Lecture 8 - Monte Carlo Methods (MD)

Instructor: GuangBing Yang, PhD

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

Introduction to Monte Carlo Methods

- What is Monte Carlo Methods?
  - They are learning methods for estimating value function and discovering optimal policies
  - Without knowing environment, only based on experience
  - Sample sequences of states, average sample returns
  - DP needs knowing the environment
  - Only for episodic tasks - experience is divided into episodes

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

Introduction to Monte Carlo Methods

- What is Monte Carlo Methods?
  - "Monte Carlo" the term in statistics means random sampling
  - Monte Carlo methods sample and average returns for each agate-action pair
  - Much like the bandit methods, but for multiple states in an interrelated way
  - Dealing with non stationary problem
  - are the model free approach

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

Monte Carlo Prediction

- Monte Carlo methods learn the state-value function for a given policy
  - Monte Carlo methods learn directly from episodes of experience
  - Monte Carlo methods have no model—unknown environment—model-free
  - No knowledge of MDP transitions and rewards, not like DP
  - Learn from complete episodes—for converging
  - Simple idea: value = average of returns
  - Each occurrence of state s in an episode is called a *visit* to s
  - The first time the state s is visited in an episode is called *first visit* to s.
  - First-visit MC is widely used in RL. This lecture focuses on it

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

First-visit MC Prediction Algorithm

- The First-visit MC prediction is for estimating $V \approx v_\pi$.
- The goal of this algorithm is learning v_π from episodes of experience under policy π, $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_t \sim \pi$
- The return G_t is the total discounted reward: $G_t = R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{T-1} R_T$
- In DP, value function is the expected return: $v_\pi(s) = E_\pi[G_t \mid S_t = s]$
- In Monte Carlo policy prediction, it uses *empirical mean* return instead of expected return.

First-visit MC Prediction Algorithm

First-visit MC prediction, for estimating $V \approx v_\pi$

Input: a policy π to be evaluated
Initialize:
  $V(s) \in \mathbb{R}$, arbitrarily, for all $s \in \mathcal{S}$
  $Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

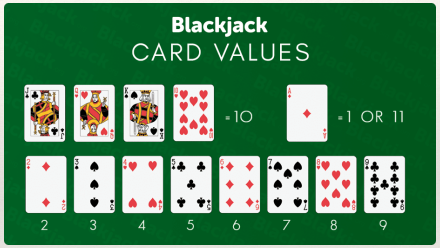
Loop forever (for each episode):
  Generate an episode following π: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$
  $G \leftarrow 0$
  Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
    $G \leftarrow \gamma G + R_{t+1}$
    Unless S_t appears in $S_0, S_1, \ldots, S_{t-1}$:
      Append G to $Returns(S_t)$
      $V(S_t) \leftarrow$ average$(Returns(S_t))$

First-visit MC Prediction Algorithm

- To evaluate state s, the first time-step t that state s is visited in an episode
- incrementally count $N(s) \leftarrow N(s) + 1$, where $N(s)$ number of times s counted.
- Increment total return $V(S_t) \leftarrow V(S_t) + G_t$, if $S_t \in S_0, S_1, \ldots, S_{t-1}$
- Return value is estimated by average $V(s) = V(s)/N(s)$
- Bay law of large numbers, $V(s) \rightarrow v)\pi(s)$ as $N(s) \rightarrow \infty$

Example of Monte Carlo First-visit Prediction

- States (200 of them):
  - Current sum (12-21)
  - Dealer's showing card
  - "used" ace? (1 or 11)
- Actions:
  - stick - stop receiving cards and terminate
  - twist: take another card (no replacement)



Blackjack card game

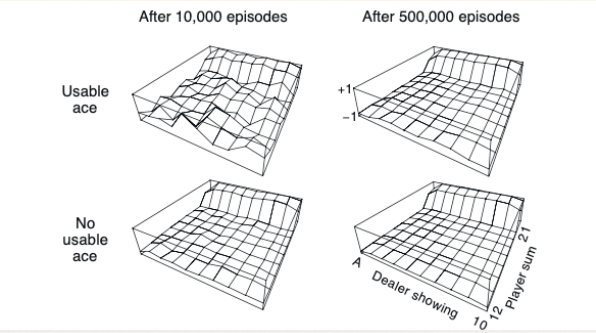
Example of Monte Carlo First-visit Prediction

- Reward for action stick:
  - +1 (win) if sum of cards > sum of dealer cards
  - 0 (draw) if sum of cards = sum of dealer cards
  - -1 (lose) if sum of cards < sum of dealer cards
- Reward for action twist:
  - -1 if sum of cards > 21 (and terminate)
  - 0 otherwise
- Transitions: automatically twist if sum of cards < 12

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

Example of Monte Carlo First-visit Prediction

- In any event, after 500,000 games the value function is very well approximated.



