Reinforcement *Learning*

Model Free

by

Giovanni Stilo, PhD.

giovanni.stilo@univaq.it

Last time Recap.

- Markov Processes
- Markov Reward Processes (MRPs)
  - Compute Value Function
- Markov Decision Processes (MDPs)
  - Evaluate a Policy
- Compute the Optimal Policy in MDP

Outline

- Estimating the expected return of a particular policy if don't have access to true MDP models
  - Dynamic programming

  - Monte Carlo policy evaluation
    - Policy evaluation when don't have a model of how the world work;
      - Given on-policy samples;

Return & Value Function

- Definition of Return, G_t (for a MRP):
  - Discounted sum of rewards from time step ***t*** to horizon

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+1} + \gamma^3 r_{t+3} + \cdots$$

- Definition of State Value Function, V(s) (for a MRP)
  - Expected return from starting in state s under policy π

$$\mathrm{V}^{\pi}(\mathrm{s}) = \mathbb{E}_{\pi}[\, G_t \mid s_t = s] = \mathbb{E}_{\pi}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+1} + \gamma^3 r_{t+3} + \cdots | s_t = \mathrm{s}]$$

- Definition of the State-action value function:
  - Expected return from starting in state s, taking action a and then following policy π

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

Dynamic Programming for Policy Evaluation

- The Dynamic Programming (Iterative);
- Initialize $V_0^\pi(s) = 0 \; \forall \; s \in S$;
- For $k \; = \; 1$ until **convergence**:
  - $\forall s \in S$

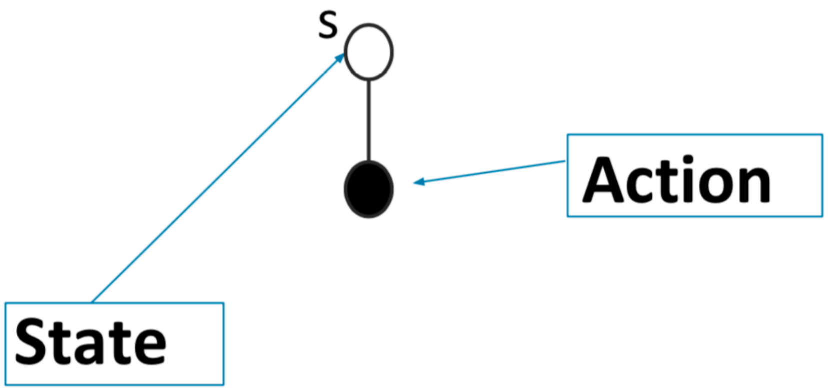
$$V_k^\pi(s) = R(s, \pi(s)) + \gamma \sum_{\{s' \in S\}} P(s'|s, \pi(s)) \, V_{k-1}^\pi(s')$$

- $V_k^\pi(s)$ is exact value of k-horizon value of state s under policy π
- $V_k^\pi(s)$ is an estimate of infinite horizon value of state s under policy π:

$$\mathrm{V}^\pi(\mathrm{s}) = \mathbb{E}_\pi[\, G_t \mid s_t = s] \approx \mathbb{E}_\pi[r_t + \gamma V_{k-1} | \; s_t = \mathrm{s}]$$

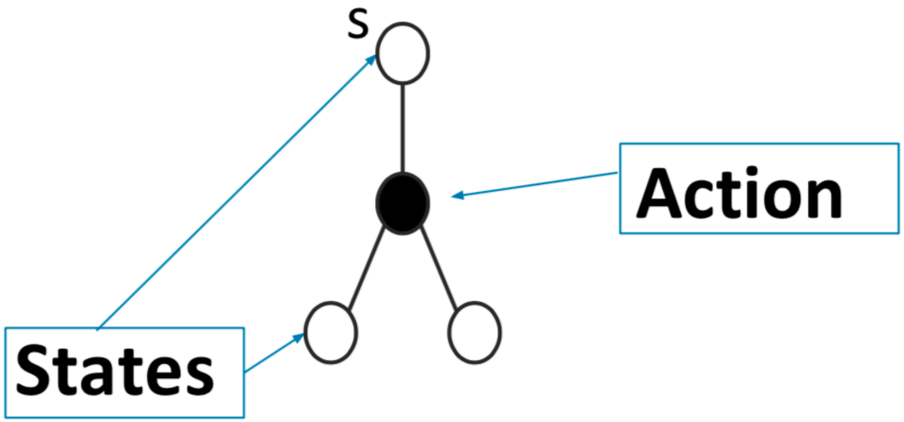
Dynamic Programming for Policy Evaluation

- $\mathrm{V}^{\pi}(\mathrm{s}) = \mathbb{E}_{\pi}[\, G_t \mid s_t = s] \approx \mathbb{E}_{\pi}[r_t + \gamma V_{k-1} \mid s_t = \mathrm{s}]$



