Reinforcement Learning (RL)

Chapter 4:

Monte-Carlo Methods on Reinforcement Learning

Saeed Saeedvand, Ph.D.

Contents

In this Chapter:

- ✓ Model-Based Learning and Model-Free Learning
- ✓ First-Visit Method MC Prediction
- ✓ Every-Visit Method MC Prediction
- ✓ Monte Carlo Exploring Starts (MC-ES)
- ✓ MC Epsilon Greedy (without Exploring Starts)

Aim of this chapter:

- ✓ Understand the differences of model-based and model-free algorithms. Learning about Monte Carlo different approaches both based on Prediction and Control.

Monte-Carlo Algorithm (Model free RL)

Idea:

✓ In policy and value iteration algorithms the assumption includes the agent that **has access to complete model of the environment**:

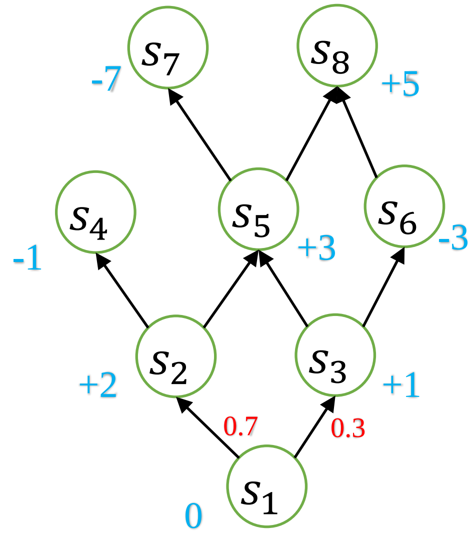
- Transition Dynamics (possible states after each action)
- No need to interact and calculations were predictable (including rewards)

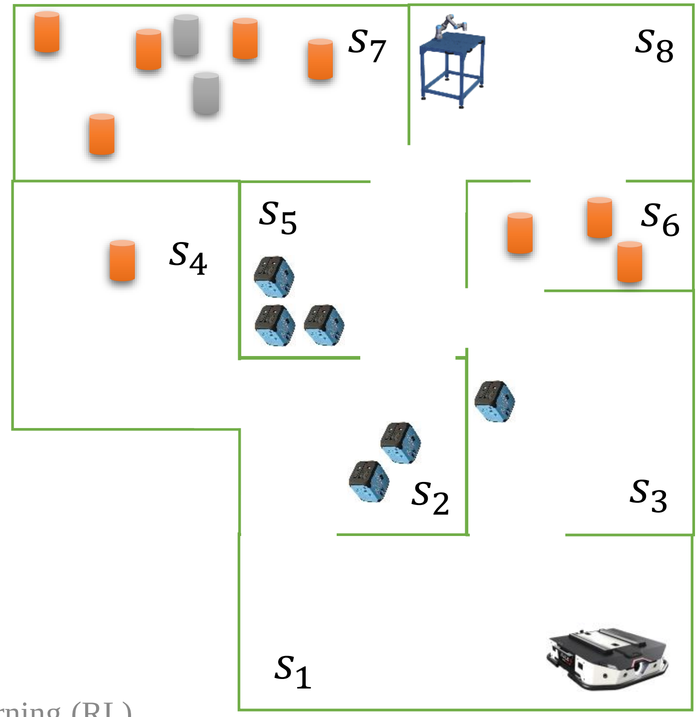
This is impractical in many scenarios and environments

Monte-Carlo Algorithm (Model free RL)

Idea:

- ✓ The agent maybe cannot have all information of the environment that we need algorithms to solve it
- ✓ So we need **agent that interacts in** the environment to **learn policy**





Monte-Carlo Algorithm

Monte Carlo term meaning:

- Any estimation method with significant random components.

MC Definition

✓ Monte Carlo methods works based on **experience sampling**

✓ Experience sampling is **sequences of** states, actions, and rewards from actual or simulated **interaction** with an environment

- Not predictable before real try in many cases.

✓ Only generate sample transitions, not the complete probability distributions of all as it was in dynamic programming.

Monte-Carlo Algorithm

MC Definition

- ✓ **Monte Carlo** methods are **ways of solving the reinforcement learning problem** based on averaging sample returns.

- ✓ Monte Carlo methods can thus be **incremental** in an **episode-by-episode sense**, but not in a step-by-step (online) sense.

Markov Decision Processes (MDPs)

What was Definition of the MDPs?

- ✓ There are set of states S, actions A
- ✓ Reward model

$$\cancel{R_t =} R(S_t, A_t)$$

- ✓ Transition Model:

$$\mathrm{P}(S_t \cancel{| S}_{t-1}, a_{t-1})$$

- ✓ Discount Factor (γ), it is between [0,1]
- ✓ Horizon (h) (episode, or time steps)

We do not have them in MC algorithms

Monte-Carlo Algorithm (RL)

Assumption

- ✓ We assume experience is divide into episodes (eventually terminate)
- ✓ Learn and **update policy based on the interaction**
- ✓ Agent **doesn't know** about the environment and the model of environment
- ✓ Agent **needs to interact** and try

Model-Based vs Model-Free Learning

Model

- ✓ A model in RL strictly refers to whether the agent is using **learning through environment** actions or not.

Model-Based Learning

- ➢ In Model-Based RL, the agent has access to a model of the environment.
- ➢ The advantage is that this allows the agent to plan ahead by prediction ahead

Model-Free learning

- ➢ In Model-Free learning, the agent does not have access to a model of the environment (**Predictions of state transition and executing to get rewards**).

Monte-Carlo Algorithm

Underlying Idea of all Monte Carlo methods

- ✓ Monte Carlo methods aim for learning the **state-value** or **action-values** function (based on which approach we are using)
- ✓ Simple average the returns observed after visits to that state
- ✓ More returns are observed, the average should converge to the expected value
- ✓ Monte Carlo for episodic RL problems with a terminal state problem

Quick Reminder

What was the state-value function:

- Value of a state is the expected return

What was the action-value function:

- Value of a **each action** in state is the expected return
- Known as Q-value

What is the expected return:

- Expected cumulative future discounted reward starting from that state

Monte-Carlo Algorithm-Exploring Starts

Different versions of Monte-Carlo Algorithm

- ✓ **First-Visit method** (MC Prediction)
  - The first-visit MC method estimates $v_\pi(s)$ as **the average of the returns following first visits to state s**
  - Based on the state-value (estimation of value function)
- ✓ **Every-Visit method** (MC Prediction)
  - The every-visit MC method estimates $v_\pi(s)$ as the **average of the all visits to state s**
  - Based on the state-value (estimation of value function)
- ✓ **Exploring Starts method** (MC Control)
  - Based on the state-action pair
- ✓ **MC Epsilon Greedy** (MC Control)
  - without Exploring Starts
- ✓ …

Monte-Carlo Algorithm-Exploring Starts

Differences between control and prediction in RL and Monte Carlo methods

RL Prediction

- ✓ A prediction task in RL is when **policy $\boldsymbol{\pi}$ already is given** and we need to measure how well it performs.
- ✓ It means action already are fixed ($\pi(a, s)$)
- ✓ Only predict expected total reward for any state

RL Control

- ✓ In control task for RL the policy is not fixed and the goal is to find the optimal policy
- ✓ It means find $\pi(a, s)$ that maximizes expected reward

Monte-Carlo Algorithm

First-Visit Method MC Prediction

Input: policy π, number of episodes n_ep
Output: value function V (if n_ep is large enough $V \approx v_\pi$; means the prediction is accurate by sampling)
Initialize: Returns(s) = 0 for all $s \in S$
Initialize: $N(s)$ = 0 for all $s \in S$
for episode e = 1 **to** n_ep **do**
    Generate an episode $(s_0, a_0, r_0), (s_1, a_1, r_1), \dots, (s_{T-1}, a_{T-1}, r_T)$ using policy π (T number of steps)
    $G = 0$
    for time step t = T – 1 **to** 0 **do** (each state of the episode)
        $G = \gamma G + R_{t+1}$
        if search(S_t) == false **then** (search from start to see if S_t is not existing at that episode anymore (finished → first occur of S_t)
            Returns(S_t) = Returns(S_t) + G (per each episode the return of each S_t will be collected)
            $N(S_t) = N(S_t) + 1$
        end if
    end for
end for
$\mathrm{V}(s) = \mathrm{Returns}(s)/N(s)$, for all $s \in S$
return (V)

