13. Policy Gradient Methods

Notes

- Here, we consider methods that learn a *parameterised policy* that can select actions without consulting a value function. A value function may still be used to *learn* the policy parameter, but is not required for action selection. We denote the policy as $\pi(a|s, \boldsymbol{\theta}) = \Pr\{A_t = a | S_t = s, \boldsymbol{\theta}_t = \boldsymbol{\theta}\}$, where $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ is the policy's parameter vector.
- We learn the policy parameter based on the gradient of some scalar performance measure $J(\boldsymbol{\theta})$ w.r.t. the policy parameter. The methods discussed seek to *maximise* performance, and undergoes *gradient ascent* in J: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \widehat{\nabla J(\boldsymbol{\theta}_t)}$, where $\widehat{\nabla J(\boldsymbol{\theta}_t)} \in \mathbb{R}^{d'}$ is a stochastic estimate whose expectation approximates the gradient of the performance measure wrt its argument $\boldsymbol{\theta}_t$. Methods that follow this general schema are called *policy gradient methods*.
- Methods that learn approximations to both policy and value functions are called *actor-critic methods*, where the 'actor' refers to the learned policy and *critic* refers to the learned value function.
- Policy Approximation and its Advantages:
  - The policy can be parameterised in any way as long as $\pi(a|s, \boldsymbol{\theta})$ is differentiable w.r.t. its parameters. To ensure exploration, we require that the policy never becomes deterministic.
  - If the action space is not too large and discrete, it is then natural to form a parameterised numerical preference $h(s, a, \boldsymbol{\theta}) \in \mathbb{R}$ for each state-action pair. The policy can then be expressed using a soft-max distribution

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

    Such a policy parameterisation is called *soft-max in action preferences*. The action preferences can be parameterised arbitrarily. They could be computed by a deep ANN where $\boldsymbol{\theta}$ are the connection weights. They could also simply be linear in features

    $h(s, a, \boldsymbol{\theta}) = \boldsymbol{\theta}^\intercal \mathbf{x}(s, a)$.
  - Advantages of parameterising policies:

        - The approximate policy can approach a deterministic policy, unlike an ϵ-greedy policy. The action preferences are driven to produce the optimal stochastic policy.
        - The policy allows selection of actions with arbitrarily probabilities, where problems often have imperfect information and the optimal play is often more than one action.
        - Policy parameterisastion might be simpler than action-value parameterisation, but this would depend on the complexity of the environment. Sometimes, the action-value function is simpler to approximate, but for others the policy could be easier to approximate.
- The Policy Gradient Theorem:
    - The theoretical advantage of policy parameterisation is that the action probabilities change smoothly as a function of the learned parameter. Thus, stronger convergence is guaranteed for policy-gradient methods than for action-value methods. This continuity also enables gradient ascent to be approximated.
    - The performance measure $J(\boldsymbol{\theta})$ is defined differently for the episodic and continuing case.
    - In the episodic case, the performance measure is defined as the value of the start state of the episode: $J(\boldsymbol{\theta}) \doteq v_{\pi_\theta}(s_0)$. Here, we assume that all episodes start at state s_0, and v_{π_θ} is the true value function for π_θ.
    - The performance depends on the action selections and the distribution of states, and both of these are affected by the policy parameter. However, the effect of the policy on the state distribution is a function of the environment and is generally unknown.
    - The *policy gradient theorem* provides an expression for the gradient of the performance that does not involve the derivative of the state distribution. It is given by (for the episodic case)

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

      where the constant of proportionality is the average length of an episode, or is 1 in the continuing case. Proof can be found in page 325.
- REINFORCE: Monte Carlo Policy Gradient:
    - The gradient of the performance measure can be rewritten as

$$\nabla J(\boldsymbol{\theta}) \propto \mathbb{E}_\pi\left[\sum_a q_\pi(S_t, a)\nabla\pi(a|S_t, \boldsymbol{\theta})\right]$$

since the on-policy distribution describes how often a state occurs under the target policy π, thus it can be written as the expectation under π.

- For an *all-actions method*, the policy parameters can be updated with

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha\sum_a \hat{q}(S_t, a, \mathbf{w})\nabla\pi(a|S_t, \boldsymbol{\theta}).$$

  However, we are more interested in the classical REINFORCE algorithm, so we continue on.