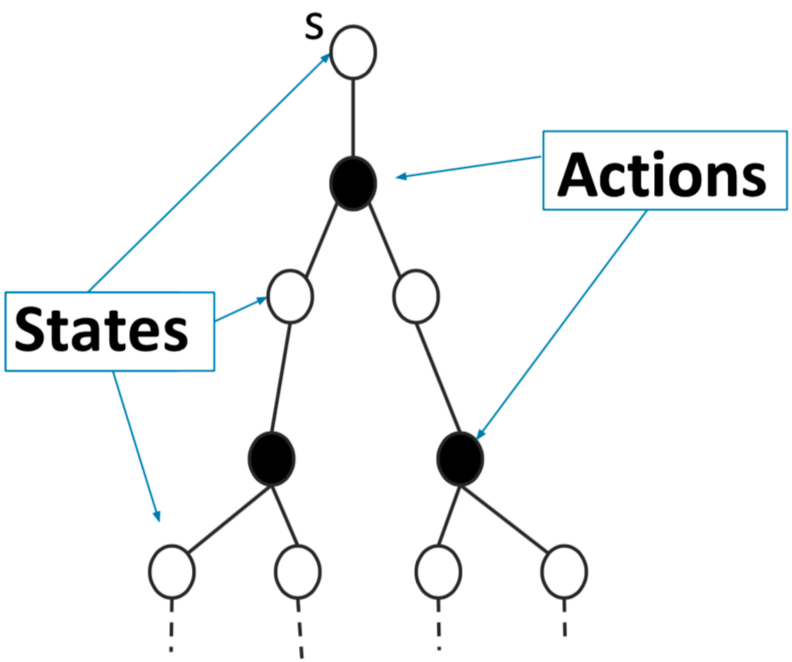
Dynamic Programming for Policy Evaluation

- $\mathrm{V}^{\pi}(\mathrm{s}) = \mathbb{E}_{\pi}[\, G_t \mid s_t = s] \approx \mathbb{E}_{\pi}[r_t + \gamma V_{k-1} \mid s_t = \mathrm{s}]$



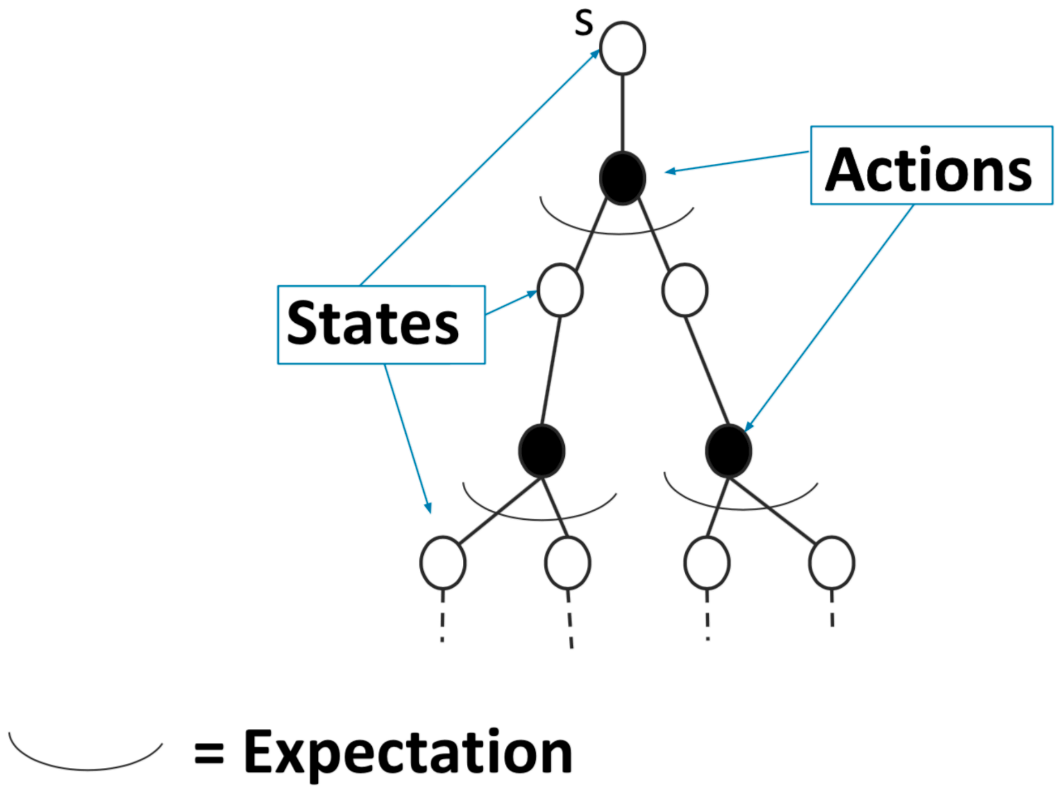
Dynamic Programming for Policy Evaluation

