IIIT Hyderabad **Scribed By: Ritabrata Podder, M Siva Durga Prasad**
Topics in Machine Learning (CS975) **Lecture #8**
Instructor: Dr. Naresh Manwani **August 23, 2019**

Outline *In this lecture we will cover Monte Carlo Methods.Monte Carlo methods are learning methods, used for estimating value functions and discovering optimal policies.Monte Carlo methods are ways of solving the reinforcement learning problem based on averaging sample returns. To ensure that well-defined returns are available, here we define Monte Carlo methods only for episodic tasks. That is, we assume experience is divided into episodes, and that all episodes eventually terminate no matter what actions are selected*

Contents

1 Naive Approach:

For each s S, run from s for m times, where the i^{th} run episode is E_i

Let i^{th} episode 0 E_i terminates at T_i.Thus,

$$E_i = S_0, A_0, R_1, S_1, A_1, R_2.....S_T \tag{1}$$

Let G_i be the return of E_i.Thus

Estimate the value of policy starting from s, by The variables G_i are independent since the runs E_i are independent

2 MC Approaches

- First Visit MC-Average returns for first time s visited
- Every Visit MC-Averages the returns following all visits to s

2.1 First Visit MC Policy Evaluation:

- Run π from s for m times, where the i^{th} run episode is E_i
- For each run E_i and state s in it. Let $G(s, E_i)$ be the return of π in run E_i from first appearance of s in E_i until the run ends (reaching T state).
- Let the value of state s under policy π

$$\hat{V}_\pi(s) = \frac{1}{m} \sum_{i=1}^{m} G(s, E_i) \tag{2}$$

- The random variable $G(s, E_i)$ for a given state s and different E_{i^s} are independent since different runs (episodes) are independent, since different runs are independent.

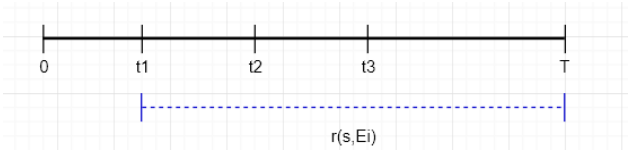

Figure 1: Return in first visit MC

2.2 Every Visit MC Policy Evaluation:

- Run π from s for m times, where the i^{th} run (episode i is E_i).

- For each run E_i and state s in it. Let $G(s, E_i)$ be the return of π in run E_i from the j^{th} appearance of s in E_i.
- Let $N_i(s)$ be the number of times state s has been visited in the run E_i
- Let the value of state s under policy π

$$\hat{V}_\pi(s) = \frac{1}{\sum_{i=1}^{m} N_i(s)} \sum_{i=1}^{m} \sum_{i=1}^{N_i(s)} G(s, E_i, j) \tag{3}$$

The random variable $G(s, E_i, j$ for a given state s and E_{i^s} are independent. Since different run are independent

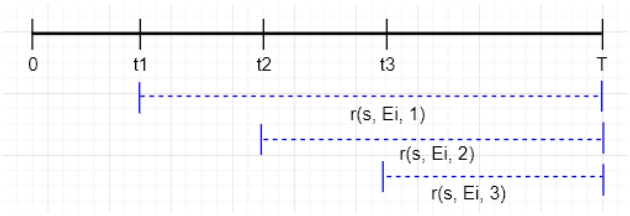

Figure 2: Return in every visit MC

- If S1 is visited at t = 2,5,10,18,....

$G(s1, E_i, 1)$ = R3 + R4 + ...

$G(s1, E_i, 2)$ = R6 + R7 + ...

