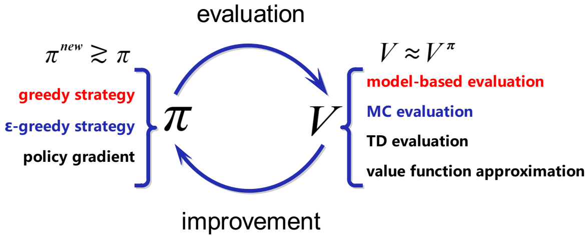
Algorithmic and Theoretical Foundations of RL

Monte Carlo (MC) Learning

Ke Wei, School of Data Science, Fudan University

With help from Jie Feng and Jiacai Liu

Policy Iteration Recap



Policy Iteration: greedy policy is improved via

$$\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}}\, \mathbb{E}_{s'} \left[r(s, a, s') + \gamma v_{\pi_k}(s') \right],$$

where $v_{\pi_k}(s')$ is evaluated via Bellman equation based on the model.

— What if system information (P and r) is not available?

— Replace model by data (model free).

— How to collect data? How to use data?

— ......

Action Value Based Policy Iteration

- Policy improvement via state value:

$$\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}}\, \mathbb{E}_{s'}\left[r(s,a,s') + \gamma v_{\pi_k}(s')\right].$$

  Given $v_{\pi_k}(s')$, still need to compute the expectation which requires model.

- Policy improvement via state value:

$$\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}}\, q_{\pi_k}(s,a).$$

  Ideal for model free RL since we can estimate $q_{\pi_k}(s,a)$ directly from data.

MC Policy Evaluation (or Prediction)

Basic idea. Given π, estimate $v_\pi(s)$ and $q_\pi(s,a)$ from sampled trajectories

$$\tau_i = \{(s_0^i, a_0^i, r_0^i, s_1^i, a_1^i, r_1^i, \cdots)\}_{i=1}^n \sim \pi.$$

- MC evaluation of $v_\pi(s)$: $s_0^i = s$,

$$v_\pi(s) \approx \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{t=0}^{\infty}\gamma^t r_t^i\right).$$

- MC evaluation of $q_\pi(s,a)$: $s_0^i = s$, $a_0^i = a$,

$$q_\pi(s,a) \approx \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{t=0}^{\infty}\gamma^t r_t^i\right).$$

Primitive MC Learning Algorithm

Algorithm 1: Primitive MC Learning

Initialization: π_0, n

for $k = 0, 1, 2, \ldots$ **do**

  for *every s* **do**

    for *every a* **do**

      Sample n episodes (finite-step trajectory) starting from (s, a) and then following π_k

$$\tau_i = \{(s_0^i, a_0^i, r_0^i, s_1^i, a_1^i, r_1^i, \cdots, s_{T-1}^i, a_{T-1}^i, r_{T-1}^i, s_T^i)\}_{i=1}^n \sim \pi_k$$

      Compute $Q_k(s, a) = \frac{1}{n} \sum_{i=1}^n \left(\sum_{t=0}^{T-1} \gamma^t r_t^i \right)$

    end

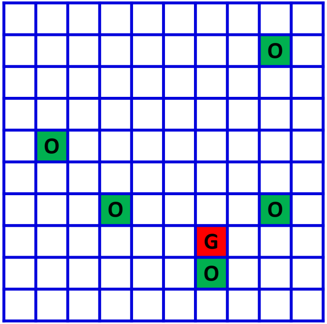
    $\pi_{k+1}(s) = \operatorname*{argmax}_a Q_k(s, a)$

  end

end

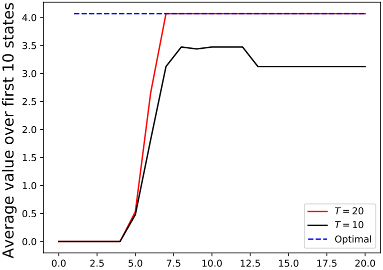
Ideally, T should be ∞ or s_T be a terminal state. In practice, T should be sufficiently large, especially for the sparse reward case.