Function bool search (state S_t)
    for $state = 0$ to $t - 1$ in episode do
        if episode(state) == S_t then **return** (true) (search for S_t in generated episode steps one by one)
        else **return** (false)

Monte-Carlo Algorithm - First-Visit (Prediction)

Example

Generate an episodes ($T = 3$):

$E_1: (s_4, U, r_4) \rightarrow (s_1, R, r_1) \rightarrow (s_2, L, r_2) \rightarrow (s_1, stop, r_1)$

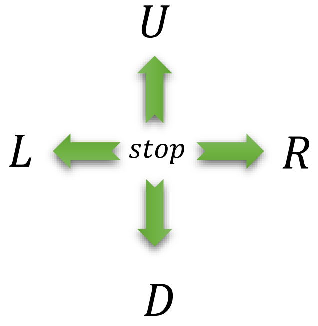
$E_2: (s_3, U, r_3) \rightarrow (s_0, stop, r_0)$

$E_3: (s_5, L, r_5) \rightarrow (s_4, U, r_4) \rightarrow (s_1, R, r_1) \rightarrow (s_2, stop, r_2)$

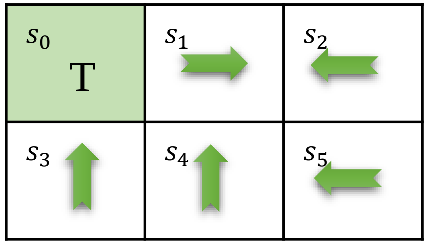
Environment

Terminal

s_0	s_1	s_2
s_3	s_4	s_5



Given Policy π



reward

+50	-1	-3
-1	-2	-4

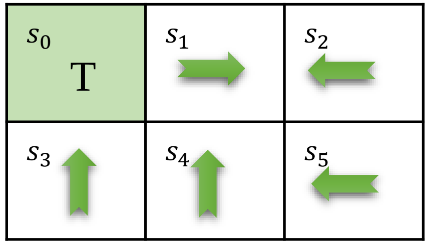
Monte-Carlo Algorithm - First-Visit (Prediction)

Example

Episodes:

$E_1: (s_4, U, r_4) \rightarrow (s_1, R, r_1) \rightarrow (s_2, L, r_2) \rightarrow (s_1, stop, r_1)$
$E_2: (s_3, U, r_3) \rightarrow (s_0, stop, r_0)$
$E_3: (s_5, L, r_5) \rightarrow (s_4, U, r_4) \rightarrow (s_1, R, r_1) \rightarrow (s_2, stop, r_2)$

Given Policy π



reward

+50	-1	-3
-1	-2	-4

First Visit Monte Carlo:

✓ First visit estimates (Value | State: s_t) as the average of the returns **following the first visit** to the state s_t

Every Visit Monte Carlo:

✓ It estimates (Value | State: s_t) as the average of returns for **every visit to** the State s_t.

Monte-Carlo Algorithm - First-Visit (Prediction)

Example

$$G = \gamma G + R_{t+1}$$

Given Policy π

s_0 T	s_1 →	s_2 ←
s_3 ↑	s_4 ↑	s_5 ←

reward

+50	-1	-3
-1	-2	-4

Episodes:

$E_1: (s_4, U, r_4) \rightarrow (\boldsymbol{s_1}, \boldsymbol{R}, \boldsymbol{r_1}) \rightarrow (s_2, L, r_2) \rightarrow (s_1, stop, r_1)$

$E_2: (s_3, U, r_3) \rightarrow (s_0, stop, r_0)$

$E_3: (s_5, L, r_5) \rightarrow (s_4, U, r_4) \rightarrow (\boldsymbol{s_1}, \boldsymbol{R}, \boldsymbol{r_1}) \rightarrow (s_2, stop, r_2)$

First Visit Monte Carlo:

✓ Summing all the rewards coming after the first visit to (S_t). **Here $\boldsymbol{s_1}$** (for simplicity $\gamma = 1$).

For episode E_1: $G = ((0 \times 1) + (-3)) + (-1) \rightarrow G = -4$

For episode E_2: Null

For episode E_3: $G = -3$

$$V(s_1) = \frac{-4 - 3}{2} = -4.5$$

Value function V

T	**-4.5**	0
0	0	0

Note: if an episode doesn't have an occurrence of $\boldsymbol{s_1}$, it won't be considered in the average

Monte-Carlo Algorithm - First-Visit (Prediction)

Example

Episodes:

$E_1: (s_4, U, r_4) \rightarrow (s_1, R, r_1) \rightarrow (\boldsymbol{s_2}, \boldsymbol{L}, \boldsymbol{r_2}) \rightarrow (s_1, stop, r_1)$

$E_2: (s_3, U, r_3) \rightarrow (s_0, stop, r_0)$

$E_3: (s_5, L, r_5) \rightarrow (s_4, U, r_4) \rightarrow (s_1, R, r_1) \rightarrow (\boldsymbol{s_2}, \boldsymbol{stop}, \boldsymbol{r_2})$

Given Policy π

s_0 T	s_1 →	s_2 ←
s_3 ↑	s_4 ↑	s_5 ←

reward

+50	-1	-3
-1	-2	-4

First Visit Monte Carlo:

✓ Calculations for $\boldsymbol{s_2}$

For episode E_1: $G = (-1) \rightarrow G = -1$

For episode E_2: Null

For episode E_3: $G = 0 \rightarrow G = 0$

$$V(s_2) = \frac{-1 + 0}{2} = -0.5$$

Value function V

T	-4.5	**-0.5**
0	0	0

Monte-Carlo Algorithm - First-Visit (Prediction)

Example

Episodes:

$E_1: (s_4, U, r_4) \rightarrow (s_1, R, r_1) \rightarrow (s_2, L, r_2) \rightarrow (s_1, stop, r_1)$
$E_2: (\boldsymbol{s_3}, \boldsymbol{U}, \boldsymbol{r_3}) \rightarrow (s_0, stop, r_0)$
$E_3: (s_5, L, r_5) \rightarrow (s_4, U, r_4) \rightarrow (s_1, R, r_1) \rightarrow (s_2, stop, r_2)$

Given Policy π

s_0 T	s_1 →	s_2 ←
s_3 ↑	s_4 ↑	s_5 ←

reward

+50	-1	-3
-1	-2	-4

First Visit Monte Carlo:

✓ Calculations for $\boldsymbol{s_3}$

For episode E_1: Null
For episode E_2: $G = 50$
For episode E_3: Null

$$V(s_3) = \frac{50}{1} = 50$$

Value function V

T	-4.5	-0.5
50	0	0

Monte-Carlo Algorithm - First-Visit (Prediction)

Example

Episodes:

E_1: $(\boldsymbol{s_4}, \boldsymbol{U}, \boldsymbol{r_4}) \rightarrow (s_1, R, r_1) \rightarrow (s_2, L, r_2) \rightarrow (s_1, stop, r_1)$
E_2: $(s_3, U, r_3) \rightarrow (s_0, stop, r_0)$
E_3: $(s_5, L, r_5) \rightarrow (\boldsymbol{s_4}, \boldsymbol{U}, \boldsymbol{r_4}) \rightarrow (s_1, R, r_1) \rightarrow (s_2, stop, r_2)$

First Visit Monte Carlo:

✓ Calculations for $\boldsymbol{s_4}$

For episode E_1: $G = (-1) + (-3) + (-1) \rightarrow G = -5$
For episode E_2: Null
For episode E_3: $G = (-1) + (-3) \rightarrow G = -4$

$$V(s_4) = \frac{-4-5}{2} = -4.5$$

Terminal

Given Policy π

s_0 T	s_1 →	s_2 ←
s_3 ↑	s_4 ↑	s_5 ←

reward

+50	-1	-3
-1	-2	-4

Value function V

T	-4.5	-0.5
50	**-4.5**	0

Monte-Carlo Algorithm - First-Visit (Prediction)

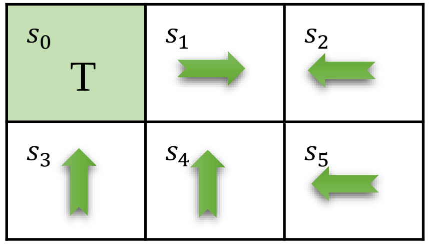
Example

Episodes:

