Machine Learning **November 11, 2025, Fall**

Lecture 9: Reinforcement Learning: A Brief Introduction

Lecturer: Zikang Shan

Scribe: Group 8

9.1 Reinforcement Learning Overview

9.1.1 Learning algorithms

- Supervised Learning: Labeled data ⇒ Predictor. Immediate feedback.
- Unsupervised Learning: Unlabeled data ⇒ Structure. No feedback.
- Reinforcement Learning: Experience ⇒ Decision Policy. Delayed scalar feedback.

9.1.2 Goal of RL(Informal)

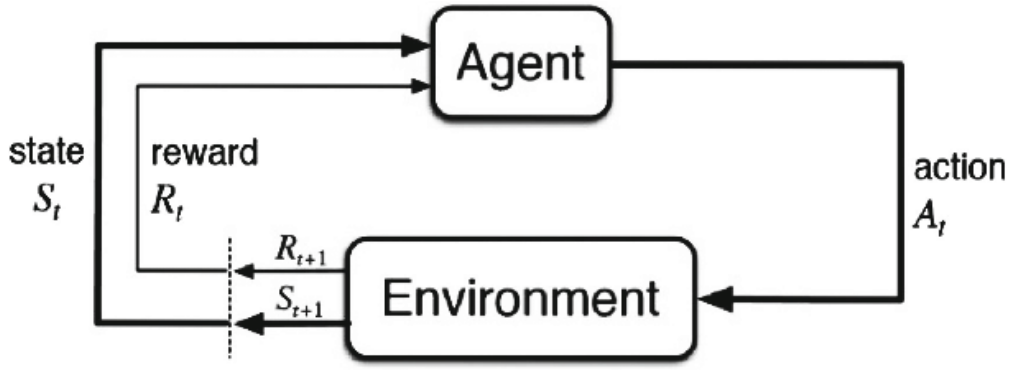

Figure 9.1: Illustration of RL

- State (S_t): The agent observes the current environment.
- Action (A_t): The agent decides an action to take.
- Reward (R_t): The agent receives a scalar feedback.
- Markov property: Both the agent and the environment are memoryless.
- Policy (π): Decide what action to take given a state.
- Goal: Learn a policy that achieves high rewards over time.

9.1.3 Challenges

- Exploration vs. Exploitation
- Credit Assignment: Determine which past actions leads to current outcome.
- Sample Complexity: Requires massive interactions to work.
- Apply MDP properly in practice: e.g. How to define a proper reward function?

9.2 Markov Decision Process (MDP)

Definition 9.1 ***Markov Decision Process (MDP)*** *is defined by* $\langle \mathcal{S}, \mathcal{A}, p, R, \gamma, \rho_0 \rangle$*, where*

- $\mathcal{S}$*: state space*
- $\mathcal{A}$*: action space*
- p*:* $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$*, transition probability function (transition dynamic)*
- R*:* $\mathcal{S} \times \mathcal{A} \to \mathbb{R}$*, reward function*
- $\gamma \in [0, 1]$*: discount factor*
- ρ_0*:* $\mathcal{S} \to \mathbb{R}$*, initial state distribution*

Notions:

- Policy: $\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a policy that determines the probability over actions to take at given states.
- Trajectory: $\tau = (S_0, A_0, R_1, S_1, A_1, R_2, \dots)$ is a sequence of states and actions generated by the MDP and the policy π.
$$p_\pi(\tau) = \rho_0(S_0) \prod_{t=0}^{\infty} \pi(A_t|S_t) p(S_{t+1}|S_t, A_t)$$
- Return: $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$ is the discounted total rewards at S_t.

The RL objective is the expected return $J(\pi) = \mathbb{E}_{\tau \sim \pi} [G_0]$.