Illustrative Example



Goal: +10, obstacle: -10; goal is terminal state.

Illustrative Example



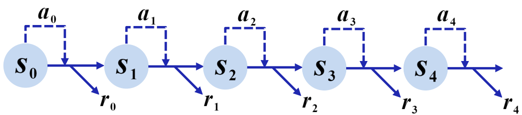
The learned policy is evaluated exactly using model.

Inefficiency of Primitive MC Learning

- A trajectory is only used for estimating one state-action value;
- Wait until all trajectories have been collected before policy update;
- Old state-action values are not reused and thus wasted (next lecture).

Table of Contents

Use Trajectory More Efficiently



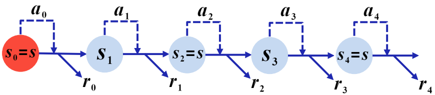
Trajectory $(s_0, a_0, r_0, s_1, a_1, r_1, \cdots) \sim \pi$ starting from s contains sub-trajectories $(s_t, a_t, r_t, s_{t+1}, a_{t+1}, r_{t+1}, \cdots)$ that starts from other states (e.g. $s_t = s'$). Thus, return from the sub-trajectory

$$G_t = \sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'}$$

can be used to build an estimator of $v_\pi(s')$. Namely, one trajectory can be used to estimate different $v_\pi(s)$.

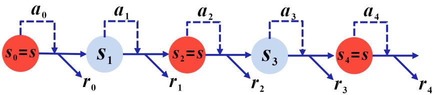
There is no difference in the MC evaluations of state value and action value in methodology. Thus discussion in this section will be mainly based on state value.

First-Visit and Every Visit



First Visit

- Only sub-trajectory that starts from the first visit of s is used in the estimation of $v_\pi(s)$; One trajectory is only used once in the evaluation of $v_\pi(s)$.



Every Visit

- All sub-trajectories that start from of s is used in the estimation of $v_\pi(s)$; One trajectory might be used many times in the evaluation of $v_\pi(s)$.

First-Visit MC Policy Evaluation

Algorithm 2: First-Visit Monte Carlo Policy Evaluation

Initialization: Counter of visited numbers $N(s) = 0$, the total return $G(s) = 0$, $\forall s \in \mathcal{S}$

for $k = 0, 1, 2, \ldots$ **do**

  Initialize s_0 and sample an episode following π:

$$(s_0, a_0, r_0, s_1, a_1, r_1, \cdots, s_{T-1}, a_{T-1}, r_{T-1}, s_T) \sim \pi$$

  $G \leftarrow 0$

  for $t = T-1, T-2, \ldots, 0$ **do**

    $G \leftarrow \gamma G + r_t$

    if s_t *does not appear in* $(s_0, s_1, \cdots, s_{t-1})$ **then**

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

      $G(s_t) \leftarrow G(s_t) + G$

      $V^{first}(s_t) \leftarrow G(s_t)/N(s_t)$

    end

  end

end

Every-Visit MC Policy Evaluation

Algorithm 3: Every-Visit Monte Carlo Policy Evaluation

Initialization: Counter of visited numbers $N(s) = 0$, the total return $G(s) = 0$, $\forall s \in \mathcal{S}$

for $k = 0, 1, 2, \dots$ **do**

  Initialize s_0 and sample an episode following π:

$$(s_0, a_0, r_0, s_1, a_1, r_1, \cdots, s_{T-1}, a_{T-1}, r_{T-1}, s_T) \sim \pi$$