- $V^{\pi}(s) = \mathbb{E}_{\pi}[\, G_t \mid s_t = s] \approx \mathbb{E}_{\pi}[r_t + \gamma V_{k-1} \mid s_t = s]$



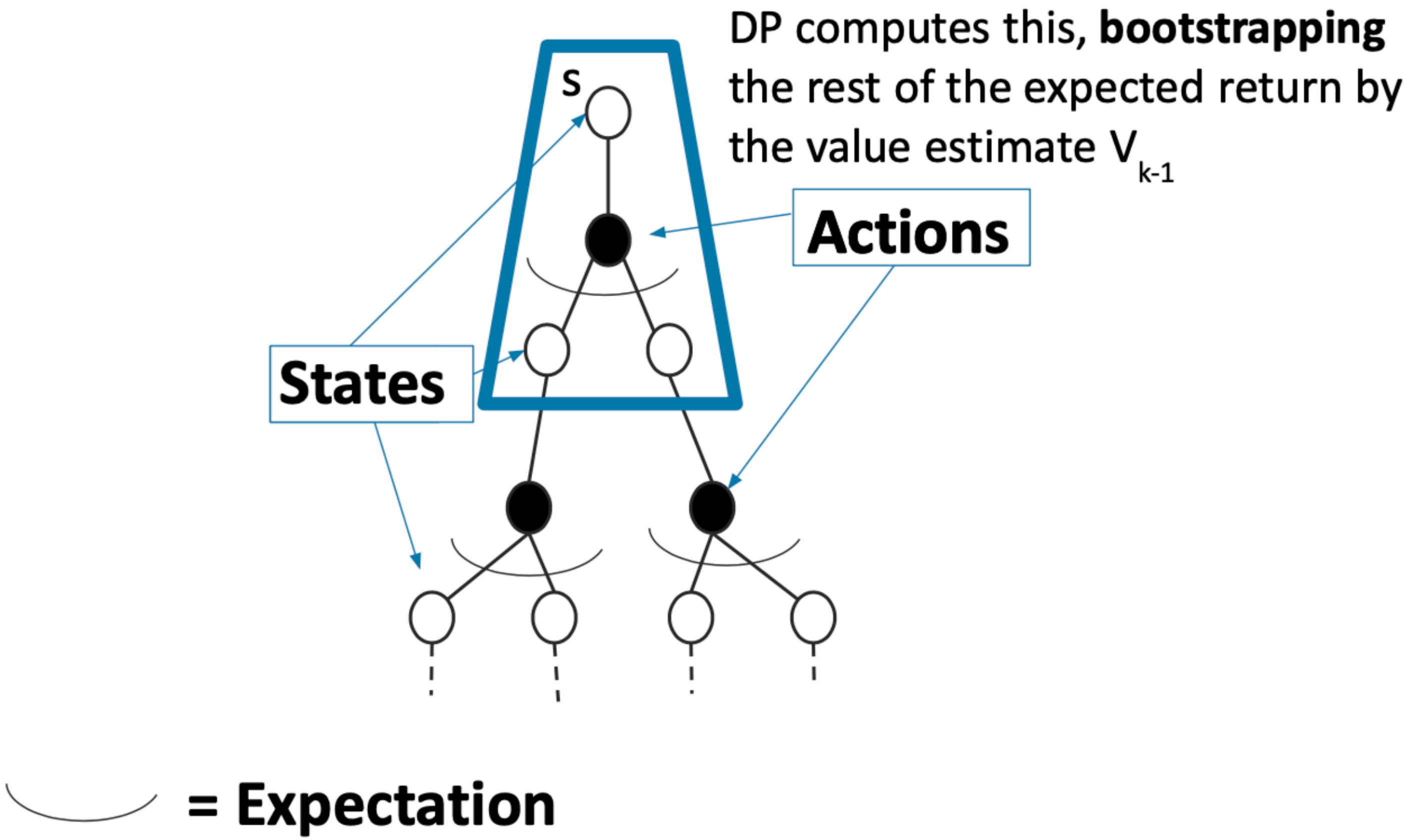
Dynamic Programming for Policy Evaluation

- $\mathrm{V}^{\pi}(\mathrm{s}) = \mathbb{E}_{\pi}[\, G_t \mid s_t = s] \approx \mathbb{E}_{\pi}[r_t + \gamma V_{k-1} \mid s_t = \mathrm{s}]$



