IIIT Hyderabad **Scribed By: Rehas Mehar Kaur Sachdeva**
- **Sairam Kolla**
Topics in Machine Learning (CSE975) **Lecture #17**
Instructor: Dr. Naresh Manwani **October 13, 2017**

Outline. *First Visit and Every Visit Monte Carlo Policy Evaluation, Monte Carlo Estimation of Action Values, Monte Carlo Control with assumption of Exploring Starts, Monte Carlo Control without Exploring Starts using On Policy and Off-Policy approaches.*

1 Monte Carlo Methods

Monte Carlo (MC) methods are learning methods, used for estimating value functions and discovering optimal policies. Unlike Dynamic Programming (DP) method, here we do not assume complete knowledge of the environment but learn from on-line experience of sample sequences of states, actions, and rewards. No model is necessary to achieve optimality as in we do not need complete probability distributions of all possible transitions that is required by DP methods.

Monte Carlo methods are based on averaging sample returns. We assume that the experience is divided into episodes. After every episode, we receive the return corresponding to it. It is only upon the completion of an episode that value estimates and policies are changed. Monte Carlo methods are thus incremental in an episode-by-episode sense, but not in a step-by-step sense.

2 Monte Carlo Policy Evaluation

Let π be a policy. Then the goal here is to estimate $v_\pi(s)$, the state value i.e. the expected return or the expected cumulative future discounted reward starting from that state, $\forall s \in S$, under the policy π. We are given some number of episodes under π which contain s. If each occurrence of a state in an episode be called a visit, then the idea here is to average returns observed after visits to s. By the law of large numbers, this average converges asymptotically to the expected value. We can categorize these approaches in two broad categories.

2.1 First Visit Monte Carlo Policy Evaluation

The first-visit MC method averages the returns following first visit to state s in an episode.

We generate m episodes using policy π which observe state s at some time step. Let i^{th} run (episode) is denoted by T_i. For each run T_i and state s in it, let $r(s, T_i)$ be the return of π in run T_i from the first appearance of s in T_i until the run ends (reaching terminal state). Then the estimate of $v_\pi(s)$ under π is,

$$\hat{v}_\pi(s) = \frac{1}{m} \sum_{i=1}^{m} r(s, T_i) \tag{1}$$

Note that the random variable $r(s, T_i)$ for a given state s and different T_i 's are independent since each episodes is generated independently from the other episodes.

- **Example - First Visit Monte Carlo Policy Evaluation**

  Consider an MDP as follows:

  - $\mathcal{S} = \{s_1, s_2, s_3\}$. Episodes can start either at s_1 or at s_2. s_3 is terminal state.
  - Transition probabilities, discount rate and the rewards are as follows.
    - $p(s_2|s_1, \pi(s_1)) = 1, r(s_1, \pi(s_1), s_2) = 1$
    - $p(s_3|s_2, \pi(s_2)) = 1, r(s_2, \pi(s_2), s_3) \in \{?3, 4\}$
    - $\gamma = 1$

  In its first 40 episodes, the following trajectories have been observed by the agent:

  - $s_1, \pi(s_1), 1, s_2, \pi(s_2), ?3, s_3$ — 15 times
  - $s_1, \pi(s_1), 1, s_2, \pi(s_2), 4, s_3$ — 5 times
  - $s_2, \pi(s_2), ?3, s_3$ — 10 times
  - $s_2, \pi(s_2), 4, s_3$ — 10 times

  Computation of $\hat{v}_\pi(s_1)$: There are 20 trajectories passing through s_1. Total reward accrued subsequent to passing s_1, is

  $$(1-3) \times 15 + (1+4) \times 5 = -5.$$

  Hence, the Monte Carlo estimate $\hat{v}_\pi(s_1) = \frac{?5}{20} = \frac{?1}{4}$.

  Computation of $\hat{v}_\pi(s_2)$: There are 40 trajectories passing through s_2. Total reward accrued subsequent to passing s_2, is

  $$-3 \times 15 + 4 \times 5 - 3 \times 10 + 4 \times 10 = -15.$$

  Hence, the Monte Carlo estimate $\hat{v}_\pi(s_2) = \frac{-15}{40} = \frac{-5}{8}$.