  $G \leftarrow 0$

  for $t = T-1, T-2, \dots, 0$ **do**

    $G \leftarrow \gamma G + r_t$

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

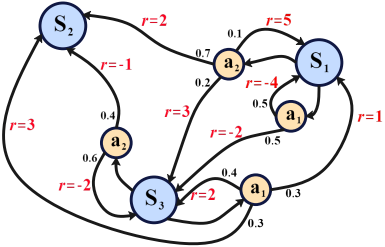
    $G(s_t) \leftarrow G(s_t) + G$

    $V^{every}(s_t) \leftarrow G(s_t)/N(s_t)$

  end

end

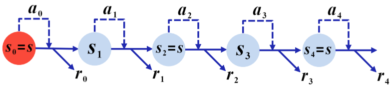
Illustrative Example



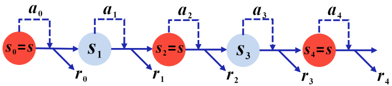
Consider the policy $\pi(a|s) = 0.5$ for each state s and each action a and $\gamma = 0.9$. Recall that $v_\pi = [-0.21, 0, 0.31]^T$.

- Consider a sampled trajectory: $(s_1, a_1, -2, s_3, a_1, 1, s_1, a_2, 3, s_3, a_2, -1, s_2)$
- First-visit policy evaluation for state s_3:
  $N(s_3) = 1, V^{first}(s_3) = (1 + 0.9 \times 3 + 0.9^2 \times (-1)) = 2.89$
- Every-visit policy evaluation for state s_3:
  $N(s_3) = 2, V^{every}(s_3) = (1 + 0.9 \times 3 + 0.9^2 \times (-1) - 1)/2 = 0.945$

First-Visit vs Every-Visit



First Visit



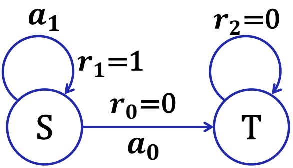
Every Visit

MSE = bias^2+variance

	Un-biased	Short MSE	Long MSE
First visit	Yes	Higher	Lower
Every visit	No	Lower	Higher

"Reinforcement learning with replacing eligibility traces" by Singh and Sutton, 1996.

Illustrative Example



$$\pi(a_1|s) = p, \quad \pi(a_0|s) = 1 - p.$$

State value of π at s is $v_\pi(s) = \frac{p}{1-p}$.

- Single trajectory

$$\mathbb{E}\left[V^{first}(s)\right] = \frac{p}{1-p}, \quad \text{MSE}\left[V^{first}\right] = \text{Var}\left[V^{first}\right] = \frac{p}{(1-p)^2};$$

$$\mathbb{E}\left[V^{every}\right](s) = \frac{p}{2(1-p)}, \quad \text{MSE}\left[V^{every}\right] = \frac{p}{4(1-p)^2}.$$

- As the number of trajectories increase, it can be shown that

$$V^{every}(s) \to \frac{p}{1-p}.$$

Incremental Monte Carlo Policy Evaluation

As demonstrated in the last lecture, mean evaluation can be conducted in an incremental way:

$$N(s_t) \leftarrow N(s_t) + 1, \quad V(s_t) \leftarrow V(s_t) + \frac{1}{N(s_t)}(G - V(s_t)).$$

Algorithm 4: First-Visit Monte Carlo Policy Evaluation (Incremental Version)

Initialization: Visited numbers $N(s) = 0$ and initialize $V(s)\ \forall s \in \mathcal{S}$.

for $k = 0, 1, 2, \ldots$ **do**

  Initialize s_0 and sample an episode following π:

$$(s_0, a_0, r_0, s_1, a_1, r_1, \cdots, s_{T-1}, a_{T-1}, r_{T-1}, s_T) \sim \pi$$

  $G \leftarrow 0$

  for $t = T-1, T-2, \ldots, 0$ **do**

    $G \leftarrow \gamma G + r_t$

    if *s_t does not appear in* $(s_0, s_1, \cdots, s_{t-1})$ **then**

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

      $V(s_t) \leftarrow V(s_t) + \frac{1}{N(s_t)}(G - V(s_t))$

    end

  end

end

Without further specification, discussion in the rest of this lecture will focus on first visit, and the superscript "first" will be omitted.