E_1: $(s_4, U, r_4) \rightarrow (s_1, R, r_1) \rightarrow (s_2, L, r_2) \rightarrow (s_1, stop, r_1)$
E_2: $(s_3, U, r_3) \rightarrow (s_0, stop, r_0)$
E_3: $(\boldsymbol{s_5}, \boldsymbol{L}, \boldsymbol{r_5}) \rightarrow (s_4, U, r_4) \rightarrow (s_1, R, r_1) \rightarrow (s_2, stop, r_2)$

Terminal

Given Policy π



reward

+50	-1	-3
-1	-2	-4

First Visit Monte Carlo:

✓ Calculations for $\boldsymbol{s_5}$

For episode E_1: Null
For episode E_2: Null
For episode E_3: $G = (-2) + (-1) + (-3) \rightarrow G = -6$

$$V(5) = \frac{-6}{1} = -6$$

Value function V

T	-4.5	-0.5
50	-4.5	**-6**

Monte-Carlo Algorithm, Every-Visit (Prediction)

Every-Visit Method MC Prediction

Input: policy π, number of episodes n_ep
Output: value function V (if n_ep is large enough $V \approx v_\pi$; means the prediction is accurate by sampling)
Initialize: Returns(s) = 0 for all $s \in S$
Initialize: $N(s)$ = 0 for all $s \in S$
for episode e = 1 **to** n_ep **do**
  Generate an episode $(s_0, a_0, r_0), (s_1, a_1, r_1), \ldots, (s_{T-1}, a_{T-1}, r_T)$ using policy π (T number of steps)
  $G = 0$
  for time step t = T – 1 **to** 0 in episode e **do** (each state of the episode)
    $G = \gamma G + R_{t+1}$
    Returns(S_t) = Returns(S_t) + G (per each episode the return of each S_t will be collected,
    S_t may visited multiple times and add its <u>following states in episode will be summed again</u>)
    $N(S_t) = N(S_t) + 1$ (per each time visit of S_t its visiting time will be increased,
    so automatically if one state is not in a episode it's counter will not increased)
  end for
end for
V(s) = Returns$(s)/N(s)$, for all $s \in S$
return (V)

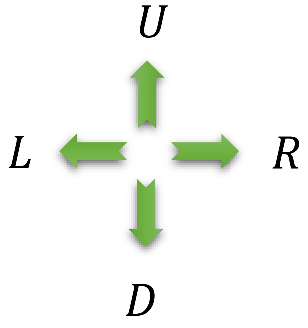
Monte-Carlo Algorithm, Every-Visit (Prediction)

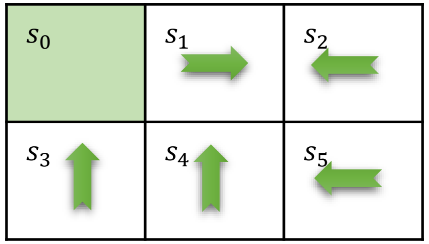
Same Example

Environment

Terminal

s_0	s_1	s_2
s_3	s_4	s_5





Given Policy π

+50	-1	-3
-1	-2	-4

reward

Generate an episode ($T = 4$): (We did 3 times to see)

$E_1: (s_4, U, r_4) \rightarrow (s_1, R, r_1) \rightarrow (s_2, L, r_2) \rightarrow (s_1, R, r_1) \rightarrow (s_2, stop, r_2)$

$E_2: (s_3, U, r_3) \rightarrow (s_0, stop, r_0)$

$E_3: (s_5, L, r_5) \rightarrow (s_4, U, r_4) \rightarrow (s_1, R, r_1) \rightarrow (s_2, L, r_2) \rightarrow (s_1, stop, r_1)$

First Visit Monte Carlo:

- ✓ First visit estimates (Value | State: s_t) as the average of the returns **following the first visit** to the state $\boldsymbol{s_t}$

Every Visit Monte Carlo:

- ✓ It estimates (Value | State: s_t) as the average of returns for **every visit to** the State $\boldsymbol{s_t}$.

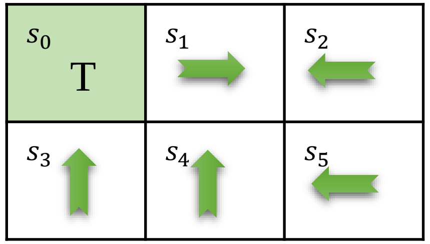
Monte-Carlo Algorithm, Every-Visit (Prediction)

Example

$$G = \gamma G + R_{t+1}$$

Given Policy π

Terminal



Episodes:

E_1: $(s_4, U, r_4) \rightarrow (\boldsymbol{s_1}, \boldsymbol{R}, \boldsymbol{r_1}) \rightarrow (s_2, L, r_2) \rightarrow (\boldsymbol{s_1}, \boldsymbol{R}, \boldsymbol{r_1}) \rightarrow (s_2, stop, r_2)$

E_2: $(s_3, U, r_3) \rightarrow (s_0, stop, r_0)$

E_3: $(s_5, L, r_5) \rightarrow (s_4, U, r_4) \rightarrow (\boldsymbol{s_1}, \boldsymbol{R}, \boldsymbol{r_1}) \rightarrow (s_2, L, r_2) \rightarrow (\boldsymbol{s_1}, \boldsymbol{stop}, \boldsymbol{r_1})$

reward

+50	-1	-3
-1	-2	-4

Every Visit Monte Carlo:

✓ Summing all the rewards coming after the first visit to (S_t). **Here $\boldsymbol{s_1}$** (for simplicity $\gamma = 1$).

For episode $\boldsymbol{E_1}$: $G = ((1 \times 0) + (-3)) \rightarrow G = -3$

For episode $\boldsymbol{E_1}$: $G = (-3) + (-1) + (-3) \rightarrow G = -7$

For episode E_2: Null

For episode $\boldsymbol{E_3}$: $G = (0) \rightarrow G = 0$

For episode $\boldsymbol{E_3}$: $G = (-3) + (-1) \rightarrow G = -4$

$$V(s_1) = \frac{(-3) + (-7) + (0) + (-4)}{4} = -3.5$$

Note again: if an episode doesn't have an occurrence of $\boldsymbol{s_1}$, it won't be considered in the average

Monte-Carlo Algorithm, Every-Visit (Prediction)

Example

Episodes:

$E_1: (s_4, U, r_4) \rightarrow (s_1, R, r_1) \rightarrow (\boldsymbol{s_2, L, r_2}) \rightarrow (s_1, L, r_1) \rightarrow (\boldsymbol{s_2, stop, r_2})$

$E_2: (s_3, U, r_3) \rightarrow (s_0, stop, r_0)$

$E_3: (s_5, L, r_5) \rightarrow (s_4, U, r_4) \rightarrow (s_1, R, r_1) \rightarrow (\boldsymbol{s_2, L, r_2}) \rightarrow (s_1, stop, r_1)$

Terminal

Given Policy π

s_0 T	s_1 →	s_2 ←
s_3 ↑	s_4 ↑	s_5 ←

reward

+50	-1	-3
-1	-2	-4

First Visit Monte Carlo:

✓ Calculations for $\boldsymbol{s_2}$

For episode E_1: $G = (0) \rightarrow G = 0$

For episode E_1: $G = (-1) + (-3) \rightarrow G = -4$

For episode E_2: Null

For episode E_3: $G = -1 \rightarrow G = -1$

$$V(s_2) = \frac{0 + (-4) + (-1)}{3} = -1.66$$

Value function V

T	**-3.5**	**-1.66**
0	0	0

Monte-Carlo Algorithm, Every-Visit (Prediction)

Example

Episodes:

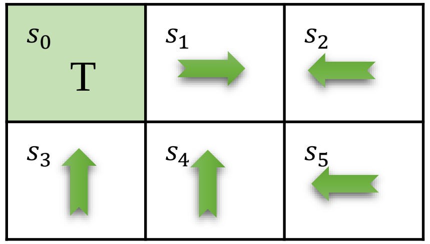
$E_1: (s_4, U, r_4) \rightarrow (s_1, R, r_1) \rightarrow (s_2, L, r_2) \rightarrow (s_1, L, r_1) \rightarrow (s_2, stop, r_2)$

$E_2: (\boldsymbol{s_3, U, r_3}) \rightarrow (s_0, stop, r_0)$

$E_3: (s_5, L, r_5) \rightarrow (s_4, U, r_4) \rightarrow (s_1, R, r_1) \rightarrow (s_2, L, r_2) \rightarrow (s_1, stop, r_1)$

Given Policy π

Terminal



reward

+50	-1	-3
-1	-2	-4

First Visit Monte Carlo:

✓ Calculations for $\boldsymbol{s_3}$

For episode E_1: Null

For episode E_2: $G = 50$

For episode E_3: Null

$$V(s_3) = \frac{50}{1} = 50$$

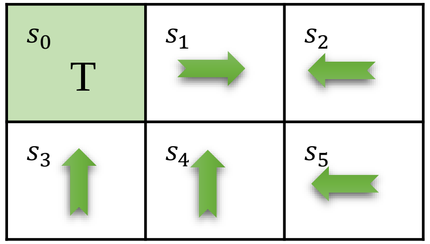
Value function V

T	-3.5	-1.66
50	0	0

Monte-Carlo Algorithm, Every-Visit (Prediction)

Example

Terminal



Given Policy π

Episodes:

E_1: $(\boldsymbol{s_4}, \boldsymbol{U}, \boldsymbol{r_4}) \rightarrow (s_1, R, r_1) \rightarrow (s_2, L, r_2) \rightarrow (s_1, L, r_1) \rightarrow (s_2, stop, r_2)$

E_2: $(s_3, U, r_3) \rightarrow (s_0, stop, r_0)$

E_3: $(s_5, L, r_5) \rightarrow (\boldsymbol{s_4}, \boldsymbol{U}, \boldsymbol{r_4}) \rightarrow (s_1, R, r_1) \rightarrow (s_2, L, r_2) \rightarrow (s_1, stop, r_1)$

reward

+50	-1	-3
-1	-2	-4

First Visit Monte Carlo:

✓ Calculations for $\boldsymbol{s_4}$

For episode E_1: $G = (-1) + (-3) + (-1) + (-3) \rightarrow G = -8$
For episode E_2: Null
For episode E_3: $G = (-1) + (-3) + (-1) \rightarrow G = -5$

$$V(s_4) = \frac{-8-5}{2} = -6.5$$

Value function V

T	-3.5	-1.66
50	**-6.5**	0

Monte-Carlo Algorithm, Every-Visit (Prediction)

Example

Episodes:

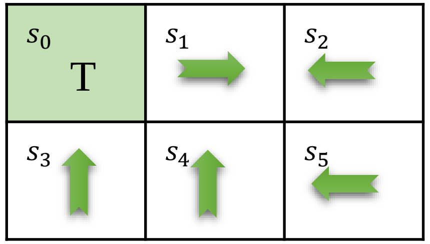
$E_1: (s_4, U, r_4) \rightarrow (s_1, R, r_1) \rightarrow (s_2, L, r_2) \rightarrow (s_1, L, r_1) \rightarrow (s_2, stop, r_2)$

$E_2: (s_3, U, r_3) \rightarrow (s_0, stop, r_0)$

$E_3: (\boldsymbol{s_5}, \boldsymbol{L}, \boldsymbol{r_5}) \rightarrow (s_4, U, r_4) \rightarrow (s_1, R, r_1) \rightarrow (s_2, L, r_2) \rightarrow (s_1, stop, r_1)$

Given Policy π

Terminal



reward

+50	-1	-3
-1	-2	-4

First Visit Monte Carlo:

✓ Calculations for $\boldsymbol{s_5}$

For episode E_1: Null

For episode E_2: Null

For episode E_3: $G = (-2) + (-1) + (-3) + (-1) \rightarrow G = -7$

$$V(5) = \frac{-7}{1} = -7$$

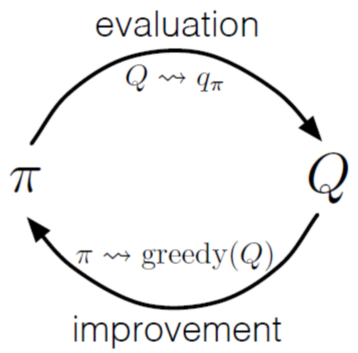
Value function V

T	-3.5	-1.66
50	-6.5	**-7**

Monte-Carlo Algorithm-Exploring Starts (Control)

Exploring Starts Method

Monte Carlo Exploring Starts (MC-ES), for estimating $\pi \approx \pi_*$



Initialize:

- $\pi(s) \in \mathcal{A}(s)$ (arbitrarily), for all $s \in \mathcal{S}$
- $Q(s, a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$
- $Returns(s, a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Loop forever (for each episode):

- Choose $S_0 \in \mathcal{S}$, $A_0 \in \mathcal{A}(S_0)$ randomly such that all pairs have probability > 0
- Generate an episode from S_0, A_0, following π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
- $G \leftarrow 0$
- Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
  - $G \leftarrow \gamma G + R_{t+1}$
  - Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:
    - Append G to $Returns(S_t, A_t)$
    - $Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$
    - $\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$

Monte-Carlo Algorithm-Exploring Starts (Control)

Follow the steps

a) Assign arbitrary value for Q-values and Policy first

Repeat Until Converge (no policy change) – optimal policy $\boldsymbol{\pi}_*$

b) Select random state s and action a pair in the environment

c) Generate an episode by policy π (T steps)

- ➢ Start from selected state s and run selected action a

d) For each state-action pair compute the discounted returns and create list

- ➢ Return is calculation of only all next states of each state in episode

$$V_\pi(s) = E_\pi[G(s)], G = \gamma g + R_{t+1}$$

e) Average the value of list and update the Q-value for state-action pairs

$$Q(s_i, a_j) = AVG(List(s_i, a_j))$$

f) Update the Policy

$$\pi(s) = argmax_a Q(s, a)$$

Monte-Carlo Exploring Starts Algorithm (Example)

Example Environment path planning

Problem: A robot needs to go from Start to Target
Reward: +100 for target, -1 for each other step
Q-value: for simplicity all zero

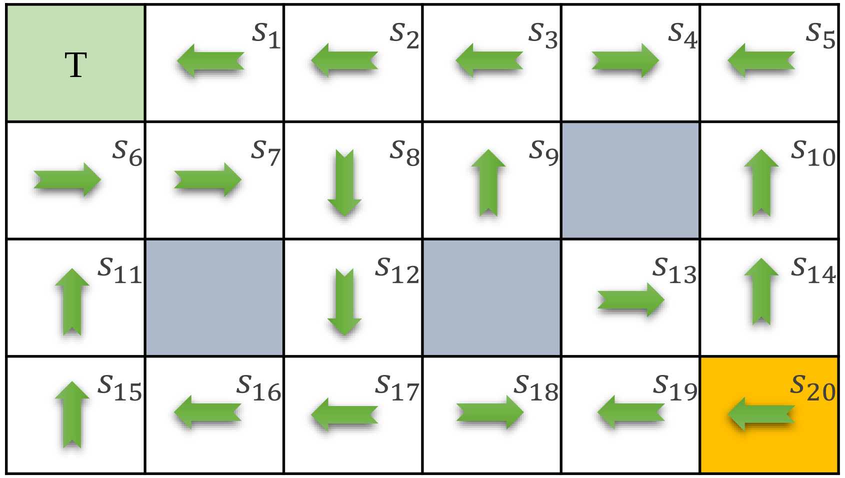
Target

s_0	s_1	s_2	s_3	s_4	s_5
s_6	s_7	s_8	s_9		s_{10}
s_{11}		s_{12}		s_{13}	s_{14}
s_{15}	s_{16}	s_{17}	s_{18}	s_{19}	s_{20}

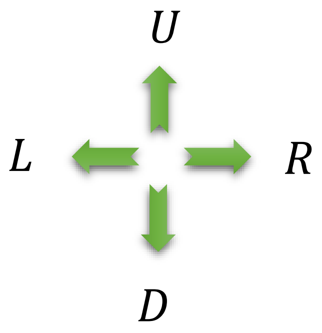
Start

Monte-Carlo Algorithm-Exploring Starts (Example)

a) We assign arbitrary Policy (up, down, left, right) and value for Q-values first.