- To derive the REINFORCE algorithm, we need to weight out actions by $\pi(a|S_t, \boldsymbol{\theta})$ in order for us to get rid of the sum and be an expectation under π. Thus

$$\begin{aligned}\nabla J(\theta) &\propto \mathbb{E}_\pi\left[\sum_a \pi(a|S_t, \boldsymbol{\theta})q_\pi(S_t, a)\frac{\nabla\pi(a|S_t, \boldsymbol{\theta})}{\pi(a|S_t, \boldsymbol{\theta})}\right]\\ &= \mathbb{E}_\pi\left[q_\pi(S_t, A_t)\frac{\nabla\pi(A_t|S_t, \boldsymbol{\theta})}{\pi(A_t|S_t, \boldsymbol{\theta})}\right]\\ &= \mathbb{E}_\pi\left[G_t\frac{\nabla\pi(A_t|S_t, \boldsymbol{\theta})}{\pi(A_t|S_t, \boldsymbol{\theta})}\right].\end{aligned}$$

  This would then give the update equation of

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha G_t\frac{\nabla\pi(A_t|S_t, \boldsymbol{\theta})}{\pi(A_t|S_t, \boldsymbol{\theta})} = \boldsymbol{\theta}_t + \alpha G_t\nabla\big(\ln\pi(A_t|S_t, \boldsymbol{\theta})\big).$$

  We note that the update is made in the direction in parameter space that most increases the probability of repeating the action A_t on future visits to state S_t. The update is proportional to the return, and inversely proportional to the action probability. This makes sense because we want to favour the actions that yield the highest return, but at the same time not give the frequently selected actions ad advantage.

- The REINFORCE algorithm is said to be an MC method since it uses the complete return, which will only be obtained at the end of the episode. Theoretically, this algorithm has good convergence property. However, the fact that it is an MC method means that it may have high variance and thus produce slow learning.

- REINFORCE with Baseline:
    - The policy gradient theorem can be generalised by including an arbitrary *baseline* $b(s)$:

      $$\nabla J(\boldsymbol{\theta}) \propto \sum_s \mu(s) \sum_a \Big(q_\pi(s,a) - b(s)\Big) \nabla \pi(a|s,\boldsymbol{\theta})$$

      where the baseline can be any function as it does not affect the equation. The update rule can then be written as

      $$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha\Big(G_t - b(S_t)\Big)\frac{\nabla \pi(A_t|S_t,\boldsymbol{\theta})}{\pi(A_t|S_t,\boldsymbol{\theta})}.$$

    - This baseline can significantly reduce the variance of the policy gradient theorem.
    - Note that the baseline function is dependent on state. A natural choice for the baseline would be the state value $\hat{v}(S_t, \mathbf{w}),\ \mathbf{w} \in \mathbb{R}^d$. We can incorporate the learning of the state-value weights into our REINFORCE algorithm as well.
- Actor-Critic Methods:
    - Actor-critic methods involves the second state of the transition, and uses the one-step return $G_{t:t+1}$. The one-step return is often superior to the actual return in terms of its variance and computational requirement.
    - When the state value function is used to assess actions in this way it is called a *critic*, and the overall policy gradient method is called an *actor-critic* method. The update can be written as

      $$\begin{aligned}\boldsymbol{\theta}_{t+1} &\doteq \boldsymbol{\theta}_t + \alpha\big(G_{t:t+1} - \hat{v}(S_t,\mathbf{w})\big)\frac{\nabla \pi(A_t|S_t,\boldsymbol{\theta})}{\pi(A_t|S_t,\boldsymbol{\theta})} \\ &= \boldsymbol{\theta}_t + \alpha\big(R_{t+1} + \gamma\hat{v}(S_{t+1},\mathbf{w}) - \hat{v}(S_t,\mathbf{w})\big)\frac{\nabla \pi(A_t|S_t,\boldsymbol{\theta})}{\pi(A_t|S_t,\boldsymbol{\theta})} \\ &= \boldsymbol{\theta}_t + \alpha\delta_t\frac{\nabla \pi(A_t|S_t,\boldsymbol{\theta})}{\pi(A_t|S_t,\boldsymbol{\theta})}.\end{aligned}$$

      It is natural to pair this with the semi-gradient TD(0). This method is fully online and incremental. The pseudocode can be found in page 332.

    - The eligibility trace version of this would use separate eligibility traces for the state-value and policy parameters. The pseudocode can be found in page 332.

- Policy Gradient for Continuing Problems:
  - In continuing problems, performance is measure in terms of the average reward per time step:

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

    Here, $\mu(s)$ is ergodic, where it is independent on the start state(s) and initial actions. Note that the value functions are defined with respect to the differential return.
  - Pseudocode can be found in page 333. Proof can be found in page 334.
- Policy Parameterisation for Continuous Actions:
  - Instead of computing probabilities for each action, we instead learn the probability distribution of the continuous actions.
  - We can let the policy be defined by a normal distribution, where the mean and standard deviation are given by parametric function approximators that depend on the state: $\mu : \mathcal{S} \times \mathbb{R}^{d'}$, $\sigma : \mathcal{S} \times \mathbb{R}^{d'} \to \mathbb{R}^+$. The policy is given as

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

  - The policy parameter vector can be written as $\boldsymbol{\theta} = [\boldsymbol{\theta}_\mu, \boldsymbol{\theta}_\sigma]^\intercal$, and the mean can be approximated as a linear function, and the standard deviation must always be positive so is better approximated as the exponential of a linear function:

$$
\mu(s, \boldsymbol{\theta}) \doteq \boldsymbol{\theta}_\mu^\intercal \mathbf{x}_\mu(s), \quad \sigma(s, \boldsymbol{\theta}) \doteq \exp\left(\boldsymbol{\theta}_\sigma^\intercal \mathbf{x}_\sigma(s)\right).
$$

