IFT 6085 - Lecture 19
Basic results on reinforcement learning

This version of the notes has not yet been thoroughly checked. Please report any bugs to the scribes or instructor.

Scribes
Winter 2019: Eugene Vorontsov & Charles Guille-Escuret
Winter 2020: Emre Onur Kahya

Instructor: Ioannis Mitliagkas

1 Summary

Reinforcement learning is concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward. An optimal *policy* defines actions by which an agent achieves its goal of maximizing its *rewards* in this environment. This lecture covers:

- Value iteration (compute states)
- Policy evaluation (expected return)
- Policy optimization (maximize expected return)

2 Basic definitions and assumptions

In this lecture, we consider state transitions and subsequent rewards induced by actions to have a Markov property: given an action in a state, the reward and the new state are independent of other states and actions. We assume the agent can be found in a finite set of states $\mathcal{S}$ with access to a set of actions $\mathcal{A}$. Furthermore, we assume that rewards are bounded: $\exists M \in \mathbb{R}$ such that $\forall s, s' \in \mathcal{S}, \forall a \in \mathcal{A}, \forall r > M, P(r, s'|s, a) = 0$.

Definition 1 (Markov Decision Process)**.** *The probability of transitioning from state $s \in \mathcal{S}$ to the new state $s' \in \mathcal{S}$ with a reward $r \in \mathbb{R}^+$ given an action $a \in \mathcal{A}$ is yielded by the distribution:*

$$P(r, s'|s, a)$$

This transition probability governs Markov Decision Process and this distribution models our interaction with the environment. In case only part of the state is observed, we have a *partially observed MDP* (POMDP).

Definition 2 (Partially Observed Markov Decision Process)**.** *Observe some y from s:*

$$y \sim P(y|s)$$

Observation : To work with an POMDP we can convert it to an MDP by assembling states from cumulative partial observations: $\tilde{s}_0 = \{y_0\}$, $\tilde{s}_1 = \{y_0, y_1\}$, $\tilde{s}_2 = \{y_0, y_1, y_2\}$, etc...

As an agent traces a trajectory of states, its goal is to find a policy for taking actions that maximize the reward along this trajectory.

Definition 3 (Policy)**.** *Actions $a \in \mathcal{A}$ are defined for a state $s \in \mathcal{S}$ with a policy:*

$$\pi(a|s)$$

States a are sampled according to the *probabilistic* policy $\pi(a|s)$. A policy is *deterministic* if $a = \pi(s)$.

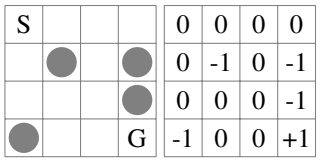

Figure 1: **Example** where an agent traverses a frozen lake from a starting point (S) to a goal (G). Moving along the grid has a 50% chance of moving in a move in a random direction. The lake grid is shown on the left and grid rewards are on the right. The agent is rewarded 1 for reaching the goal, -1 for falling in a hole (gray circles, left), and 0 elsewhere. Since an agent dies in a hole, this state is a *sink state* in that it no longer changes. One *episode* traces a trajectory from the start to the goal or a sink state.

3 Value Iteration

A dynamic programming approach called value iteration computes an optimal MDP policy and its value. This requires the computation of an expected return. Depending on the task, the expectation can be computed by dynamic programming over a finite horizon (task ends) or an infinite horizon (task continues).

3.1 Finite Horizon

In the case of a finite horizon, the trajectory reaches the goal or sink state in a finite number of steps.