Arbitrary Policy

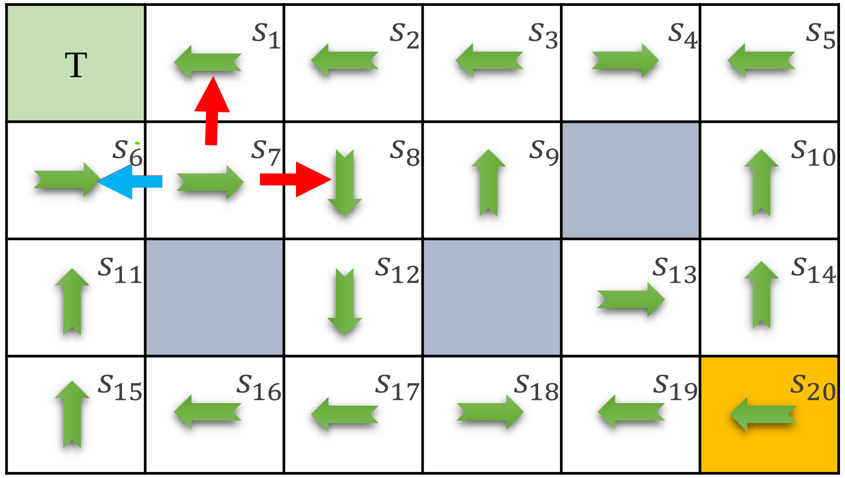


Monte-Carlo Algorithm-Exploring Starts (Example)

b) Select **random state *s*** and **random action *a*** pair in the environment

Note: Consider only possible actions from each state

(s_7, L)

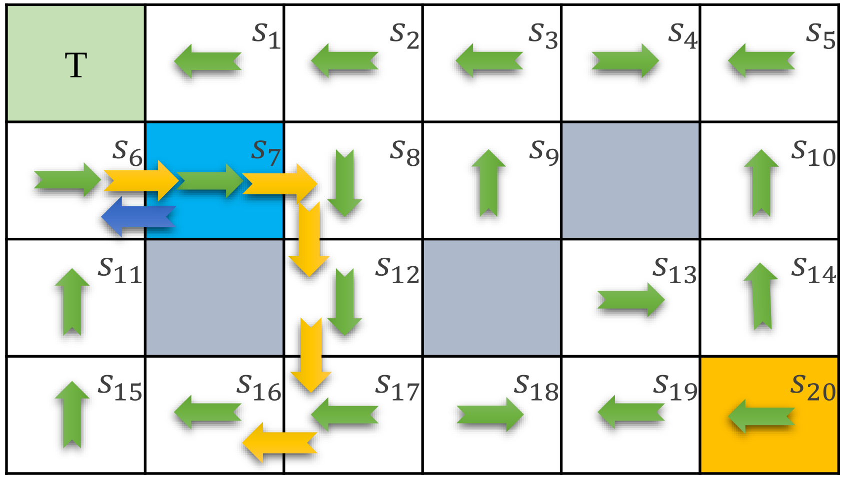


Monte-Carlo Algorithm-Exploring Starts (Example)

c) Generate an episode by policy π (T steps, 6 here)

- ➢ Start from **selected random state** $\boldsymbol{s}$ and run **selected random action** $\boldsymbol{a}$ first

E1 **Generate episode (1):** $(s_7\,, L)$



$$(\boldsymbol{s_7}\,, \boldsymbol{L}) \rightarrow (s_6\,, R) \rightarrow (s_7\,, R) \rightarrow (s_8\,, D) \rightarrow (s_{12}\,, D) \rightarrow (s_{17}\,, L) \rightarrow (s_{16}\,, L)$$

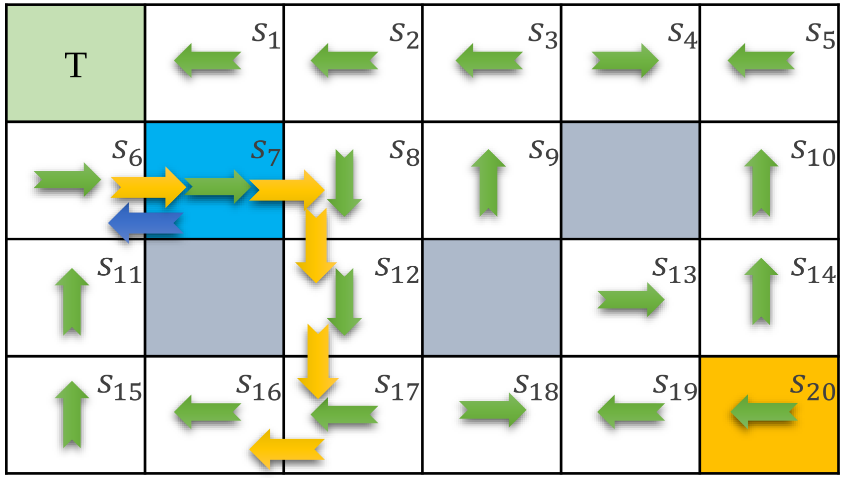
Monte-Carlo Algorithm-Exploring Starts (Example)

d) For each state-action pair compute the discounted returns and create list:

- Return is calculation of only all next states of each state in episode

Discounted returns: $\boxed{G = \gamma G + R_{t+1}}$ $\gamma = 0.9$

$$(\boldsymbol{s_7}, \boldsymbol{L}) \rightarrow (s_6, R) \rightarrow (s_7, R) \rightarrow (s_8, D) \rightarrow (s_{12}, D) \rightarrow (s_{17}, L) \rightarrow (s_{16}, L)$$



$(s_{17}, L) = (0.9 \times 0) - 1 = -1$
$(s_{12}, D) = (0.9 \times -1) - 1 =$ -1.9
$(s_8, D) = (0.9 \times -1.9) - 1 =$ -2.71
$(s_7, R) = (0.9 \times -2.71) - 1 =$ -3.439
$(s_6, R) = (0.9 \times -3.349) - 1 =$ -4.095
$(s_7, L) = (0.9 \times -4.095) - 1 =$ -4.685

Returns_list(s_{17}, L)=[-1]
Returns_list(s_{12}, D)=[-1.9]
Returns_list(s_8, D)=[-2.71]
Returns_list(s_7, R)=[-3.439]
Returns_list(s_6, R)=[-4.095]
Returns_list(s_7, L)=[-4.685]

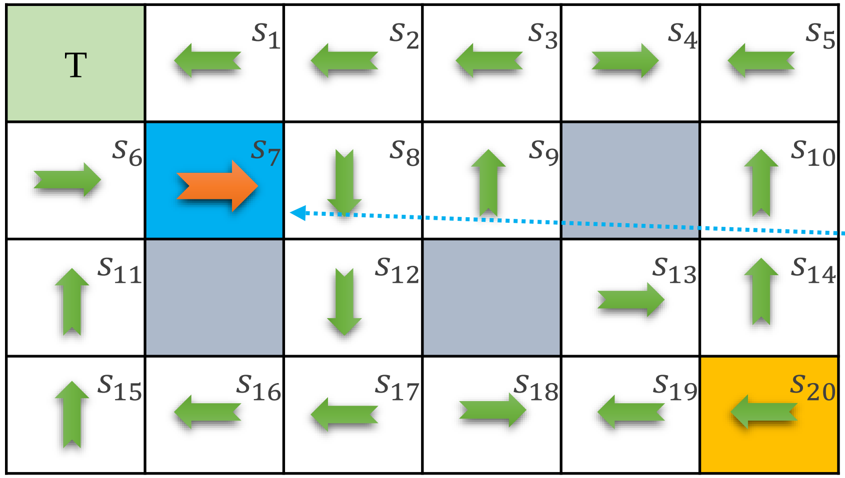
Note: since we do **did not run action for the last state**, do not need to include.