Approx. state-value function of the blackjack policy: stick if 20 or 21, else twist

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

Backup diagram for Monte Carlo Methods

- For Monte Carlo estimation of v_π, the root is a state node, and below it is the entire trajectory of transitions along a particular single episode, ending at the terminal state, as shown to the right.
- If DP, backup diagram shows all possible transitions, and only one step transition.

backup diagram for MC

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

Monte Carlo Estimation of Action Values

- MC has no model.
  - evaluate state value alone is insufficient for policy estimation
  - evaluate each action directly for the policy π using $q_\pi(s,a) \leftarrow q_\pi(s,a) + \frac{1}{N(s)}(G_t - q_\pi(s,a))$, in each episode.
- The first-visit MC method averages the returns following the first time in each episode that the state was visited and the action was selected.
- By the law of large number, this method converges as the number of visits to each state-action pair approaches infinity.

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

Problem of Maintaining Exploration

- Issue of MC method, particular to first-visit method, is that many state-action pairs may never be visited under deterministic π.
- Only one of the actions form each state is selected for the average returns.
- need to estimate the value of all the actions from each state, not just the one we currently favor.
- specifying that the episodes start in a state–action pair, select a pair having a nonzero probability of being selected as the start.
- This is called the assumption of *exploring starts*

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

13

Problem of Maintaining Exploration

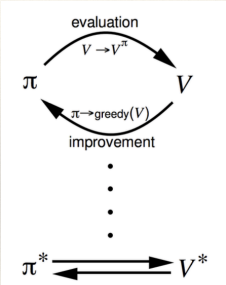
- A more general approach is to consider only policies that are stochastic with a nonzero probability of selecting all actions in each state.
- There are two important variants of this approach: on-policy and off-policy Monte Carlo control.

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

14

Monte Carlo Control

- How can Monte Carlo estimation be used in control—approximate optimal policies?
- The answer is: based on the generalized policy iteration (GPI)
- GPI maintains both an approximate policy and an approximate value function.
- The value function is repeatedly altered to more closely approximate the value function for the current policy, and the policy is repeatedly improved with respect to the current value function, as suggested by the diagram to the right.



Generalized policy iteration

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

15

Monte Carlo Control

- Both of changes work against each other until optimized.
- Policy improvement is done by this operation: $\pi(s) = \text{argmax}_a q(s, a)$
- Or, by constructing each π_{k+1}as the greedy policy with respect to q_{π_k}

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

16

Monte Carlo Control

- Based on the policy improvement theorem discussed previously, the improved policy is:

$$\begin{aligned} q_{\pi_k}(s, \pi_{k+1}(s)) &= q_{\pi_k}(s, \text{argmax}_a q_{\pi_k}(s,a)) \\ &= \max_a q_{\pi_k}(s,a) \\ &\geq q_{\pi_k}(s, \pi_k(s)) \\ &\geq v_{\pi_k}(s), \ \ \forall s \in S \end{aligned}$$

Two unlikely assumptions:

exploring starts and infinite number of episodes.

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

Monte Carlo Control

- For practice applications, two assumptions have to be removed:
  - approach one: every update of improvement is shrinking to a tiny number (a threshold) with many iterations
  - Monte Carlo ES — Monte Carlo with Exploring Starts.
- The pseudocode of Monte Carlo ES is:

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

The pseudocode of Monte Carlo ES

Monte Carlo ES (Exploring Starts), for estimating $\pi \approx \pi_*$

Initialize:
  $\pi(s) \in \mathcal{A}(s)$ (arbitrarily), for all $s \in \mathcal{S}$
  $Q(s,a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$
  $Returns(s,a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Loop forever (for each episode):
  Choose $S_0 \in \mathcal{S}$, $A_0 \in \mathcal{A}(S_0)$ randomly such that all pairs have probability > 0
  Generate an episode from S_0, A_0, following π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
  $G \leftarrow 0$
  Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
    $G \leftarrow \gamma G + R_{t+1}$
    Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:
      Append G to $Returns(S_t, A_t)$
      $Q(S_t, A_t) \leftarrow$ average$(Returns(S_t, A_t))$
      $\pi(S_t) \leftarrow \text{argmax}_a Q(S_t, a)$

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

Monte Carlo Control without Exploring Starts

- To remove the assumption of the exploring starts:
  - general approach is that ensures all actions are selected continually.
  - Two approaches: on-policy and off-policy
- On-policy methods evaluate or improve a policy that is used for making decisions
- Off-policy methods evaluate or improve a policy different from that used to generate data and made decisions.

