RL Ch13 Policy Gradient Methods

Le-Rong Hsu

July 2023

1 Introduction

Goal

Learn a **parameterized policy** $\pi(a|s,\theta)$ where $\theta \in \mathbb{R}^{d'}$ is the parameter vector. A value function can be used to **learn** the policy parameter, but is not required for action selection.

Methods

We seek to maximize the performance, through "gradient methods" (but not exactly same)

$$\theta_{t+1} = \theta_t + \alpha\widehat{\nabla J(\theta_t)} \tag{13.1}$$

where $\widehat{\nabla J(\theta_t)} \in \mathbb{R}^{d'}$ which requires that **the expectation of the sample gradient is proportional to the actual gradient** of the performance measure. All these methods following this schema are called **policy gradient methods**. Methods that learn approximations to both policy and value functions are often called **actor–critic methods**, where '**actor**' is a reference to the **learned policy**, and '**critic**' refers to the **learned value function**.

2 Policy Approximation and its Advantages

Assume $\nabla\pi(a|s,\theta)$ (column vector of partial derivatives w.r.t. the components of θ) exists and is finite for all $s \in \mathcal{S}, a \in \mathcal{A}$ and $\theta \in \mathbb{R}^{d'}$.

To **ensure exploration**, we hope $\pi(a|s,\theta) > 0$ for all s, a, θ. The most common parameterization for **discrete actions space** (but not too large) is through **exponential soft-max** distribution,

$$\pi(a|s,\theta) \doteq \frac{e^{h(s,a,\theta)}}{\sum_b e^{h(s,b,\theta)}}, \tag{13.2}$$

under which the actions with highest preference is selected.

The action preferences themselves can be parameterized arbitrarily. The preferences could simply be linear in features,

$$h(s, a, \theta) = \theta^T \mathbf{x}(s, a) \tag{13.3}$$

using feature vectors $\mathbf{x}(s, a) \in \mathbb{R}^{d'}$.

soft-max vs. ε-greedy

soft-max:

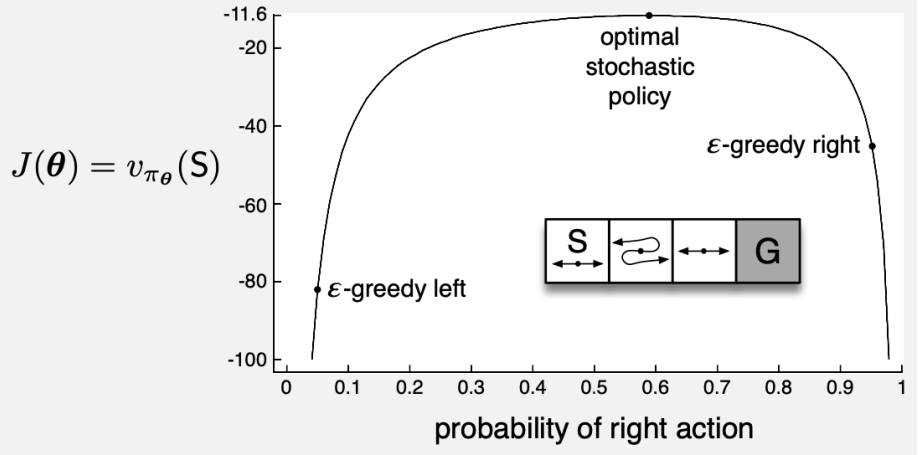
- the approximate policy can approach a deterministic policy
- enables the selection of actions with arbitrary probabilities. In some situations, the best approximate policy may be stochastic.

ε-greedy:

- always has ε probability to choose a random action
- not flexible (which action has prob. $1 - \varepsilon$?)

Example 13.1 The ε-greedy policy is forced to choose which action, **left** or right, has prob. $1 - \varepsilon$.

- reward = -1 for all steps
- each (nonterminal) state has two actions: **left** or **right**



Example 13.1 Short corridor with switched actions

3 The Policy Gradient Theorem

We treat the episodic case. Without loss of generality, assume that every episode starts in some particular (non-random) state s_0.

First, we have to **measure the performance** of a policy determined by θ. We define the performance as

$$J(\theta) \doteq v_{\pi_\theta}(s_0) \tag{13.4}$$

where v_{π_θ} is the true value of π_θ, s_0 is the initial state.