2.2 Every Visit Monte Carlo Policy Evaluation

The every-visit MC method averages the returns following every visit to state s in an episode. Let m be the number of episodes generated using policy π in which state s is observed at least once. Let i^{th} run (episode) is denoted by T_i. For each run T_i and state s in it, let $r(s, T_i, j)$ be the return from the j^{th} appearance of s in T_i. Let $N_i(s)$ be the number of times state s has been visited in the run T_i. Then the estimate of $v_\pi(s)$ under π is,

$$\hat{v}_\pi(s) = \frac{1}{\sum_{i=1}^{m} N_i(s)} \sum_{i=1}^{m} \sum_{j=1}^{N_i(s)} r(s, T_i, j) \tag{2}$$

Note that the random variable $r(s, T_i, j)$ for a given state s and different T_i 's are dependent for different j.

3 Monte Carlo Estimation of Action Values

If a model is not available, then it is particularly useful to estimate action values rather than state values. With a model, state values alone are sufficient to determine a policy; one simply looks ahead one step and chooses whichever action leads to the best combination of reward and next state, as in the DP method. Without a model, however, state values alone are not sufficient. One must explicitly estimate the value of each action in order for the values to be useful in suggesting a policy.

So we want to learn q_*. The policy evaluation problem for action values is to estimate $q_\pi(s, a)$, the expected return when starting in state s, taking action a, and thereafter following policy π. We can extend the first visit and every visit approaches for state value estimation to state-action value estimation. The only complication is that many relevant state-action pairs may never be visited. If π is a deterministic policy, then in following π one will observe returns only for one of the actions from each state. With no returns to average, the Monte Carlo estimates of the other actions will not improve with experience.

Hence we need to maintain exploration, as discussed in the n-armed bandit problem. One way maintain exploration is by specifying that the first step of each episode starts at a state-action pair, and that every such pair has a nonzero probability of being selected as the start. We call this the assumption of *exploring starts*. Also, we need to visit each state action pair infinitely often.

4 Monte Carlo Control

We use Monte Carlo estimation to approximate optimal policies. The overall idea is same as in DP method, that is to maintain 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 (Generalized policy iteration), as shown.

$$\pi_0 \longrightarrow q_{\pi_0} \longrightarrow \pi_1 \longrightarrow q_{\pi_1} \longrightarrow ... \longrightarrow \pi_* \longrightarrow q_{\pi_*} \tag{3}$$

Note that we make two assumptions here,

- We observe an infinite number of episodes.
- Episodes are generated with exploring starts.

Policy Improvement Step: Policy improvement is done by making the policy greedy with respect to the current value function.

$$\pi_{k+1}(s) = \arg\max_a q_{\pi_k}(s, a)$$

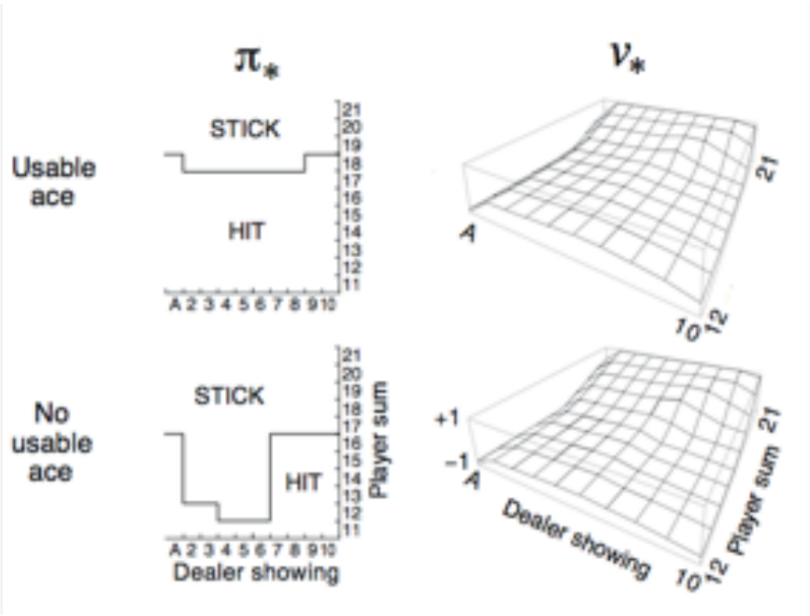
This requires $q_{\pi_k}(s, a)$ values for every (s, a) pair. But π_k itself is a greedy policy, due to which we will only get the values of $q_{\pi_k}(s, \pi_k(s)),\ \forall s \in \mathcal{S}$ using sampling.

Algorithm 1 Monte Carlo ES: A Monte Carlo control algorithm assuming exploring starts.