Definition 4 (State Value Function)**.**

$$V_t^\pi(s) = \mathbb{E}\left[\sum_{\tau=0}^{t} r_\tau | s_0 = s\right]$$

Here $V_t^\pi(s)$ quantifies the expected return of a given a policy $\pi(a|s)$ starting from state s and applying it for t steps, where r_τ are the reward variables for every step from 0 to t. Particularly, if we start with state s and take a as our very first action we can give the following definition:

Definition 5 (State-Action Value Function)**.**

$$Q_t^\pi(s,a) = \mathbb{E}\left[\sum_{\tau=0}^{t} r_\tau | s_0 = s, a_0 = a\right]$$

Definition 6 (Expected Return)**.** *For a trajectory with T steps and reward r_i at each step $i \in [1, ..., T]$:*

$$\mathbb{E}\left[\sum_{i=0}^{T-1} r_i + V_T(s_T)\right],$$

To solve for the optimal policy and its value, we consider the state value function with the finite horizon expected return:

$$V(s_0) = \max_{\pi_0} \max_{\pi_1} ... \max_{\pi_{T-1}} \mathbb{E}[r_0 + r_1 + ... + r_{T-1} + V_T(s_T)]$$

Since for any i, r_i is independent of $\pi_{i+1}, \pi_{i+2}, ..., \pi_{T-1}$ (the reward does not depend of the decision we take in the future), we can successively take the max out of the expectation and obtain:

$$V(s_0) = \max_{\pi_0} \mathbb{E}[r_0 + \max_{\pi_1} \mathbb{E}[r_1 + ... + \max_{\pi_{T-1}} \mathbb{E}[r_{T-1} + V_T(s_T)]]]$$

Here notice that each one of these expectations is an expectation for a specific randomness. As an example the above series the last r_{T-1} is a random variable and it is the reward that we get for the state that we end up in after the action dictated by policy π_{T-1}. Therefore this expectation is with respect to randomness only that will appear starting at step

$T-1$ in future. So the first expectation $\max_{\pi_0} \mathbb{E}[r_0 + ...$ is with respect to all randomness for the problem. Using Fubini–Tonelli theorem we were able to move the integrals that appear in multiple expectations. To be able to do that we assumed a regularity condition, i.e. our random variable rewards live in a compact space.
We can then solve by applying for i following $\{T, ..., 1\}$:

$$\begin{aligned}
&\forall s \in \mathcal{S} \\
&\pi_{i-1}(s) = \underset{a}{\text{argmax}}(\mathbb{E}_{s_i}[r_{i-1} + V_i(s_i)] \\
&V_{i-1}(s) = \max_a(\mathbb{E}_{s_i}[r_{i-1} + V_i(s_i)]
\end{aligned}$$

Algorithm 1 Finite Horizon Value Iteration

for $t = T-1, T-2, .., 0$ **do**
  for $s \in \mathcal{S}$ **do**
    $\pi_t(s), V_t(s) = \underset{a}{\text{maximize}}(\mathbb{E}[r_t + V_{t+1}(s_{t+1})])$
  end
end

3.2 Infinite Horizon

For some games we do not have a predetermined end time and the game can keep going forever. Therefore finite horizon algorithms do not apply to these cases. Also in real life problems one usually has the following situation: it is better to find a good solution soon then to find a great solution much later. For these two reasons one would need to have the case of an infinite horizon where future rewards are increasingly discounted (reward is bounded).

Definition 7 (State Value Function)**.**

$$V_t^\pi(s) = \mathbb{E}\left[\sum_{\tau=0}^{t} \gamma^\tau r_\tau | s_0 = s\right]$$

Here the γ is called the discount factor and its value ($0 < \gamma < 1$) determines how much we care about future rewards.

Definition 8 (State-Action Value Function)**.**

$$Q_t^\pi(s, a) = \mathbb{E}\left[\sum_{\tau=0}^{t} \gamma^\tau r_\tau | s_0 = s, a_0 = a\right]$$

Definition 9 (Expected Return)**.** *For a trajectory with T steps and reward r_i at each step $i \in [1, ..., T]$:*

$$\mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i r_i\right],$$

Observation : We can interpret the rewards in a discounted setting as the non-discounted rewards from an MDP, by adding a sink state $\tilde{s}$ that traps the agent indefinitely with reward 0, and using the transition:

$$\tilde{P}(s'|s, a) = \begin{cases} P(s'|s, a) & \text{with probability } \gamma \\ \tilde{s} & \text{with probability } 1 - \gamma \end{cases}$$