Goal

Show that the policy gradient methods indeed improve the policy (by changing the policy parameter).

Observation

- The **action selections** and **distribution of states** are affected by the policy parameter.
- Given a state, the effect of the policy parameter on the actions can be computed in a relatively straightforward.
- The state distribution is typically **unknown** (since the environment is unknown).

Proof

The most interesting thing in this proof is that it doesn't involve the derivative of the state distribution. Some formulae are applied in the proof:

$$p(s'|s,a) \doteq P_r\{S_t = s'|S_{t-1}, A_{t-1} = a\} = \sum_{r\in\mathcal{R}} p(s', r|s, a) \tag{3.4}$$

$$\eta(s) = h(s) + \sum_{\bar{s}} \eta(\bar{s}) \sum_a \pi(a|\bar{s}) p(s|\bar{s}, a) \tag{9.2}$$

$$\mu(s) = \frac{\eta(s)}{\sum_{s'} \eta(s')} \tag{9.3}$$

The policy gradient theorem finally establishes that

$$\nabla J(\theta) \propto \sum_s \mu(s) \sum_a q_\pi(s, a) \nabla \pi(a|s, \theta) \tag{13.5}$$

Proof of the Policy Gradient Theorem (episodic case)

With just elementary calculus and re-arranging of terms, we can prove the policy gradient theorem from first principles. To keep the notation simple, we leave it implicit in all cases that π is a function of $\boldsymbol{\theta}$, and all gradients are also implicitly with respect to $\boldsymbol{\theta}$. First note that the gradient of the state-value function can be written in terms of the action-value function as

$$
\begin{aligned}
\nabla v_\pi(s) &= \nabla \left[\sum_a \pi(a|s) q_\pi(s,a) \right], \quad \text{for all } s \in \mathcal{S} && \text{(Exercise 3.18)} \\
&= \sum_a \Big[\nabla \pi(a|s) q_\pi(s,a) + \pi(a|s) \nabla q_\pi(s,a) \Big] && \text{(product rule of calculus)} \\
&= \sum_a \Big[\nabla \pi(a|s) q_\pi(s,a) + \pi(a|s) \nabla \sum_{s',r} p(s',r|s,a) \big(r + v_\pi(s')\big) \Big] \\
& && \text{(Exercise 3.19 and Equation 3.2)} \\
&= \sum_a \Big[\nabla \pi(a|s) q_\pi(s,a) + \pi(a|s) \sum_{s'} p(s'|s,a) \nabla v_\pi(s') \Big] && \text{(Eq. 3.4)} \\
&= \sum_a \Big[\nabla \pi(a|s) q_\pi(s,a) + \pi(a|s) \sum_{s'} p(s'|s,a) && \text{(unrolling)} \\
&\qquad \sum_{a'} \big[\nabla \pi(a'|s') q_\pi(s',a') + \pi(a'|s') \sum_{s''} p(s''|s',a') \nabla v_\pi(s'') \big] \Big] \\
&= \sum_{x \in \mathcal{S}} \sum_{k=0}^{\infty} \Pr(s \to x, k, \pi) \sum_a \nabla \pi(a|x) q_\pi(x,a),
\end{aligned}
$$

after repeated unrolling, where $\Pr(s \to x, k, \pi)$ is the probability of transitioning from state s to state x in k steps under policy π. It is then immediate that

$$
\begin{aligned}
\nabla J(\boldsymbol{\theta}) &= \nabla v_\pi(s_0) \\
&= \sum_s \left(\sum_{k=0}^{\infty} \Pr(s_0 \to s, k, \pi) \right) \sum_a \nabla \pi(a|s) q_\pi(s,a) \\
&= \sum_s \eta(s) \sum_a \nabla \pi(a|s) q_\pi(s,a) && \text{(box page 199)} \\
&= \sum_{s'} \eta(s') \sum_s \frac{\eta(s)}{\sum_{s'} \eta(s')} \sum_a \nabla \pi(a|s) q_\pi(s,a) \\
&= \sum_{s'} \eta(s') \sum_s \mu(s) \sum_a \nabla \pi(a|s) q_\pi(s,a) && \text{(Eq. 9.3)} \\
&\propto \sum_s \mu(s) \sum_a \nabla \pi(a|s) q_\pi(s,a) && \text{(Q.E.D.)}
\end{aligned}
$$