Table of Contents

Simply Combine MC Policy Evaluation with Greedy Policy

Algorithm 5: MC Learning with Greedy Policy

Initialization: $Q(s,a) = 0, N(s,a) = 0, \forall s, a$; Initialize π_0.

for $k = 0, 1, 2, \dots$ **do**

Initialize s_0 and sample an episode following π_k:

$$(s_0, a_0, r_0, s_1, a_1, r_1, \cdots, s_{T-1}, a_{T-1}, r_{T-1}, s_T) \sim \pi_k$$

$G \leftarrow 0$

for $t = T-1, T-2, \dots, 0$ **do**

$G \leftarrow \gamma G + r_t$

if (s_t, a_t) *does not appear in* $(s_0, a_0, s_1, a_1, \dots, s_{t-1}, a_{t-1})$ **then**

$N(s_t, a_t) \leftarrow N(s_t, a_t) + 1$

$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \frac{1}{N(s_t,a_t)}(G - Q(s_t, a_t))$

$\pi_{k+1}(s_t) = \operatorname*{argmax}_a Q(s_t, a)$ [update policy of visited state]

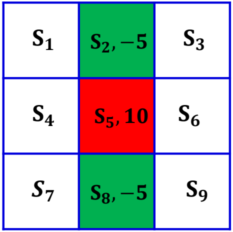
end

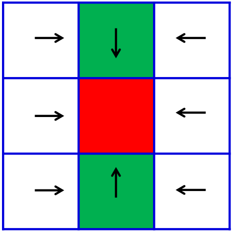
end

end

- Compared with primitive MC learning (i.e., Algorithm 1), Algorithm 5 updates policy after each episode is collected.

An Example Algorithm 5 Fails to Work





Consider the gridworld problem (left) where $\gamma = 0.9$. Assume $Q(s, a) = 0$ for all s, a and π_0 is given in the right plot. It can be verified that π_0 does not change for Algorithm 5.

Exploration and Exploitation

Indeed, how to collect data (or interaction with environment) is very important for the success of RL algorithms. We mainly consider the following intersection protocol: Start from an initialized state and then sample an episode following a policy (behavior policy). Eventually, we hope the data enables us to evaluate the state-action values of the target policy for all state-action pairs (recall that in model based policy iteration, action values are all equally evaluated for every action (fully exploration) or the first action is independent of policy). However, the behavior policy may bias towards some actions , for example the greedy policy. On the one hand, collect data from a biased behavior policy may reduce the ability of exploration. One the other hand, if the behavior policy can provide good experiences, it should be able to provide good instruction to improve the target policy. This forms the tradeoff between exploration and exploitation.

- How to encourage exploration?
  - Explore state-action pairs when sampling episodes.
  - ϵ-greedy policy
  - Off-policy learning

ϵ-Greedy Policy

With small probability ϵ randomly choose an action to ensure exploration:

$$\pi(a|s) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|\mathcal{A}|} & \text{if } a = \underset{a}{\operatorname{argmax}} Q_k(s, a'), \\ \frac{\epsilon}{|\mathcal{A}|} & \text{otherwise.} \end{cases}$$

Theorem 1

For any ϵ-greedy policy π, the ϵ-greedy policy π' with respect to q_π is an improvement, $v_{\pi'}(s) \geq v_\pi(s)$.

MC Learning with ϵ-Greedy Policy

Algorithm 6: MC Learning with ϵ-Greedy Exploration

Initialization: $N(s,a) = 0, Q(s,a) = 0, \forall s, a, \pi_0$

for $k = 0, 1, 2, \ldots$ **do**

Initialize s_0 and sample an episode following π_k:

$$(s_0, a_0, r_0, s_1, a_1, r_1, \cdots, s_{T-1}, a_{T-1}, r_{T-1}, s_T) \sim \pi_k$$

$G \leftarrow 0$

for $t = T-1, T-2, \ldots, 0$ **do**

$G \leftarrow \gamma G + r_t$

if (s_t, a_t) *does not appear in* $(s_0, a_0, s_1, a_1, \ldots, s_{t-1}, a_{t-1})$ **then**

