Reinforcement Learning

Lecture 8

Lecturer: Haim Permuter *Scribe: Gal Rattner*

I. Introduction

In the previous lectures we saw methods which selects actions according to an estimation of an action-value function. In this lecture we instead consider methods that uses a parameterized policy that selects actions using the learned policy weights. A value function approximation might still be used to learn the policy weights, but action selection is defined by the current state of the weights solely. This lecture is based on the book by S.Sutton and A.Barto [2] and lecture number 7 in D. Silver course [1].

II. Policy Gradient

In policy gradient methods the policy is determined by a parameter set. Consider $\boldsymbol{\theta} \in \mathbb{R}^n$ to be the policy weights vector. Thus the policy is given by

$$\pi(a|s,\boldsymbol{\theta}) = Pr\{A_t = a|S_t = s, \boldsymbol{\theta}_t = \boldsymbol{\theta}\}, \tag{1}$$

as the probability that action a is taken at time t, given that the state at time t is s and the weights vector is $\boldsymbol{\theta}$. The policy gradient methods are learning the policy weights based on the gradients of some performance measure $\eta(\boldsymbol{\theta})$. These methods seek to maximize the performance, by training the weights vector, therefore the weights update rule is given by gradient ascent, i.e.

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha\widehat{\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta}_t)}, \tag{2}$$

where α is the step size parameter, and $\widehat{\nabla\eta(\boldsymbol{\theta}_t)}$ is the stochastic estimation of the gradient of $\eta(\boldsymbol{\theta}_t)$ with respect to the weights vector $\boldsymbol{\theta}$. In policy gradient methods, the policy can by parameterized in many ways, as long as $\pi(a|s,\boldsymbol{\theta})$ is differentiable with respect to its parameters, i.e. as long as $\nabla_\theta\pi(a|s,\boldsymbol{\theta})$ exists. In order to ensure exploration, we demand that the policy will never become deterministic, i.e. $\pi(a|s,\boldsymbol{\theta}) \in (0,1)$.

Example 1 (Softmax Policy) Consider the case of discrete and finite action space, where each action-state pair receives a numerical preference $h(s, a, \boldsymbol{\theta}) \in \mathbb{R}$ according to the learned policy parameters $\boldsymbol{\theta}$. The preference is given by a linear transformation

$$h(s, a, \boldsymbol{\theta}) = \boldsymbol{\theta}^T \boldsymbol{\phi}(s, a), \tag{3}$$

where the features vector $\boldsymbol{\phi}(s, a)$ is given by a feature extraction. We would like to give higher probability to actions with higher preference value, using a softmax function, i.e.

$$\begin{aligned} \pi(a|s, \boldsymbol{\theta}) &= \frac{\exp\left(h(s, a, \boldsymbol{\theta})\right)}{\sum_{a'} \exp\left(h(s, a', \boldsymbol{\theta})\right)} \\ &= \frac{\exp\left(\boldsymbol{\theta}^T \boldsymbol{\phi}(s, a)\right)}{\sum_{a'} \exp\left(\boldsymbol{\theta}^T \boldsymbol{\phi}(s, a')\right)}, \end{aligned} \tag{4}$$

where for each state s the entire actions probabilities sums to one. Notice that the softmax function satisfies the exploration demand, as all softmax outputs are within range $(0, 1)$, such that a deterministic probability is never reached.

III. Policy Gradient Theorem

Consider the episodic case where the performance measurement, $\eta(\boldsymbol{\theta})$, is given by the term

$$\eta(\boldsymbol{\theta}) = v_{\pi_\theta}(s_0), \tag{5}$$

where $v_{\pi_\theta}(s_0)$ is the true value function for the policy π_θ determined by the parameters set $\boldsymbol{\theta}$, starting from the deterministic state s_0. According to equation (2), the parameters are updated with the approximation of the performance measure gradient. The policy gradient theorem provide us with the analytic expression to the performance measure gradient.