Algorithm 2 Infinite Horizon Value Iteration

Initialize $V^{(0)}$ arbitrarily
for $n = 0, 1, 2, ...$ *until termination condition* **do**
  for $s \in \mathcal{S}$ **do**
    $\pi^{(n+1)}(s), V^{(n+1)}(s) = \underset{a}{\text{maximize}}(\mathbb{E}_{s' \sim P(s'|s,a)}[r + \gamma V^n(s')])$
  end
end

There is a fundamental difference between infinite horizon and the finite horizon cases. When we have the finite horizon version of the problem we impose the boundary conditions. Then it makes a big difference if we are near the beginning of the game or end of the game. Therefore our optimal policy changes from each time step. But for the infinite horizon version of the problem, there is no end point and no sense of absolute time. Hence an optimal policy is always optimal no matter how far we are in the game.

Theorem 10. *By assuming a horizon $T \in \mathbb{N}$, an infinite horizon value iteration problem can be converted into a finite horizon one with error bounded by:*

$$\epsilon \leq r_{max} \frac{\gamma^T}{(1-\gamma)}$$

Proof.

$$\begin{aligned}
\sum_{t=T}^{\inf} \gamma r_t &\leq \sum_{t=T}^{\inf} \gamma r_{max} \\
&= \sum_{t=0}^{\inf} \gamma r_{max} - \sum_{t=0}^{T} \gamma r_{max} \\
&= r_{max} \left(\frac{1}{1-\gamma} - \frac{1-\gamma^T}{1-\gamma} \right) && \text{Geometric series.} \\
&= r_{max} \frac{\gamma^T}{1-\gamma}
\end{aligned}$$

□

Since we are dealing with discounted rewards where $\gamma < 1$, we can interpret a value iteration update as application of an operator that has a fixed point to which iteration converges at the limit.

Definition 11 (Backup Operator)**.**

$$\mathcal{T} : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$$

$$[\mathcal{T}V](s) = \max_a \mathbb{E}_{s',r|s,a}[r + \gamma V(s')]$$

Using the above definition, we will be able to express one iteration of the infinite horizon algorithm as an operator acting on a state value function. With the help of operator algebra, one can also show the following: If we have two state value functions V and W and we apply the same T operator to both of them. Then the new state value functions $\mathcal{T}V$ and $\mathcal{T}W$ are closer compared to V and W.

Theorem 12. *Backup operator $\mathcal{T}$ is a contraction with modulus γ under ∞-norm*

$$||\mathcal{T}V - \mathcal{T}W||_\infty \leq \gamma ||V - W||_\infty$$

Finally we can show that repeated application of the operator T will always converge state value function.

Theorem 13. *The backup operator $\mathcal{T}$ has a fixed point V^* and :*

$$(\mathcal{T}^i V) \underset{i\to\infty}{\to} V^*$$

Proof : This is a direct consequence of Theorem 12 and Banach's Fixed Point theorem.

We can then rewrite Algorithm 2 in a cleaner manner:

Algorithm 3 Infinite Horizon Value Iteration with operator

Initialize $V^{(0)}$ arbitrarily
for $n = 0, 1, 2, ...$ *until termination condition* **do**
| $V^{(n+1)} = \mathcal{T}V^{(n)}$
end

4 Policy Evaluation and Iteration

While value iteration allows us to evaluate the return from states according to the optimal policy, policy iteration seeks to update each policy to increase the expected return. We again use a backup operator but this time without maximizing over actions.

