[B(A-B)] = \mathbb{E}[\mathbb{E}[B(A-B)|B]] = \mathbb{E}[B^2 - B^2] = 0.$$

Then,

$$\begin{aligned}
\text{Var}(A) &= \mathbb{E}[(A - \mathbb{E}[A])^2] \\
&= \mathbb{E}[(A - \mathbb{E}[B])^2] \\
&= \mathbb{E}[((B - \mathbb{E}[B]) + (A - B))^2] \\
&= \mathbb{E}[(B - \mathbb{E}[B])^2 + 2(B - \mathbb{E}[B])(A - B) + (A - B)^2] \\
&= \text{Var}(B) + \mathbb{E}[(A - B)^2]
\end{aligned}$$

□

Now, consider the expression for policy gradient estimate at $\hat{\theta}$ (for infinite horizon discounted case) in (5). For simplicity of notation, let's consider a single sample and omit the superscript i.

$$\hat{\mathbf{g}} = \sum_{t=1}^{T} \gamma^{t-1} (\hat{Q}_t - b_t(s_t)) \; \nabla_\theta \log(\pi_\theta(s_t, a_t; \hat{\theta}))$$

Let

$$A_t = F'_t = (\hat{Q}_t - b_t(s_t)) \nabla_\theta \log(\pi_\theta(s_t, a_t)).$$

And,

$$B_t = \mathbb{E}[A_t | s_1, a_1, \ldots, s_t, a_t] = (Q_t(s_t, a_t) - b_t(s_t)) \nabla_\theta \log(\pi_\theta(s_t, a_t))$$

Note that $\mathbb{E}[B_t|s_t] = \sum_a Q^{\pi_\theta}(s_t, a) \nabla_\theta \pi_\theta(s_t, a)$.

Also, the conditions in the above lemma are satisfied by A_t, B_t for $\mathcal{F} = \{s_1, a_1, \ldots, s_t, a_t\}$ Therefore, using the lemma above:

$$\begin{aligned}
\mathrm{Var}(F'_t|s_t) = \mathrm{Var}(A_t|s_t) &= \mathbb{E}[(A_t - B_t)^2|s_t] + \mathrm{Var}(B_t|s_t) \\
&= \mathbb{E}\left[\left((\hat{Q}_t - \mathbb{E}[\hat{Q}_t|s_t, a_t])\nabla_\theta \log(\pi(s_t, a_t))\right)^2 |s_t\right] \\
&\quad + \mathbb{E}[B_t^2|s_t] - \mathbb{E}[B_t|s_t]^2 \\
&= \mathbb{E}\left[\left((\hat{Q}_t - Q^{\pi_\theta}(s_t, a_t))\nabla_\theta \log(\pi(s_t, a_t))\right)^2 |s_t\right] \\
&\quad + \mathbb{E}\left[((Q^{\pi_\theta}(s_t, a_t) - b_t(s_t))\nabla_\theta \log(\pi(s_t, a_t)))^2 |s_t\right] - (\sum_a Q^{\pi_\theta}(s_t, a)\nabla_\theta \pi_\theta(s_t, a))^2
\end{aligned}$$

This is minimized by baseline:

$$b_t(s_t) = \frac{\mathbb{E}\left(Q^{\pi_\theta}(s_t, a_t)\nabla_\theta \log(\pi(s_t, a_t))^2|s_t\right)}{\mathbb{E}\left(\log(\pi(s_t, a_t))^2|s_t\right)}$$

The value function $V^{\pi_\theta}(s) = \mathbb{E}\left(Q^{\pi_\theta}(s_t, a_t)|s_t\right)$ is an approximation of the above optimal baseline.

The difference $Q^{\pi_\theta}(s, a) - V^{\pi_\theta}(s)$ is also referred to as **Advantage** of action a in state s. This terminology appears in algorithms like 'Asynchronous Advantage Actor Critic Algorithm (A3C)' [Mnih et al., 2016] and 'Generalized Advantage Estimation (GAE)' [Schulman et al., 2015]. See Greensmith et al. [2002, 2004] for a more detailed discussion on the impact of baseline on variance and optimal baseline.

However, even this baseline can only be estimated, and needs to be updated every time the policy changes (i.e., as θ changes). It seems natural to use Q-function/value function approximation methods as a subroutine to make these estimations, thus combining the two categories of methods – policy-gradient and value-function based. We will discuss further motivations for combining the two when studying actor-critic methods.

3 Infinite horizon average reward case

Let's consider the case when $\rho^\pi(s)$ is long-term average reward of (randomized) policy π.

$$\rho^\pi(s_1) = \lim_{T\to\infty} \frac{1}{T}\mathbb{E}[r_1 + r_2 + \ldots + r_T|\pi, s_1] = \sum_s d^\pi(s) \sum_a \pi(s, a)R(s, a)$$

where $d^\pi(s) = \lim_{t\to\infty} \Pr(s_t = s|s_1, \pi)$ is the stationary distribution over states for policy π, on starting from state s_1. We assume that this distribution exists and is independent of the starting state for all policies.