We wish to obtain a policy that maximizes $J(\pi)$, i.e.

$$\pi^* = \arg\max_\pi J(\pi)$$

9.2.1 Bellman Expectation Equation

Definition 9.2 *The* ***state-value (value) function*** v_π *is the expected return starting from state* s *and then following* π*:*

$$v_\pi(s) = \mathbb{E}_\pi [G_t | S_t = s]$$

Definition 9.3 *The **action-value (q) function** q_π is the expected return starting from state s, taking action a, and then following π:*

$$q_\pi(s,a) = \mathbb{E}_\pi \left[G_t | S_t = s, A_t = a \right]$$

It is easy to find the connection between v_π and q_π:

$$v_\pi(S_t) = \mathbb{E}_{A_t \sim \pi(\cdot|S_t)} \left[q_\pi(S_t, A_t) \right]$$

and

$$q_\pi(S_t, A_t) = R(S_t, A_t) + \gamma \mathbb{E}_{S_{t+1} \sim p(\cdot|S_t, A_t)} \left[v_\pi(S_{t+1}) \right]$$

Combining the two equations, we have the **Bellman expectation equation**:

$$v_\pi(S_t) = \mathbb{E}_{A_t \sim \pi(\cdot|S_t)} \left[R(S_t, A_t) + \gamma \mathbb{E}_{S_{t+1} \sim p(\cdot|S_t, A_t)} \left[v_\pi(S_{t+1}) \right] \right]$$

and

$$q_\pi(S_t, A_t) = R(S_t, A_t) + \gamma \mathbb{E}_{S_{t+1} \sim p(\cdot|S_t, A_t), A_{t+1} \sim \pi(\cdot|S_{t+1})} \left[q_\pi(S_{t+1}, A_{t+1}) \right]$$

Note that, if the MDP is tabular:

- finite small spaces: $|\mathcal{S}|$ and $|\mathcal{A}|$ are finite and small.
- known: transition function p and reward function R are known.

Then the Bellman Equation is directly solvable by linear algebra. (For value function, we have $|\mathcal{S}|$ equations with $|\mathcal{S}|$ unknowns, and for action-value function, we have $|\mathcal{S}| \times |\mathcal{A}|$ equations with $|\mathcal{S}| \times |\mathcal{A}|$ unknowns.)

9.2.2 Bellman Optimality Equation

Definition 9.4 *The optimal value function v_* (q_*) is the maximum value function over all policies Π:*

$$v_*(s) = \max_{\pi \in \Pi} v_\pi(s)$$

$$q_*(s,a) = \max_{\pi \in \Pi} q_\pi(s,a)$$

Define a partial ordering over policies:

$$\pi' \geq \pi \iff v_{\pi'}(s) \geq v_\pi(s), \forall s \in \mathcal{S}$$

Then a policy π^* is optimal if $\pi^* \geq \pi, \forall \pi \in \Pi$.

The optimal poclicy exists by greedying on q_*:

$$\pi(a|s) = \begin{cases} 1, & a = \arg\max_{a' \in \mathcal{A}} q_*(s,a') \\ 0, & \text{otherwise} \end{cases}$$

However, the optimal policy may not be unique, but all of them share the same v_* and q_*. The proof will be given later.

Then we have the **Bellman optimality equation**:

$$v_*(S_t) = \max_{A_t \in \mathcal{A}} q_*(S_t, A_t)$$

$$q_*(S_t, A_t) = R(S_t, A_t) + \gamma \mathbb{E}_{S_{t+1} \sim p(\cdot|S_t, A_t)} \left[v_*(S_{t+1}) \right]$$

$$v_*(S_t) = \max_{A_t \in \mathcal{A}} \left(R(S_t, A_t) + \gamma \mathbb{E}_{S_{t+1} \sim p(\cdot|S_t, A_t)} \left[v_*(S_{t+1}) \right] \right)$$

$$q_*(S_t, A_t) = R(S_t, A_t) + \gamma \mathbb{E}_{S_{t+1} \sim p(\cdot|S_t, A_t)} \left[\max_{A_{t+1} \in \mathcal{A}} q_*(S_{t+1}, A_{t+1}) \right]$$

Apparently, the Bellman optimality equation is non-linear, so it has no closed form solution. The numberical solutions usually conclude dynamic programming methods.