Definition 14 (Backup Policy Operator)**.**

$$\mathcal{T}^\pi : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$$

$$[\mathcal{T}^\pi V](s) = \mathbb{E}_{s'|s,a\sim\pi(s)}[r + \gamma V(s')]$$

Instead, we evaluate the expected return for every state, induced by each policy, and then select the actions that maximize the expected return for that policy. We obtain the state valuation induced by π by using $V^* = \mathcal{T}^\pi V^*$. This induces a linear equation that can be solved exactly for *policy evaluation*:

$$V(s) = \sum_{s'} P(s'|s, a \sim \pi(s))[r(s, a, s') + \gamma V(s')]$$

Policy iteration is then performed by alternating between policy evaluation for each policy π and a greedy update of the policies actions:

Algorithm 4 Policy Iteration

Initialize $\pi^{(0)}$ arbitrarily
for $n = 1, 2, ...$ **do**
| $V^{\pi^{(n-1)}} = Solve[V = \mathcal{T}^{\pi^{(n-1)}} V]$ $\pi^{(n)} = \underset{a}{\text{argmax}}(\mathbb{E}_{s'|s,a}[r + V^{\pi^{(n-1)}}(s')])$
end

Note that for a finite MDP, Algorithm 4 should converge in a finite number of iterations since the number of policies is finite [2]. The optimization of the expected reward can be seen as solving the bellman's equation for an utility function V, that is described below. We have that for an infinite-horizon decision problem, the value function $V(s_0)$ reaches equilibrium when the Bellman's equation is satisfied.

Definition 15 (Bellman's equation)**.**

$$V(s_0) = \max_{a_{t=0}^\infty} \sum_{t=0}^{\infty} \gamma^t F(s_t, a_t)$$

Where $F(a_t, s_t)$ defines the expected reward of a state-action pair, submitted to constraints $s_{t+1} \sim \pi(s_t, a_t)$ and discount factor $0 < \gamma < 1$.

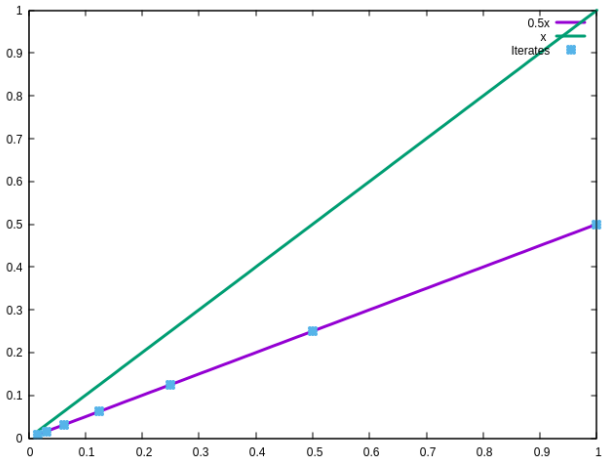

Figure 2: Iterates of the simplest contraction mapping $f(x) = \alpha x$ with $\alpha \leq 1$ (here, $\alpha = 0.5$) at starting point $x_0 = 1$ tend toward the fixed point $f(x) = x$ at $(0, 0)$. Green: the line $f(x) = x$. Blue: the contraction mapping $f(x) = 0.5x$, respecting $d(0.5x, 0.5y) \leq 0.5d(x, y)$ with $d(x, y) = |y - x|$. Points: iterates of $f(x)$ with $x_0 = 1$.

Note that V is a function of the optimal sequence of actions $a_{t=0}^{\infty}$ **as well as** the optimal/maximal value of the reward allowed by the environment. We will see that this is equivalent to solving a function with a singular fixed point. Additionally, this equation is an algorithm in itself to compute the reward and can be easily translated into code.

In the next sections, we will analyze what happens when the whole distribution of rewards is considered rather than just the expected reward, a setting called distributional RL.

5 Banach Fixed Point Theorem

In this section we discuss a central theorem in the analysis of RL algorithms: the Banach fixed point theorem[1].
The Banach fixed point theorem gives convergence guarantees to a unique fixed point under iterated contraction mappings. Thus, if the update equation of an RL algorithm can be shown to be a contraction mapping, it will eventually converge. Moreover, regardless of initial values, convergence leads to the same fixed point (per uniqueness).