$N(s_t, a_t) \leftarrow N(s_t, a_t) + 1$

$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \frac{1}{N(s_t,a_t)}(G - Q(s_t, a_t))$

Update policy of visited state via ϵ_k-greedy:

$$\pi_{k+1}(a|s_t) = \begin{cases} 1 - \epsilon_k + \frac{\epsilon_k}{|\mathcal{A}|} & \text{if } a = \operatorname*{argmax}_{a'} Q(s_t, a'), \\ \frac{\epsilon_k}{|\mathcal{A}|} & \text{otherwise.} \end{cases}$$

end

end

end

GLIE and Convergence of MC Learning with ϵ-Greedy Policy

Definition 1 (Greedy in the Limit with Infinite Exploration (GLIE))

The policies $\{\pi_k\}$ are called GLIE if:

- *All state-action pairs are explored infinitely many times,*

$$\lim_{k \to \infty} N_k(s, a) = \infty, \quad \forall s \in \mathcal{S}, \forall a \in \mathcal{A}.$$

- *The policy converges on a greedy policy,*

$$\lim_{k \to \infty} \pi_k(a|s) = \mathbf{1}\left(a = \underset{a' \in \mathcal{A}}{\mathrm{argmax}} Q_k\left(s, a'\right)\right).$$

For example, ϵ_k-greedy is GLIE if $\epsilon_k = \frac{1}{k}$.

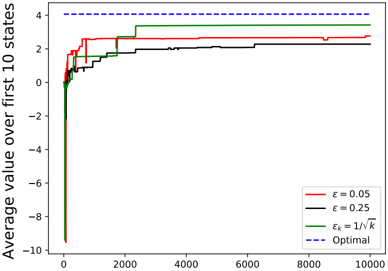
Theorem 2

GLIE MC learning converges to the optimal state-action value function, i.e.,

$$\lim_{k \to \infty} Q_k(s, a) \to q^*(s, a).$$

"Convergence results for single-step on-policy reinforcement-learning algorithms" by Singh et al., 2000.

Illustrative Example



For the previously mentioned 10×10 gridworld problem.

Table of Contents

Off-Policy Monte Carlo Evaluation

- On-policy learning vs Off-policy learning
  - On-policy: Learn target policy π from experience sampled from π;
  - Off-policy: Learn target policy π from experience sampled from b.
- On-policy ϵ-greedy method which is not deterministic need to behave non-optimally in order to explore all actions.
- Off-policy method attempts to learn a deterministic optimal policy, but from data generated by another exploratory policy.

Importance Sampling for Off-Policy MC Evaluation

Let $\tau_t = \{s_t, a_t, r_t, s_{t+1}, a_{t+1}, r_{t+1}, \cdots\}$ be a sub-trajectory from t induced by π. Let $(s_t, a_t) = (s, a)$ and Pr^{π}_{t+1} be the distribution of τ_{t+1}. By importance sampling,

$$\begin{aligned} q_\pi(s,a) &= \mathbb{E}_{P_t^\pi(r_t, r_{t+1}, \cdots)}\left[G_t\right] \\ &= \mathbb{E}_{P_t^b(r_t, r_{t+1}, \cdots)}\left[\frac{P_t^\pi(r_t, r_{t+1}, \cdots)}{P_t^b(r_t, r_{t+1}, \cdots)} G_t\right], \end{aligned}$$

where $G_t = \sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'}$ and

$$\frac{P_t^\pi(r_t, r_{t+1}, \cdots)}{P_t^b(r_t, r_{t+1}, \cdots)} = \frac{P\left(s_{t+1}|s_t, a_t\right) \prod_{k=t+1}^{\infty} P\left(s_{k+1}|s_k, a_k\right) \pi\left(a_k|s_k\right)}{P\left(s_{t+1}|s_t, a_t\right) \prod_{k=t+1}^{\infty} P\left(s_{k+1}|s_k, a_k\right) b\left(a_k|s_k\right)} = \prod_{k=t+1}^{\infty} \frac{\pi\left(a_k|s_k\right)}{b\left(a_k|s_k\right)}$$

is known as importance-sampling ratio.

