DISTRIBUTIONAL REINFORCEMENT LEARNING

SPRING 2024

Hyunin Lee
Ph.D. student
UC Berkeley
hyunin@berkeley.edu

Contents

1 Chapter 2

1.1 Random Variables and Their Probability Distributions

1.2 Markov Decision Processes

Definition 1.1 (Transition dynamics)**.** We define transition dynamics $\boldsymbol{P} : \mathcal{X} \times \mathcal{A} \to \mathscr{P}(\mathbb{R} \times \mathcal{X})$ that provides the joint probabiltiy distirbuiotn of R_t and X_{t+1} in ertns of state X_t and action A_t.

$$R_t, X_{t+1} \sim \boldsymbol{P}(\cdot, \cdot | X_t, A_t)$$

Definition 1.2 (Reward distribution)**.** $R_t \sim \boldsymbol{P}_{\mathcal{R}}(\cdot \mid X_t, A_t)$

Definition 1.3 (Transition kernel)**.** $X_{t+1} \sim \boldsymbol{P}_{\mathcal{X}}(\cdot \mid X_t, A_t)$

Definition 1.4 (Markov Decision Process (MDP))**.** MDP is a tuple $(\mathcal{X}, \mathcal{A}, \xi_0, \boldsymbol{P}_{\mathcal{X}}, \boldsymbol{P}_{\mathcal{R}})$

Definition 1.5 (Policy)**.** A policy is a maaping $\pi : \mathcal{X} \to \mathscr{P}(\mathcal{A})$ rom state to probabilty distributions over actions.

$$A_t \sim \pi(\cdot | X_t)$$

1.3 The Pinball Model

1.4 The Return

Definition 1.6 (Return G)**.** $G = \sum_{t=0}^{\infty} \gamma^t R_t$

The return is a sum of scaled, real-valued random variables and is therefore itself a random variable.

Assumption 1.7. For each state $x \in \mathcal{X}$ and action $a \in \mathcal{A}$, the reward distribution $\boldsymbol{P}_{\mathcal{R}}(\cdot \mid x, a)$ has finite first moment. This is if $R \sim \boldsymbol{P}_{\mathcal{R}}(\cdot \mid x, a)$, then

$$\mathbb{E}\left[|R|\right] < \infty.$$

Proposition 1.8. Under Assumption 1.7, the random return G exists and is finite with proabbility 1, in the sense that

$$\mathbb{P}_{\pi}\left(G \in (-\infty, \infty)\right) = 1.$$

1.5 Properties of the Random Trajectory

Definition 1.9 (Probablity distribution of random variable Z)**.** We denote $\mathcal{D}(Z)$ as the probability distribution of random variable Z. When Z is real-valued, then for $S \in \mathbb{R}$, we have

$$\mathcal{D}(Z)(S) = \mathbb{P}(Z \in S)$$

Also, we denote $\mathcal{D}_\pi(Z)$ as

$$\mathcal{D}_\pi(Z)(S) = \mathbb{P}_\pi(Z \in S)$$

1.6 The Random-Variable Bellman Equation

Definition 1.10 (Return-variable function)**.** $G^\pi = \sum_{t=0}^{\infty} \gamma^t R_t,\ X_0 = x$.

Formally, G^π is a collection of random variables indexed by an initial state x, each generated by a random trajectory $(X_t, A_t, R_t)_{t\geq 0}$ under the distribution $\boldsymbol{P}(\cdot|X_0 = x)$.

Proposition 1.11 (The random-variable Bellman equation)**.** Let G^π be the return-variable function of policy π. For a sample transition $(X = x, A, R, X')$, it holds that for any state $x \in \mathcal{X}$,

$$G^\pi(x) \stackrel{\mathcal{D}}{=} R + \gamma G^\pi(X')$$

1.7 From Random Variables to Probability Distributions

Recall the notation that for a real-valued cariable Z with probablity distribution $\nu \in \mathscr{P}(\mathbb{R})$, we define

$$\nu(S) = \mathbb{P}(Z \in S),\ S \subseteq \mathbb{R}.$$

In a same way, for each state $x \in \mathcal{X}$, let us denote the distribution of the random variable $G^\pi(x)$ by $\eta^\pi(x)$. Using this notation ,we have

$$\eta^\pi(x)(S) = \mathbb{P}(G^\pi(x) \in S),\ S \subseteq \mathbb{R}.$$

We call the collection of these per-state distribution the return-distirbuion function. Note that $\eta^\pi(x) \in \mathscr{P}(\mathbb{R})^{\mathcal{X}}$.