4 REINFORCE: Monte Carlo Policy Gradient

Recall that

$$\theta_{t+1} = \theta_t + \alpha\widehat{\nabla J(\theta_t)} \tag{13.1}$$

requires a way of sampling such that the expectation of the sample gradient should be proportional to the actual gradient. Since step size α can absorb the proportionality, the sample gradients need only be proportional to the gradient.

$$\begin{aligned}\nabla J(\theta) &\propto \sum_s \mu(s) \sum_a q_\pi(s,a)\nabla\pi(a|s,\theta) \\ &= \mathbb{E}_\pi\left[\sum_a q_\pi(S_t,a)\nabla\pi(a|S_t,\theta)\right] \end{aligned} \tag{13.6}$$

We then can write down the SGD algorithm (13.1)

$$\theta_{t+1} \doteq \theta_t + \alpha \sum_a \hat{q}(S_t,a,\mathbf{w})\nabla\pi(a|S_t,\theta) \tag{13.7}$$

where $\hat{q}$ is some learned approximation to q_π. This algorithm is called an **all-actions method** since its update involves all of the actions.

REINFORCE algorithm

The update at time t involves just A_t, the one action actually taken at time t.

$$\begin{aligned}\nabla J(\theta) &\propto \mathbb{E}_\pi\left[\sum_a \pi(a|S_t,\theta) q_\pi(S_t,a)\frac{\nabla\pi(a|S_t,\theta)}{\pi(a|S_t,\theta)}\right] \\ &= \mathbb{E}_\pi\left[q_\pi(S_t,A_t)\frac{\nabla\pi(A_t|S_t,\theta)}{\pi(A_t|S_t,\theta)}\right] && \text{replacing } a \text{ by the sample } A_t \sim \pi \\ &= \mathbb{E}_\pi\left[G_t\frac{\nabla\pi(A_t|S_t,\theta)}{\pi(A_t|S_t,\theta)}\right] && \text{because } \mathbb{E}_\pi[G_t|S_t,A_t] = q_\pi(S_t,A_t)\end{aligned}$$

Then the final expression in brackets is a quantity that can be sampled on each time step. The sample yields the REINFORCE update:

$$\theta_{t+1} \doteq \theta_t + \alpha G_t \frac{\nabla\pi(A_t|S_t,\theta)}{\pi(A_t|S_t,\theta)} \tag{13.8}$$

Remark. The vector is the direction in parameter space that most increases the probability of repeating the action A_t on future visits to state S_t.

REINFORCE: Monte-Carlo Policy-Gradient Control (episodic) for π_*

Input: a differentiable policy parameterization $\pi(a|s,\boldsymbol{\theta})$
Algorithm parameter: step size $\alpha > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):
  Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot,\boldsymbol{\theta})$
  Loop for each step of the episode $t = 0, 1, \ldots, T-1$:
    $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$ $\qquad (G_t)$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \gamma^t G \nabla \ln \pi(A_t|S_t,\boldsymbol{\theta})$

Note that we use $\nabla \ln \pi(A_t|S_t,\theta_t)$ to express $\frac{\nabla \pi(A_t|S_t,\theta)}{\pi(A_t|S_t,\theta)}$. The expressions are equivalent because $\nabla \ln x = \frac{\nabla x}{x}$. We will refer to it simply as the eligibility vector.