Dynamic Programming for Policy Evaluation

- $V^{\pi}(s) = \mathbb{E}_{\pi}[\, G_t \mid s_t = s] \approx \mathbb{E}_{\pi}[r_t + \gamma V_{k-1} \mid s_t = s]$

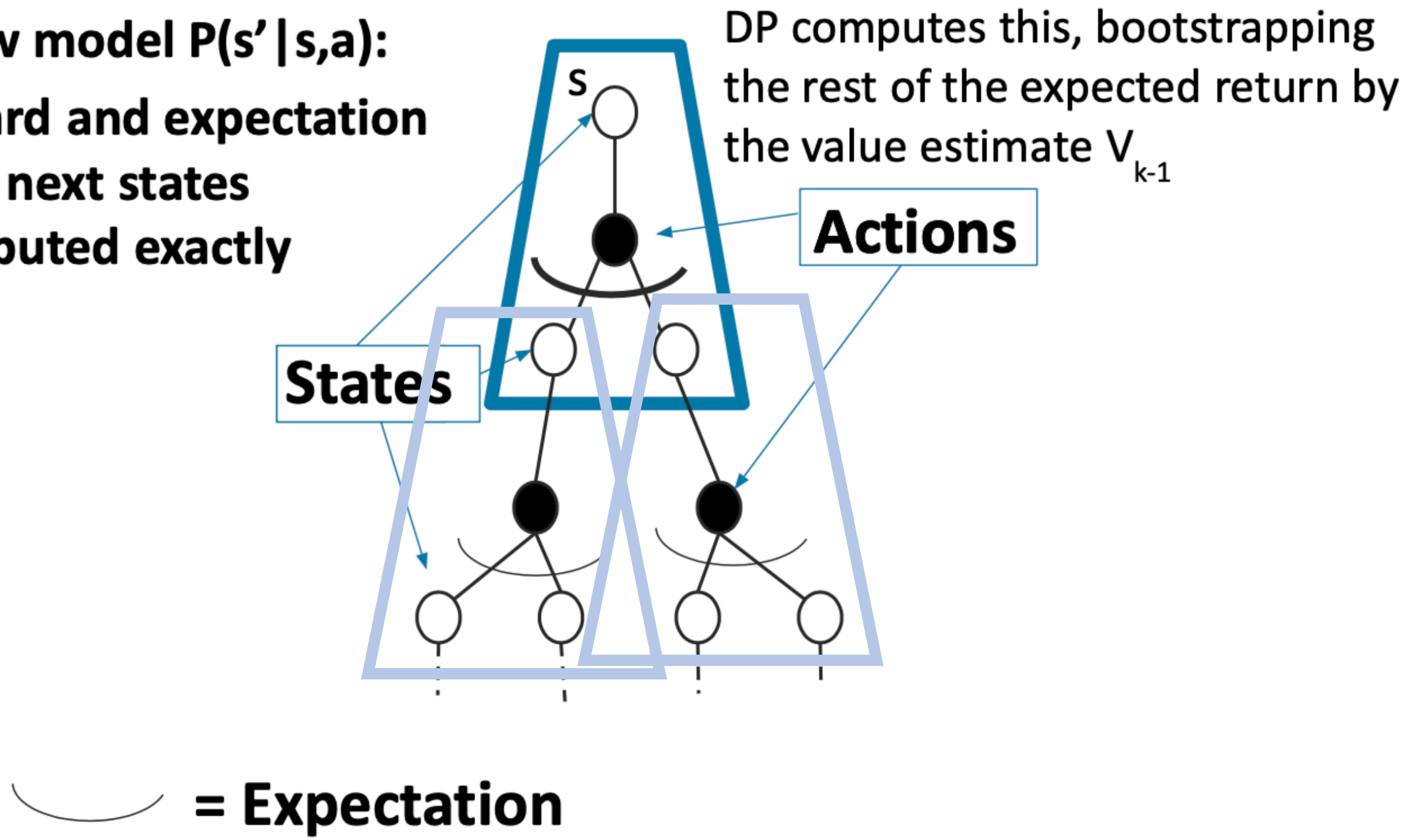


Dynamic Programming for Policy Evaluation

- $V^{\pi}(s) = \mathbb{E}_{\pi}[\, G_t \mid s_t = s] \approx \mathbb{E}_{\pi}[r_t + \gamma V_{k-1} \mid s_t = s]$

Know model P(s'|s,a):

reward and expectation over next states computed exactly



Policy Evaluation

- $G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+1} + \gamma^3 r_{t+3} + \cdots$
- Dynamic Programming:
  - $\mathrm{V}^{\pi}(\mathrm{s}) = \mathbb{E}_{\pi}[r_t + \gamma V_{k-1} | s_t = s]$

  - Requires model of MDP ***M***
  - Bootstraps future return using value estimate;
  - Requires Markov assumption: bootstrapping regardless of history;

What if don't know dynamics model P and/ or reward model R?