9.3 Dynamic Programming

Dynamic programming is applicable if the problem has two attributes:

- Optimal substructure: Optimal solution decomposes to optimal solutions of subproblems.
- Overlapping subproblems: Space of subproblems are small.

9.3.1 Bellman Operator

Definition 9.5 *Define Bellman expectation operator $\mathcal{B}_\pi : \mathcal{V} \to \mathcal{V}$ by:*

$$\mathcal{B}_\pi v(s) = \mathbb{E}_{a \sim \pi(\cdot|s)} \left[R(s, a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)} \left[v(s') \right] \right]$$

where $\mathcal{V} = \{v : \mathcal{S} \to \mathbb{R}\}$ is the set of all value functions.

Definition 9.6 *Similarly, define Bellman optimality operator $\mathcal{B}_* : \mathcal{V} \to \mathcal{V}$ by:*

$$\mathcal{B}_* v(s) = \max_{a \in \mathcal{A}} \left(R(s, a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)} \left[v(s') \right] \right)$$

Then we have v_π (or v_*) is a fixed point of $\mathcal{B}_\pi$ (or $\mathcal{B}_*$), if exists. (And we will show the existence later.)

Lemma 9.7 *The Bellman expectation operator $\mathcal{B}_\pi$ and the Bellman optimality operator $\mathcal{B}_*$ are γ-contraction mappings with respect to the sup-norm $\| \cdot \|_\infty$:*

$$\|\mathcal{B}_\pi v_1 - \mathcal{B}_\pi v_2\|_\infty \leq \gamma \|v_1 - v_2\|_\infty, \forall v_1, v_2 \in \mathcal{V}$$

$$\|\mathcal{B}_* v_1 - \mathcal{B}_* v_2\|_\infty \leq \gamma \|v_1 - v_2\|_\infty, \forall v_1, v_2 \in \mathcal{V}$$

Proof:

$$\begin{aligned}
&\|\mathcal{B}_\pi v_1 - \mathcal{B}_\pi v_2\|_\infty \\
=&\max_{s\in\mathcal{S}} \left|\mathbb{E}_{a\sim\pi(\cdot|s)}\left[R(s,a)+\gamma\mathbb{E}_{s'\sim p(\cdot|s,a)}\left[v_1(s')\right]\right] - \mathbb{E}_{a\sim\pi(\cdot|s)}\left[R(s,a)+\gamma\mathbb{E}_{s'\sim p(\cdot|s,a)}\left[v_2(s')\right]\right]\right| \\
=&\gamma\max_{s\in\mathcal{S}} \left|\mathbb{E}_{a\sim\pi(\cdot|s),s'\sim p(\cdot|s,a)}\left[v_1(s')-v_2(s')\right]\right| \\
\leq&\gamma\max_{s\in\mathcal{S}} |v_1(s)-v_2(s)| = \gamma\|v_1-v_2\|_\infty
\end{aligned}$$

$$\begin{aligned}
&\|\mathcal{B}_* v_1 - \mathcal{B}_* v_2\|_\infty \\
=&\max_{s\in\mathcal{S}} \left|\max_{a\in\mathcal{A}}\left(R(s,a)+\gamma\mathbb{E}_{s'\sim p(\cdot|s,a)}\left[v_1(s')\right]\right) - \max_{a\in\mathcal{A}}\left(R(s,a)+\gamma\mathbb{E}_{s'\sim p(\cdot|s,a)}\left[v_2(s')\right]\right)\right| \\
\leq&\max_{s\in\mathcal{S}}\max_{a\in\mathcal{A}} \left|\gamma\mathbb{E}_{s'\sim p(\cdot|s,a)}\left[v_1(s')-v_2(s')\right]\right| \\
\leq&\gamma\max_{s\in\mathcal{S}} |v_1(s)-v_2(s)| = \gamma\|v_1-v_2\|_\infty
\end{aligned}$$

■

Theorem 9.8 *When $\gamma < 1$, let $v_0 \in \mathcal{V}$ be any value function, and define the sequence $\{v_k\}$ by $v_{k+1} = \mathcal{B}_\pi v_k$ (or $v_{k+1} = \mathcal{B}_* v_k$). Then v_k converges to a unique fixed point v_π (or v_*) as $k \to \infty$.*

Proof: Let's prove for Bellman expectation operator $\mathcal{B}_\pi$. The proof for Bellman optimality operator $\mathcal{B}_*$ is similar.