Monte-Carlo Algorithm-Exploring Starts (Example)

f) Update the Policy

$$\pi(s) = argmax_a Q(s,a)$$

Average of the list:



Update the Policy

Q(s_{17}, L)=[−1]
Q(s_{12}, L)=[−1.9]
Q(s_8, D)=[−2.71]
Q(s_7, R)=[−3.439]
Q(s_6, R)=[−4.095]
Q(s_7, L)=[−4.685]

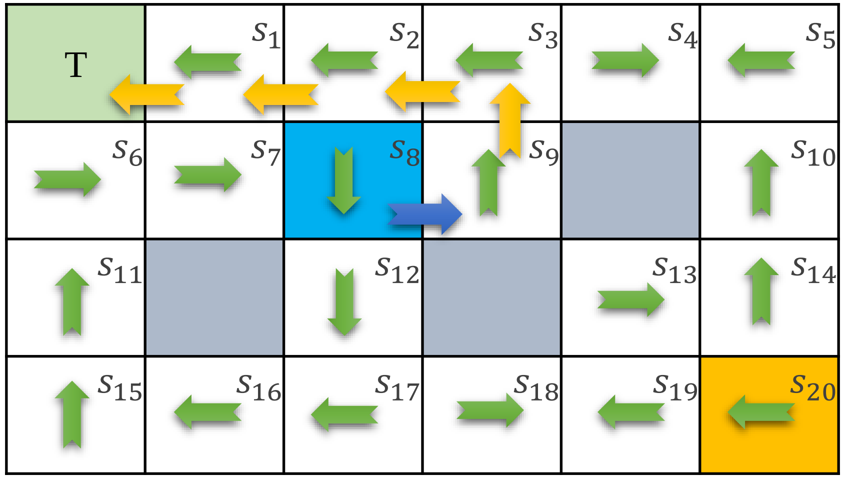
Returns_list(s_{17}, L)=[−1]
Returns_list(s_{12}, D)=[−1.9]
Returns_list(s_8, D)=[−2.71]
Returns_list(s_7, R)=[−3.439]
Returns_list(s_6, R)=[−4.095]
Returns_list(s_7, L)=[−4.685]

Monte-Carlo Algorithm-Exploring Starts (Example)

E2 Generate another episode by policy π (T steps, 6 here)

Random (s_8, R)

Generate episode (2):

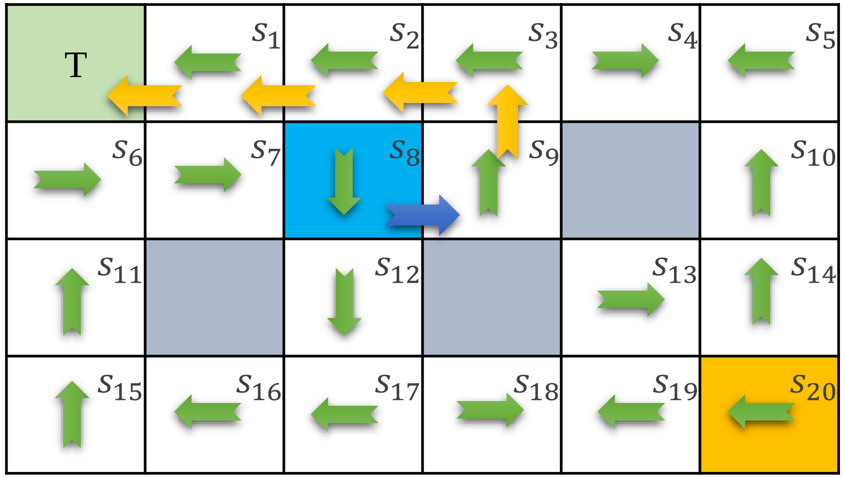


$$(\boldsymbol{s_8}, \boldsymbol{R}) \rightarrow (s_9, U) \rightarrow (s_3, L) \rightarrow (s_2, L) \rightarrow (s_1, L) \rightarrow (s_0, stop)$$

Monte-Carlo Algorithm-Exploring Starts (Example)

Discounted returns: $\boxed{G = \gamma G + R_{t+1}}$ $\gamma = 0.9$

$$(\boldsymbol{s_8}, \boldsymbol{R}) \rightarrow (s_9, U) \rightarrow (s_3, L) \rightarrow (s_2, L) \rightarrow (s_1, L) \rightarrow (s_0, stop)$$



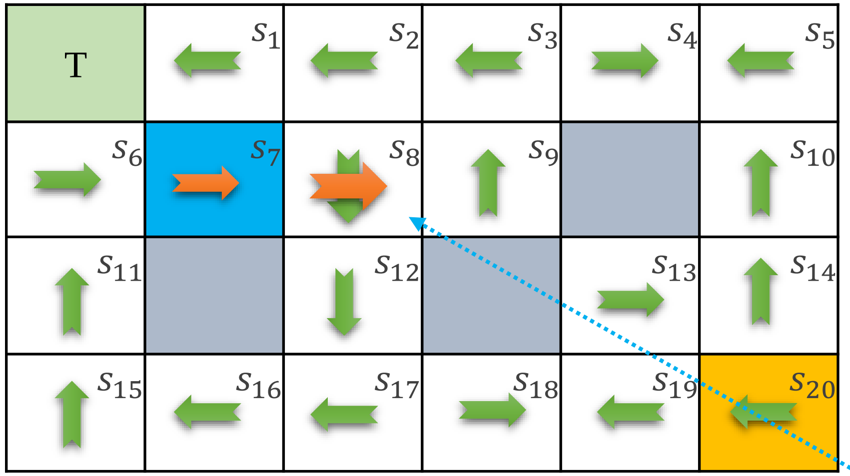
$(s_1, L) = (0.9 \times 0) + 100 = 100$
$(s_2, L) = (0.9 \times 100) - 1 = 89$
$(s_3, L) = (0.9 \times 89) - 1 = 79.1$
$(s_9, U) = (0.9 \times 79.1) - 1 = 70.19$
$(s_8, R) = (0.9 \times 70.19) - 1 = 62.171$

Returns_list(s_{17}, L)=[-1]
Returns_list(s_{12}, D)=[-1.9]
Returns_list(s_8, D)=[-2.71]
Returns_list(s_7, R)=[-3.439]
Returns_list(s_6, R)=[-4.095]
Returns_list(s_7, L)=[-4.685]
Returns_list(s_1, L)=[100]
Returns_list(s_2, L)=[89]
Returns_list(s_3, L)=[79.1]
Returns_list(s_9, U)=[70.19]
Returns_list(s_8, R)=[62.171]

Monte-Carlo Algorithm-Exploring Starts (Example)

f) Update the Policy

$$\pi(s) = argmax_a Q(s,a)$$



Update the Policy

Q(s_{17}, L)=[-1]
Q(s_{12}, D)=[-1.9]
$\boxed{\text{Q}(s_8, D)=[-2.71]}$
$\boxed{\text{Q}(s_7, R)=[-3.439]}$
Q(s_6, R)=[-4.095]
$\boxed{\text{Q}(s_7, L)=[-4.685]}$
Q(s_1, L)=[100]
Q(s_2, L)=[89]
Q(s_3, L)=[79.1]
Q(s_9, U)=[70.19]
$\boxed{\textbf{Q}(\boldsymbol{s_8}, \boldsymbol{R})=[\mathbf{62.171}]}$

Average of the list:

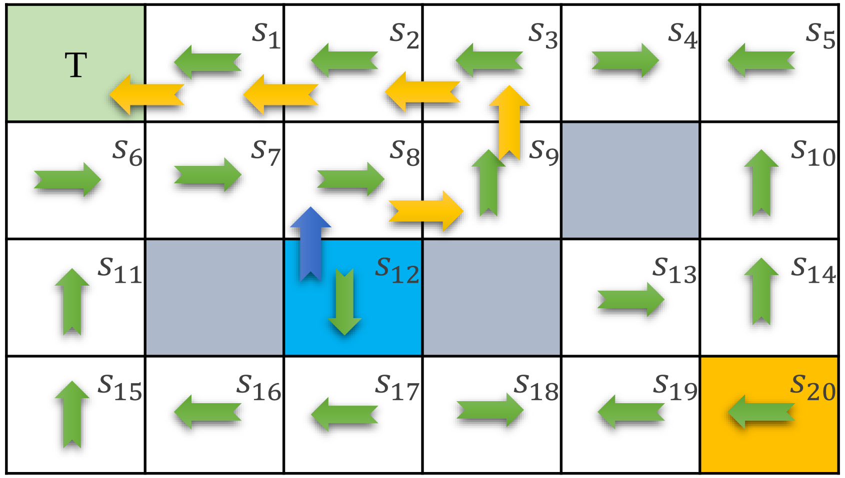
Returns_list(s_{17}, L)=[-1]
Returns_list(s_{12}, D)=[-1.9]
Returns_list(s_8, D)=[-2.71]
Returns_list(s_7, R)=[-3.439]
Returns_list(s_6, R)=[-4.095]
Returns_list(s_7, L)=[-4.685]
Returns_list(s_1, L)=[100]
Returns_list(s_2, L)=[89]
Returns_list(s_3, L)=[79.1]
Returns_list(s_9, U)=[70.19]
Returns_list(s_8, R)=[62.171]

Monte-Carlo Algorithm-Exploring Starts (Example)

E3 Generate another episode by policy π (T steps, 6 here)

(s_{12}, U)

Generate episode (3):