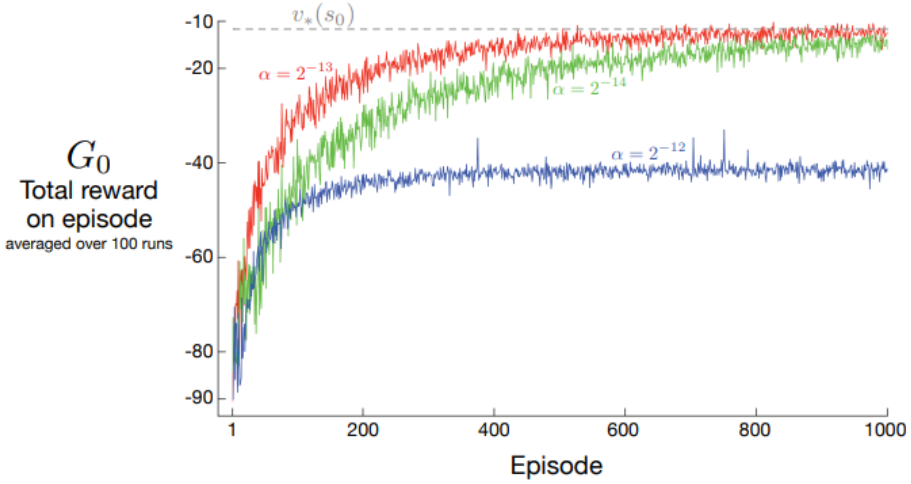


Figure 13.1 With a good step size, the total reward per episode approaches the optimal value of the start state.

5 REINFORCE with Baseline

The policy gradient theorem (13.5) can be generalized to include a comparison of the action value to an arbitrary **baseline** $b(s)$:

$$\nabla J(\theta) \propto \sum_s \mu(s) \sum_a (q_\pi(s,a) - b(s)) \nabla \pi(a|s,\theta) \qquad (13.10)$$

The baseline can be any function, even a random variable, as long as it does not vary with a.

$$\sum_a b(s)\nabla\pi(a|s,\theta) = b(s)\nabla\sum_a \pi(a|s,\theta) = b(s)\nabla(1) = 0.$$

REINFORCE with baseline:

$$\theta_{t+1} \doteq \theta_t + \alpha(G_t - b(S_t))\frac{\nabla\pi(A_t|S_t,\theta_t)}{\pi(A_t|S_t,\theta_t)}. \tag{13.11}$$

In general, the baseline leaves the expected value of the update unchanged, but it can have a large effect on its variance.

One natural choice for the baseline is an estimate of the state value $\hat{v}(S_t, \mathbf{w})$.

Remark. Only one state S_t is needed.

REINFORCE with Baseline (episodic), for estimating $\pi_\theta \approx \pi_*$

Input: a differentiable policy parameterization $\pi(a|s,\boldsymbol{\theta})$
Input: a differentiable state-value function parameterization $\hat{v}(s,\mathbf{w})$
Algorithm parameters: step sizes $\alpha^{\boldsymbol{\theta}} > 0$, $\alpha^{\mathbf{w}} > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ and state-value weights $\mathbf{w} \in \mathbb{R}^d$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):
  Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot,\boldsymbol{\theta})$
  Loop for each step of the episode $t = 0, 1, \ldots, T-1$:
    $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$ $\qquad (G_t)$
    $\delta \leftarrow G - \hat{v}(S_t,\mathbf{w})$
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}}\delta\nabla\hat{v}(S_t,\mathbf{w})$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha^{\boldsymbol{\theta}}\gamma^t\delta\nabla\ln\pi(A_t|S_t,\boldsymbol{\theta})$

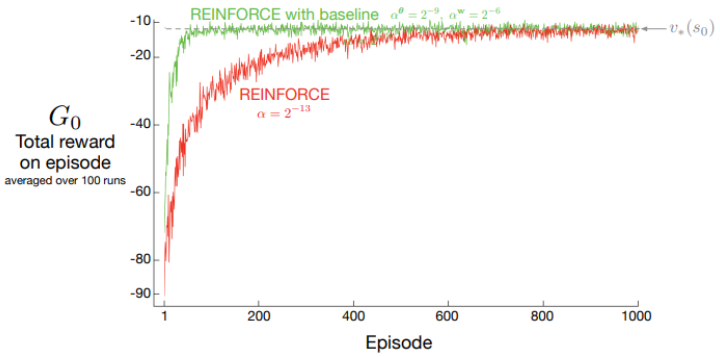


Figure 13.2 Adding a baseline to REINFORCE can make it learn much faster. The step size for REINFORCE is that at which it performs best.