- Policy evaluation without a model:
  - Given data and/or ability to **interact** in the environment
  - Efficiently compute a good estimate of a policy π

Monte Carlo (MC) Policy Evaluation

- in MDP $\boldsymbol{M}$ under policy π:
  - $G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+1} + \gamma^3 r_{t+3} + \cdots$

- Dynamic Programming:
  - $\mathrm{V}^{\pi}(\mathrm{s}) = \mathbb{E}_{T\sim\pi}[G_t | s_t = s]$
  - **Expectation over all trajectories** *T* generated by following *π* according to **the transition model *P***

Idea: *Value = Mean of **the return***

- If ***all*** trajectories are finite:
  - sample a set of trajectories;
  - average returns;

Monte Carlo (MC) Policy Evaluation - PRO

- If trajectories are ***all*** finite:
  - sample set of trajectories;
  - average returns;

- Does not require MDP dynamics/rewards;

- Does not assume state is Markov;

- No bootstrapping;

- Can only be applied to ***episodic*** MDPs:
  - Averaging over returns from a complete episode
  - Requires each episode to terminate

Atari Games – 2015 DeepMind

https://www.youtube.com/watch?v=V1eYniJ0Rnk

Monte Carlo (MC) Policy Evaluation

- Aim: estimate $V_\pi(s)$ given episodes generated under policy π (actions are **sampled** from) π:
  - $s_1, a_1, r_1, s_2, a_2, r_2, \ldots$

- in MDP ***M*** under policy π:
  - $G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+1} + \gamma^3 r_{t+3} + \cdots$
  - $\mathrm{V}^\pi(\mathrm{s}) = \mathbb{E}_\pi[G_t | s_t = s]$

- Monte Carlo computes ***empirical*** mean return

- Often do this in an incremental fashion:
  - after each episode, update estimate of V_π

First-Visit Monte Carlo (MC) On-Policy Evaluation

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

Loop

- Sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots, s_{i,T_i}$

- Define the return of the i^{th} episode from time step t onwards:
  - $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots + \gamma^{T_i-1} r_{i,T_i}$

- For each state s visited in episode i:
  - For **first** time t that state s is visited in episode i:
    - Increment counter of total first visits: $N(s) = N(s) + 1$
    - Increment total return $G(s) = G(s) + G_{i,t}$
    - Update estimate $V^{\pi}(s) = G(s)/N(s)$

Bias, Variance and MSE

- Consider a statistical model that is parameterized by θ which 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:

$$Bias_\theta(\hat{\theta}) = E_{x|\theta}[\hat{\theta}] - \theta$$

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

$$Var_\theta\,(\hat{\theta}) = E_{x|\theta}[\hat{\theta} - E_{x|\theta}\left[\hat{\theta}\right]^2]$$

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

$$\mathrm{MSE}\left(\hat{\theta}\right) = Bias_\theta\left(\hat{\theta}\right)^2 + Var_\theta\,(\hat{\theta})$$

First-Visit Monte Carlo (MC) On-Policy Evaluation

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

Loop

- Sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots, s_{i,T_i}$

- Define the return of the i^{th} episode from time step t onwards:
  - $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots + \gamma^{T_i - 1} r_{i,T_i}$

- For each state s visited in episode i:
  - For **first** time t that state s is visited in episode i:
    - Increment counter of total first visits: $N(s) = N(s) + 1$
    - Increment total return $G(s) = G(s) + G_{i,t}$
    - Update estimate $V^{\pi}(s) = \frac{G(s)}{N(s)}$

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

Every-Visit MC On-Policy Evaluation

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

Loop

- Sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots, s_{i,T_i}$

- Define the return of the i^{th} episode from time step t onwards:
  - $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots + \gamma^{T_i - 1} r_{i,T_i}$

- For each state s visited in episode i:
  - For **every** time t that state s is visited in episode i:
    - Increment counter of total visits: $N(s) = N(s) + 1$
    - Increment total return $G(s) = G(s) + G_{i,t}$
    - Update estimate $V^{\pi}(s) = \frac{G(s)}{N(s)}$

Every-Visit MC On-Policy Evaluation

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

Loop

- Sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots, s_{i,T_i}$

- Define the return of the i^{th} episode from time step t onwards:
  - $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots + \gamma^{T_i - 1} r_{i,T_i}$

- For each state s visited in episode i:
  - For **every** time t that state s is visited in episode i:
    - Increment counter of total visits: $N(s) = N(s) + 1$
    - Increment total return $G(s) = G(s) + G_{i,t}$
    - Update estimate $V^\pi(s) = \frac{G(s)}{N(s)}$