1.7.1 Mixing

Recall that for return-variable G^π and return-distribution function η^π, we have defined

$$\mathcal{D}_\pi(G^\pi(X')|X = x)(S) \stackrel{\text{def}}{=} \mathbb{P}_\pi(G^\pi(X') \in S|X = x).$$

Now, let's take a look at $\mathbb{P}_\pi$ term.

$$
\begin{aligned}
\mathcal{D}_\pi(G^\pi(X')|X=x)(S) &\overset{\text{def}}{=} \mathbb{P}_\pi(G^\pi(X') \in S|X=x) \\
&= \sum_{x' \in \mathcal{X}} \mathbb{P}_\pi(X'=x'|X=x)\mathbb{P}_\pi(G^\pi(X') \in S|X'=x', X=x) \\
&= \sum_{x' \in \mathcal{X}} \mathbb{P}_\pi(X'=x'|X=x)\mathbb{P}_\pi(G^\pi(x') \in S) \\
&= \left(\sum_{x' \in \mathcal{X}} \mathbb{P}_\pi(X'=x'|X=x)\eta^\pi(x') \right)(S)
\end{aligned}
$$

Therefore, we can conclude that

$$
\begin{aligned}
\mathcal{D}_\pi(G^\pi(X')|X=x)(S) &= \sum_{x' \in \mathcal{X}} \mathbb{P}_\pi(X'=x'|X=x)\eta^\pi(x') \\
&= \mathbb{E}_\pi\left[\eta^\pi(X') \mid X=x\right]
\end{aligned}
$$

The indexing step (S) also has a simple expression in terms of cumulative distribution functions as follows. Let $X = (\infty, z]$. Then we have

$$
\begin{aligned}
\mathbb{P}_\pi(G^\pi(X') \in S \mid X=x) &= P_\pi(G^\pi(X') \le z \mid X=x) \\
&= \sum_{x' \in \mathcal{X}} P_\pi(X'=x' \mid X=x)P_\pi(G^\pi(x') \le z \mid X=x) \\
&= \sum_{x' \in \mathcal{X}} P_\pi(X'=x' \mid X=x)P_\pi(G^\pi(x') \le z)
\end{aligned}
$$

Then if we let $F_{G^\pi(X')}(z)$ to be the c.d.f of random variable $G^\pi(X')$ up to$z$, we have

$$
F_{G^\pi(X')}(z) = \sum_{x' \in \mathcal{X}} P_\pi(X'=x' \mid X=x)F_{G^\pi(x')}(z)
$$

1.7.2 Scaling and translation

Suppose we konw the distribution of $G^\pi(X')$. Then what is the distribution of $R+\gamma G^\pi(X')$? This is an instance of a more general question: given a random variable$Z \sim \nu$ and a transformation $f : \mathbb{R}ß\mathbb{R}$, how should we express the distribution of $f(Z)$in terms of f and ν? Within this sense, we define *pushforward distrbution* as $f_\#\nu := \mathcal{D}(f(Z))$. Now, for $r \in \mathbb{R}$ and $\gamma \in [0, 1)$, we define bootstarp function $b_{r,\gamma}z \mapsto r + \gamma z$. Then we have

$$
(b_{r,\gamma})_\#\nu = \mathcal{D}(r + \gamma Z)
$$

where $Z \sim \nu$. Now, let's regard that $\nu = \eta^\pi(x')$ as a return distribution of state x' and we have correspoding random variable $G^\pi(x')$, i,e. $Z = G^\pi(x')$. Then, we have

$$
(b_{r,\gamma})_\#\eta^\pi(x') = \mathcal{D}(r + \gamma G^\pi(x')).
$$

Proposition 1.12 (The distributional Bellman equation)**.** Let η^π be the return-distribution function of policy π. Then, for any state $x \in \mathcal{X}$, we have

$$\eta^\pi(x) = \mathbb{E}_\pi \left[(b_{r,\gamma})_\# \eta^\pi(X') \mid X = x \right] \tag{1}$$

Just want to leave remark that $\mathbb{E}_\pi [g(X') \mid X = x] = \sum_{x' \in \mathcal{X}} \mathbb{P}_\pi(X' = x' \mid X = x) g(x')$ for any real-value function $g : \mathcal{X} \to \mathbb{R}$.

