Model Free Prediction : Monte Carlo Methods

Easwar Subramanian

TCS Innovation Labs, Hyderabad

Email : easwar.subramanian@tcs.com / cs5500.2020@iith.ac.in

September 13, 2021

Overview

Review

Policy Evaluation : Prediction

Given a Markov decision process is a tuple $< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$ and a policy π, we have,

- State value function of policy π:

$$V^\pi(s) \stackrel{\text{def}}{=} \mathbb{E}_\pi(G_t | s_t = s) = \mathbb{E}_\pi\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right)$$

- Bellman evaluation equation:

$$V^\pi(s) \quad = \quad \mathbb{E}_\pi\left[r_{t+1} + \gamma V^\pi(s_{t+1}) | s_t = s\right] = \sum_a \pi(a|s) \sum_{s'} \mathcal{P}_{ss'}^a \left[\mathcal{R}_{ss'}^a + \gamma V^\pi(s')\right]$$

- Iterative policy evaluation:

$$V_{k+1}(s) \quad \leftarrow \quad \sum_a \pi(a|s) \sum_{s'} \mathcal{P}_{ss'}^a \left[\mathcal{R}_{ss'}^a + \gamma V_k(s')\right]$$

$$V_k(s) \rightarrow V^\pi(s)$$

Finding Optimal Policy : Control

- Bellman Optimality Equation

$$V_*(s) = \max_\pi \left[\mathbb{E}_\pi \left[r_{t+1} + \gamma V^\pi(s_{t+1}) | s_t = s\right]\right] = \max_a \left[\sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma V_*(s')\right)\right]$$

- Value Iteration

$$V_{k+1}(s) \leftarrow \max_a \left[\sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma V_k(s')\right)\right]$$

$$V_k(s) \rightarrow V_*(s)$$

- Policy Iteration
  - ★ Policy evaluation for policy π_k (k-th iteration)
  - ★ Policy Improvement $\pi_{k+1} \leftarrow \text{greedy}(\pi_k)$

$$\pi_k(s) \rightarrow \pi_*(s)$$

Convergence Aspects

- Iterative policy evaluation converges to V_π
- Value iteration converges to V^*
- And therefore that policy iteration converges to π_*
- Rate of convergence of these algorithms depends on the discount factor γ
- **Contraction mapping theorem** or **Banach fixed point theorem** are used in resolvimg these issues

DP Algorithms : A Closer Look

DP Algorithms : Terminology

Let us consider the policy evaluation formula

$$V_{k+1}(s) \leftarrow \sum_{a} \pi(s,a) \sum_{s'} \mathcal{P}^{a}_{ss'} \left[\mathcal{R}^{a}_{ss'} + \gamma V_k(s')\right]$$

DP Algorithms : Terminology

Let us consider the policy evaluation formula

One-step lookahead

$$V_{k+1}(s) \leftarrow \sum_{a} \pi(s,a) \sum_{s'} \mathcal{P}^{a}_{ss'} \left[\mathcal{R}^{a}_{ss'} + \gamma V_k(s')\right]$$

DP Algorithms : Terminology

Let us consider the policy evaluation formula

One-step lookahead

$$V_{k+1}(s) \leftarrow \sum_{a} \pi(s, a) \sum_{s'} \mathcal{P}^{a}_{ss'} \left[\mathcal{R}^{a}_{ss'} + \gamma V_k(s')\right]$$

Backup

DP Algorithms : Terminology

Let us consider the policy evaluation formula

One-step lookahead

$$V_{k+1}(s) \leftarrow \sum_a \pi(s, a) \boxed{\sum_{s'}} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma V_k(s')\right]$$

Full backup

Backup

DP Algorithms : Terminology

Let us consider the policy evaluation formula

Bootstrap

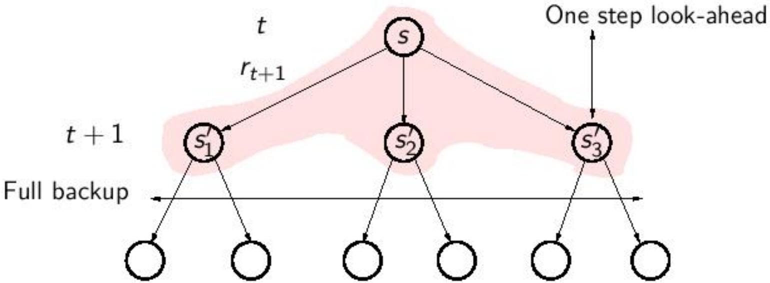
One-step lookahead

$$V_{k+1}(s) \leftarrow \sum_a \pi(s,a) \boxed{\sum_{s'}} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma V_k(s')\right]$$

