NOTES ON REINFORCEMENT LEARNING

1. Model-based learning

1.1. (Sequential) Markov Decision Process.

Definition 1.1. $\{(S_t, A_t, R_{t+1})\}_t$ random variables. States S_t are random variables on $\mathcal{S}$. ($\mathcal{S}^+$ is $\mathcal{S}$ with the terminal state.) Actions A_t are random variables on $\mathcal{A}$. Rewards R_t are random variables on $\mathcal{R}$.

Definition 1.2. The Markov property tells us that the conditional joint probability can be expressed using the distributions blow:

$$p_t(s', r|s, a) := \mathbf{P}(S_{t+1} = s', R_{t+1} = r|S_t = s, A_t = a)$$

Definition 1.3.

$$r_t(s, a) := \mathbb{E}(R_{t+1}|S_t = s, A_t = a) := \int_{\mathbb{R}} r \int_{\mathcal{S}} p_t(s', r|s, a) ds' dr$$

State transition probabilities:

$$p_t(s'|s, a) := \mathbf{P}(S_{t+1} = s', |S_t = s, A_t = a) = \int_{\mathcal{R}} p_t(s', r|s, a) dr$$

$$r_t(s, a, s') := \mathbb{E}(R_{t+1}|S_t = s, S_{t+1} = s', A_t = a) := \frac{\int_{\mathbb{R}} p_t(s', r|s, a) r dr}{p_t(s'|s, a)}$$

Remark 1.4. The subscript t might be omitted. We use the same $\mathbf{P}$ to indicate the probability measures on different space.

Remark 1.5. If the random variables are finite, then one can just replace the integrals in the formulas with summations.

Definition 1.6. The accumulated discounted reward is

$$G_t = \gamma^k R_{t+k+1},$$

where $\gamma \in [0, 1]$ is the discount.

Definition 1.7. Policy π is either a map $\mathcal{S} \to \mathcal{A}$, or a probability density function $\pi(a|s)$. The state value function for policy π is

$$v_{\pi,t}(s) := \mathbb{E}_\pi(G_t|S_t = s) = \mathbb{E}_\pi(R_{t+1}|S_t = s) + \gamma \sum_{k=0}^{\infty} \gamma^k \mathbb{E}_\pi(R_{t+k+2}|S_t = s)$$

where

$$\mathbb{E}_\pi(R_{t+1}|S_t = s) = \int_{\mathcal{S}} \int_{\mathcal{A}} p_t(s_1|s, a)\pi(a|s)\mathbb{E}(R_{t+1}|S_t = s, A_t = a, S_{t+1} = s_1) da ds_1$$

$$= \int_{\mathcal{A}} \int_{\mathcal{S}} p_t(s_1|s, a)\pi(a|s) r_t(s, a, s_1) ds_1 da = \int_{\mathcal{A}} \pi(a|s) r_t(s, a) da$$

and

$$\mathbb{E}_\pi(R_{t+k+2}|S_t = s) :=$$
$$\int_{\mathcal{S}^{k+2}} \int_{\mathcal{A}^{k+2}} \pi(a_0|s)p_t(s_1|s,a_0)\pi(a_1|s_1)p_{t+1}(s_2|s_1,a_1)...\pi(a_{k+1}|s_{k+1})p_{t+k+1}(s_{k+2}|s_{k+1},a_{k+1})$$
$$\mathbb{E}(R_{t+k+2}|S_t = s, A_t = a_0, S_{t+1} = s_1, A_{t+1} = a_1, ..., A_{t+k+1} = a_{k+1}, S_{t+k+2} = s_{k+2})da_0...da_{k+1}ds_1...ds_{k+2}$$
$$= \int_{\mathcal{S}^{k+2}} \int_{\mathcal{A}^{k+2}} \pi(a_0|s)p_t(s_1|s,a_0)\pi(a_1|s_1)p_{t+1}(s_2|s_1,a_1)...\pi(a_{k+1}|s_{k+1})p_{t+k+1}(s_{k+2}|s_{k+1},a_{k+1})$$
$$r_{t+k+1}(s_{k+1},a_{k+1},s_{k+2})da_0...da_{k+1}ds_1...ds_{k+2}$$