Definition 16 (Contraction Mapping)**.** *Let (X, d) be a metric space with metric d on space X. Then a function $T : X \to X$ is a contraction mapping on X if there exists $q \in [0, 1)$ such that*

$$\forall x, y \in X : d(T(x), T(y)) \leq qd(x, y)$$

Definition 16 is reminiscent of the definition of L-Lipschitz functions. Indeed, a contraction mapping is simply an L-Lipschitz function (where the image is a subset of the domain) for some $L \in [0, 1)$. That is, beyond restricting the maximum growth rate of the function, we also have that the function must grow arbitrarily slower as the function is repeatedly applied to its images. Figure 2 illustrates iteration of the contraction mapping $f(x) = 0.5x$.

Theorem 17 (Banach Fixed Point Theorem)**.** *Let (X, d) be a non-empty complete metric space with a contraction mapping $T : X \to X$, then T admits a unique fixed point $x^* \in X$. Moreover, for any $x_0 \in X$, $T^n(x_0) \to x^*$ as $n \to \infty$*

Proof. Let $x_0 \in X$ be an arbitrary point and let $\{x_n\}$ be the sequence of iterates such that $\forall n \in \mathcal{N}^+, x_n = T(x_{n-1})$. Then we have

$$\begin{aligned} d(x_{n+1}, x_n) &\leq qd(x_n, x_{n-1}) \\ &\leq \dots \\ &\leq q^n d(x_1, x_0) \end{aligned}$$

[1]The derivations and proofs are mostly taken from Wikipedia [3]

Let $m, n \in \mathcal{N}^+$ such that $m > n$, then, because (X, d) is a metric space and T is a contraction mapping on X,

$$\begin{aligned} d(x_m, x_n) &\leq d(x_m, x_{m-1}) + \ldots + d(x_{n+1}, x_n) \\ &\leq q^{m-1} d(x_1, x_0) + \ldots + q^n d(x_1, x_0) \\ &= q^n d(x_1, x_0) \sum_{k=0}^{m-n-1} q^k \\ &\leq q^n d(x_1, x_0) \sum_{k=0}^{\infty} q^k \\ &= \frac{q^n}{1-q} d(x_1, x_0) \end{aligned}$$

For some arbitrary ϵ, since $q \in [0, 1)$, there is some N such that

$$q^N < \frac{\epsilon(1-q)}{d(x_1, x_0)}$$

Choosing $m, n > N$ gives

$$d(x_m, x_n) \leq \frac{q^n}{1-q} d(x_1, x_0) < \frac{\epsilon(1-q) d(x_1, x_0)}{(1-q) d(x_1, x_0)} = \epsilon$$

Thus, $\{x_n\}$ is a Cauchy sequence with a limit $x^* \in X$, and it is a fixed point of T because

$$\begin{aligned} x^* &= \lim_{n \to \infty} x_n \\ &= \lim_{n \to \infty} T(x_{n-1}) \\ &= T(\lim_{n \to \infty} x_{n-1}) && \text{Valid because } T \text{ is continuous} \\ &= T(x^*) \end{aligned}$$

Finally, it can be shown by contradiction that the fixed point is unique. Suppose there exists $p_1, p_2 \in X$ such that $p_1 \neq p_2$ and p_1 and p_2 are fixed points of T. Then

$$\begin{aligned} d(T(p_1), T(p_2)) = d(p_1, p_2) &\leq q d(p_1, p_2) \\ &\iff q = 0 && q \in [0, 1) \\ &\implies d(p_1, p_2) \leq 0 \end{aligned}$$

Since $d(p_1, p_2) \geq 0$ by definition, this implies $d(p_1, p_2) = 0 \implies p_1 = p_2$, which is a contradiction. □

Theorem 17 tells us that given any point $x_0 \in X$, repeated application of the contraction mapping not only leads to convergence, but also converges to the unique fixed point $x^* \in X$. This will be used in the next section to demonstrate convergence of distributional RL.