Theorem 1 (Policy Gradient Theorem) The policy gradient theorem states that

$$\nabla \eta(\boldsymbol{\theta}) = \sum_s d_\pi(s) \sum_a Q_\pi(s, a) \nabla_{\boldsymbol{\theta}} \pi(a|s, \boldsymbol{\theta}), \tag{6}$$

where $Q_\pi(s, a)$ is the action-value function for policy π and $d_\pi(s)$ is the weighting of state s given by its steady-state distribution, i.e.

$$d_\pi(s) \triangleq \lim_{t \to \infty} Pr\{S_t = s | A_{0:t-1} \sim \pi\}. \tag{7}$$

Notice that $d_\pi(s)$ is determined by how often state s is encountered starting at any S_0 and following policy π, and this limit is assumed to exist and be independent of S_0.

Proof 1 Assuming episodic case, with the performance measure as defined above and the discount factor γ, the gradient is given by

$$\begin{aligned}
\nabla v_{\pi_\theta}(s_0) &= \nabla_{\boldsymbol{\theta}} \left[\sum_a \pi(a|s) Q_\pi(s,a) \right], \ \forall s \in \mathcal{S} \\
&\overset{(a)}{=} \sum_a \left[\nabla_{\boldsymbol{\theta}} \pi(a|s) Q_\pi(s,a) + \pi(a|s) \nabla_{\boldsymbol{\theta}} Q_\pi(s,a) \right] \\
&\overset{(b)}{=} \sum_a \left[\nabla_{\boldsymbol{\theta}} \pi(a|s) Q_\pi(s,a) + \pi(a|s) \nabla_{\boldsymbol{\theta}} \sum_{s',r} p(s',r|s,a)(r + \gamma v_\pi(s')) \right] \\
&\overset{(c)}{=} \sum_a \left[\nabla_{\boldsymbol{\theta}} \pi(a|s) Q_\pi(s,a) + \pi(a|s) \sum_{s'} \gamma p(s'|s,a) v_\pi(s') \right] \\
&\overset{(d)}{=} \sum_a \Bigg[\nabla_{\boldsymbol{\theta}} \pi(a|s) Q_\pi(s,a) + \pi(a|s) \sum_{s'} \gamma p(s'|s,a) \\
&\qquad \sum_{a'} \left[\nabla_{\boldsymbol{\theta}} \pi(a'|s') Q_\pi(s',a') + \pi(a'|s') \sum_{s''} \gamma p(s''|s',a') v_\pi(s'') \right] \Bigg] \\
&\overset{(e)}{=} \sum_{x \in \mathcal{S}} \sum_{k=0}^{\infty} \gamma^k Pr(s \to x, k, \pi) \sum_a \nabla_{\boldsymbol{\theta}} \pi(a|x) Q_\pi(x,a), \qquad (8)
\end{aligned}$$

where

(a) - follows the product rule,

(b) - follows the definition of $Q_\pi(s,a)$,

(c) - follows that $\boldsymbol{\theta}$ is independent of r,

(d) - follows single unrolling step (positioning the value function derivation argument once),

(e) - follows multiple unrolling steps, where $Pr(s \to x, k, \pi)$ is the probability to reach from initial state s to state x after k steps, following policy π.

Now it is immediate to obtain

$$\nabla_{\boldsymbol{\theta}} \eta(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} v_{\pi_\theta}(s_0)$$

$$
\begin{aligned}
&= \sum_s \left(\sum_{k=0}^{\infty} \gamma^k Pr(s_0 \to s, k, \pi) \right) \sum_a \nabla_{\boldsymbol{\theta}} \pi(a|s) Q_\pi(s,a) \\
&\overset{(f)}{=} \sum_s d_\pi(s) \sum_a Q_\pi(s,a) \nabla_{\boldsymbol{\theta}} \pi(a|s),
\end{aligned}
\tag{9}
$$

where (f) is the on-policy distribution (episode relative amount of steps spent in state s in average). ■

