Model-Free policy evaluation

Introduction to Reinforcement Learning

Bogdan Ivanyuk-Skulskyi, Dmytro Kuzmenko

Department of Mathematics,
National University of Kyiv-Mohyla Academy

February 27, 2023

Monte Carlo (MC) Policy Evaluation

- $G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + ..$ in MDP under policy π
- $V^\pi(s) = \mathbb{E}_{T\sim\pi}\left[G_t | s_t = s\right]$
  - Expectation over trajectories T generated by following π
- Simple idea: Value = mean return
- If trajectories are all finite, sample set of trajectories average returns

Monte Carlo (MC) Policy Evaluation

- If trajectories are all finite, sample set of trajectories average returns
- Does not require MDP dynamics / rewards
- Does not assume state in Markov
- Can be applied to episodic MDPs
  - Averaging over returns from a complete episode
  - Requires each episode to terminate

Monte Carlo (MC) on Policy Evaluation

- Aim: estimate $V^{\pi}(s)$ given episodes generated under policy π
  - $s_1, a_1, r_1, s_2, a_2, r_2, ...$ where the actions are samples from π
- $G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + ..$ in MDP under policy π
- $V^{\pi}(s) = \mathbb{E}_{T\sim\pi}\left[G_t | s_t = s\right]$
- MC computes empirical mean return
- Perfomed in incremental fashion
  - After each episode, update estimate of V^{π}

First-Visit Monte Carlo on Policy evaluation

Init $N(s) = 0$, $G(s) = 0, \forall s \in S$
Loop

- Sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, ...s_{i,T_i}, a_{i,T_i}, r_{i,T_i}$
- Define $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + ...\gamma^{T_i - 1} r_{i,T_i}$ as return from time step t onwards in ith episode
- For each time step t till the end of the episode i
  - If this is the **first** time t that state s is visited in episode i
    - Increment counter of total first visits: $N(s) += 1$
    - Increment total retrun $G(s) += G_{i,t}$
    - Update estimate $V^{\pi}(s) = G(s)/N(s)$

Evaluation of the Quality of a Policy Estimation Approach: Bias, Variance and MSE

- Consider a statistical model that is parametrized by θ and that determines a probability distribution over observed data $P(x|\theta)$
- Consider a statistic $\hat{\theta}$ that provides an estimate of θ and is a function of observed data x
    - E.g. for a Gaussian distribution with known variance, the average of a set of i.i.d data points is an estimate of the mean of the Gaussian
- Definition: the bias of an estimator $\hat{\theta}$ is :

$$Bias_{\theta}(\hat{\theta}) = \mathbb{E}_{x|\theta}[\hat{\theta}] - \theta$$

- Definition: the variance of an estimator $\hat{\theta}$ is :

$$Var(\hat{\theta}) = \mathbb{E}_{x|\theta}\left[(\hat{\theta} - \mathbb{E}[\hat{\theta})^2\right]$$

- Definition: mean squared error (MSE) of an estimator $\hat{\theta}$ is:

$$MSE(\hat{\theta}) = Var(\hat{\theta}) + Bias_{\theta}(\hat{\theta})^2$$

Evaluation of the Quality of a Policy Estimation Approach: Bias, Variance and MSE

- Consider a statistical model that is parametrized by θ and that determines a probability distribution over observed data $P(x|\theta)$
- Consider a statistic $\hat{\theta}$ that provides an estimate of θ and is a function of observed data x
- Definition: the bias of an estimator $\hat{\theta}$ is :

$$Bias_{\theta}(\hat{\theta}) = \mathbb{E}_{x|\theta}[\hat{\theta}] - \theta$$

- Let n be the number of data points x used to estimate the parameter θ and call the resulting estimate of θ using that data $\hat{\theta}_n$
- Then the estimator $\hat{\theta}_n$ is consistent if, for all $\epsilon > 0$

$$\lim_{n \to \infty} Pr(|\hat{\theta}_n - \theta| > \epsilon) = 0$$

First-Visit Monte Carlo on Policy evaluation

Init $N(s) = 0$, $G(s) = 0$, $\forall s \in S$
Loop

- Sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots s_{i,T_i}, a_{i,T_i}, r_{i,T_i}$
- Define $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \ldots \gamma^{T_i - 1} r_{i,T_i}$ as return from time step t onwards in ith episode
- For each time step t till the end of the episode i
  - If this is the **first** time t that state s is visited in episode i
    - Increment counter of total first visits: $N(s) += 1$
    - Increment total retrun $G(s) += G_{i,t}$
    - Update estimate $V^\pi(s) = G(s)/N(s)$