Initialize for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$:
    $q(s, a)$ arbitrary
    $\pi(s)$ arbitrary
    $Returns(s, a)$ empty list
Repeat forever:
    Choose $s_0 \in \mathcal{S}$ and $a_0 \in \mathcal{A}(s_0)$ s.t. all pairs have probability greater than 0.
    Generate an episode starting from s_0, a_0 following π.
    For each pair s, a appearing in the episode:
        $G =$ return following the first occurrence of s, a.
        Append G to $Returns(s, a)$.
        $q(s, a) = average(Returns(s, a))$.
    For each s in the episode:
        $\pi(s) = \arg\max_a q(s, a)$.

Exploring Starts Example : Black Jack

Since the episodes are all simulated games, it is easy to arrange for exploring starts that include all possibilities. In this case one simply picks the dealer's cards, the player's sum, and whether or not the player has a usable ace, all at random with equal probability. As the initial policy we use the policy that sticks only on 20 or 21. The initial action-value function can simply be zero for all state-action pairs. Below figure shows the optimal policy for blackjack found by Monte Carlo ES.



5 Removing the assumption of infinitely many episodes

We need infinitely many episodes from π to approximate the state values. However, we can relax this assumption with the help of Hoeffding's inequality as follows.

$$Pr\left[|v_\pi(s) - \hat{v}_\pi(s)| > \epsilon\right] \leq 2e^{-2\epsilon^2 n}$$

where $\hat{v}_\pi(s) = \frac{1}{n}\sum_{i=1}^{n} G_i(s)$ and $G_i(s)$ is the return of ith episode after observing s. Thus we can find the value of n such that the probability above is below some predefined threshold.

6 Monte Carlo Control without Exploring Starts

We can remove the assumption of exploring starts by ensuring that all state-actions are selected sufficiently many times. This can be done in two ways,

- **On Policy Method:** Evaluate and improve the same policy which is being used for the exploration/data generation.
- **Off Policy Method:** One policy is used for exploration called behavior policy. The policy being learned is called target policy.

6.1 On-Policy Monte Carlo Control

In on-policy control methods the policy being used is soft (an ϵ-soft policy π is such that $\pi(a \mid s) > 0,\ \forall s \in \mathcal{S},\ \forall a \in \mathcal{A}(s)$.

ϵ-greedy approach: In this approach, we chose the action that has maximal estimated action value with probability $(1-\epsilon)$ and with probability ϵ we instead select an action at random. That is, all non-greedy actions are given the minimal probability of selection, $\frac{\epsilon}{|\mathcal{A}(s)|}$, and the remaining bulk of the probability, $1 - \epsilon + \frac{\epsilon}{|\mathcal{A}(s)|}$, is given to the greedy action. The ϵ-greedy policies are examples of ϵ-soft policies. The complete algorithm for on-policy first visit Monte Carlo control is described in Algorithm **??**.

We now describe the policy improvement theorem for the algorithm described.

Policy improvement theorem: It assures that any ϵ-greedy policy by this method, (denote by π') is an improvement over any ϵ-soft policy, (denote by π).

$$\begin{aligned} q_\pi(s, \pi'(s)) &= \sum_a \pi'(a|s) q_\pi(s,a) \\ &= \frac{\epsilon}{\mid \mathcal{A}(s) \mid} \sum_a q_\pi(s,a) + (1-\epsilon) \max_a q_\pi(s,a) \\ &\geq \frac{\epsilon}{\mid \mathcal{A}(s) \mid} \sum_a q_\pi(s,a) + (1-\epsilon) q_\pi(s,a) \end{aligned}$$

Algorithm 2 On Policy First Visit Monte Carlo Control (for ϵ-soft policies)

Initialize for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$:
  $q(s,a)$ arbitrary
  $\pi(a \mid s)$ an arbitrary ϵ-soft policy.
  $Returns(s,a)$ empty list
Repeat forever:
  Generate an episode using π.
  For each pair s,a appearing in the episode:
    G = return following the first occurrence of s,a.
    Append G to $Returns(s,a)$.
    $q(s,a) = average(Returns(s,a))$.
  For each s in the episode:
    $A^* = \arg\max_a \ q(s,a)$.
    For all $a \in \mathcal{A}(s)$:

$$\pi(a \mid s) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|\mathcal{A}(s)|} & \text{if } a = A^* \\ \frac{\epsilon}{|\mathcal{A}(s)|} & \text{if } a \neq A^* \end{cases}$$