Proof. □

It is also possible to omit these random variables and write Equation (1) purely in terms of probability distributions, by making the expectation explicit:

$$\eta^\pi(x) = \sum_{a \in \mathcal{A}} \pi(a \mid x) \sum_{x' \in \mathcal{X}} \boldsymbol{P}(x' \mid x, a) \int_{\mathbb{R}} \boldsymbol{P}_{\mathbb{R}}(dr|x, a)(b_{r,\gamma})_\# \eta^\pi(x')$$

2 Chapter 3

2.1 The Monte Carlo Backup

Suppose we have K sample trajectories for state x and action a and reward r where each trajectory have total T_k steps as follows.

$$\{(x_{k,t}, a_{k,t}, x_{k,t})_{t=0}^{T_k - 1}\}_{k=1}^{K} \tag{2}$$

For now, assume that $T_k = T$ and $x_{k,0} = x_0$ for all k. We are interested in estimating the expected return

$$\mathbb{E}_\pi \left[\sum_{t=0}^{T-1} \gamma^t R_t \right] = V^\pi(x_0).$$

Monte Carlo methods estimate the expected return by averaging the outcomes of observed trajecoteries. Let us denote the sample reutnr for kth trajeoctyr as g_k which is defined as

$$g_k = \sum_{t=0}^{T-1} \gamma^t r_{k,t} \tag{3}$$

Then the sample-mean Monte Carlo estimate is the average of these K sample returns

$$\hat{V}^\pi(x_0) = \frac{1}{K} \sum_{k=1}^{K} g_k \tag{4}$$

2.2 Incremental Learning

Rather than after sample K samples, then compute all at once, it is much more useful to consider a learning model under which sample trajectories are processed sequentially. We call this algorihtm as *incremental algorithms*. Consdier an infinite sequence of sample trajectories

$$\{(x_{k,t}, a_{k,t}, x_{k,t})_{t=0}^{T_k - 1}\}_{k \geq 0} \tag{5}$$

suppose that initial states $\{(x_{k,0})_{k\geq 0}\}$ may be different. At kth stage, the agent is given a kth trajectory, and the algorihtm compues the sample return g_k (Equation (4)) which we called as *Monte Carlo target*. It then adjusts the value function of initial state $x_{k,0}$ toward this target (g_k) by the following *update rule*,

$$V(x_{k,0}) \leftarrow (1-\alpha_k)V(x_{k,0}) + \alpha_k g_k$$

where α_k is a time-varying step size.

Note that this *incremental Monte Carlo Update rule* only depends on the stating state and the sampel return pairs:

$$(x_k, g_k)_{k\geq 0} \tag{6}$$

We asume that the sample return g_k is assumed drawn from the return distribution $\eta^\pi(x_k)$. Then we have the following update rule

$$V(x_k) \leftarrow (1-\alpha_k)V(x_k) + \alpha_k g_k \tag{7}$$

This could be more expressed by

$$\begin{aligned} V_{k+1}(x_k) &= (1-\alpha_k)V_k(x_k) + \alpha_k g_k \\ V_{k+1}(x) &= V_k(x) \text{ for } x \neq x_k \end{aligned} \tag{8}$$

2.3 Temporal-Difference Learning

Incremental learning algorihtms are useful since they update for eveyr episode. Tempoarl-differnet learning (TD learning) is more fine-grained update version. It learn from sample transitions, rather than entire trajectories.

Let us consdier a seuqen of smpale ransitions drwn independently as follows

$$(x_k, a_k, r_k, x'_k)_{k\geq 0} \tag{9}$$

As with the incremental Monte Carlo algoithm, the update rule of temporal differnece learning is

$$V(x_k) \leftarrow (1-\alpha_k)V(x_k) + \alpha_k(r_k + \gamma V(x'_k)) \tag{10}$$

We call the term $r_k + \gamma V(x'_k)$ as the *temporal-difference target*, and by arrangin the term , we call the term $r_k + \gamma V(x'_k) - V(x_k)$ as the *temproal-differnec error* as

$$V(x_k) \leftarrow V(x_k)\alpha_k(r_k + \gamma V(x'_k) - V(x'_k)).$$

Incremental Monte Carlo algorithm updates its value function estimate toward a fixed target g_k, but in TD learning we don't have such fixed target. Temporal-difference learning instead depends on the value function at the next state $V(x'_k)$being approximately correct. As such, it is said to *bootstrap* from its own value function estimate.