By Lemma 9.7, for any $m > n$, we have:

$$\|v_m - v_n\|_\infty \leq \sum_{k=n}^{m-1} \|v_{k+1} - v_k\|_\infty = \sum_{k=n}^{m-1} \|\mathcal{B}_\pi^k v_1 - \mathcal{B}_\pi^k v_0\|_\infty \leq \sum_{k=n}^{m-1} \gamma^k \|v_1 - v_0\|_\infty \leq \frac{\gamma^n}{1-\gamma}\|v_1 - v_0\|_\infty$$

Hence $\{v_k\}$ is a Cauchy sequence, so $\{v_k\}$ converges to some limit $\bar{v}$.

And we have

$$\bar{v} = \lim_{k\to\infty} v_{k+1} = \lim_{k\to\infty} \mathcal{B}_\pi v_k = \mathcal{B}_\pi \bar{v}$$

Then $\bar{v}$ is a fixed point of $\mathcal{B}_\pi$.

Finally, we show the uniqueness of the fixed point. Suppose there is another fixed point $\tilde{v}$, then

$$\|\bar{v} - \tilde{v}\|_\infty = \|\mathcal{B}_\pi \bar{v} - \mathcal{B}_\pi \tilde{v}\|_\infty \leq \gamma\|\bar{v} - \tilde{v}\|_\infty$$

Since $\gamma < 1$, we have $\|\bar{v} - \tilde{v}\|_\infty = 0$, i.e. $\bar{v} = \tilde{v}$. ■

9.3.2 Iterative Policy Evaluation

Now we will talk about tabular MDPs. We want to iteratively improve initial policy through:

1. Policy Evaluation Phase (E): Compute v_π with iterative policy evaluation.
2. Policy Improvement Phase (I): Construct $\pi' \geq \pi$ with greedy policy improvement.

Goal: Given a policy π, compute v_π.

Iterative Policy Evaluation:

- Initialize v_0 arbitrarily.
- For $k = 0, 1, 2, \ldots$, update Iteratively:

$$v_{k+1}(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left(R(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s,a) v_k(s') \right)$$

  That is, $v_{k+1} = \mathcal{B}_\pi v_k$.

By Theorem 9.8, v_k converges to v_π as $k \to \infty$.

9.3.3 Greedy Policy Improvement

Goal: Given v_π, construct a new policy π' such that $\pi' \geq \pi$.

Let

$$\pi'(s) = \arg\max_a q_\pi(s,a)$$

Lemma 9.9 *For any policy π and $v_1 \geq v_2$, we have $\mathcal{B}_\pi v_1 \geq \mathcal{B}_\pi v_2$,*

where $v_1 \geq v_2$ if and only if $v_1(s) \geq v_2(s), \forall s \in \mathcal{S}$.

Proof: $\forall s \in \mathcal{S}$,

$$\begin{aligned} \mathcal{B}_\pi v_1(s) &= \sum_{a \in \mathcal{A}} \pi(a|s) \left(R(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s,a) v_1(s') \right) \\ &\geq \sum_{a \in \mathcal{A}} \pi(a|s) \left(R(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s,a) v_2(s') \right) = \mathcal{B}_\pi v_2(s) \end{aligned}$$

■

Theorem 9.10 *For any policy π, let π' be the greedy policy with respect to v_π. Then $\pi' \geq \pi$.*

Proof: $\forall s \in \mathcal{S}$,

$$\begin{aligned} \mathcal{B}_{\pi'} v_\pi(s) &= \sum_{a \in \mathcal{A}} \pi'(a|s) \left(R(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s,a) v_\pi(s') \right) \\ &= \max_{a \in \mathcal{A}} \left(R(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s,a) v_\pi(s') \right) \\ &\geq \sum_{a \in \mathcal{A}} \pi(a|s) \left(R(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s,a) v_\pi(s') \right) = v_\pi(s) \end{aligned}$$