Properties:

- V^π estimator is an unbiased estimator of true $\mathbb{E}_\pi [G_t | s_t = s]$
- By law of large numbers, as $N(s) \to \infty$, $V^\pi(s) \to \mathbb{E}_\pi [G_t | s_t = s]$

Every-Visit Monte Carlo on Policy evaluation

Init $N(s) = 0$, $G(s) = 0, \forall s \in S$
Loop

- Sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots s_{i,T_i}, a_{i,T_i}, r_{i,T_i}$
- Define $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \ldots \gamma^{T_i - 1} r_{i,T_i}$ as return from time step t onwards in ith episode
- For each time step t till the end of the episode i
  - If this is the **every** time t that state s is visited in episode i
    - state s is the state visited at time step t in episode i
    - Increment counter of total visits: $N(s) += 1$
    - Increment total return $G(s) += G_{i,t}$
    - Update estimate $V^{\pi}(s) = G(s)/N(s)$

EVERY-VISIT MONTE CARLO ON POLICY EVALUATION

Init $N(s) = 0$, $G(s) = 0, \forall s \in S$
Loop

- Sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, ...s_{i,T_i}, a_{i,T_i}, r_{i,T_i}$
- Define $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + ...\gamma^{T_i-1} r_{i,T_i}$ as return from time step t onwards in ith episode
- For each time step t till the end of the episode i
  - If this is the **every** time t that state s is visited in episode i
    - state s is the state visited at time step t in episode i
    - Increment counter of total visits: $N(s) += 1$
    - Increment total return $G(s) += G_{i,t}$
    - Update estimate $V^\pi(s) = G(s)/N(s)$

Properties:

- V^π every-visit MC estimator is a **biased** estimator of V^π
- But consistent estimator and often has better MSE

EXAMPLE: FIRST-VISIT MC ON POLICY EVALUATION

Init $N(s) = 0$, $G(s) = 0$, $\forall s \in S$
Loop

- Sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots s_{i,T_i}, a_{i,T_i}, r_{i,T_i}$
- Define $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \ldots \gamma^{T_i - 1} r_{i,T_i}$ as return from time step t onwards in ith episode
- For each time step t till the end of the episode i
  - If this is the **first** time t that state s is visited in episode i
    - Increment counter of total first visits: $N(s) += 1$
    - Increment total retrun $G(s) += G_{i,t}$
    - Update estimate $V^\pi(s) = G(s)/N(s)$

- Mars rover: $R(s) =$ [+1 0 0 0 0 0 +10]
- $\pi(s) = a_1$, $\forall s$, $\gamma = 1$ any action from s_1 and s_7 terminates episode
- Trajectory = $(s_3, a_1, 0, s_2, a_1, 0, s_2, a_1, 0, s_1, a_1, 1, \textit{terminal})$

Example: First-Visit MC on Policy Evaluation

Init $N(s) = 0$, $G(s) = 0$, $\forall s \in S$
Loop

- Sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots s_{i,T_i}, a_{i,T_i}, r_{i,T_i}$
- Define $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \ldots \gamma^{T_i-1} r_{i,T_i}$ as return from time step t onwards in ith episode
- For each time step t till the end of the episode i
  - If this is the **first** time t that state s is visited in episode i
    - Increment counter of total first visits: $N(s) += 1$
    - Increment total retrun $G(s) += G_{i,t}$
    - Update estimate $V^{\pi}(s) = G(s)/N(s)$

- Mars rover: $R(s) = [+1\ 0\ 0\ 0\ 0\ 0\ +10]$
- Trajectory = $(s_3, a_1, 0, s_2, a_1, 0, s_2, a_1, 0, s_1, a_1, 1, \textit{terminal})$
- Let $\gamma = 1$. First visit MC estimate of V of each state?

