DOTE 6635: Artificial Intelligence for Business Research (Spring 2026)

Model-Free Reinforcement Learning

Renyu (Philip) Zhang

1

Core Challenges of RL

- Temporal credit assignment
  - Which earlier actions/deserves deserve "credit" or "blame" for outcomes that happen later?
- Exploration
  - How to efficiently explore to gain information?
- Generalization
  - How policies learned in one environment could generalize?
- RL to solve a large planning problem using a learning approach
  - AlphaGo, AlphaStar, simulated robotics, LLM coding, LLM math, etc.
  - Transition dynamics is known, but the very very large state space.
  - Run the simulator to collect data.
  - RL proved to be very successful.
- RL to solve a learning problem
  - Adaptive medical treatment, deep research agent, AI scientist, etc.
  - Transition dynamics unknown, reward unknown (and too many states).
  - Interact with the environment to collect data.
  - Great potential for RL, but not realized yet.

2

2

Agenda

- **Monte Carlo Method**
- Temporal Difference

3

3

Model-Free RL

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-4-model-free-prediction-.pdf

- So far we assume that we know all four important sets of information:
  - Action Space A
  - State Space S
  - Transition Matrix P
  - Reward Function R
- Now, we relax this assumption and assume we do not know the reward function and transition matrix.

4

4

Monte-Carlo RL

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-4-model-free-prediction-.pdf

- Model-free learning:
  - You do not know reward function and transition matrix.
  - However, you can take actions and experiment with the environment to observe the outcomes.
- Goal: Given a policy, learn the value function from data.
- Monte-Carlo methods:
  - MC RL learn directly from episodes of experience. All episode must terminate.
  - MC is model-free, without any knowledge of MDP transitions or rewards.
  - MC leverages the simplest possible idea that value function = mean return.

5

5

MC Policy Evaluation

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-4-model-free-prediction-.pdf

- Objective: Estimate the value function v_π under policy π:

$$S_0, A_0, R_0, \cdots, S_T \sim \pi$$

$$G_t = R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{T-1} R_T \qquad v_\pi(s) = \mathbb{E}_\pi \left[G_t \mid S_t = s \right]$$

- The idea of Monte Carlo: use empirical mean return the approximate the expected return.
- First-visit MC: v_π is estimated by the average return following each first visit to a state s in a set of episodes.
- Every-visit MC: v_π is estimated by the average return following each visit to a state s in a set of episodes.
- If there is an absorbing state, everything is well-defined; otherwise, we need to specify a finite T, which is larger if γ is larger.

6

6

First-Visit MC Policy Evaluation

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-4-model-free-prediction-.pdf

- **Initialization**:
  N (counter), $N(s) \leftarrow 0$ for all $s \in \mathcal{S}$
  $\text{Returns}(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$
- **Repeat**:
  Generate an episode following policy π
  For each distinct s appearing in the episode
    $G \leftarrow$ return following the first occurrence of s
    $N(s) \leftarrow N(s) + 1$
    $\text{Returns}(s) \leftarrow \text{Returns}(s) + G$
- **Output**:
  For each distinct s
    $N^{-1}(s)\text{Returns}(s) \longrightarrow v_\pi(s)$ as $N(s) \rightarrow$ +inf

- First-visit evaluation is unbiased, and it converges to the true value function.

7

7

Every-Visit MC Policy Evaluation

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-4-model-free-prediction-.pdf

- **Initialization**:
  $N \leftarrow$ counter, $N(s) \leftarrow 0$ for all $s \in \mathcal{S}$
  $\text{Returns}(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$
- **Repeat**:
  Generate an episode following policy π
  For each s appearing in the episode
    $G \leftarrow$ return following the occurrence of s
    $N(s) \leftarrow N(s) + 1$
    $\text{Returns}(s) \leftarrow \text{Returns}(s) + G$
- **Output**:
  For each distinct s
    $N^{-1}(s)\text{Returns}(s) \longrightarrow v_\pi(s)$ as $N(s) \rightarrow$ +inf

- Every-visit evaluation is biased, but it still converges to the true value function. And it usually has a smaller MSE because the sample size is much larger.

8

8

Incremental MC Policy Evaluation

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-4-model-free-prediction-.pdf

- An incremental way to compute mean value:

$$\begin{aligned}\mu_k &= \frac{1}{k}\sum_{j=1}^{k} x_j \\ &= \frac{1}{k}\left(x_k + \sum_{j=1}^{k-1} x_j\right) \\ &= \frac{1}{k}\left(x_k + (k-1)\mu_{k-1}\right) \\ &= \mu_{k-1} + \frac{1}{k}\left(x_k - \mu_{k-1}\right)\end{aligned}$$

- Similarly, for MC policy evaluation:

$$N(S_t) \leftarrow N(S_t) + 1$$

$$V(S_t) \leftarrow V(S_t) + \frac{1}{N(S_t)}\left(G_t - V(S_t)\right)$$

9

9

Incremental MC Policy Evaluation

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-4-model-free-prediction-.pdf

- This incremental update also means less memory usage.