We notice that $\sum_a \frac{\pi(a|s) - \frac{\epsilon}{|\mathcal{A}(s)|}}{1-\epsilon} = \frac{1}{(1-\epsilon)}(\sum_a \pi(a|s) - \frac{\epsilon}{|\mathcal{A}(s)|}\sum_a 1) = \frac{1}{(1-\epsilon)}(1-\epsilon) = 1$. Thus,

$$\begin{aligned} q_\pi(s, \pi'(s)) &\geq \frac{\epsilon}{\mid \mathcal{A}(s) \mid} \sum_a q_\pi(s,a) + (1-\epsilon) \sum_a \frac{\pi(a|s) - \frac{\epsilon}{|\mathcal{A}(s)|}}{1-\epsilon} q_\pi(s,a) \\ &= \frac{\epsilon}{\mid \mathcal{A}(s) \mid} \sum_a q_\pi(s,a) - \frac{\epsilon}{\mid \mathcal{A}(s) \mid} \sum_a q_\pi(s,a) + \sum_a \pi(a|s) q_\pi(s,a) \\ &= v_\pi(s) \end{aligned}$$

6.2 Off-Policy Monte Carlo Control

As described earlier, the problem in Monte Carlo control is that for a given deterministic policy, we can't get the action values for all the (s,a) pairs. However, this problem can be resolved with the help of importance sampling. We quickly describe the importance sampling idea.

Importance Sampling: Let x be a random variable and $q(x)$ be a probability distribution on x. We want to find $\mathbb{E}_{x\sim q}[x]$. But, we can't simulate $q(x)$ and hence can't get the estimate of expected value of x under q. However, we have the access to another distribution $p(.)$ on x. We can generate samples from the distribution p. To estimate $\mathbb{E}_{x\sim q}[x]$, we use the following trick. Let $\hat{x} = \frac{xq(x)}{p(x)}$ be another random variable. We see that

$$\mathbb{E}_{x\sim p}[\hat{x}] = \sum_x p(x)\frac{xq(x)}{p(x)} = \sum_x xq(x) = \mathbb{E}_{x\sim q}[x]$$

Thus, we can see that the expected value of $\hat{x}$ under p is same as the expected value of x under q.

This is exactly the problem we are facing here. We want to find the expected value of return (v_π or q_π)under a policy π (called target policy) but we can't sample return values for every (s, a) pair under pi. Let us assume that we have the access to μ ($\mu \neq \pi$) which is a behavior policy. μ is such that $\mu(a|s) > 0, \forall s \in \mathcal{S}, \forall a \in \mathcal{A}(s)$. Thus, we can generate episodes following μ. As an example, suppose you want to test your new Black Jack strategy, but as poor student you don?t have enough money to play at the casino. Off-policy methods allow you to assess your strategy observing other players.

Assumption of coverage: Every action which is taken under policy π must have a non-zero probability to be taken as well under policy μ. i.e. $\pi(a \mid s) > 0 \implies \mu(a \mid s) > 0$.

Typically the target policy π is a greedy (deterministic) policy with respect to the current action-value function. Given a state S_t , the probability of a subsequent state-action trajectory $A_t, S_{t+1}, A_{t+1}, ..., S_T$ occurring under policy π is

$$P_\pi(S_t, A_t, ..., S_T) = \Pi_{k=t}^{T-1}\pi(A_k \mid S_k)p(S_{k+1} \mid S_k, A_k) \tag{4}$$

where p is the state-transition probability. The relative probability of the trajectory under the target and behavior policies, or the importance sampling ratio, is:

$$\rho_t^T = \frac{\Pi_{k=t}^{T-1}\pi(A_k \mid S_k)p(S_{k+1} \mid S_k, A_k)}{\Pi_{k=t}^{T-1}\mu(A_k \mid S_k)p(S_{k+1} \mid S_k, A_k)} \tag{5}$$

Note that the trajectory probabilities p depends on the MDP, and are generally unknown. Let $\tau(s)$ denote the set of all time steps in which state s is visited in an episode for every-visit method. For first visit Monte Carlo $\tau(s)$ is the time of first visit to s in an episode. Let $T(t)$ denote the first time of termination following time t, and G_t denote the return after t upto $T(t)$. Then $\{G_t\}_{t\in\tau(s)}$ are returns corresponding to state s. $\{\rho_t^{T(t)}\}_{t\in\tau(s)}$ are the corresponding importance sampling ratios.

Ordinary importance sampling estimates $v_\pi(s)$ by scaling the returns by the number of times we visited state s.

$$\hat{v}_\pi(s) = \frac{\sum_{t\in\tau(s)}\rho_t^T(t)G_t}{\mid \tau(s) \mid} \tag{6}$$

References

[1] Sutton, Richard S., and Andrew G. Barto, *Reinforcement learning: An introduction* Vol. 1. No. 1. Cambridge: MIT press, 1998.