6 Actor–Critic Methods

- **critic**: **state-value function** is used to assess actions
- REINFORCE with baseline estimates the value of the only the first state of each state transition.
- $G_{t:t+1} = R_{t+1} + \gamma\hat{v}(S_{t+1}, \mathbf{w}_t)$ is a useful estimate of the actual return
- n-step returns and eligibility traces is available in actor-critic methods

$$\theta_{t+1} \doteq \theta_t + \alpha\Big(G_{t:t+1} - \hat{v}(S_t, \mathbf{w})\Big)\frac{\nabla\pi(A_t|S_t,\theta_t)}{\pi(A_t|S_t,\theta_t)} \tag{13.12}$$

$$= \theta_t + \alpha\Big(R_{t+1} + \gamma\hat{v}(S_{t+1}, \mathbf{w}) - \hat{v}(S_t, \mathbf{w})\Big)\frac{\nabla\pi(A_t|S_t,\theta_t)}{\pi(A_t|S_t,\theta_t)} \tag{13.13}$$

$$= \theta_t + \alpha\delta_t\frac{\nabla\pi(A_t|S_t,\theta_t)}{\pi(A_t|S_t,\theta_t)} \tag{13.14}$$

The generalizations to the forward view of n-step methods and then to a λ-

One-step Actor–Critic (episodic), for estimating $\pi_{\boldsymbol{\theta}} \approx \pi_*$

Input: a differentiable policy parameterization $\pi(a|s, \boldsymbol{\theta})$
Input: a differentiable state-value function parameterization $\hat{v}(s,\mathbf{w})$
Parameters: step sizes $\alpha^{\boldsymbol{\theta}} > 0$, $\alpha^{\mathbf{w}} > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ and state-value weights $\mathbf{w} \in \mathbb{R}^d$ (e.g., to $\mathbf{0}$)
Loop forever (for each episode):
  Initialize S (first state of episode)
  $I \leftarrow 1$
  Loop while S is not terminal (for each time step):
    $A \sim \pi(\cdot|S, \boldsymbol{\theta})$
    Take action A, observe S', R
    $\delta \leftarrow R + \gamma\hat{v}(S',\mathbf{w}) - \hat{v}(S,\mathbf{w})$ (if S' is terminal, then $\hat{v}(S',\mathbf{w}) \doteq 0$)
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}}\delta\nabla\hat{v}(S,\mathbf{w})$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha^{\boldsymbol{\theta}} I\delta\nabla\ln\pi(A|S,\boldsymbol{\theta})$
    $I \leftarrow \gamma I$
    $S \leftarrow S'$

return algorithm are straightforward. The one-step return in (13.12) is merely replaced by $G_{t:t+n}$ or G_t^λ respectively.

Actor–Critic with Eligibility Traces (episodic), for estimating $\pi_{\boldsymbol{\theta}} \approx \pi_*$

Input: a differentiable policy parameterization $\pi(a|s,\boldsymbol{\theta})$
Input: a differentiable state-value function parameterization $\hat{v}(s,\mathbf{w})$
Parameters: trace-decay rates $\lambda^{\boldsymbol{\theta}} \in [0,1]$, $\lambda^{\mathbf{w}} \in [0,1]$; step sizes $\alpha^{\boldsymbol{\theta}} > 0$, $\alpha^{\mathbf{w}} > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ and state-value weights $\mathbf{w} \in \mathbb{R}^{d}$ (e.g., to $\mathbf{0}$)
Loop forever (for each episode):
  Initialize S (first state of episode)
  $\mathbf{z}^{\boldsymbol{\theta}} \leftarrow \mathbf{0}$ (d'-component eligibility trace vector)
  $\mathbf{z}^{\mathbf{w}} \leftarrow \mathbf{0}$ (d-component eligibility trace vector)
  $I \leftarrow 1$
  Loop while S is not terminal (for each time step):
    $A \sim \pi(\cdot|S,\boldsymbol{\theta})$
    Take action A, observe S', R
    $\delta \leftarrow R + \gamma \hat{v}(S',\mathbf{w}) - \hat{v}(S,\mathbf{w})$ (if S' is terminal, then $\hat{v}(S',\mathbf{w}) \doteq 0$)
    $\mathbf{z}^{\mathbf{w}} \leftarrow \gamma \lambda^{\mathbf{w}} \mathbf{z}^{\mathbf{w}} + \nabla \hat{v}(S,\mathbf{w})$
    $\mathbf{z}^{\boldsymbol{\theta}} \leftarrow \gamma \lambda^{\boldsymbol{\theta}} \mathbf{z}^{\boldsymbol{\theta}} + I \nabla \ln \pi(A|S,\boldsymbol{\theta})$
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}} \delta \mathbf{z}^{\mathbf{w}}$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha^{\boldsymbol{\theta}} \delta \mathbf{z}^{\boldsymbol{\theta}}$
    $I \leftarrow \gamma I$
    $S \leftarrow S'$