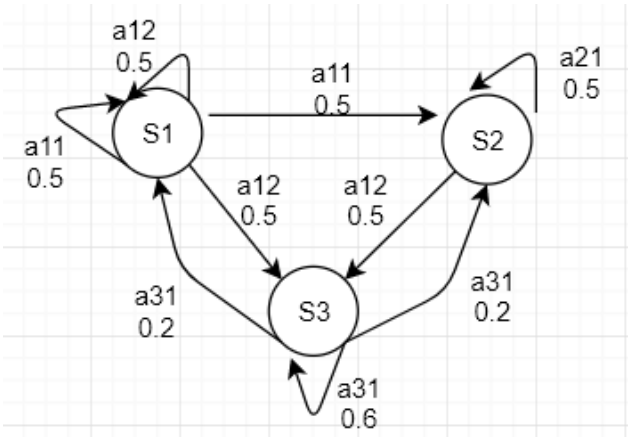

Figure 3: Example

$$Let : S_0 = s_2, \quad S_1 = s_3, \quad S_2 = s_1$$

$$S_0A_0R_1S_1A_1R_2S_2A_2R_3S_3A_3R_4S_4A_4R_5\ldots$$

$$G(s_1, E_i) = R_3 + \gamma R_4 + \ldots$$

$$G(s_2, E_i) = R_1 + \gamma R_2 + \ldots$$

$$G(s_3, E_i) = R_2 + \gamma R_3 + \ldots$$

3 Dynamic Programming v/s Monte Carlo

3.1 Dynamic Programming (DP)

- We need full knowledge of environment.

$$P_r\{R_{t+1} = r, S_{t+1} = s' | S_t = s, A_t = a\} \quad \forall s \quad s' \in S, \forall a \quad \in A(s) \quad \forall r\} \tag{4}$$

- All of the expected rewards and transition probabilities must be computed before DP can be applied.
- Its complex and error prone.

3.2 Monte Carlo (MC)

- Here generating samples is easy
- MC methods can be better even if complete knowledge of environment dynamics not known
- State value estimates for each state are independent
- The estimate of one state does not depend upon estimate of any other state
- MC methods do not bootstrap
- Computational expense of estimating the value of a single state is independent of number of states
- One can generate the episodes starting from state of interest,averaging the returns from only these states ignoring others

4 MC estimation of Action Values

MC is most useful when a model is not available, then it is particularly useful to estimate action values (the values of state–action pairs) 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. 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. Thus, one of our primary goals for Monte Carlo methods is to estimate q_*

It estimates $q_\pi(s, a)$ the expected return starting from state s, taking action a, then following policy π

5 MC Control

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. In this method, we perform alternating complete steps of policy evaluation and policy improvement, beginning with an arbitrary policy π and ending with the optimal policy and optimal action-value

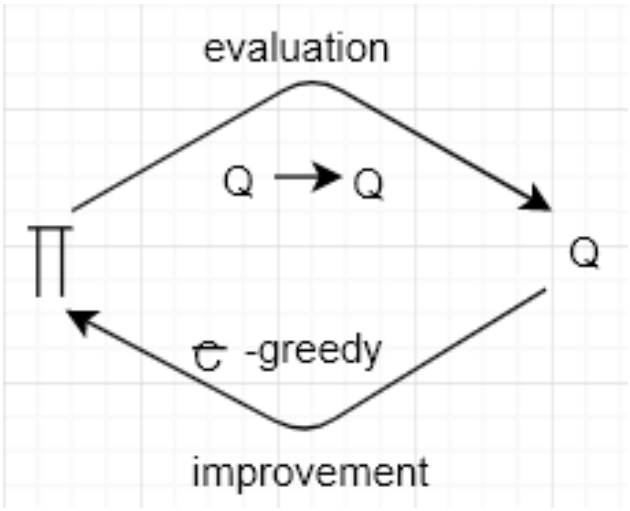

Figure 4: Generalized policy iteration

function:

$$\pi_0 \to \pi_1 \to q_{\pi_2} \to \dots \pi_* \to q_* \tag{5}$$

Policy improvement is done by making the policy greedy with respect to the current value function. In this case we have an action-value function, and therefore no model is needed to construct the greedy policy. For any action- value function q, the corresponding greedy policy is the one that, $\forall s \in S$, deterministic-ally chooses an action with maximal action-value:

$$\pi(s) = \arg\max_a q_{\pi_k}(s, a) \tag{6}$$

5.1 There are 2 possibilities here

- On Policy: Estimate and improve the same policy which is being used for the exploration/data generation.

- Off Policy: One policy is used for exploration called behaviour policy.The policy being learned is called target policy.

In the next lecture we will go with On-Policy and Off-Policy MC controls and important sampling

The recommended textbooks for the course are Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction [1] and Masashi Sugiyama - Statistical reinforcement learning: modern machine learning approaches [2]. Cite as demonstrated in this document if you take any content verbatim, optionally using the `main.bib` file for new sources.

References

[1] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[2] Masashi Sugiyama. *Statistical reinforcement learning: modern machine learning approaches*. Chapman and Hall/CRC, 2015.