6 Distributional RL

Distributional Reinforcement Learning was first introduced in Bellemare et al. [1]. We will discuss it in the context of infinite horizon. First we define The following are all the random variables involved in the MDP:

1. $r_t, s_t \sim \mathrm{P}(\cdot, \cdot \,|s_{t-1}, a_{t-1})$ (non-deterministic reward and transition)
2. $a_t \sim \pi(\cdot \,|s_{t-1})$ (non-deterministic policy)

Using these variables, we can define what is known as the return:

Definition 18 (Return). *Let $(s_t, a_t)_{t=1}^{\infty}$ be the state-action pairs of an infinite horizon MDP with a discount factor $0 < \gamma \leq 1$. The **return** is defined as*

$$\Phi^{\pi}(s, a) = r_0 + \sum_{t=1}^{\infty} \gamma^t r_t \quad \text{with } s_0 = s \text{ and } a_0 = a \tag{1}$$

which is a function that takes an initial state and action pair as input and output a random variable on the initial state and action.

As return, $\Phi^{\pi}(s, a)$, is a random variable, it has a law, or a distribution. Let $\mathcal{D}$ denote **the space of all distributions of return**. We define the value distribution as a mapping from the state-action space to the space of distributions of return below.

Definition 19 (Value Distribution). *Then we define the value distribution as*

$$Z^{\pi} : \mathcal{S} \times \mathcal{A} \to \mathcal{D}$$

such that

$$\Phi^{\pi}(s, a) \sim Z^{\pi}(s, a)$$

$\mathcal{D}^{|\mathcal{S}| \times |\mathcal{A}|}$. *Since $Z^{\pi}(a, s)$ is a distribution, we can write in the discrete case*

$$P(\Phi = \phi) = Z^{\pi}(a, s)(\phi)$$

It is important to understand that for given a and s, $\Phi^{\pi}(s, a)$ is a random variable and $Z^{\pi}(s, a)$ is the distribution of that random variable. The return and value distributions are linked to the state-action value function described last lecture via the formula

$$Q^{\pi}(s, a) := \mathbb{E}_{Z^{\pi}(s,a)}[\Phi(s, a)] \tag{2}$$

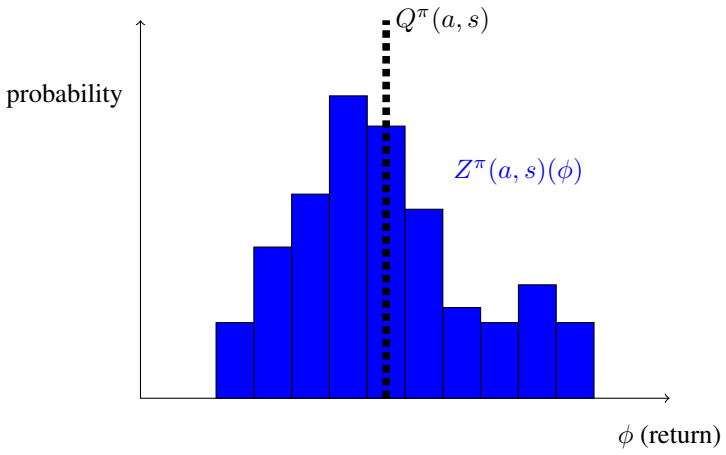

Figure 3: example of distribution $Z^{\pi}(a, s)(\phi)$

But why would we want to use the distribution instead of its expectation. The problem with expectation is that we lose information about the random variables.

For example lets take two dice (A, B) and play a game. The user must choose a die and throw it once. If die A lands on an even number, the player gets 1 dollar and he loses 1 otherwise. If die B land on $1, 2, 3, 4, 5$ he gets 20 dollars and loses 100 dollars if it lands on 6. Both dice have an expectation of 0, but the two choices involve different levels of risk as they have different distributions. This risk cannot be characterized by simply looking at the expectation because both dice are indistinguishable in the sense of expectation, which does not reflect the other statistics of the distribution

(such as variance).