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

On-policy control methods

- On-policy control methods is a *soft* approach, which is $\pi(a\,|\,s) > 0,\ \ \forall s \in S,\ \ \forall a \in A, \pi(a\,|\,s) \to \pi_*(a\,|\,s)$, close to optimal policy.
- The algorithm of on-policy is given as:

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

On-policy Algorithm

Monte Carlo ES (Exploring Starts), for estimating $\pi \approx \pi_*$

Initialize:
$\pi(s) \in \mathcal{A}(s)$ (arbitrarily), for all $s \in \mathcal{S}$
$Q(s,a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$
$Returns(s,a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Loop forever (for each episode):
Choose $S_0 \in \mathcal{S}$, $A_0 \in \mathcal{A}(S_0)$ randomly such that all pairs have probability > 0
Generate an episode from S_0, A_0, following π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
$G \leftarrow 0$
Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
$G \leftarrow \gamma G + R_{t+1}$
Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:
Append G to $Returns(S_t, A_t)$
$Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$
$\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$

On-policy Algorithm

- The conditions of the policy improvement theorem apply because for any state s:

$$q_\pi(s, \pi'(s)) = \sum_a \pi'(a\,|\,s) q_\pi(s,a)$$

$$= \frac{\epsilon}{|A(s)|} \sum_s q_\pi(s,a) + (1-\epsilon)\max_a q_\pi(s,a)$$

$$\geq \frac{\epsilon}{|A(s)|} \sum_s q_\pi(s,a) + (1-\epsilon) \sum_a \frac{\pi(a\,|\,s) - \frac{\epsilon}{|A(s)|}}{1-\epsilon} q_\pi(s,a)$$

- $$= \frac{\epsilon}{|A(s)|} \sum_s q_\pi(s,a) - \frac{\epsilon}{|A(s)|} \sum_s q_\pi(s,a) + \sum_a \pi(a\,|\,s) q_\pi(s,a)$$

$$= v_\pi(s)$$

- , $\forall s \in S$

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

Off-policy Prediction via Importance Sampling

- This algorithm tries to solve the problem of learning optimal policies conditional on non-optimal actions in order to satisfy the exploration desires.
- How can they learn about the optimal policy while behaving according to an exploratory policy?
- On-policy algorithm is one approach using a compromise—it learns action values not for the optimal policy, but for a near-optimal policy that still explores.
- Off-policy algorithm is another one— using two policies: *target policy* and *behaviour policy*. Target is for optimal, behaviour is for generate policy to improve.

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

Off-policy Prediction via Importance Sampling

- On-policy approach is simple and straightforward.
- Off-policy approach is complicated and often slower to converge.
- Off-policy methods are more general and powerful.
- On-policy is a special case of the off-policy, when target policy is the same as the behaviour policy.
- Off-policy methods are used widely in applications.

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

Off-policy Prediction via Importance Sampling

- Off-policy methods utilize importance sampling
  - estimate expected values by samples from another distribution
  - weighting the probability of the trajectories occurring under the target and policies,
  - $Pr\{A_t, S_{t+1}, \ldots, S_T | S_t, A_{t:T-1} \sim \pi\} = \prod_{k=t}^{T-1} \pi(A_k | S_k) p(S_{k+1} | S_k, A_k)$,
  - where p is the state-transition probability function, thus,
  - the importance sampling ratio is: $\rho_{t:T-1} = \frac{\prod_{k=t}^{T-1} \pi(A_k | S_k) p(S_{k+1} | S_k, A_k)}{\prod_{k=t}^{T-1} b(A_k | S_k) p(S_{k+1} | S_k, A_k)} = \prod_{k=t}^{T-1} \frac{\pi(A_k | S_k)}{b(A_k | S_k)}$

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

Off-policy Prediction via Importance Sampling

- Off-policy methods utilize importance sampling that the ratio $\rho_{t:T-1}$ transforms the returns to have the right expected value $E[\rho_{t:T-1} G_t | S_t = s] = v_\pi(s)$
- To estimate $v_\pi(s)$, just simply scale the returns by the importance sampling ratios and average the results: $V(s) = \frac{\sum_{t \in J(s)} \rho_{t:T(t)-1} G_t}{|J(s)|}$, where T(t) denote the first time of termination following time t, and Gt denote the return after t up to T(t).