That is $\mathcal{B}_{\pi'} v_\pi \geq v_\pi$, then by Lemma 9.9, we have $\forall k, \mathcal{B}_{\pi'}^{k+1} v_\pi \geq \mathcal{B}_{\pi'}^k v_\pi$. Then

$$v_{\pi'} = \lim_{k \to \infty} \mathcal{B}_{\pi'}^k v_\pi \geq v_\pi$$

■

Theorem 9.11 *The fixed point solution is optimal, i.e.* $\pi' = \pi \implies v_\pi = v_*$.

Proof: $v_\pi = v_{\pi'} \implies v_\pi = v_{\pi'} = \mathcal{B}_{\pi'} v_{\pi'} = \mathcal{B}_{\pi'} v_\pi = \mathcal{B}_* v_\pi$, the last equality holds because π' is greedy with respect to v_π.

Hence v_π is a fixed point of $\mathcal{B}_*$. As v_* is the unique fixed point of $\mathcal{B}_*$, we have $v_\pi = v_*$. ■

Algorithm 1: Policy Iteration: Practical Algorithm

$\pi_0 \leftarrow$ arbitrary policy
for $k = 0, 1, \ldots$ **do**
  `// Policy Evaluation Phase`
  $v_k \leftarrow$ iterative policy evaluation of π_k
  `// Policy Improvement Phase`
  $\pi_{k+1}(s) \leftarrow \arg\max_a \left(R(s,a) + \gamma \sum_{s'} p(s'|s,a) v_k(s')\right), \forall s$
  if $\pi_{k+1} = \pi_k$ **then**
    return π_k
  end
end

Remark:

- Since the set of greedy policies is finite, the policy iteration algorithm converges to an optimal policy in a finite number of iterations.
- For the policy evaluation phase, we can iterate for a fixed large enough k instead of until convergence, since once the approximate value function is accurate enough that the greedy action in each state matches the true best action—that is, its error is smaller than half of the action-value gap—the resulting policy updates are identical to those from exact policy iteration.

9.4 Value Iteration

When $k = 1$ in the policy evaluation phase of policy iteration, we get value iteration:

- Initialize v_0 arbitrarily.
- For $k = 0, 1, 2, \ldots$, update Iteratively:

$$\pi_k(s) = \arg\max_a \left(R(s,a) + \gamma \sum_{s'} p(s'|s,a) v_k(s') \right)$$

$$v_{k+1}(s) = \sum_{a \in \mathcal{A}} \pi_k(a|s) \left(R(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s,a) v_k(s') \right)$$

Since π_k is always a greedy policy, by making it implicit, we have:

Algorithm 2: Value Iteration: Practical Algorithm

$v_0 \leftarrow$ arbitrary value function
for $k = 0, 1, 2, \ldots$ **do**
  $v_{k+1}(s) \leftarrow \max_a \left(R(s,a) + \gamma \sum_{s'} p(s'|s,a) v_k(s')\right), \forall s$
  if $\|v_{k+1} - v_k\|_\infty \leq \epsilon$ **then**
    return $\pi^*(s) = \arg\max_a \left(R(s,a) + \gamma \sum_{s'} p(s'|s,a) v_{k+1}(s')\right), \forall s$
  end
end

- convergence: By Theorem 9.8, v_k converges to v_* as $k \to \infty$.
- stopping condition: We can stop when π_k converges. Policies can converge before value functions converge.
- Efficiency: The efficiency of policy iteration and value iteration are not directly comparable. (Notices that v_k is not always equal to v_{π_k}.)

9.5 Building Blocks of RL Methods

Iterative until performance convergence:

- Data collection step: Collect trajectory samples by executing some policy.
- Policy evaluation: Evaluate the policy on the collected samples.
- Policy improvement: Based on evaluation results, update the policy.

References

[MRT13] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar, The Foundation of Machine Learning.