7 Policy Gradient for Continuing Problems

We need to define performance in terms of the average rate of reward per time step:

$$\begin{aligned} J(\theta) \doteq r(\pi) &\doteq \lim_{h\to\infty} \frac{1}{h} \sum_{t=1}^{h} \mathbb{E}[R_t | S_0, A_{0:t-1} \sim \pi] \\ &= \lim_{t\to\infty} \mathbb{E}[R_t | S_0, A_{0:t-1} \sim \pi] \\ &= \sum_s \mu(s) \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a) r \end{aligned} \tag{13.15}$$

where μ is the steady-state distribution under π: $\mu(s) \doteq \lim_{t\to\infty} \Pr\{S_t = s | A_{0:t} \sim \pi\}$. Also, recall that if you select actions according to π, you remain in the same distribution:

$$\sum_s \mu(s) \sum_a \pi(a|s,\theta) p(s'|s,a) = \mu(s'), \text{ for all } s' \in \mathcal{S}. \tag{13.16}$$

Actor–Critic with Eligibility Traces (continuing), for estimating $\pi_{\boldsymbol{\theta}} \approx \pi_*$

Input: a differentiable policy parameterization $\pi(a|s,\boldsymbol{\theta})$
Input: a differentiable state-value function parameterization $\hat{v}(s,\mathbf{w})$
Algorithm parameters: $\lambda^{\mathbf{w}} \in [0,1]$, $\lambda^{\boldsymbol{\theta}} \in [0,1]$, $\alpha^{\mathbf{w}} > 0$, $\alpha^{\boldsymbol{\theta}} > 0$, $\alpha^{\bar{R}} > 0$
Initialize $\bar{R} \in \mathbb{R}$ (e.g., to 0)
Initialize state-value weights $\mathbf{w} \in \mathbb{R}^d$ and policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ (e.g., to $\mathbf{0}$)
Initialize $S \in \mathcal{S}$ (e.g., to s_0)

$\mathbf{z}^{\mathbf{w}} \leftarrow \mathbf{0}$ (d-component eligibility trace vector)
$\mathbf{z}^{\boldsymbol{\theta}} \leftarrow \mathbf{0}$ (d'-component eligibility trace vector)
Loop forever (for each time step):
  $A \sim \pi(\cdot|S,\boldsymbol{\theta})$
  Take action A, observe S', R
  $\delta \leftarrow R - \bar{R} + \hat{v}(S',\mathbf{w}) - \hat{v}(S,\mathbf{w})$
  $\bar{R} \leftarrow \bar{R} + \alpha^{\bar{R}}\delta$
  $\mathbf{z}^{\mathbf{w}} \leftarrow \lambda^{\mathbf{w}}\mathbf{z}^{\mathbf{w}} + \nabla\hat{v}(S,\mathbf{w})$
  $\mathbf{z}^{\boldsymbol{\theta}} \leftarrow \lambda^{\boldsymbol{\theta}}\mathbf{z}^{\boldsymbol{\theta}} + \nabla \ln \pi(A|S,\boldsymbol{\theta})$
  $\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}}\delta\mathbf{z}^{\mathbf{w}}$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha^{\boldsymbol{\theta}}\delta\mathbf{z}^{\boldsymbol{\theta}}$
  $S \leftarrow S'$

In the continuing case, it's natural to define the differential return:

$$G_t \doteq R_{t+1} - r(\pi) + R_{t+2} - r(\pi) + R_{t+3} - r(\pi) \cdots \tag{13.17}$$

Proof of the Policy Gradient Theorem (continuing case)