- More importantly, it allows for non-stationary policy (i.e., the transition matrices may change over time):
  - We only consider stationary policy so far, because a stationary MDP with finite states is guaranteed to have a stationary optimal policy.
  - If we go beyond this setting, i.e., policy at time t is different from policy at time $t+1$.

$$V(S_t) \leftarrow V(S_t) + \alpha\left(G_t - V(S_t)\right)$$

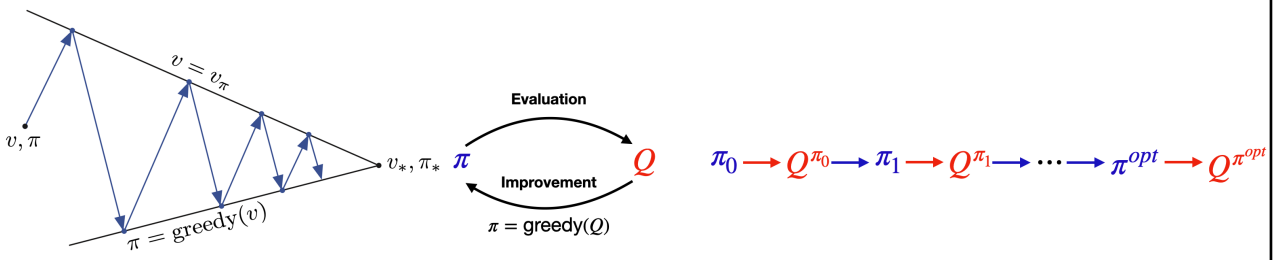
10

10

MC Policy Iteration

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-5-model-free-control-.pdf

- Objective: use MC estimation to learn the optimal policy.
- MC marries policy iteration:



- Policy Evaluation: Compute v_π via MC policy evaluation.
- Policy Iteration: Improve the current policy w.r.t. the current state-action value function Q_π.

- Challenge: under a policy, many state-action pairs may never be visited.
- Solution: epsilon-greedy.

11

11

MC Policy Iteration with ϵ-Greedy

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-5-model-free-control-.pdf

- **Initialization**:

  $\boldsymbol{N}$ (counter), $\boldsymbol{N}(\boldsymbol{s}, \boldsymbol{a}) \leftarrow \mathbf{0}$ for all $\boldsymbol{s} \in \mathcal{S}$, $\boldsymbol{a} \in \mathcal{A}$

  Returns$(\boldsymbol{s}, \boldsymbol{a}) \leftarrow$ empty lists, for all $\boldsymbol{s} \in \mathcal{S}$, $\boldsymbol{a} \in \mathcal{A}$

  $\pi \leftarrow$ arbitrary ε-greedy policy

  $\boldsymbol{Q} \leftarrow$ arbitrary

- **Repeat**:

  Generate an episode using exploring starts and policy π

  For each distinct $(\boldsymbol{s}, \boldsymbol{a})$ appearing in the episode

  $\boldsymbol{G} \leftarrow$ return following the first occurrence of $(\boldsymbol{s}, \boldsymbol{a})$

  $\boldsymbol{N}(\boldsymbol{s}, \boldsymbol{a}) \leftarrow \boldsymbol{N}(\boldsymbol{s}, \boldsymbol{a}) + \mathbf{1}$

  Returns$(\boldsymbol{s}, \boldsymbol{a}) \leftarrow$ **Returns**$(\boldsymbol{s}, \boldsymbol{a}) + \boldsymbol{G}$

  $\boldsymbol{Q}(\boldsymbol{s}, \boldsymbol{a}) \leftarrow$ **Returns**$(\boldsymbol{s}, \boldsymbol{a}) / \boldsymbol{N}(\boldsymbol{s}, \boldsymbol{a})$

  For each distinct $\boldsymbol{s}$:

$$\pi(\boldsymbol{a}|\boldsymbol{s}) \leftarrow \begin{cases} \varepsilon/\boldsymbol{m} + \mathbf{1} - \varepsilon, & \text{if } \boldsymbol{a} = \arg\max \boldsymbol{Q}(\boldsymbol{s}, \boldsymbol{a}) \\ \varepsilon/\boldsymbol{m}, & \text{otherwise} \end{cases}$$

Thm: The ϵ-greedy policy iteration π' with respect to q_π is an improvement over any ϵ-greedy policy π, i.e.,

$$v_{\pi'}(s) \geq v_\pi(s)$$

12

12

Agenda

- Monte Carlo Method
- **Temporal Difference**

13

13

Temporal-Difference Learning

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-4-model-free-prediction-.pdf

- "If one had to identify one idea as central and novel to RL, it would undoubtedly be temporal-difference (TD) learning." – Sutton and Barto (2017)
- TD is a combination of MC and Bellman operator.
- TD is a model-free method.
- TD can be used in episodic or non-episodic settings.
- TD updates a guess towards a guess.