$$V = [1, 1, 1, 0, 0, 0, 0]$$

- Not let $\gamma = 0.9$. Compare the first visit and every visit MC estimates of s_2
  - First visit

$$V^{MC}(s_2) = \gamma^2$$

  - Every visit

$$V^{MC}(s_2) = \frac{\gamma^2 + \gamma}{2}$$

Incremental MC on Policy Evaluation

After each episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots$

- Define $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \ldots$ as return from time step t onward in ith episode
- For state s visited at time step t in episode i
  - Increment counter of total visits: $N(s) + = 1$
  - Update estimate

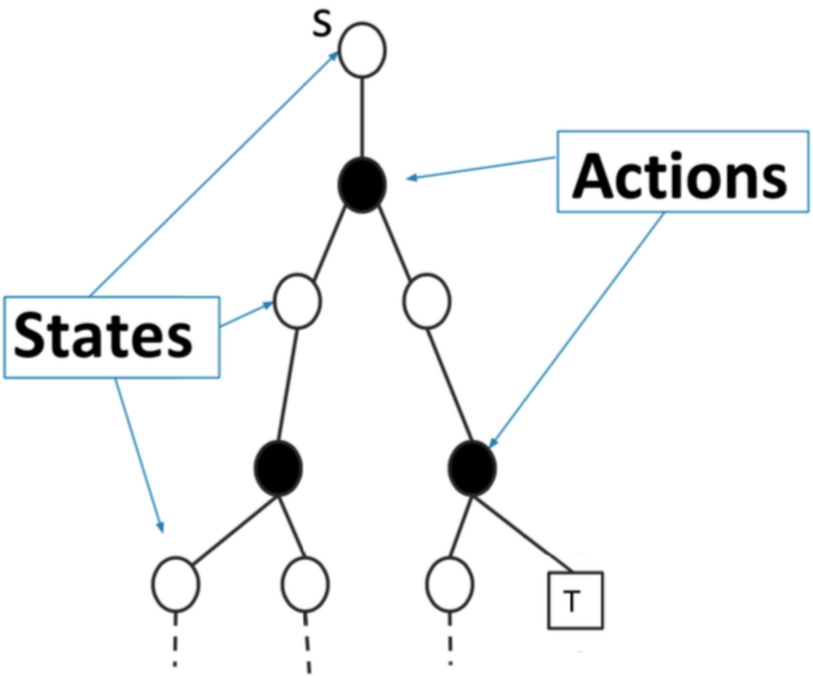
$$V^\pi(s) = V^\pi(s)\frac{N(s)-1}{N(s)} + \frac{G_{i,t}}{N(s)} = V^\pi + \frac{1}{N(s)}(G_{i,t} - V^\pi(s))$$

Incremental MC on Policy Evaluation

- Sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, ..., s_{i,T_i}$
- $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + ...\gamma^{T_i-1} r_{i,T_i}$
- for $i = 1 : T_i$ where T_i is the length of the i-th episode

$$V^\pi(s_{it}) = V^\pi(s_{it}) + \alpha(G_{i,t} - V^\pi(s_{it}))$$

Policy Evaluation Diagram



MC Policy Evaluation key limitations

- Generally high variance estimator
  - Reducing variance can require a lot of data
  - In cases where data is very hard or expensive to acquire, or the stakes are high, MC may be impractical
- Requires episodic settings
  - Episode must end before data from episode can be used to update V

MC Policy Evaluation Summary

- Aim: estimate $V^\pi(s)$ given episodes generated under policy π
  - $s_1, a_1, r_1, s_2, a_2, r_2, \ldots$ where the actions are sampled from π
- $G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots$ under policy π
- $V^\pi(s) = \mathbb{E}_\pi [G_t | s_t = s]$
- Simple: Estimates expectation by empirical average (given episodes sampled from policy of interest)
- Updates V estimate using **sample** of return to approximate the expectation
- Does not assume Markov process
- Converges to true value under some (generally mild) assumptions

Temporal Difference Learning

- Combination of Monte Carlo and dynamic programming methods
- Model-free
- Can be used in episodic or infinite-horizon non-episodic settings
- Immediately updates estimate of V after each (s, a, r, s') tuple