Full backup

Backup

DP Algorithms : Schematic View



$$V^{\pi}(s) = \sum_{a} \pi(a|s) \sum_{s'} \mathcal{P}^{a}_{ss'} \left[\mathcal{R}^{a}_{ss'} + \gamma V^{\pi}(s') \right]$$

$$V_{k+1}(s) \leftarrow \sum_{a} \pi(a|s) \sum_{s'} \mathcal{P}^{a}_{ss'} \left[\mathcal{R}^{a}_{ss'} + \gamma V_{k}(s') \right]$$

Drawbacks of DP Algorithms

- Requires full prior knowledge of the dynamics of the environment
- Can be implemented only on small or medium sized discrete state spaces
  - ★ For large problems, DP suffers from Bellman's curse of dimensionality
- DP uses full width back-ups
  - ★ Every successor state and action is considered

Key Idea

$$
\begin{aligned}
V^\pi(s) &\stackrel{\text{def}}{=} \mathbb{E}_\pi(G_t|s_t = s) = \mathbb{E}_\pi\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|s_t = s\right) \\
&= \mathbb{E}_\pi\left[r_{t+1} + \gamma V^\pi(s_{t+1})|s_t = s\right]
\end{aligned}
$$

How can we estimate the expectations?

Use samples!

Approximate Methods: Monte Carlo Algorithms

Monte Carlo Policy Evaluation

- Goal : Evaluate $V^{\pi}(s)$ using experiences (or trajectories) under policy π

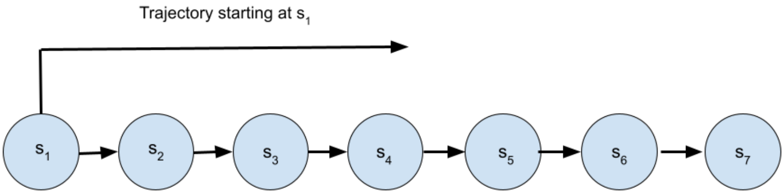
$$s_0, a_0, r_1, s_1, a_1, r_2, s_3, \cdots, s_T$$

- Recall that

$$V^{\pi}(s) = \mathbb{E}_{\pi}(G_t | s_t = s) = \mathbb{E}_{\pi}\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right)$$

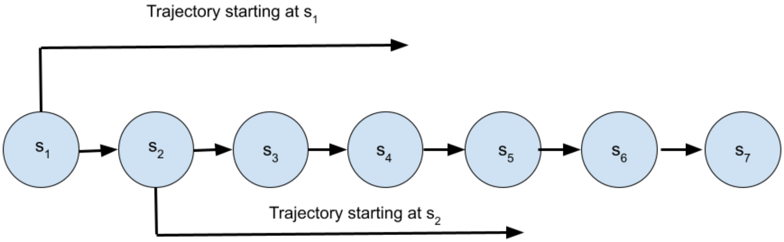
- The idea is to calculate **sample** mean return (G_t) starting from state s instead of expected mean return

Monte Carlo Evalution : Schematics



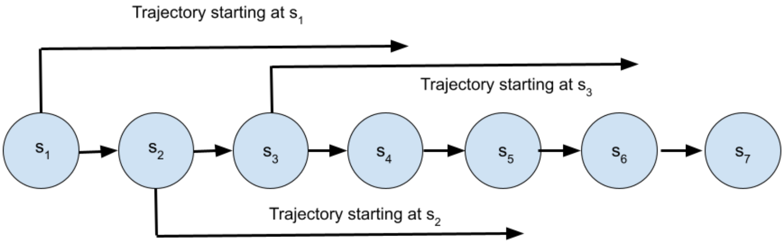
- Use G_1 to update $V^\pi(s_1)$

Monte Carlo Evalution : Schematics



- Use G_1 to update $V^\pi(s_1)$
- Use G_2 to update $V^\pi(s_2)$

Monte Carlo Evalution : Schematics



- Use G_1 to update $V^\pi(s_1)$
- Use G_2 to update $V^\pi(s_2)$
- Use G_3 to update $V^\pi(s_3)$

First-visit Monte Carlo Policy Evaluation

- To evaluate $V^{\pi}(s)$ for some given state s, repeat over several episodes
  - ★ The first time t that $s_t = s$ in the episode
    1. Increment counter for number of visits to s: $N(s) \leftarrow N(s) + 1$
    2. Increment running sum of total returns with return from current episode: $S(s) \leftarrow S(s) + G_t$
- Monte Carlo estimate of value function $V(s) \leftarrow S(s)/N(s)$

By the law of large numbers $V(s) \rightarrow V^{\pi}(s)$ as number of episodes increases