Theorem 1.8. *(Bellman Equation)*

$$v_{\pi,t}(s) = \int_{\mathcal{A}} \pi(a|s)(r_t(s,a) + \gamma \int_S p_t(s'|s,a)v_{\pi,t+1}(s')ds')da$$

Proof. Because

$$\mathbb{E}_\pi(R_{t+k+2}|S_t = s) =$$
$$\int_{\mathcal{S}} \int_{\mathcal{A}} \pi(a_0|s)p_t(s_1|s,a_0)(\int_{\mathcal{S}^{k+1}} \int_{\mathcal{A}^{k+1}} \pi(a_1|s_1)p_{t+1}(s_2|s_1,a_1)...\pi(a_{k+1}|s_{k+1})p_{t+k+1}(s_{k+2}|s_{k+1},a_{k+1})$$
$$r_{t+k+1}(s_{k+1},a_{k+1},s_{k+2})da_1...da_{k+1}ds_1...ds_{k+2})ds_1da_0$$
$$= \int_{\mathcal{S}} \int_{\mathcal{A}} \pi(a_0|s)p_t(s_1|s,a_0)\mathbb{E}_\pi(R_{t+k+2}|S_{t+1} = s_1)da_0ds_1$$

we have

$$v_{\pi,t}(s) = \int_{\mathcal{A}} \pi(a|s)r_t(s,a)da + \gamma \sum_{k=0}^{\infty} \gamma^k \int_{\mathcal{S}} \int_{\mathcal{A}} \pi(a_0|s)p_t(s_1|s,a_0)\mathbb{E}_\pi(R_{t+k+2}|S_{t+1} = s_1)da_0ds_1$$
$$= \int_{\mathcal{A}} \pi(a|s)(r_t(s,a) + \int_{\mathcal{S}} p_t(s_1|s,a)(\gamma \sum_{k=0}^{\infty} \gamma^k \mathbb{E}_\pi(R_{t+k+2}|S_{t+1} = s_1))ds_1)da$$
$$= \int_{\mathcal{A}} \pi(a|s)(r_t(s,a) + \gamma \int_{\mathcal{S}} p_t(s_1|s,a)v_{\pi,t+1}(s_1)ds_1)da$$

□

Definition 1.9. The Q-function is

$$q_{\pi,t}(s,a) := \mathbb{E}_\pi(G_t|S_t = s, A_t = a) =$$
$$\mathbb{E}(R_{t+1}|S_t = s, A_t = a) + \gamma \sum_{k=0}^{\infty} \gamma^k \mathbb{E}_\pi(R_{t+k+2}|S_t = s, A_t = a)$$

where

$\mathbb{E}_\pi(R_{t+k+2}|S_t = s, A_t = a) =$

$$\int_{\mathcal{S}^{k+2}} \int_{\mathcal{A}^{k+1}} p_t(s_1|s,a_0)\pi(a_1|s_1)p_{t+1}(s_2|s_1,a_1)...\pi(a_{k+1}|s_{k+1})p_{t+k+1}(s_{k+2}|s_{k+1},a_{k+1})$$

$\mathbb{E}(R_{t+k+2}|S_t = s, A_t = a_0, S_{t+1} = s_1, A_{t+1} = a_1, ..., A_{t+k+1} = a_{k+1}, S_{t+k+2} = s_{k+2})da_1...da_{k+1}ds_1...ds_{k+2}$

$$= \int_{\mathcal{S}^{k+2}} \int_{\mathcal{A}^{k+1}} p_t(s_1|s,a_0)\pi(a_1|s_1)p_{t+1}(s_2|s_1,a_1)...\pi(a_{k+1}|s_{k+1})p_{t+k+1}(s_{k+2}|s_{k+1},a_{k+1})$$

$r_{t+k+1}(s_{k+1}, a_{k+1}, s_{k+2})da_1...da_{k+1}ds_1...ds_{k+2}$

$$= \int_{\mathcal{S}} p_t(s_1|s,a_0)\mathbb{E}_\pi(R_{t+k+2}|S_{t+1} = s_1)ds_1$$