Before anything else, we define some algebraic operations between distributions:

Definition 20. *Let Z_1, Z_2 be two independant univariate distributions and let $\gamma \in \mathbb{R}$ be a constant We define:*

1. $(Z_1 + Z_2)(x) := (Z_1 * Z_2)(x)$ *(convolutionnal product)*
2. $(\gamma\, Z_1)(x) := \frac{1}{\gamma} Z_1(\gamma x)$ *(contracting the function horizontally)*
3. $(\gamma + Z_1)(x) := Z_1(x - \gamma)$ *(horizontal translation)*

Now we turn to policy evaluation using distributional RL. The policy π is fixed and we want to find the value distribution $Z^\pi(s, a)$ for a given (s, a) pair. First we define the transition operator:

Definition 21 (Transition Operator)**.**

$$P^\pi : \mathcal{D} \to \mathcal{D}$$

$$P^\pi Z(s, a) = Z(S', A') \tag{3}$$

where $S' \sim P(.|s, a)$ and $A' \sim \pi(.|S')$. Capital letter are used to emphasize the randomness of the new state and action.

The output of this operator can be seen as a mixture distribution with weight $P(S' = s', A' = a'|a, s)$ for the each component distribution $Z^\pi(s', a')$. In what follows, we assume the randomness in the reward and the transition P^π are independent. Another operator that we define below is the distributional Bellmann operator:

Definition 22 (Distributional Bellmann operator)**.**

$$\mathcal{T}^\pi : \mathcal{D} \to \mathcal{D}$$

$$\mathcal{T}^\pi Z(s, a) = \mathit{Distr}(r) + \gamma \mathcal{P}^\pi Z(s, a) \tag{4}$$

where Dist(r) is the distribution of rewards at a given state s and by doing the action a. We could also change the input and output spaces of the operator so it acts on random variables instead of their distributions

$$\mathcal{T}^\pi \Phi^{\pi'}(s, a) \overset{D}{=} r + \gamma \mathcal{P}^\pi \Phi^{\pi'}(s, a) \tag{5}$$

These two definitions are equivalent (in the sense of distribution).

Here is a graphical explanation of the operator with deterministic reward:

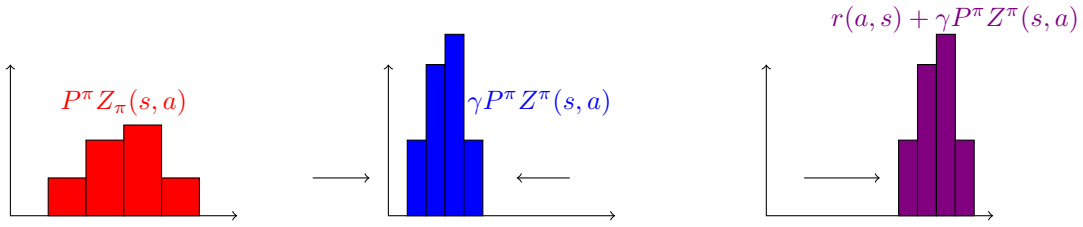

Figure 4: Illustration of how $\mathcal{T}^\pi$ operates on a distribution

The final operation is simply a translation because r is deterministic in that example. In a more general case, the rightmost distribution would have been obtained with a convolutional product. It is important to note that $\mathcal{T}^\pi$ resembles the Bellman equations for expected reward. In the spirit of policy evaluation, we want to show that the operator is a contraction mapping with respect to some metric. Let's first introduce the Wasserstein metric between distributions.

Definition 23. *Let Z_1 and Z_2 be two distributions; i.e. $Z_1, Z_2 \in \mathcal{D}$. For $p \geq 1$, the p-Wassertein metric is defined as*

$$d_p(Z_1, Z_2) = \inf_{D \in \prod(Z_1, Z_2)} \mathbb{E}_{(z_1, z_2) \sim D}[||z_1 - z_2||_p^p]^{\frac{1}{p}} \tag{6}$$

where $\prod(Z_1, Z_2)$ is the set of all joint distributions with marginals Z_1 and Z_2. The metric has the following properties (for $\gamma \in \mathbb{R}$ and a, a random variable independent of z_1 and z_2 that follows distribution A):