IV. REINFORCE: MONTE CARLO POLICY GRADIENT

The policy gradient theorem in Equation (10) gives us an exact expression of the gradient needed for applying gradient ascent step. The gradient expression can be approximated by summing over the states following policy π weighted by γ times the number of steps it takes to get to these states. Thus

$$
\begin{aligned}
\nabla \eta(\boldsymbol{\theta}) &= \sum_s d_\pi(s) \sum_a Q_\pi(s,a) \nabla_{\boldsymbol{\theta}} \pi(a|s, \boldsymbol{\theta}), \\
&= \mathbb{E}_\pi \left[\gamma^t \sum_a Q_\pi(S_t, a) \nabla_{\boldsymbol{\theta}} \pi(a|S_t, \boldsymbol{\theta}) \right].
\end{aligned}
\tag{10}
$$

Now we have an expression that include summation over all actions for each state S_t. In order to replace the sum over a we multiply and divide by the probability to select each action A_t according to π, to obtain

$$
\begin{aligned}
\nabla \eta(\boldsymbol{\theta}) &= \mathbb{E}_\pi \left[\gamma^t \sum_a \pi(a|S_t, \boldsymbol{\theta}) Q_\pi(S_t, a) \frac{\nabla_{\boldsymbol{\theta}} \pi(a|S_t, \boldsymbol{\theta})}{\pi(a|S_t, \boldsymbol{\theta})} \right] \\
&\overset{(a)}{=} \mathbb{E}_\pi \left[\gamma^t Q_\pi(S_t, A_t) \frac{\nabla_{\boldsymbol{\theta}} \pi(A_t|S_t, \boldsymbol{\theta})}{\pi(A_t|S_t, \boldsymbol{\theta})} \right] \\
&\overset{(b)}{=} \mathbb{E}_\pi \left[\gamma^t G_t \frac{\nabla_{\boldsymbol{\theta}} \pi(A_t|S_t, \boldsymbol{\theta})}{\pi(A_t|S_t, \boldsymbol{\theta})} \right] \\
&\overset{(c)}{=} \mathbb{E}_\pi \left[\gamma^t G_t \nabla_{\boldsymbol{\theta}} \ln \pi(A_t|S_t, \boldsymbol{\theta}) \right],
\end{aligned}
\tag{11}
$$

where (a) is the replacement of a with an action $A_t \sim \pi$, (b) follows that

$$
\mathbb{E}_\pi[G_t|S_t, A_t] = Q_\pi(S_t, A_t), \tag{12}
$$

and (c) follows the equality $\nabla \ln x = \frac{\nabla x}{x}$.

We now hold a quantity as shown in Equation (11) that can be sampled each time step, whose expectation equals to the gradient according to the policy gradient theorem. Applying the update rule as a gradient ascent algorithm yields

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \gamma^t G_t \nabla_{\boldsymbol{\theta}} \ln \pi_\theta(A_t|S_t, \boldsymbol{\theta}_t), \tag{13}$$

where α is a fixed step size parameter. This algorithm is called REINFORCE after [3]. The update rule as described in Equation (13) increases the parameter vector in the direction that increases the probability of taking A_t when returning to S_t in the future, and is proportional to the return G_t at each time step. Moreover, the division of the gradient vector by the probability of taking action A_t eliminates the advantage given to actions that are taken more often. During this lecture, the return G_t at each step t is given by

$$\begin{aligned} G_t &= R_{t+1} + \gamma R_{t+2} + \cdots \\ &= \sum_{s=t}^{T-1} \gamma^{s-t} R_{s+1}, \end{aligned} \tag{14}$$

where $t \in \{1, \ldots, T-1\}$. REINFORCE is a Monte-Carlo algorithm, well defined only for the episodic case (as we learned in Lecture 3) where all updates are made retrospectively after each episode is completed, as described in the algorithm below.