Exercises

13.1 Not to sure what is meant by exact symbolic expression.

13.3 With linear action preferences, the policy is given by $\pi(a|s,\boldsymbol{\theta}) \doteq e^{\boldsymbol{\theta}^\top \mathbf{x}(s,a)} / \sum_b e^{\boldsymbol{\theta}^\top \mathbf{x}(s,b)}$.

$$\begin{aligned}
\nabla \ln \pi(a|s,\boldsymbol{\theta}) &= \nabla \ln \left(\frac{e^{\boldsymbol{\theta}^\top \mathbf{x}(s,a)}}{\sum_b e^{\boldsymbol{\theta}^\top \mathbf{x}(s,b)}} \right) \\
&= \nabla \ln e^{\boldsymbol{\theta}^\top \mathbf{x}(s,a)} - \nabla \ln \left(\sum_b e^{\boldsymbol{\theta}^\top \mathbf{x}(s,b)} \right) \\
&= \mathbf{x}(s,a) - \sum_b \frac{e^{\boldsymbol{\theta}^\top \mathbf{x}(s,b)}}{\sum_b e^{\boldsymbol{\theta}^\top \mathbf{x}(s,b)}} \mathbf{x}(s,b) \\
&= \mathbf{x}(s,a) - \sum_b \pi(b|s,\boldsymbol{\theta}) \mathbf{x}(s,b)
\end{aligned}$$

13.4

$$\begin{aligned}
\frac{\nabla \pi(A_t|S_t,\boldsymbol{\theta}_\mu)}{\pi(A_t|S_t,\boldsymbol{\theta})} &= \frac{1}{\sigma(s,\boldsymbol{\theta})\sqrt{2\pi}} \exp\left(\frac{-(a-\mu(s,\boldsymbol{\theta}))^2}{2\sigma(s,\boldsymbol{\theta})^2} \right) \frac{(a-\mu(s,\boldsymbol{\theta}))\mathbf{x}_\mu(s)}{\sigma(s,\boldsymbol{\theta})^2} \frac{1}{\pi(A_t|S_t,\boldsymbol{\theta})} \\
&= \frac{(a-\mu(s,\boldsymbol{\theta}))}{\sigma(s,\boldsymbol{\theta})^2} \mathbf{x}_\mu(s) \\
\frac{\nabla \pi(A_t|S_t,\boldsymbol{\theta}_\sigma)}{\pi(A_t|S_t,\boldsymbol{\theta})} &= \left(-\mathbf{x}_\sigma(s) + \frac{-(a-\mu(s,\boldsymbol{\theta}))^2}{2\sigma(s,\boldsymbol{\theta})^2} \left(-2\mathbf{x}_\sigma(s) \right) \right) \\
&= \left(\frac{(a-\mu(s,\boldsymbol{\theta}))^2}{\sigma(s,\boldsymbol{\theta})^2} - 1 \right) \mathbf{x}_\sigma(s)
\end{aligned}$$

13.5

a)

$$\begin{aligned}
\pi(1|S_t,\boldsymbol{\theta}_t) &= \frac{e^{h(s,1,\boldsymbol{\theta}_t)}}{e^{h(s,0,\boldsymbol{\theta}_t)} + e^{h(s,1,\boldsymbol{\theta}_t)}} = \frac{1}{e^{h(s,0,\boldsymbol{\theta}_t) - h(s,1,\boldsymbol{\theta}_t)} + 1} \\
&= \frac{1}{e^{\boldsymbol{\theta}_t^\top \mathbf{x}(S_t)} + 1}
\end{aligned}$$

b)

$$\begin{aligned}
\boldsymbol{\theta}_{t+1} &\doteq \boldsymbol{\theta}_t + \alpha G_t \frac{\nabla \pi(A_t|S_t,\boldsymbol{\theta})}{\pi(A_t|S_t,\boldsymbol{\theta})} \\
&= \boldsymbol{\theta} + \alpha G_t
\end{aligned}$$

c)

$$\begin{aligned}
&\text{let } P = \pi(1|s,\boldsymbol{\theta}) : \\
&\nabla \ln \pi(1|s,\boldsymbol{\theta}) = \nabla \ln P = \frac{\nabla P}{P} \\
&\qquad\qquad\qquad = \frac{\cancel{P}(1-P)}{\cancel{P}} \nabla\big(\boldsymbol{\theta}^\intercal \mathbf{x}(s)\big) \\
&\qquad\qquad\qquad = (1-P)\mathbf{x}(s) \\
&\qquad\qquad\qquad = (1-\pi(1|s,\boldsymbol{\theta}))\mathbf{x}(s)
\end{aligned}$$

$$\begin{aligned}
&\text{This means} \\
&\nabla \ln \pi(a|s,\boldsymbol{\theta}) = (1-\pi(a|s,\boldsymbol{\theta}))\mathbf{x}(s)
\end{aligned}$$