The proof of the policy gradient theorem for the continuing case begins similarly to the episodic case. Again we leave it implicit in all cases that π is a function of $\boldsymbol{\theta}$ and that the gradients are with respect to $\boldsymbol{\theta}$. Recall that in the continuing case $J(\boldsymbol{\theta}) = r(\pi)$ (13.15) and that v_π and q_π denote values with respect to the differential return (13.17). The gradient of the state-value function can be written, for any $s \in \mathcal{S}$, as

$$\begin{aligned}
\nabla v_\pi(s) &= \nabla \left[\sum_a \pi(a|s) q_\pi(s,a) \right], \quad \text{for all } s \in \mathcal{S} && \text{(Exercise 3.18)} \\
&= \sum_a \Big[\nabla\pi(a|s) q_\pi(s,a) + \pi(a|s) \nabla q_\pi(s,a) \Big] && \text{(product rule of calculus)} \\
&= \sum_a \Big[\nabla\pi(a|s) q_\pi(s,a) + \pi(a|s) \nabla \sum_{s',r} p(s',r|s,a) \big(r - r(\boldsymbol{\theta}) + v_\pi(s')\big) \Big] \\
&= \sum_a \Big[\nabla\pi(a|s) q_\pi(s,a) + \pi(a|s) \big[-\nabla r(\boldsymbol{\theta}) + \sum_{s'} p(s'|s,a) \nabla v_\pi(s') \big] \Big].
\end{aligned}$$

After re-arranging terms, we obtain

$$\nabla r(\boldsymbol{\theta}) = \sum_a \Big[\nabla\pi(a|s) q_\pi(s,a) + \pi(a|s) \sum_{s'} p(s'|s,a) \nabla v_\pi(s') \Big] - \nabla v_\pi(s).$$

Notice that the left-hand side can be written $\nabla J(\boldsymbol{\theta})$, and that it does not depend on s. Thus the right-hand side does not depend on s either, and we can safely sum it over all $s \in \mathcal{S}$, weighted by $\mu(s)$, without changing it (because $\sum_s \mu(s) = 1$):

$$\begin{aligned}
\nabla J(\boldsymbol{\theta}) &= \sum_s \mu(s) \left(\sum_a \Big[\nabla\pi(a|s) q_\pi(s,a) + \pi(a|s) \sum_{s'} p(s'|s,a) \nabla v_\pi(s') \Big] - \nabla v_\pi(s) \right) \\
&= \sum_s \mu(s) \sum_a \nabla\pi(a|s) q_\pi(s,a) \\
&\quad + \sum_s \mu(s) \sum_a \pi(a|s) \sum_{s'} p(s'|s,a) \nabla v_\pi(s') - \sum_s \mu(s) \nabla v_\pi(s) \\
&= \sum_s \mu(s) \sum_a \nabla\pi(a|s) q_\pi(s,a) \\
&\quad + \sum_{s'} \underbrace{\sum_s \mu(s) \sum_a \pi(a|s) p(s'|s,a)}_{\mu(s') \text{ (13.16)}} \nabla v_\pi(s') - \sum_s \mu(s) \nabla v_\pi(s) \\
&= \sum_s \mu(s) \sum_a \nabla\pi(a|s) q_\pi(s,a) + \sum_{s'} \mu(s') \nabla v_\pi(s') - \sum_s \mu(s) \nabla v_\pi(s) \\
&= \sum_s \mu(s) \sum_a \nabla\pi(a|s) q_\pi(s,a). \qquad \text{Q.E.D.}
\end{aligned}$$

8 Policy Parameterization for Continuous Actions

For large action sets or continuous spaces with an infinite number of actions, we instead learn statistics of the probability distribution.

$$p(x) \doteq \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{13.18}$$

Mean and standard deviation can be parameterized by state s and policy parameter $\theta \in \mathbb{R}^{d'}$:

$$\pi(a|s,\theta) \doteq \frac{1}{\sigma(s,\theta)\sqrt{2\pi}} \exp\left(-\frac{(a-\mu(s,\theta)^2}{2\sigma(s,\theta)^2}\right) \tag{13.19}$$

where $\mu : \mathcal{S} \times \mathbb{R}^{d'} \to \mathbb{R}$ and $\sigma : \mathcal{S} \times \mathbb{R}^{d'} \to \mathbb{R}^+$ are parameterized function approximators. We divide the policy's parameter vector into two parts, $\theta = [\theta_\mu, \theta_\sigma]^T$.

$$\mu(s,\theta) \doteq \theta_\mu^T \mathbf{x}_\mu(s) \quad \text{and} \quad \sigma(s,\theta) \doteq \exp\left(\theta_\sigma^T \mathbf{x}_\sigma(s)\right) \tag{13.20}$$

where $\mathbf{x}_\mu(s)$ and $\mathbf{x}_\sigma(s)$ are state feature vectors.