- **Properties:**
  - V^π estimator is *a* ***biased*** estimator of true $\mathbb{E}_\pi[G_t | s_t = s]$
  - But consistent estimator and often has better ***MSE*** (< variance);

Incremental (MC) On Policy Evaluation

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

Loop

- Sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots, s_{i,T_i}$

- Define the return of the i^{th} episode from time step t onwards:
  - $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots + \gamma^{T_i-1} r_{i,T_i}$

- For each state s visited in episode i at time t:
  - Increment counter of total first visits: $N(s) = N(s) + 1$
  - ~~Increment total return $G(s) = G(s) + G_{i,t}$~~
  - Update estimate:

$$V_k^\pi(s) = V_{k-1}^\pi(s)\frac{N(s)-1}{N(s)} + \frac{G_{i,t}}{N(s)} = V_{k-1}^\pi(s)\frac{N(s)}{N(s)} - \frac{V_{k-1}^\pi(s)}{N(s)} + \frac{G_{i,t}}{N(s)} =$$
$$V_{k-1}^\pi(s) + \frac{1}{N(s)}\left(G_{i,t} - V_{k-1}^\pi(s)\right) = V_{k-1}^\pi(s) + \boldsymbol{\alpha}\left(G_{i,t} - V_{k-1}^\pi(s)\right)$$

  - $\alpha = \frac{1}{N(s)}$: identical to the every visit behaviour
  - $\alpha > \frac{1}{N(s)}$: forget older data (can be helpful for non-stationary domains);

Monte Carlo (MC) Policy Evaluation Limitations

- Generally **high variance** estimator:
  - Reducing variance can require a lot of data;
- Requires **episodic** settings:
  - to update the value function the episode must end before data from that episode can be used;

Monte Carlo (MC) Policy Evaluation Summary

- Aim: estimate $V_\pi(s)$ given episodes generated under policy π (actions are **sampled** from) π:
  - $s_1, a_1, r_1, s_2, a_2, r_2, \ldots$
- in MDP ***M*** under policy π:
  - $G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+1} + \gamma^3 r_{t+3} + \cdots$
  - $\mathrm{V}^\pi(\mathrm{s}) = \mathbb{E}_\pi[G_t | s_t = s]$
- **Simple:**
  - Given episodes sampled from policy of interest:
    - Estimates expectation by empirical average
    - or reweighted empirical average (importance sampling);
- Updates value estimate by using a **sample** of return to approximate the expectation;
- No bootstrapping
- Converges to true value under some (generally mild) assumptions

Reinforcement *Learning*

Model Free Control

by

Giovanni Stilo, PhD.

giovanni.stilo@univaq.it

Outline Last/Today

- **Last time:** Estimating the expected return of a particular policy if don't have access to true MDP models:
  - Monte Carlo policy evaluation
    - Policy evaluation when don't have a model of how the world work;

- **Today:** Control (making decisions) without a model of how the world works. how can we learn a good policy?

Recall RL Involves

- Optimization
- Delayed consequences
- Exploration
- Generalization

Today: Learning to Control Involves

- **Optimization:** Goal is to identify a policy with high expected rewards (similar to computing an optimal policy given decision process models)

- **Delayed consequences:** May take many time steps to evaluate whether an earlier decision was good or not

- **Exploration:** Necessary to try different actions to learn what actions can lead to high rewards

Today: Model-free Control

- **Generalized policy improvement**
- **Importance of exploration**
- **Monte Carlo control**

Why Model-free Control?

- Many applications can be modeled as a MDP
  - Backgammon, Go, Robot locomotion, Helicopter flight, Robocup soccer, Autonomous driving, Customer ad selection, Invasive species management, Patient treatment;

- For many of these and other problems either:
  1. MDP model is **unknown** but can be sampled
  2. MDP model is known but it is **computationally infeasible** to use directly, except through sampling

On and Off-Policy Learning

- **On-policy learning:**
  - Direct experience
  - Learn to estimate and evaluate a policy from experience obtained from following that policy;

- **Off-policy learning:**
  - Learn to estimate and evaluate a policy using experience gathered from following certain different policy;

Recall Policy Iteration

- Initialize policy π
- Repeat:
  - Policy evaluation: compute V^{π}
  - Policy improvement: update π

$$\pi_{i+1}(S) = arg\ max_a\ Q^{\pi_i}(s,a) =$$
$$arg\ max_a R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^{\pi}(s')$$

- Now **we need** to do the above two steps **without** access to the true **dynamics** and **reward models**

- Last lecture introduced methods for model-free policy evaluation

Model Free Policy Iteration

We can perform policy iteration but how to improve the policy?