Importance Sampling for Off-Policy MC Evaluation

Given an

$$\{s_0, a_0, r_0, s_1, a_1, r_1, \cdots, s_{T-1}, a_{T-1}, r_{T-1}, s_T\} \sim b,$$

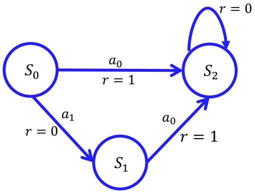
off-policy MC evaluation has the following form:

$$N(s_t, a_t) \leftarrow N(s_t, a_t) + 1$$

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \frac{1}{N(s_t, a_t)} \left(G_t \frac{P_t^{\pi}}{P_t^{b}} - Q(s_t, a_t) \right)$$

Weight for Initial Pair Should Not Be Included

Note when defining $q_\pi(s,a)$, action a is independent of the policy π. Thus, when computing importance sampling weight for (s_t,a_t), $\frac{\pi(a_t|s_t)}{b(a_t|s_t)}$ should not be included.



Suppose $\gamma < 1$. Optimal policy for s_0 is $\pi^*(s_0) = a_0$. Set $Q(s,a) = 0$ for all (s,a), $\pi_0(s_0) = a_1$ and $\pi_0(s_1) = a_0$. Two possible episodes for an exploratory behavior policy b:

$$(s_0, a_0, 1, s_2) \quad \text{and} \quad (s_0, a_1, 0, s_1, a_0, 1, s_2).$$

It is easy to verify that π_0 will not be updated if $\frac{\pi_0(a_0|s_0)}{b(a_0|s_0)} = 0$ is included in the computation of importance sampling weight. In contrast, π_0 will be updated if $\frac{\pi_0(a_0|s_0)}{b(a_0|s_0)} = 0$ is not included.

Example is from Wenye Li.

Off-Policy MC Learning

Algorithm 7: Off-policy MC Learning

Initialization: $\forall s, a$, arbitrarily initial $Q(s,a)$, $\pi_0(s) = \mathrm{argmax}_a\, Q(s,a)$, $N(s,a) = 0$.

for $k = 0, 1, 2, \ldots$ **do**

  $b_k \leftarrow$ any soft policy, i.e., $b_k(a|s) > 0, \forall s, a$

  Initialize s_0 and sample an episode following b_k:

$$(s_0, a_0, r_0, s_1, a_1, r_1, \cdots, s_{T-1}, a_{T-1}, r_{T-1}, s_T) \sim b_k$$

  $G \leftarrow 0, W \leftarrow 1$

  for $t = T-1, T-2, \ldots, 0$ **do**

    $G \leftarrow r_t + \gamma G$

    if (s_t, a_t) *does not appear in* $(s_0, a_0, s_1, a_1, \ldots, s_{t-1}, a_{t-1})$ **then**

      $N(s_t, a_t) \leftarrow N(s_t, a_t) + 1$

      $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \frac{1}{N(s_t, a_t)}(W \cdot G - Q(s_t, a_t))$

      $\pi_{k+1}(s_t) \leftarrow \mathrm{argmax}_a\, Q(s_t, a)$

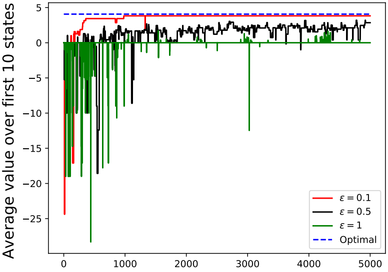
    end

    $W \leftarrow W \frac{\pi_k(a_t|s_t)}{b_k(a_t|s_t)}$

  end

end

Illustrative Example



For the previously mentioned 10×10 gridworld problem.

Questions?