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

Off-policy Prediction via Importance Sampling

- Ordinary importance sampling: $V(s) = \frac{\sum_{t \in J(s)} \rho_{t:T(t)-1} G_t}{|J(s)|}$
- Weighted importance sampling: $V(s) = \frac{\sum_{t \in J(s)} \rho_{t:T(t)-1} G_t}{\sum_{t \in J(s)} \rho_{t:T(t)-1}}$

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

Off-policy Incremental Implementation

- In weighted importance sampling: $V(s) = \dfrac{\sum_{t\in J(s)} \rho_{t:T(t)-1} G_t}{\sum_{t\in J(s)} \rho_{t:T(t)-1}}$
- Given $G_1, G_2, \ldots G_{n-1}$, W_i $(W_i = \rho_{t_i:T(t_i)-1})$,
- $V_n = \dfrac{\sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k}, n \geq 2,$
- apply incremental approach,

  $V_{n+1} = V_n + \dfrac{W_n}{C_n}[G_n - V_n], n \geq 1,$ and $C_{n+1} = C_n + W_{n+1}, C_0 = 0$

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

Off-policy Incremental Implementation

Off-policy MC prediction (policy evaluation) for estimating $Q \approx q_\pi$

Input: an arbitrary target policy π
Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
  $Q(s,a) \in \mathbb{R}$ (arbitrarily)
  $C(s,a) \leftarrow 0$

Loop forever (for each episode):
  $b \leftarrow$ any policy with coverage of π
  Generate an episode following b: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
  $G \leftarrow 0$
  $W \leftarrow 1$
  Loop for each step of episode, $t = T-1, T-2, \ldots, 0$, while $W \neq 0$:
    $G \leftarrow \gamma G + R_{t+1}$
    $C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
    $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)}[G - Q(S_t, A_t)]$
    $W \leftarrow W \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$

Off-policy Monte Carlo Control

- The second approach of the Monte Carlo Control is for off-policy.
- The behaviour policy has a nonzero probability of selecting all actions for exploration.
- The target policy is deterministic.
- Based on GPI and weighted importance sampling, estimate π_* and q_π.

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

Off-policy Monte Carlo Control

Off-policy MC control, for estimating $\pi \approx \pi_*$

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
  $Q(s,a) \in \mathbb{R}$ (arbitrarily)
  $C(s,a) \leftarrow 0$
  $\pi(s) \leftarrow \text{argmax}_a Q(s,a)$ (with ties broken consistently)

Loop forever (for each episode):
  $b \leftarrow$ any soft policy
  Generate an episode using b: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
  $G \leftarrow 0$
  $W \leftarrow 1$
  Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
    $G \leftarrow \gamma G + R_{t+1}$
    $C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
    $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)}[G - Q(S_t, A_t)]$
    $\pi(S_t) \leftarrow \text{argmax}_a Q(S_t, a)$ (with ties broken consistently)
    If $A_t \neq \pi(S_t)$ then exit inner Loop (proceed to next episode)
    $W \leftarrow W \frac{1}{b(A_t|S_t)}$

Recap

- Monte Carlo Methods are learning methods for estimating value function and discovering optimal policies, without knowing environment, only based on experience by sampling sequences of states, then average sample returns
- Not like the DP, which needs knowing the environment
- In this lecture, the MC is only for episodic tasks - experience is divided into episodes.
- MC methods deal with non stationary problem
- They are the model free approach

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

33

33

Recap

- Monte Carlo Methods learn the state-value function for a given policy
- First-visit MC is widely used in RL. This lecture focuses on it.
- The First-visit MC prediction is for estimating $V \approx v_\pi$
- Examples include Blackjack card games
- There are two unlikely assumptions: exploring starts and infinite number of episodes
- To release these assumptions, adapt on-policy and off-policy methods for Monte Carlo Control.
- MC controls follows the generalized policy iteration (GPI).

yguangbing@gmail.com, Guang.B@chula.ac.th Mar 16 2021

34

34

Questions and Lab

35

35