- Initialize policy π
- Repeat:
  - Policy evaluation: compute V^{π}
  - Policy improvement: update π

Incremental (MC) On Policy Evaluation

Initialize $N(s) = 0,\ G(s) = 0\ \forall s \in S$
Loop
- Sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots, s_{i,T_i}$

- Define the return of the i^{th} episode from time step t onwards:
  - $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots + \gamma^{T_i - 1} r_{i,T_i}$

- For each state s visited in episode i at time t:
  - Increment counter of total first visits: $N(s) = N(s) + 1$
  - Update estimate:

$$V_k^{\pi}(s) = V_{k-1}^{\pi}(s) + \boldsymbol{\alpha}\left(G_{i,t} - V_{k-1}^{\pi}(s)\right)$$

  - $\alpha = \frac{1}{N(s)}$: identical to the every visit behaviour
  - $\alpha > \frac{1}{N(s)}$: forget older data (can be helpful for non-stationary domains);

MC for On Policy Q Evaluation

Initialize $N(s,a) = 0,\ G(s,a) = 0, Q^{\pi}(s,a) = 0, \forall s \in S, \forall A$

Loop

- Using policy π sampling $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots, s_{i,T_i}$
- $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots + \gamma^{T_i - 1} r_{i,T_i}$

- For each **state $\boldsymbol{s}$** and **action $\boldsymbol{a}$** $(\boldsymbol{s}, \boldsymbol{a})$ visited in episode i at time t:
  - $N(s,a) = N(s,a) + 1, G(s,a) = G(s,a) + G_{i,t}$
  - Update estimate $Q^{\pi}(s,a) = G(s,a)/N(s,a)$

Model-free Generalized Policy Improvement

- Given an estimate $Q^{\pi_i}(s, a) \; \forall \; s, a$
- Update new policy

$$\pi_{i+1}(s) = \arg\max_a Q^{\pi_i}(s, a)$$

Model-free Policy Iteration (issue)

- Initialize policy π
- Repeat:
  - Policy evaluation: compute Q^{π}
  - Policy improvement: update π given Q^{π}

- May need to modify policy evaluation:
  - If π is **deterministic**, can't compute $Q(s, a)$ for any $a \neq \pi(s)$

- How to interleave policy evaluation and improvement?
  - Policy improvement is now using an estimated Q

So?

Policy Evaluation with Exploration

- Want to compute a model-free estimate of Q^{π}
- In general seems subtle:
  - Need to try all (s, a) pairs but then follow π
  - Want to ensure resulting estimate Q^{π} is good enough so that policy improvement is a monotonic operator
- For certain classes of policies can ensure all (s, a) pairs are tried such that asymptotically Q^{π} converges to the true value;

ε-greedy Policies

- Simple idea to balance exploration and exploitation:

- Let $|A|$ be the number of actions
- Then an ε-greedy policy w.r.t. a state-action value $Q(s,a)$ is:

$\pi(a|s)$ = $[\arg\max_a Q(s,a), \quad w.prob\ 1-\epsilon; a\ w.prob\ \frac{\epsilon}{|A|}]$

  - With prob. $\frac{\epsilon}{|A|}$ we action a; // New Action
  - With prob. $1-\epsilon$ we use $arg\ \max_a Q(s,a)$ *// Old Policy*

Check Your Understanding: MC for On Policy Q Evaluation

Initialize $N(s,a) = 0,\ G(s,a) = 0, Q^{\pi}(s,a) = 0, \forall s \in S, \forall A$

Loop

- Using policy π sampling $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots, s_{i,T_i}$
- $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots + \gamma^{T_i - 1} r_{i,T_i}$
- For each **state** $\boldsymbol{s}$ and **action** $\boldsymbol{a}$ $(\boldsymbol{s}, \boldsymbol{a})$ visited in episode i at time t:
  - $N(s,a) = N(s,a) + 1, G(s,a) = G(s,a) + G_{i,t}$
  - Update estimate $Q^{\pi}(s,a) = G(s,a)/N(s,a)$

- Mars rover with two actions:
  - $r(-, a_1) = [\,1\ 0\ 0\ 0\ 0\ 0\ +10],\ r(-, a_2) = [\,0\ 0\ 0\ 0\ 0\ 0\ +5],\ \gamma = 1.$
- Assume current greedy $\pi(s) = a1\ \forall s,$
- $\varepsilon = 0.5$
- Sample trajectory from ε-greedy policy
- Trajectory $= (s_3, a_1, 0;\ s_2, a_2, 0;\ s_3, a_1, 0;\ s_2, a_2, 0;\ s_1, a_1, 1;\ terminal)$

- Compute the first visit MC estimate of Q of each (s,a) pair?