Lemma 1.10. *The relationship between the Q-function and the value function is that*

$$\int_{\mathcal{A}} \pi(a|s)q_{\pi,t}(s,a)da = v_{\pi,t}(s)$$

and

$$q_{\pi,t}(s,a) = r_t(s,a) + \gamma \int_S p_t(s'|s,a)v_{\pi,t+1}(s')ds'$$

Definition 1.11. The optimal state-value function is

$$v_{*,t}(s) := \sup_\pi(v_{\pi,t}(s))$$

The optimal action-value function is

$$q_{*,t}(s,a) = \sup_\pi(q_{\pi,t}(s,a))$$

Their relationship is

$$q_{*,t}(s,a) = r_t(s,a) + \gamma \int_S p_t(s'|s,a)v_{*,t+1}(s')ds'$$

Proposition 1.12. *(Bellman optimality equation)*

(1)

$$v_{*,t} = \sup_{a\in\mathcal{A}}(\int_{\mathbb{R}} \int_S p_t(s',r|s,a)(r + \gamma v_{\pi^*,t+1}(s'))ds'dr)$$

(2)

$$q_{*,t}(s) = \int_{\mathbb{R}} \int_S p_t(s',r|s,a)(r + \gamma \sup_{a'} q_{\pi^*,t+1}(s',a'))ds'dr$$

Proof. Let π^* be an optimal policy

$$v_{*,t}(s) = \sup_{a\in\mathcal{A}} q_{\pi^*,t}(s,a) = \sup_a(r_t(s,a) + \gamma \int_S p_t(s'|s,a)v_{\pi^*,t+1}(s')ds')$$

$$= \sup_a(\int_{\mathbb{R}} \int_S p_t(s',r|s,a)(r + \gamma v_{\pi^*,t+1}(s'))ds'dr)$$

$$q_{*,t}(s) = r_t(s,a) + \gamma \int_S p_t(s'|s,a) \sup_{a'} q_{\pi^*,t+1}(s',a')ds')$$

$$= \int_{\mathbb{R}} \int_S p_t(s',r|s,a)(r + \gamma \sup_{a'} q_{\pi^*,t+1}(s',a'))ds'dr$$

□

2. Dynamic programing

In reinforcement learning, value functions are used to organize and structure the search for good policies. The Bellman equation offers the following iterative algorithm to compute the value functions.

Algorithm 2.1. *(Iterative Policy Evaluation) Assume the action space and state space are discrete.*

Input π, the policy to be evaluated
Initialize an array $V(s) = 0$, for all $s \in \mathcal{S}^+$
repeat
  $\Delta \leftarrow 0$
  for *each $s \in \mathcal{S}^+$* ***do***
    $v \leftarrow V(s)$
    $V(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)(r + \gamma V(s'))$
    $\Delta \leftarrow \max(\Delta, \|v - V(s)\|)$
  end for
until $\Delta < \epsilon$ *(A small positive number) Ouput $V \approx v_\pi$*

The following proposition tells us how to improve a given deterministic policy using value function and Q-function.

Proposition 2.2. *(Policy Improvement) Assume the action space and state space are discrete. Let π be a deterministic policys. If $\pi'(s) = \text{argmax}_a(q_\pi(s, a))$ and $\pi'(s') = \pi(s)$ when $s' \neq s$, then $v_{\pi'}(s) \geq v_\pi(s)$.*

Proof.

$$\begin{aligned} v_{\pi,t}(s) &\leq q_{\pi,t}(s, \pi'(s)) \\ &= r_t(s, \pi'(s)) + \gamma \sum_{s'} p_t(s'|s, \pi'(s)) v_{\pi,t+1}(s') \\ &= r_t(s, \pi'(s)) + \gamma \sum_{s'} p_t(s'|s, \pi'(s)) q_{\pi,t+1}(s', \pi(s')) \\ &\leq r_t(s, \pi'(s)) + \gamma \sum_{s'} p_t(s'|s, \pi'(s)) v_{\pi',t+1}(s') \\ &= v_{\pi',t}(s) \end{aligned}$$

□

Combing value iteration and policy improvement we get our first reinforcement learning algorithm:

Algorithm 2.3. *(Policy Iteration) Assume the action space and state space are discrete. Let's consider deterministic policies.*