In the average reward case, recall that the bias of a state given a policy π is defined as:

$$h^\pi(s) = \lim_{T\to\infty} \mathbb{E}[\sum_{t=1}^T (r_t - \rho^\pi(s_t))|s_1 = s, a_t = \pi(s_t), t = 1, \ldots, T] = \sum_a \pi(s, a)A^\pi(s, a)$$

where we define

$$A^\pi(s, a) := \lim_{T\to\infty} \mathbb{E}[\sum_{t=1}^T (r_t - \rho^\pi(s_t))|s_1 = s, a_1 = a, a_t = \pi(s_t), t = 2, \ldots, T]$$

(Again, we assume that the limit exists.)

Theorem 4. **[Policy gradient theorem [Sutton et al., 1999]]** *For infinite horizon MDP average reward case,*

$$\nabla_\theta \rho(\pi_\theta, s_1) = \sum_s d^{\pi_\theta}(s) \sum_a A^{\pi_\theta}(s, a)\nabla_\theta \pi_\theta(s, a) = \mathbb{E}_{s\sim d^{\pi_\theta}, a\sim\pi_\theta(s)}[A^{\pi_\theta}(s, a)\nabla_\theta \log(\pi_\theta(s, a))]$$

That is,the gradient of gain with respect to θ can be expressed in terms of the gradient of policy function with respect to θ.

Proof.

$$\begin{aligned}
A^{\pi}(s,a) &= R(s,a) - \rho^{\pi}(s) + \sum_{s'} P(s,a,s')h^{\pi}(s') \\
h^{\pi}(s) &= \sum_{a} \pi(s,a)A^{\pi}(s,a) \\
\nabla_{\theta} h^{\pi}(s) &= \sum_{a} A^{\pi}(s,a)\nabla_{\theta}\pi(s,a) + \pi(s,a)\nabla_{\theta}A^{\pi}(s,a) \\
&= \sum_{a} A^{\pi}(s,a)\nabla_{\theta}\pi(s,a) + \sum_{a}\pi(s,a)\left[\sum_{s'} P(s,a,s')\nabla_{\theta}h^{\pi}(s') - \nabla_{\theta}\rho^{\pi}(s)\right] \\
\nabla_{\theta}\rho^{\pi}(s) &= \sum_{a} A^{\pi}(s,a)\nabla_{\theta}\pi(s,a) + \sum_{a}\pi(s,a)\left[\sum_{s'} P(s,a,s')\nabla_{\theta}h^{\pi}(s')\right] - \nabla_{\theta}h^{\pi}(s)
\end{aligned}$$

$$\begin{aligned}
\sum_{s} d^{\pi}(s)\nabla_{\theta}\rho^{\pi}(s) &= \sum_{s} d^{\pi}(s)\sum_{a} A^{\pi}(s,a)\nabla_{\theta}\pi(s,a) + \sum_{s} d^{\pi}(s)\sum_{a}\pi(s,a)\left[\sum_{s'} P(s,a,s')\nabla_{\theta}h^{\pi}(s')\right] - \sum_{s} d^{\pi}(s)\nabla_{\theta}h^{\pi}(s) \\
&\quad \text{since } d^{\pi} \text{ is the stationary distribution of } \pi \\
&= \sum_{s} d^{\pi}(s)\sum_{a} A^{\pi}(s,a)\nabla_{\theta}\pi(s,a) + \sum_{s'} d^{\pi}(s')\nabla_{\theta}h^{\pi}(s') - \sum_{s} d^{\pi}(s)\nabla_{\theta}h^{\pi}(s) \\
&= \sum_{s} d^{\pi}(s)\sum_{a} A^{\pi}(s,a)\nabla_{\theta}\pi(s,a)
\end{aligned}$$

By assumption that the infinite horizon average reward does not depend on the starting state for all policies, we get the theorem statement. □

4 Examples

A key contribution of policy gradient theorem is to reduce the computation of gradient of gain with respect to θ to computation of gradient of policy function (or log of policy function $\nabla_{\theta}\log(\pi(s,a;\theta))$) with respect to θ. So, it is desirable that the gradient of policy function is efficiently computable. Here are some examples of policy functions where this is efficiently implementable.

4.1 Softmax policies

Consider policy set parameterized by $\theta \in \mathbb{R}^d$ such that given $s \in S$, probability of picking action $a \in A$ is given by:

$$\pi_{\theta}(s,a) = \frac{e^{\theta^{\top}\phi_{sa}}}{\sum_{a' \in A} e^{\theta^{\top}\phi_{sa'}}}$$

where each ϕ_{sa} is an d-dimensional feature vector characterizing state-action pair s, a. This is a popular form of policy space called softmax policies. Here,

$$\nabla_{\theta}\log(\pi_{\theta}(s,a)) = \phi_{sa} - \left(\sum_{a' \in A} \phi_{sa'}\pi_{\theta}(s,a')\right) = \phi_{sa} - \mathbb{E}_{a' \sim \pi(s)}[\phi_{sa'}]$$