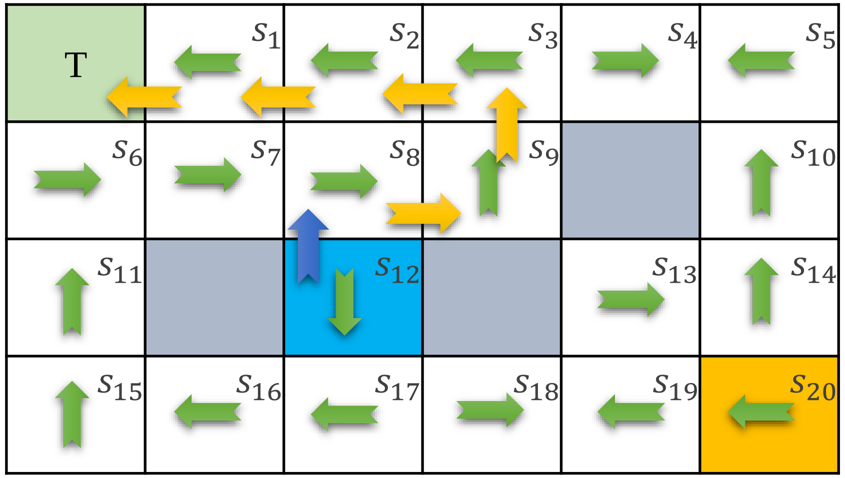


$$(\boldsymbol{s_{12}}, \boldsymbol{U}) \rightarrow (s_8, R) \rightarrow (s_9, U) \rightarrow (s_3, L) \rightarrow (s_2, L) \rightarrow (s_1, L) \rightarrow (s_0, stop)$$

Monte-Carlo Algorithm-Exploring Starts (Example)

Discounted returns: $\boxed{G = \gamma G + R_{t+1}}$ $\gamma = 0.9$

$$(\boldsymbol{s_{12}}, \boldsymbol{U}) \rightarrow (s_8, R) \rightarrow (s_9, U) \rightarrow (s_3, L) \rightarrow (s_2, L) \rightarrow (s_1, L) \rightarrow (s_0, stop)$$



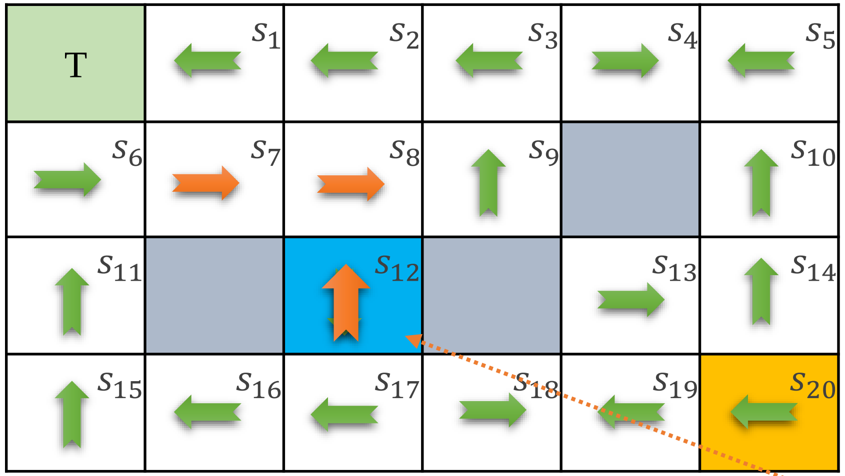
$$(s_1, L) = (0.9 \times 0) + 100 = 100$$
$$(s_2, L) = (0.9 \times 100) - 1 = 89$$
$$(s_3, L) = (0.9 \times 89) - 1 = 79.1$$
$$(s_9, U) = (0.9 \times 79.1) - 1 = 70.19$$
$$(s_8, R) = (0.9 \times 70.19) - 1 = 62.171$$
$$(s_{12}, U) = (0.9 \times 62.171) - 1 = 54.953$$

Returns_list(s_{17}, L)=[-1]
Returns_list(s_{12}, D)=[-1.9]
Returns_list(s_8, D)=[-2.71]
Returns_list(s_7, R)=[-3.439]
Returns_list(s_6, R)=[-4.095]
Returns_list(s_7, L)=[-4.685]
Returns_list(s_1, L)=[100, **100**]
Returns_list(s_2, L)=[89, **89**]
Returns_list(s_3, L)=[79.1, **79.1**]
Returns_list(s_9, U)=[70.19, **70.19**]
Returns_list(s_8, R)=[62.171, $\boldsymbol{62.171}$]
Returns_list($\boldsymbol{s_{12}}$, $\boldsymbol{U}$)=[$\boldsymbol{54.953}$]

Monte-Carlo Algorithm-Exploring Starts (Example)

f) Update the Policy

$$\pi(s) = argmax_a Q(s,a)$$



Update the Policy

$Q(s\_17, L)=[-1]$
$Q(s\_12, D)=[-1.9]$
$Q(s\_8, D)=[-2.71]$
$\mathbf{Q(s\_7, R)=[-3.439]}$
$Q(s\_6, R)=[-4.095]$
$Q(s\_7, L)=[-4.685]$
$Q(s_1, L)=[100]$
$Q(s_2, L)=[89]$
$Q(s_3, L)=[79.1]$
$Q(s_9, U)=[70.19]$
$\mathbf{Q(s_8, R)=[62.171]}$
$\mathbf{Q(s_{12}, U)=[54.953]}$

Average of the list:

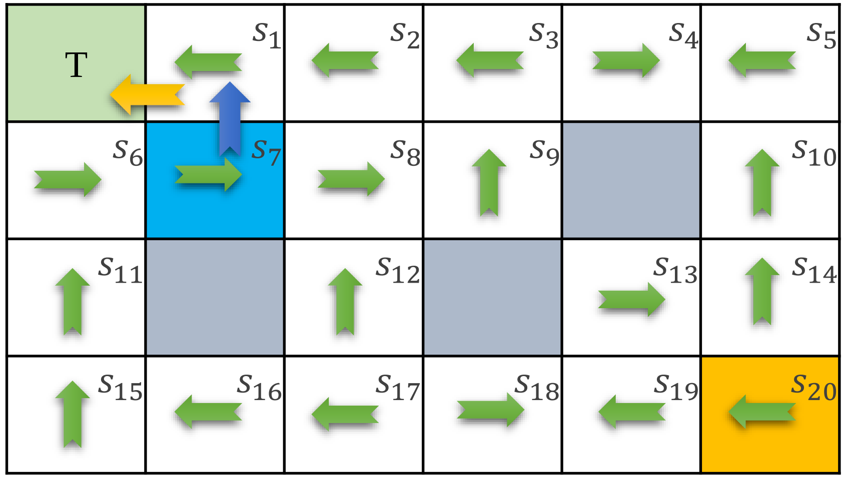
Returns_list(s_{17}, L)=[−1]
Returns_list(s_{12}, D)=[−1.9]
Returns_list(s_8, D)=[−2.71]
Returns_list(s_7, R)=[−3.439]
Returns_list(s_6, R)=[−4.095]
Returns_list(s_7, L)=[−4.685]
Returns_list(s_1, L)=[100, **100**]
Returns_list(s_2, L)=[89, **89**]
Returns_list(s_3, L)=[79.1, **79.1**]
Returns_list(s_9, U)=[70.19, **70.19**]
Returns_list(s_8, R)=[62.171, **62.171**]
Returns_list(s_{12}, U)=[54.953]

Monte-Carlo Algorithm-Exploring Starts (Example)

E4 Generate another episode by policy π (n steps, 6 here)

$(s_7\,, U)$

Generate an episode (4):

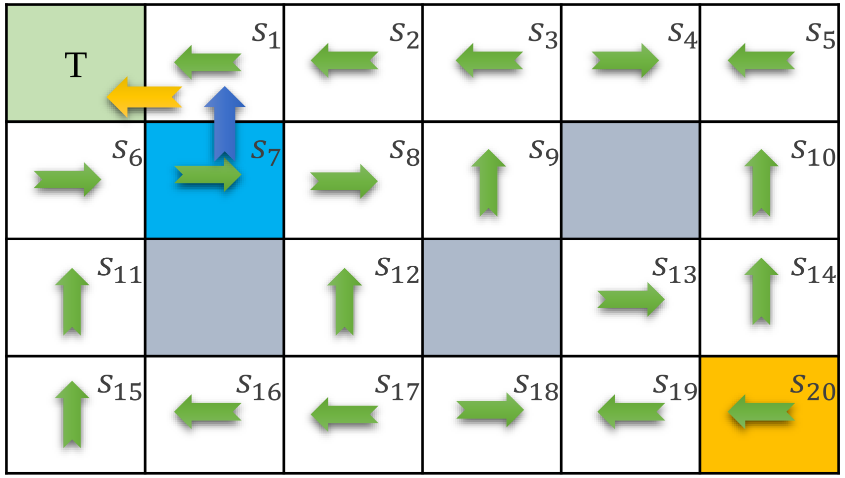


$$(\boldsymbol{s_7}\,, \boldsymbol{U}) \rightarrow (s_1\,, L) \rightarrow (s_0\,, stop)$$