Temporal Difference Learning for Estimating V

- Aim: estimate $V^\pi(s)$ given episodes generated under policy π
- $G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots$ in MDP under policy π
- $V^\pi(s) = \mathbb{E}_\pi [G_t | s_t = s]$
- Recall Bellman operator

$$B^\pi V(s) = r(s, \pi(s)) + \gamma \sum_{s' \in S} P(s'|s, \pi(s)) V(s')$$

- In incremental every-visit MC, update estimate using 1 sample of return (for the current ith episode)

$$V^\pi(s) = V^\pi(s) + \alpha(G_{i,t} + V^\pi(s))$$

- Insight: have an estimate of V^π, use to estimate expected return

$$V^\pi(s) = V^\pi(s) + \alpha([r_t + \gamma V^\pi(s_{t+1})] - V^\pi(s))$$

Temporal Difference [TD(0)] learning

- Aim: estimate $V^{\pi}(s)$ given episodes generated under policy π
  - $s_1, a_1, r_1, s_2, a_2, r_2, \ldots$ where the actions are sampled from π
- Simplest TD learning: update value towards estimated value

$$V^{\pi}(s_t) = V^{\pi}(s_t) + \alpha([r_t + \gamma V^{\pi}(s_{t+1})] - V^{\pi}(s_t))$$

- TD error:

$$\delta_t = r_t + \gamma V^{\pi}(s_{t+1}) - V^{\pi}(s_t)$$

- Can immediately update value estimate after (s, a, r, s') tuple
- Don't need episodic setting

TEMPORAL DIFFERENCE [TD(0)] LEARNING ALGORITHM

- Input: α
- Init $V^{\pi}(s) = 0, \forall s \in S$
- Loop
  - Sample tuple (s_t, a_t, r_t, s_{t+1})
  - Calculate

$$V^{\pi}(s_t) = V^{\pi}(s_t) + \alpha([r_t + \gamma V^{\pi}(s_{t+1})] - V^{\pi}(s_t))$$

Temporal Difference [TD(0)] Learning Algorithm

- Input: α
- Init $V^\pi(s) = 0, \forall s \in S$
- Loop
  - Sample tuple (s_t, a_t, r_t, s_{t+1})
  - Calculate

$$V^\pi(s_t) = V^\pi(s_t) + \alpha([r_t + \gamma V^\pi(s_{t+1})] - V^\pi(s_t))$$

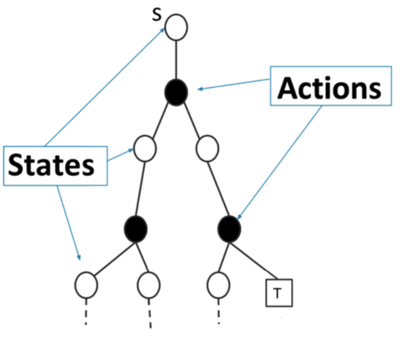
Example Mars rover: R=[1, 0, 0, 0, 0, 0, +10]

- $\pi(s) = a_1 \; \forall s$, $\gamma = 1$. any action from s_1 and s_7 terminates episode
- Trajectory = $(s_3, a_1, 0, s_2, a_1, 0, s_2, a_1, 0, s_1, a_1, 1, \textit{terminal})$
- TD estimate of all states (init at 0) with $\alpha = 1$: V = $[1, 0, 0, 0, 0, 0, 0, 0]$
- First-visit MC estimate of V of each state: $[1, 1, 1, 0, 0, 0, 0]$

Temporal Difference (TD) Policy Evaluation

$$V^\pi(s_t) = r(s_t, \pi(s_t)) + \gamma \sum_{s_{t+1}} P(s_{t+1}|s_t, \pi(s_t)) V^\pi(s_{t+1})$$

$$V^\pi(s_t) = V^\pi(s_t) + \alpha([r_t + \gamma V^\pi(s_{t+1})] - V^\pi(s_t)$$



- TD updates the value estimate using a sample of s_{t+1} to approximate an expectation
- TD updates the value estimate by bootstrapping, uses estimate of $V(s_{t+1})$

Summary: Temporal Difference Learning

- Combination of Monte Carlo dynamic programming methods
- Model-free
- Bootstraps and samples
- Can be used in episodic or infinite-horizon non-episodic settings
- Immediately updates estimate of V after each (s, a, r, s') tuple