1. $d_p(\gamma Z_1, \gamma Z_2) \leq |\gamma| d_p(Z_1, Z_2)$
2. $d_p(A + Z_1, A + Z_2) \leq d_p(Z_1, Z_2)$
3. $d_p(AZ_1, AZ_2) \leq ||a||_p d_p(Z_1, Z_2)$

Note that the Wasserstein metrics are metrics of distributions, whereas value distributions are mappings (from the space of state-action pairs to the space of distributions), the former are not yet metrics of the latter. Let $\mathcal{Z}$ denote the **space of value distributions** (with bounded moments). Let us define a uniform form of the Wasserstein distance as

$$\bar{d}_p(Z_1, Z_2) = \sup_{s,a} d_p(Z_1(s,a), Z_2(s,a))$$

for $Z_1, Z_2 \in \mathcal{Z}$. Then we can establish the following result. $\bar{d}_p$ is a metric over value distributions. The only nontrivial part to prove is triangle inequality of a metric.

Proof. For $Y \in \mathcal{Z}$, we have

$$\begin{aligned}
\bar{d}_p(Z_1, Z_2) &= \sup_{s,a} d_p(Z_1(s,a), Z_2(s,a)) \\
&\leq \sup_{s,a} d_p(Z_1(s,a), Y(s,a)) + d_p(Y(s,a), Z_2(s,a)) \\
&\leq \sup_{s,a} d_p(Z_1(s,a), Y(s,a)) + \sup_{s,a} d_p(Y(s,a), Z_2(s,a)) \\
&= \bar{d}_p(Z_1, Y) + \bar{d}_p(Y, Z_2)
\end{aligned}$$

where the first inequality is because d_p is a metric which admits triangle inequality over the space of distributions. □

Now consider the metric space $(\mathcal{Z}, \bar{d}_p)$. Considering the iterative process $Z_{k+1} := \mathcal{T}^\pi Z_k$ with some initial value distirbution $Z_0 \in \mathcal{Z}$, we now show that "distributional" Bellman operator is a contraction mapping. $\mathcal{T}^\pi : \mathcal{Z} \to \mathcal{Z}$ is a γ-contraction in $\bar{d}_p$.

Proof. Let $Z_1, Z_2 \in \mathcal{Z}$.

$$\begin{aligned}
\bar{d}_p(\mathcal{T}^\pi(Z_1), \mathcal{T}^\pi(Z_2)) &= \sup_{s,a} d_p(\mathcal{T}^\pi(Z_1), \mathcal{T}^\pi(Z_2)) \\
&= \sup_{s,a} d_p(Distr(r; s, a) + \gamma P^\pi Z_1(s,a), Distr(r; s, a) + \gamma P^\pi Z_2(s,a)) \\
&\leq \sup_{s,a} \gamma d_p(P^\pi Z_1(s,a), P^\pi Z_2(s,a)) \\
&\leq \sup_{s,a} \gamma \sup_{s',a'} d_p(Z_1(s',a'), Z_2(s',a')) \\
&= \gamma \bar{d}_p(Z_1, Z_2)
\end{aligned}$$

where the first two lines are just the definitions of $\bar{d}_p$ and $\mathcal{T}^\pi$; the third line is due to the properies of Wasserstein distance (Definition 23) and the (conditional) independence of reward and the transition; the fourth line is due to taking the sup rather than taking a random next state-action pair. □

By construction, Z^π is a fixed point of the Bellman equation, and by the Banach fixed point theorem, the sequence $(Z_k)_{k \geq 1}$ will converge in $\bar{d}_p$ to Z^π.

7 Summary

In this lecture, we saw the infinite horizon variant of RL. We introduced the Banach fixed point theorem, a central theorem to many convergence results in and outside RL algorithms, and used it to demonstrate convergence for distributed RL methods.

References

[1] M. G. Bellemare, W. Dabney, and R. Munos. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 449–458. JMLR.org, 2017.

[2] R. S. Sutton and A. G. Barto. Reinforcement learning: An introduction. 2011.

[3] Wikipedia. Wikipedia. `https://en.wikipedia.org/wiki/Banach_fixed-point_theorem`, April 2019.