Monte-Carlo Algorithm-Exploring Starts (Example)

Discounted returns: $\boxed{G = \gamma G + R_{t+1}}$ $\gamma = 0.9$

$$(\boldsymbol{s_7}, \boldsymbol{U}) \rightarrow (s_1, L) \rightarrow (s_0, stop)$$



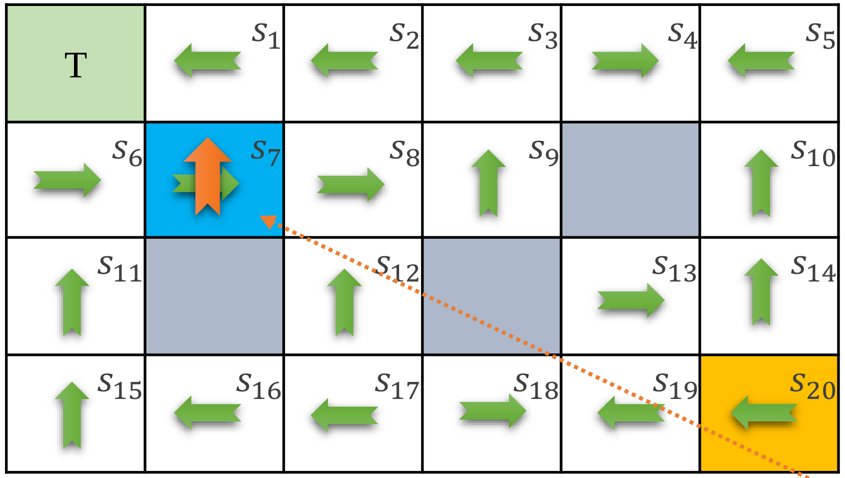
$(s_1, L) = (0.9 \times 0) + 100 = 100$
$(s_7, U) = (0.9 \times 100) - 1 = 89$

Returns_list(*s_17*, *L*)=[−1]
Returns_list(*s_12*, *D*)=[−1.9]
Returns_list(*s_8*, *D*)=[−2.71]
Returns_list(*s_7*, *R*)=[−3.439]
Returns_list(*s_6*, *R*)=[−4.095]
Returns_list(*s_7*, *L*)=[−4.685]
Returns_list(s_1, L)=[100, 100, **100**]
Returns_list(s_2, L)=[89, 89]
Returns_list(s_3, L)=[79.1, 79.1]
Returns_list(s_9, U)=[70.19, 70.19]
Returns_list(s_8, R)=[62.171, 62.171]
Returns_list(s_{12}, U)=[54.953]
Returns_list($\boldsymbol{s_7}$, $\boldsymbol{U}$)=[89]

Monte-Carlo Algorithm-Exploring Starts (Example)

f) Update Policy

$$\pi(s) = argmax_a Q(s,a)$$



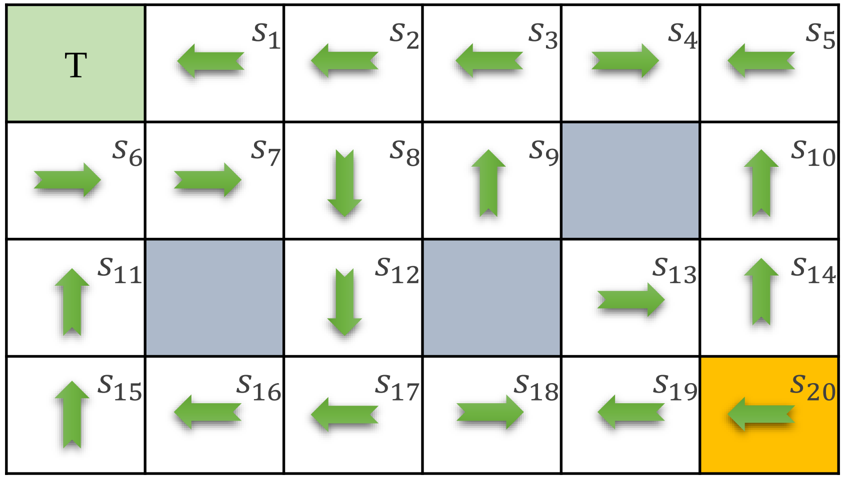
Update the Policy

Q(s_{17}, L)=[-1]
Q(s_{12}, D)=[-1.9]
Q(s_8, D)=[-2.71]
Q(s_7, R)=[-3.439]
Q(s_6, R)=[-4.095]
Q(s_7, L)=[-4.685]
Q(s_1, L)=[100]
Q(s_2, L)=[89]
Q(s_3, L)=[79.1]
Q(s_9, U)=[70.19]
Q(s_8, R)=[62.171]
Q(s_{12}, U)=[54.953]
Q(s_7, U)=[89]

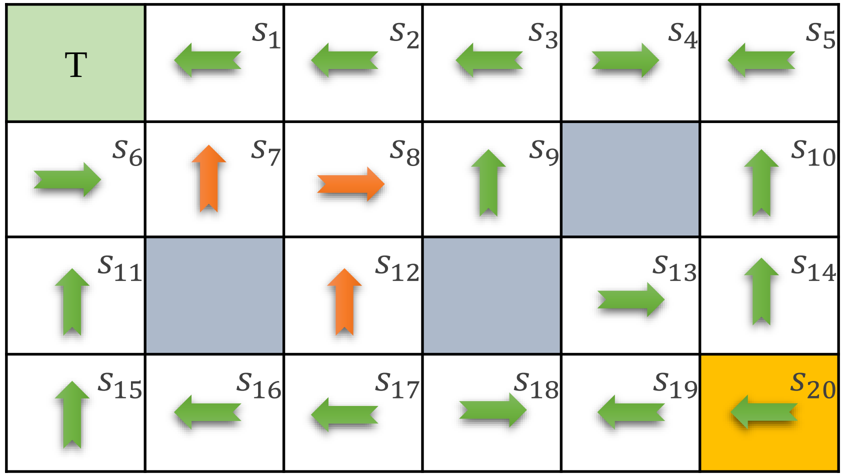
Average of the list:

Returns_list(s_{17}, L)=[-1]
Returns_list(s_{12}, D)=[-1.9]
Returns_list(s_8, D)=[-2.71]
Returns_list(s_7, R)=[-3.439]
Returns_list(s_6, R)=[-4.095]
Returns_list(s_7, L)=[-4.685]
Returns_list(s_1, L)=[100, 100, **100**]
Returns_list(s_2, L)=[89, 89]
Returns_list(s_3, L)=[79.1, 79.1]
Returns_list(s_9, U)=[70.19, 70.19]
Returns_list(s_8, R)=[62.171, 62.171]
Returns_list(s_{12}, U)=[54.953]
Returns_list(s_7, U)=[89]

Monte-Carlo Algorithm-Exploring Starts (Example)



Arbitrary Policy



Updated Policy after 4 episodes

MC Epsilon Greedy algorithm (without Exploring Starts): (Control)

Idea

- ✓ In some problems we can not calculate all edge cases.
  - ➢ for example in an application that resetting the environment always goes back to one state not random one!
- ✓ In such cases MC Exploring start becomes infeasible!

Solution

- ✓ We **eliminate randomly selection** for all **starting points** from MC Exploring Starts.
- ✓ **Apply random policy** sometimes by **Epsilon-Greedy** technique

MC Approaches

Problem

To update the value and policy we need to **wait until the end of episode**.

Is there any better idea?

- ✓ Solution is Temporal differences (TD) RL

Summery

- ✓ We discussed Model-Based and Model-Free Learnings
- ✓ We discussed State-value function and state-action function
- ✓ We undestood examples of First-Visit Method MC Prediction
- ✓ We undestood Every-Visit Method MC Prediction
- ✓ We undestood Monte Carlo Exploring Starts (MC-ES)
- ✓ We undestood MC Epsilon Greedy (without Exploring Starts)