14

14

Temporal-Difference Learning

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-4-model-free-prediction-.pdf

- Recall that, given a policy π, the value function can be obtained as:

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \cdots \text{ in MDP } M \text{ under policy } \pi$$

$$V^\pi(s) = \mathbb{E}_\pi[G_t | s_t = s]$$

- Bellman operator and value iteration:

$$B^\pi V(s) = r(s, \pi(s)) + \gamma \sum_{s' \in S} p(s'|s, \pi(s)) V(s')$$

- The incremental MC: $V(S_t) \leftarrow V(S_t) + \alpha\,(G_t - V(S_t))$

- Simplest temporal-difference learning algorithm: TD(0)
  - Update value $V(S_t)$ toward *estimated* return $R_{t+1} + \gamma V(S_{t+1})$

$$V(S_t) \leftarrow V(S_t) + \alpha\,(R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$$

  - $R_{t+1} + \gamma V(S_{t+1})$ is called the *TD target*
  - $\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$ is called the *TD error*

TD can be thought of as a bootstrap method to update based on an existing estimate.

15

15

TD(0) Algorithm

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-4-model-free-prediction-.pdf

- **Input**: π policy to be evaluated, step size α
- **Initialization**: V arbitrary
- **Repeat** for each episode:
  - **Initialize** state s
  - **Repeat** for each step of the episode:
    - $a \leftarrow$ action given by π for s
    - Take action a, observe reward r and next state s'
    - $V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)]$
    - $s \leftarrow s'$
  - **until** s is a terminal state

16

16

MC vs. TD

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-4-model-free-prediction-.pdf

- Recall under the MDP model:

$$\begin{aligned} V^{\pi}(s) &= \mathbb{E}^{\pi}(G_t | S_t = s) && (1) \\ &= \mathbb{E}^{\pi}\left(\sum_{k=0}^{\infty} \gamma^k R_{t+k} | S_t = s\right) \\ &= \mathbb{E}^{\pi}[R_t + \gamma V^{\pi}(S_{t+1}) | S_t = s] && (2) \end{aligned}$$

- MC methods use G_t in (1) as the target.

- TD methods use $R_t + \gamma v_{\pi}(S_{t+1})$ in (2) as the target.

Immediate Reward

Estimate of Future Value

17

17

MC vs. TD: Pros and Cons

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-4-model-free-prediction-.pdf

- MC must wait until the end of an episode.
- MC learns from complete sequences.
- MC only works for episodic environments.
- TD can learn online after each step.
- TD can learn from incomplete sequences.
- TD works in continuing environments.

18

18

MC vs. TD: Bias-Variance Trade-off

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-4-model-free-prediction-.pdf

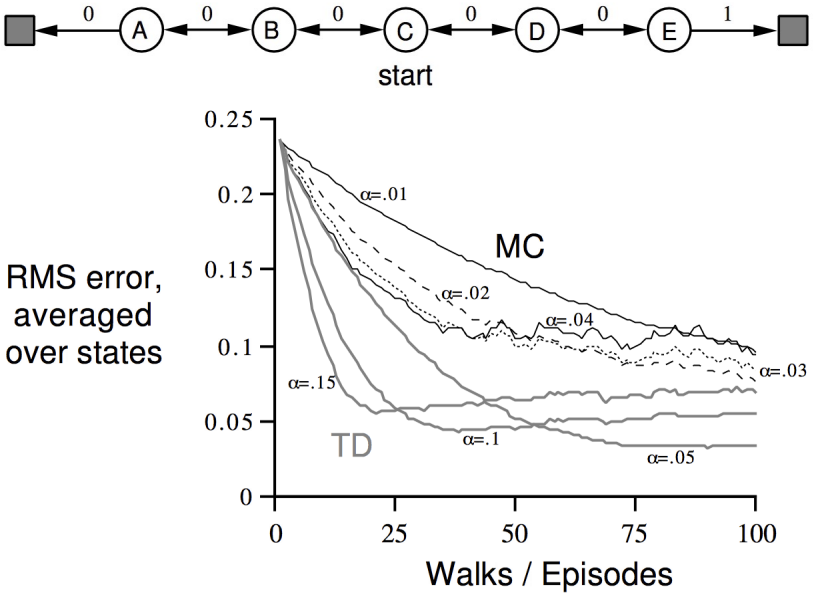
- MC updates on G_t after each episode, which is an unbiased estimate of $v_\pi(S_t)$.
- MC has high variance, zero bias.
  - Good convergence even with function approximation (using, e.g., DNN).
  - Not sensitive to initial values.
  - Simple idea without even knowing Bellman equation.
  - More efficient in non-Markov environments.
- TD updates on $R_t + \gamma v(S_{t+1})$, not $R_t + \gamma v_\pi(S_{t+1})$, which is a biased estimate of $v_\pi(S_t)$.
- TD has much lower variance, some bias.
  - Return depends on many random actions/transitions/rewards.
  - TD target only depends on one random action/transition/reward.
  - More efficient than MC, because it exploits Markov property and the Bellman equation.
  - TD(0) converges (but not always with function approximations).
  - More sensitive to initial values.

19

19

MC vs. TD: An Example

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-4-model-free-prediction-.pdf



20

20