Algorithm 1 REINFORCE (episodic)

input: $\pi(a|s, \boldsymbol{\theta})$ - a differentiable policy parameterization.

initialize parameters vector $\boldsymbol{\theta} \in \mathbb{R}^d$.

repeat forever:

Generate episode states and take actions according to $\pi(\cdot|\cdot, \boldsymbol{\theta})$, receive sequence $\{S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T\}$.

for each step in $t = 0 \ldots T-1$ **do**

$G \leftarrow$ return from step t

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \gamma^t G \nabla_{\boldsymbol{\theta}} \ln \pi(A_t|S_t, \boldsymbol{\theta})$

A. REINFORCE with baseline

The policy gradient theorem in Equation (10) can be modified to a general form that includes a comparison of the action-value function to some arbitrary baseline $b(s)$, i.e.

$$\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta}) = \sum_s d_\pi(s) \sum_a \left(Q_\pi(s,a) - b(s)\right) \nabla_{\boldsymbol{\theta}}\pi(a|s,\boldsymbol{\theta}). \tag{15}$$

Adding such a baseline argument can reduce the .Notice that the baseline function can be any function or random variable, as long as it does not vary with a. The validity of Equation (15) is true for the internal summation subtraction always sums to zero for any $b(s)$, i.e.

$$\begin{aligned}\sum_a b(s)\nabla_{\boldsymbol{\theta}}\pi(a|s,\boldsymbol{\theta}) &= b(s)\nabla_{\boldsymbol{\theta}} \sum_a \pi(a|s,\boldsymbol{\theta}) \\ &= b(s)\nabla_{\boldsymbol{\theta}} 1 \\ &= 0. \end{aligned} \tag{16}$$

The generalized policy gradient theorem with baseline yields the REINFORCE with baseline gradient update rule, i.e.

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha\gamma^t \left(G_t - b(S_t)\right) \nabla_{\boldsymbol{\theta}} \ln \pi_\theta(A_t|S_t,\boldsymbol{\theta}_t), \tag{17}$$

where using a baseline which is uniformly zero, leaves us with the previous REINFORCE update rule as described in Equation (13).

Using the REINFORCE with baseline algorithm, a natural choice for the baseline will be the value function estimation $\hat{v}(S_t,\boldsymbol{w})$, where $\boldsymbol{w} \in \mathbb{R}^m$ is the weight vector learned for the estimation, as described in the previous lecture. Using the value function estimation as a baseline may improve convergence and speed it up as shown in ([2]), the algorithm is described below.

Algorithm 2 REINFORCE with baseline (episodic)

input: $\pi(a|s, \boldsymbol{\theta})$ - a differentiable policy parameterization.

input: $\hat{v}(S_t, \boldsymbol{w})$ - a differentiable state-value parameterization.

parameters: $\alpha_{\boldsymbol{\theta}}, \alpha_{\boldsymbol{w}}$ - step sizes.

initialize parameters vector $\boldsymbol{\theta} \in \mathbb{R}^d$ and state-value weights $\boldsymbol{w} \in \mathbb{R}^m$.

repeat forever:

Generate episode states and take actions according to $\pi(\cdot|\cdot, \boldsymbol{\theta})$, receive sequence $\{S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T\}$.

for each step in $t = 0 \ldots T-1$ **do**

$G \leftarrow$ return from step t

$\delta \leftarrow G - \hat{v}(S_t, \boldsymbol{w})$

$\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha_{\boldsymbol{w}} \gamma^t \delta \nabla_{\boldsymbol{w}} \hat{v}(S_t, \boldsymbol{w})$

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha_{\boldsymbol{\theta}} \gamma^t \delta \nabla_{\boldsymbol{\theta}} \ln \pi(A_t|S_t, \boldsymbol{\theta})$

REFERENCES

[1] D. Silver. Lecture notes in advanced topics in reinforcement learning, lecture 7, January 2015.

[2] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 2017.

[3] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*. Springer, 1992.