$$Q^{\epsilon-\pi}(-, a_1) = [\_\,,\_\,,\_\,,\_\,,\_\,,\_\,,\_]$$
$$Q^{\epsilon-\pi}(-, a_2) = [\_\,,\_\,,\_\,,\_\,,\_\,,\_\,,\_]$$

Check Your Understanding: MC for On Policy Q Evaluation

Initialize $N(s,a) = 0$, $G(s,a) = 0, Q^{\pi}(s,a) = 0, \forall s \in S, \forall A$

Loop

- Using policy π sampling $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots, s_{i,T_i}$
- $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots + \gamma^{T_i - 1} r_{i,T_i}$
- For each **state** $\boldsymbol{s}$ and **action** $\boldsymbol{a}$ $(\boldsymbol{s}, \boldsymbol{a})$ visited in episode i at time t:
  - $N(s,a) = N(s,a) + 1, G(s,a) = G(s,a) + G_{i,t}$
  - Update estimate $Q^{\pi}(s,a) = G(s,a)/N(s,a)$

- Mars rover with two actions:
  - $r(-,a_1) = [\,1\ 0\ 0\ 0\ 0\ 0\ +10],\ r(-,a_2) = [\,0\ 0\ 0\ 0\ 0\ 0\ +5],\ \gamma = 1.$
- Assume current greedy $\pi(s) = a_1\ \forall s,$
- $\varepsilon = 0.5$
- Sample trajectory from ε-greedy policy
- Trajectory = $(s_3, a_1, 0;\ s_2, a_2, 0;\ s_3, a_1, 0;\ s_2, a_2, 0;\ s_1, a_1, 1;\ terminal)$

Type equation here.

- Compute the first visit MC estimate of Q of each (s,a) pair?

$$Q^{\epsilon-\pi}(-,a_1) = [1, 0, 1, 0, 0, 0, 0]$$
$$Q^{\epsilon-\pi}(-,a_2) = [0, 1, 0, 0, 0, 0, 0]$$

Subtleties of Policy Improvement

- Note that when we first introduced policy improvement with a given MDP dynamics and reward model, policy evaluation was computed exactly;
- In this case monotonic improvement was guaranteed for each policy improvement step;
- In this lecture we will often be considering computing a Q using samples gathered from many policies;
- Beautifully, generalized policy iteration using MC and TD methods still converge under some mild conditions;
- For more technical details, proofs of the convergence of Q-learning for different scenarios can be found here:
  - Q-Learning. Watkins and Dayan;
  - Machine Learning. 1992;
  - Asynchronous Stochastic Approximation and Q-Learning; Tsitsiklis. Machine Learning. 1994

Greedy in the Limit of Infinite Exploration (GLIE)

- Definition of **GLIE**:
  - All state-action pairs are visited an infinite number of times
    $$\lim_{i \longrightarrow \infty} N_i(s,a) \longrightarrow \infty$$
  - Behaviour of the policy converges to the greedy policy one:
    $\lim_{i \longrightarrow \infty} \pi(a|s) \longrightarrow arg \max_a Q(s,a)$ with probabilty 1

NOTE: A ε-greedy where $\varepsilon \longrightarrow 0$ is **GLIE**:

- Typically with the following rate: $\varepsilon_i = \frac{1}{i}$

Monte Carlo Online Control / On Policy Improvement

1. Initialize $Q(s,a) = 0, \mathrm{N(s, a)} \forall (s,a), set\ \epsilon = 1\ , k = 1$
2. $\pi_k \ = \ \varepsilon$-greedy(Q) // Create initial ε-greedy policy
3. **loop**
4. Sample k^{th} episode $(s_{k,1}, a_{k,1}, r_{k,1} s_{k,2}, a_{k,2}, r_{k,2}, \ldots\ , s_{k,T})$ given π_k
5. $G_{\mathrm{k},t} = \ r_{\mathrm{k},t} + \gamma r_{\mathrm{k},t+1} + \gamma^2 r_{\mathrm{k},t+2} + \cdots + \gamma^{T_k - 1} r_{\mathrm{k},T_k}$
6. **for** $t \ = \ 1, \ldots, T$ **do**
7. **if** First visit to (s,a) in episode k **then**:
8. $N(s,a) = \ N(s,a) + \ 1$
9. $Q'(s_t, a_t) = Q(s_t, a_t) + \frac{1}{N(s,a)}(G_{k,t} - Q(s_t, a_t))$
10. **end if**
11. **end for**
12. $k \ = \ k + 1, \epsilon = \frac{1}{k}$
13. $\pi_k \ = \ \varepsilon$-greedy(Q) // Policy improvement
14. **End loop**

Summary

- **Generalized policy improvement**
- **Importance of exploration**
- **Monte Carlo control**

AlphaGo - The Movie | Full Documentary

https://www.youtube.com/watch?v=WXuK6gekU1Y