Every-visit Monte Carlo Policy Evaluation

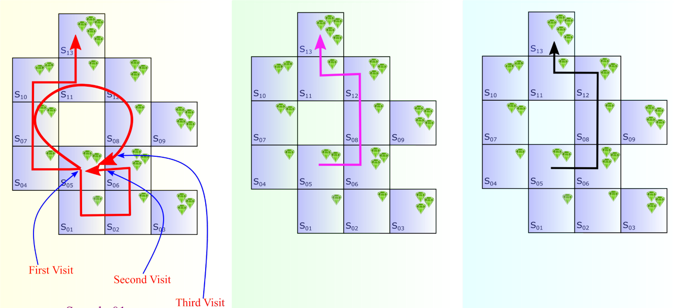
- To evaluate $V^{\pi}(s)$ for some given state s, repeat over several episodes
  - ★ Every time t that $s_t = s$ in the episode
    1. Increment counter for number of visits to s: $N(s) \leftarrow N(s) + 1$
    2. Increment running sum of total returns with return from current episode: $S(s) \leftarrow S(s) + G_t$
- Monte Carlo estimate of value function $V(s) \leftarrow S(s)/N(s)$

By the law of large numbers $V(s) \rightarrow V^{\pi}(s)$ as number of episodes increases

Monte Carlo Method : Example

- Consider an MDP with two states $\mathcal{S} = \{A, B\}$ with $\gamma = 1$
- $\mathcal{P}$ and $\mathcal{R}$ are unknown
- Consider a policy π that gives rise to following state-reward sequence
  - ★ $A(+3), A(+2), B(-4), A(+4), B(-3)$
  - ★ $B(-2), A(+3), B(-3)$
- What is $V^\pi(A)$ and $V^\pi(B)$ if we use first visit MC and every visit MC respectively ?
- First visit MC : $V(A) = \frac{1}{2}(2+0) = 1$; $V(B) = \frac{1}{2}(-3-2) = -5/2$
- Every visit MC : $V(A) = \frac{1}{4}(2-1+1+0) = 1/2$; $V(B) = \frac{1}{4}(-3-3-3-2) = -11/4$

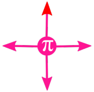
First Visit Monte Carlo Method : Example



Sample 01
Return for the state S_{05} = 33 gems
$N(S_{05})$ = 1 (only first visit is counted)

Sample 02
Return for the state S_{05} = 15 gems
$N(S_{05})$ = 2 (added 1 in this episode)

Sample 03
Return for the state S_{05} = 15 gems
$N(S_{05})$ = 3 (added 1 in this episode)

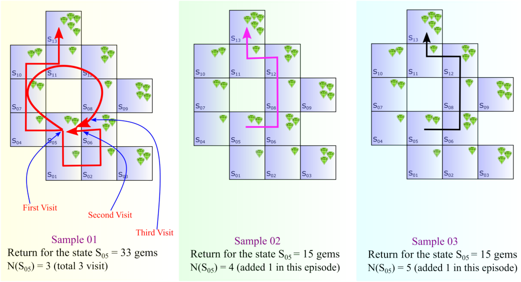


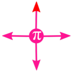
$$v(S_{05}) = (33 + 15 + 15) / 3 = 21 \text{ gems}$$

First Visit Monte Carlo State Value Evaluation of State S_{05}

Figure Source: Baijayanta Roy : Towards Data Science

Every Visit Monte Carlo Method : Example





$v(S_{05}) = (33 + 15 + 15) / 5 = 12.6$ gems

Every Visit Monte Carlo State Value Evaluation of State S_{05}

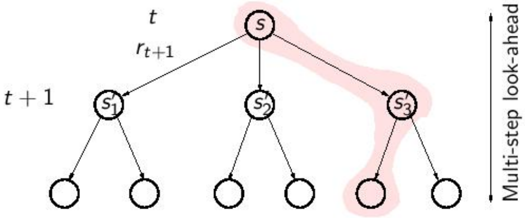
- The purpose of this example (which is borrowed from a blogpost) is to reinforce that the two MC algorithms can give different results on same trajectories. In addition, the number of terms in sample 01 is wrong in the figure.
- Guess, it should be $(33 + 26 + 13 + 15 + 15)/5 = 20.4$

Monte Carlo Methods : Bias and Variance

- Both first visit MC and every visit MC converge to V^π as number of trajectories go to infinity
- In first visit MC this is easy to see as each return sample is independent of the another
- By the law of large numbers the sequence of averages of these estimates converges to their expected value
- Each average is itself an unbiased estimate, and the standard deviation of its error falls as $\sqrt{1/N}$ where n is the number of returns averaged
- The convergence of every visit MC is less straight forward to see but it also converges at a quadratic rate to V^π

In both MC methods, it is possible that we may leave out computing $V^\pi(s)$ for some $s \in \mathcal{S}$ because the state s was never visited by any of the trajectories

Monte Carlo Algorithms: A Schematic View



- Uses experience, rather than model
- Uses only experience; does not bootstrap
- Needs complete sequences; suitable only for episodic tasks
- Suited for off-line learning
- Time required for one estimate does not depend on total number of states
- Estimates for each state are independent