2.4 From Values to Probabilities

We are highly interested in how we can learn the return-distribution function η^π. Let's first take a scenario for binary reward, i.e. $R_t \in \{0,1\}$ and we are intesreind in distribution of undiscounted finite-horizon return function

$$G^{\pi}(x) = \sum_{t=0}^{H-1} R_t,\ X_0 = x. \tag{11}$$

Since the $G^{\pi}(x)$ takes an integer value between 0 to H, these form the support of the probability distribution $\eta^{\pi}(x)$. To learn $\eta^{\pi}(x)$, we assigns a probability $p_i(x) \geq 0$ where $\sum_{i=0}^{H} p_i(x) = 1$ as

$$\eta(x) = \sum_{i=0}^{H} p_i(x)\delta_i \tag{12}$$

We call this equation *categortical representation.* It's kind of classification problem for given state x. Now, let us consider the problem that we have a state-return pairs $(x_k, g_k)_{k\geq 0}$ where each g_k is drawn from the distribution $\eta^{\pi}(x_k)$. Now, we have *categorical update rule* as

$$\begin{aligned} p_{g_k}(x_k) &\leftarrow (1-\alpha_k)p_{g_k}(x_k) + \alpha_k \\ p_i(x_k) &\leftarrow (1-\alpha_k)p_i(x_k) \text{ for } i \neq g_k \end{aligned} \tag{13}$$

Combining equations (12) and (13) provide the following equation

$$\eta(x_k) \leftarrow (1-\alpha_k)\eta(x_k) + \alpha_k\delta_{g_k} \tag{14}$$

We call Equation (14) as *undiscounted finite-horizon categorical Monte Carlo algorithm.*

2.5 The Projection Step

For H steps binary rewards ($N_{\mathcal{R}} = 2$), the number of possible returns is $N_G = H + 1$. However, what if $N_{\mathcal{R}} > 2$ or if we have discounted factor γ? Noe that whwen γ is introduced, then N_G grows exponentially on H.

To handle this large set of possible returns, we inset a *projection step* prior to the mixture update on Equation (14). We will consider return distributions that assign probability mass to $m \geq 2$ evenly spaced values or locations $\theta_1 \leq \theta_2 \leq \cdots \leq \theta_m$ where the gap $\zeta_m := \theta_{i+1} = \theta_i$ is identical. A common design is take $\theta_1 = V_{\min}, \theta_m = V_{\max}$ and set

$$\vartheta_m = \frac{V_{\max} - V_{\min}}{m-1}$$

which is just identical gap. We express the corresponding return distribution $\eta(x)$ as weighted sum of Dirac deltas as follows.

$$\eta(x) = \sum_{i=1}^{m} p_i(x)\delta_{\theta_i}$$

Now, consider a sample return $g \sim \eta(x)$ and we denote the g falls between θ_{i^*} and θ_{i^*+1} which could be defined as $i^* = \arg\max_{i\in\{0,\cdots,m\}}\{\theta_i : \theta_i \leq g\}$. We write

$$\Pi_-(g) = \theta_{i^*},\ \Pi_+(g) = \theta_{i^*+1}.$$

Then define $\zeta(g)$ term corresponds to the distance of g to the two closest elements of the support, scaled to lie in the interval $[0, 1]$ as

$$\zeta(g) = \frac{g - \Pi_-(g)}{\Pi_+(g) - \Pi_-(g)}.$$

Then, we define *stocastic projection* of g as

$$\Pi_\pm(g) = \begin{cases} \Pi_-(g) \text{ with probability } 1 - \zeta(g) \\ \Pi_+(g) \text{ with probability } \zeta(g) \end{cases}$$

Use this projection to construct the update rule as

$$\eta(x) \leftarrow (1 - \alpha)\eta(x) + \alpha\delta_{\Pi_\pm(g)}$$

which is similar to Equation (14). We could also write as

$$\begin{aligned} p_{i^\pm}(x) &\leftarrow (1 - \alpha)p_{i^\pm}(x) + \alpha \\ p_i(x) &\leftarrow (1 - \alpha)p_i(x) \text{ for } i \neq i^\pm \end{aligned}$$

where $i^\pm$ is the index of location $\Pi_\pm g$. Note that the stochastic projection could be improved by putting both $\Pi_-(g)$ and $\Pi_+(g)$ information. We define *deterministic projection* as

$$\eta(x) \leftarrow (1 - \alpha)\eta(x) + \alpha\left[(1 - \zeta(g))\delta_{\Pi_-(g)} + \zeta(g)\delta_{\Pi_+(g)}\right] \tag{15}$$