4.2 Gaussian policy for continuous action spaces

In continuous action spaces, it is natural to use Gaussian policies. Given state s, the probability of action a is given as:

$$\pi_{\theta}(s,a) = \mathcal{N}(\phi(s)^T\theta, \sigma^2)$$

for some constant σ. Here $\phi(s)$ is a feature representation of s. Then,

$$\nabla_\theta \log(\pi_\theta(s,a)) = \nabla_\theta \frac{-(a-\theta^\top \phi(s))^2}{2\sigma^2} = \frac{(a-\theta^\top\phi(s))}{\sigma^2}\phi(s)$$

4.3 Deep neural networks: backpropagation for gradient computation

In deep reinforcement learning, the policy function is computed by a multi-layer neural network. The independent-layer structure of deep neural network allows the gradient computations efficiently through **backpropagation**. (In the reinforcement learning context) Backpropagation refers to simply a computational implementation of the chain rule : an algorithm that determines the relevant partial derivatives by means of the backward pass.

Suppose π_θ is given by a deep neural network whose input is representation of state s, and output is a representation of action selection probabilities $\pi(s,a)$. (We can add another layer giving logarithmic of the action selection probabilities.) The weights of the neural network form the parameters θ of the policy. Specifically let the quantity of interest $G_\theta(s,\cdot) \in \mathbb{R}^A$ is given by a neural network with $k-1$ hidden layers and weights $\theta = (W_1, \ldots, W_K)$ and input $\mathbf{x}$ being a representation of state s, so that

$$G_\theta(s,\cdot) = \log(\pi(s,\cdot)) = \log(\sigma(W^L(\sigma(\cdots W^3\sigma(W^2 \cdot \sigma(W^1\mathbf{x}))))))$$

Or, (let $L(\cdot)$ denote the log function)

$$\begin{gathered} G(s,a) = \log(\pi(s,a)) = L(h_a^K) \\ h^k = \sigma(z^k), z^k = W^k h^{k-1} \\ h^{k-1} = \sigma(z^{k-1}), z^{k-1} = W^{k-1}h^{k-2}, \\ \ldots, \\ h^1 = \sigma(z^1), z^1 = W^1\mathbf{x} \end{gathered}$$

Now, gradient with respect of $G(s,a)$ with respect to parameter W^ℓ_{ab} is:

$$\frac{\partial G(s,a)}{\partial W^\ell_{ab}} = \frac{\partial \log(h_a^K)}{\partial h_a^K}\frac{\partial h_a^K}{\partial W^\ell_{ab}}$$

Then, basic observation is as follows. For a neuron r in layer k, $\ell < k$:

$$\begin{aligned} \frac{\partial h_r^k}{\partial W^\ell_{ab}} &= \sigma'(z_i^k)\frac{\partial z_r^k}{\partial W^\ell_{ab}} \\ &= \sigma'(z_r^k)\frac{\partial W_r^k h^{k-1}}{\partial W^\ell_{ab}} \\ &= \sigma'(z_r^k)\sum_{i\in \text{parents}(r)} W_{ri}^k \frac{\partial h_i^{k-1}}{\partial W^\ell_{ab}} \end{aligned}$$

Thus, the gradients can be back propagated over the network, until we reach layer ℓ, so that

$$\frac{\partial h_i^\ell}{\partial W^\ell_{ab}} = \begin{cases} \sigma'(z_a^\ell)h_b^{\ell-1}, & \text{if } i = a \\ 0 & \text{otherwise} \end{cases}$$

This follows from layer structure of the neural network so that:

$$\begin{aligned} \frac{\partial z_a^\ell}{\partial W^\ell_{ab}} &= \frac{\partial \sum_i W^\ell_{ai} h_i^{\ell-1}}{\partial W^{\ell-1}_{ab}} \\ &= h_b^{\ell-1} + \sum_i W^\ell_{ai}\frac{\partial h_i^{\ell-1}}{\partial W^\ell_{ab}} \\ &= h_b^{\ell-1} \end{aligned}$$

as layer $h^{\ell-1}$ is independent of layer ℓ parameters.

References

Evan Greensmith, Peter L. Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 1507–1514. MIT Press, 2002.

Evan Greensmith, Peter L. Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *J. Mach. Learn. Res.*, 5:1471–1530, December 2004. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=1005332.1044710`.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. *CoRR*, abs/1602.01783, 2016. URL `http://arxiv.org/abs/1602.01783`.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *CoRR*, abs/1506.02438, 2015. URL `http://arxiv.org/abs/1506.02438`.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Proceedings of the 12th International Conference on Neural Information Processing Systems*, pages 1057–1063, 1999.

R. J. Williams. Toward a theory of reinforcement-learning connectionist systems. Technical Report NU-CCS-88-3, College of Comp. Sci., Northeastern University, Boston, MA, 1988.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, May 1992. ISSN 1573-0565. doi: 10.1007/BF00992696. URL `https://doi.org/10.1007/BF00992696`.