1. Initialization
Initialize an array $V(s) \in \mathbb{R}$, and $\pi(s) \in \mathcal{A}$ arbitrarily.
2. Policy Evaluation
repeat
  $\Delta \leftarrow 0$
  for *each $s \in \mathcal{S}^+$* ***do***
    $v \leftarrow V(s)$
    $V(s) \leftarrow \sum_{s',r} p(s', r|s, \pi(s))(r + \gamma V(s'))$

$\Delta \leftarrow \max(\Delta, \|v - V(s)\|)$
end for
until $\Delta < \epsilon$
3. Policy Improvement
policy_stable $\leftarrow True$
for *each* $s \in \mathcal{S}$ ***do***
$old\_action \leftarrow \pi(s)$
$\pi(s) \leftarrow \text{argmax}_a \left(\sum_{s',r} p(s', r|s, a)(r + \gamma V(s'))\right)$
if $old\_action \neq \pi(s)$ ***then***
policy_stable $\leftarrow False$
end if
end for
if *policy_stable* ***then***
stop and return $V \approx v_*$ *and* $\pi \approx \pi_*$
else
go to 2.
end if

Note that in each iteration of Policy Iteration, we need to wait for Policy Evaluation to converge. If we combine one iteration of Policy Evaluation and Policy Improvement, we get the following:

Algorithm 2.4. *(Value Iteration) Assume the action space and state space are discrete. Let's consider deterministic policies.*

Initialize an array $V(s) \in \mathbb{R}$.
repeat
$\Delta \leftarrow 0$
for *each* $s \in \mathcal{S}^+$ ***do***
$v \leftarrow V(s)$
$V(s) \leftarrow \max_a \sum_{s',r} p(s', r|s, a)(r + \gamma V(s'))$
$\Delta \leftarrow \max(\Delta, \|v - V(s)\|)$
end for
until $\Delta < \epsilon$
$\pi(s) = \text{argmax}_a \sum_{s',r} p(s', r|s, a)(r + \gamma V(s'))$
Output $\pi \approx \pi_*$

3. Monte Carlo Methods

4. Some probability theory

Recall that distribution function of X is $F_X(\alpha) = \mathbb{P}(X^{-1}(-\infty, \alpha))$. The density of a distribution function is f_X, for all $\alpha \in \mathbb{R}$,

$$F_X(\alpha) = \int_{-\infty}^{\alpha} f_X(x)dx.$$

Theorem 4.1. [1] *Theorem 1.3.61 (Change of Variables) Let* $X : \Omega \to \mathbb{R}$ *be a random variable on* $(\Omega, \mathcal{F}, \mathbf{P})$*, and* h *a Borel measurable function on such that* $\mathbb{E}(h_+(X)) < \infty$ *or* $\mathbb{E}(h_-(X)) < \infty$*. Then*

$$\int_\Omega h(X(w))d\mathbf{P}(w) = \int_{\mathbb{R}} h(x)d\mathcal{P}_X(x) = \int_{\mathbb{R}} h(x)f_X(x)dx.$$

Here $\mathcal{P}_X(B) = \mathbf{P}(X^{-1}(B))$*, for Borel measurable* $B \subseteq \mathbb{R}$*.*

Theorem 4.2. *(Fubini's theorem) Suppose* $\mu = \mu_1 \times \mu_2$ *is the product of* σ*-finite measures* μ_1 *on* $(X, \mathcal{F}_1)$ *and* μ_2 *on* $(X, \mathcal{F}_2)$*, and* $\mathcal{F} = \mathcal{F}_1 \times \mathcal{F}_2$ *is the* σ*-algebra on* $X \times Y$*. If* $h \in m\mathcal{F}$*, such that* $h \geq 0$ *or* $\int |h|\mu < \infty$*, then*

$$\begin{aligned}\int_{X\times Y} h d\mu &= \int_X (\int_Y h(x,y) d\mu_2(y)) d_{\mu_1}(x) \\ &= \int_Y (\int_X h(x,y) d\mu_1(x)) d_{\mu_2}(y)\end{aligned}$$

References

[1] Amir Dembo, Probability Theory: STAT310/MATH230; August 27, 2013
[2] Richard S. Sutton and Andrew G. Barto, Reinforcement Learning: An Introduction Draft2016