Within this sense, we deinfe projection operator Π_c that applies to the distribution δ_g as

$$\Pi_c\delta_g = (1 - \zeta(g))\delta_{\Pi_-(g)} + \zeta(g)\delta_{\Pi_+(g)}$$

We call this method the *categorical Monte Carlo algorithm.*

Under the right condition, Equation (15) is correlated with a return distribution $\hat{\eta}^\pi(x)$ where we have $\hat{\eta}^\pi(x) = \mathbb{E}\left[\Pi_c\delta_{G^\pi(x)}\right]$. In fact, we may write as

$$\mathbb{E}\left[\Pi_c\delta_{G^\pi(x)}\right] = \Pi_c\eta^\pi(x)$$

where $\Pi_c\eta^\pi(x)$ is a distribution supported on $\{\theta_1, \cdots, \theta_m\}$ produced by projecting all possible outcomes under distribution $\eta^\pi(x)$.

2.6 Categorical Temporal-Difference Learning

What TD learning do is

- learn from sample transition rather than full trajectory
- It learns by bootstrapping from its current return function estimates.

Suppse we have a transition data (x, a, r, x'). CTD maintains a return fiction estaimte $\eta(x)$ supported on evenly spaced locations $\{\theta_1, \cdots, \theta_m\}$. Let the return distribution of x' as

$$\eta(x') = \sum_{i=1}^{m} p_i(x')\delta_{\theta_i}$$

then the intermediate target is

$$\tilde{\eta}(x) = \sum_{i=1}^{m} p_i(x')\delta_{r+\gamma\theta_i}$$

which can also be expressed in terms of a pushforward distribution as

$$\tilde{\eta}(x) = (b_{r,\gamma})_{\#}\eta(x') \tag{16}$$

Note that each particles of $\eta(x')$ are supports of $\{\theta_1, \cdots, \theta_m\}$, but pushing forward those particles actually does not makes liying in the support of the original distribution. This motivates the use of projection step. Then, we have

$$\begin{aligned}
\Pi_c\tilde{\eta}(x) &= \Pi_c \sum_{j=1}^{m} p_j(x')\delta_{r+\gamma\theta_i} \\
&= \sum_{j=1}^{m} p_j(x')\Pi_c\delta_{r+\gamma\theta_i} \\
&= \sum_{j=1}^{m} p_j(x') \left[(1-\zeta(\tilde{\theta}_j))\delta_{\Pi_-(\tilde{\theta}_j)} + \zeta(\tilde{\theta}_j)\delta_{\Pi_+(\tilde{\theta}_j)}\right] \\
&= \sum_{i=1}^{m} \delta_{\theta_i} \left(\sum_{j=1}^{m} p_j(x')\zeta_{i,j}(r)\right)
\end{aligned}$$

where $\zeta_{i,j}(r) = (1-\zeta(\tilde{\theta}_j))\mathbf{1}_{\{\Pi_-(\tilde{\theta}_j)=\theta_j\}} + \zeta(\tilde{\theta}_j)\mathbf{1}_{\{\Pi_+(\tilde{\theta}_j)=\theta_j\}}$. Also, the last line highlights that the CTD target lies on a support of $\{\theta_1, \cdots, \theta_m\}$. Note that the assignment is obtained by weighting the next-state probabilities $p_j(x')$ by the coefficients $\zeta_{i,j}(r)$. Using the projected intermediate target, i.e. $\Pi_c\tilde{\eta}(x)$, we have the following CTD update rule:

$$\begin{aligned}
\eta(x) &\leftarrow (1-\alpha)\eta(x) + \alpha(\Pi_c\tilde{\eta}(x)) \\
&\leftarrow (1-\alpha)\eta(x) + \alpha(\Pi_c(b_{r,\gamma}\eta(x')))
\end{aligned} \tag{17}$$

Now, note that $\eta(x)$ and $\eta(x')$ are the categorical distribution which is a mixture of dirac-delta function, we have the following update rule:

$$p_i(x) \leftarrow (1-\alpha)p_i(x) + \alpha\sum_{j=1}^{m} \zeta_{i,j}(r)p_j(x') \tag{18}$$

With this form, we see that the CTD update rule adjusts each probability $p_i(x)$ of the return distribution at state x toward a mixture of the probabilities of the return distribution at the next state.