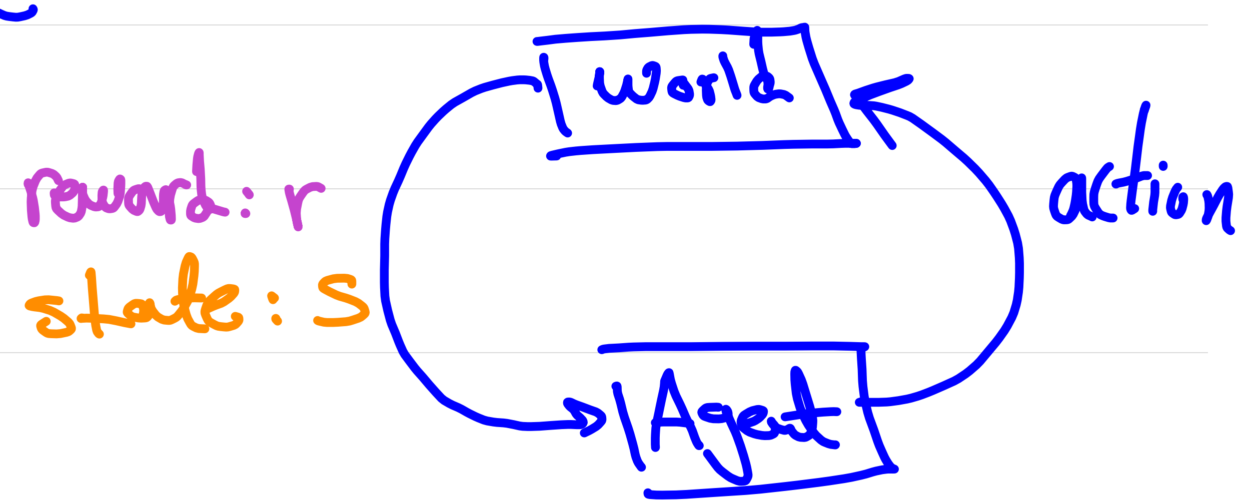
- Outline:
  - MDP V.S. RL
  - Active RL V.S. Passive RL
  - Active RL
    - Model based Learning
    - Model Free Learning
      - Direct Evaluation
      - Temporal Difference Learning
      - Q-Learning

RL: Unknown Transition and Reward Model

- We saw how MDP can be solved
- What if the environment model is not entirely Known?
  - How can we find a Policy?
- Here comes RL
  - Learning what?
    - The MDP model OR some parts of the model (e.g., value functions or Q-functions which are enough to find the optimal policy
  - Learning from what?
    - From Samples (aka episodes) of the process (i.e. transition of states)

Reinforcement Learning (RL)

- We still assume a MDP
  - (S, A, T, R, γ)
- We are still looking for a good policy
- We have a new challenge
  - We don't know T or R
    - Must explore new states and actions to observe environment
    - Basic Idea: Learn how to maximize expected rewards based on observed samples of transitions



MDP vs. RL

- MDP
  - Solving MDP is an offline problem.
    - The model is Known
  - The goal is to find a policy that collects maximum utility
- Reinforcement Learning
  - RL is an online problem.
    - The model is not Known.
  - The goal is to perform actions to learn the model and collect rewards

Passive RL vs. Active RL

- Passive RL:
  - How to Learn from already given experiences
    - Similar to supervise learning: learns from already given labeled datapoints
- Active RL:
  - How to collect new experience and learn from them

Passive RL

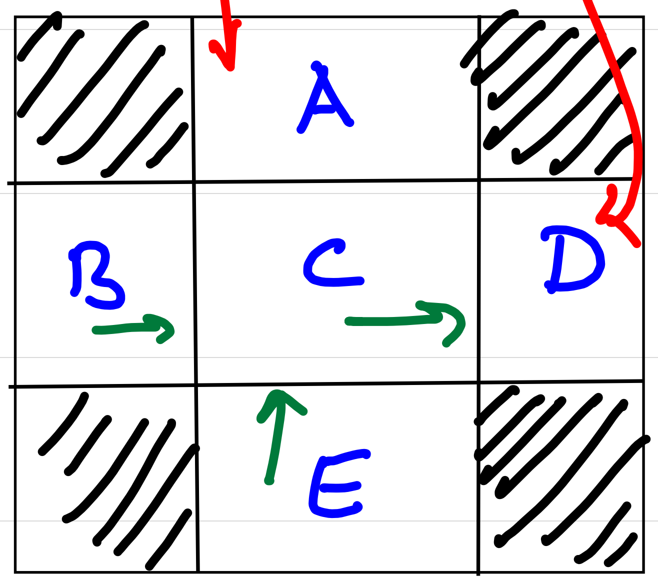
- Approaches to Passive RL
  - Model-Based Learning
    - Learn the MDP Dynamics (i.e. T) and Reward (i.e. R) from Samples
    - Then, Solve the MDP (i.e., use Policy iteration or value iteration)
  - Model-Free Learning
    - Directly learns $V(s)$ or $Q(s,a)$ from experience
    - Uses $V(s)$ or $Q(s,a)$ to make decisions

Model-based Learning

- Model-based learning:
  - Basic idea:
    - Learn an estimate MDP model (i.e. T) based on experience
    - Then, Solve the approximate MDP
- Step 1: Learn empirical MDP model $T(s,a,s') = P(s'|s,a)$
  - Count outcomes s' for each s,a
    - use the count to estimate $\hat{T}(s,a,s') = \hat{P}(s'|s,a)$
- Similarly estimate $\hat{R}(s,a,s')$
- Step 2: Solve the learned MDP (e.g., with value iteration or policy iteration)

Example: Model-based Learning Example

Terminal states



Input policy π

(Assume $\gamma = 1$)

Episode 1:

B, east, C, -1,
C, east, D, -1,
D, exit, "end", +10

Episode 2

B, east, C, -1,
C, east, D, -1,
D, exit, "end", +10

Episode 3

E, east, C, -1,
C, east, D, -1,
D, exit, "end", +10

Episode 4

E, east, C, -1,
C, east, A, -1,
A, exit, "end", 10

Learned Model

$\hat{T}(B, east, C) = 1$

$\hat{T}(C, east, D) = 3/4$

$\hat{T}(C, east, A) =$

$\vdots$

$\hat{R}(B, east, C) =$

$\hat{R}(C, east, D) =$

$\hat{R}(C, east, A) =$

$\vdots$

Pros and Cons of Model-based Learning

- Pro:
  - Makes efficient use of experience (low sample Complexity)
- Con:
  - May not scale to large state space
    - Learns model on state-action pair at a time
    - Cannot Solve MDP for very large $|S|$
  - Much harder when the environment is partially observable

Basic Idea Behind Model-Free Learning

- To approximate expectations w.r.t. a distribution, we can either
  - Estimate the distribution from samples, then compute the expectation based on the estimated distribution
  - or, bypass the distribution and estimate the expectation from the samples directly

Let's See an Example

- Consider the task of estimating the expected age of UofT students: $E[A]$
- If the probability distribution of A was known, we could find it easily: $E[A] = \sum_a P(a) \cdot a$
- Without $P(A)$, we have to collect samples: $[a_1, a_2, \ldots, a_N]$
  - Model-Based approach:
    - Estimat $P(A)$ first: $\hat{P}[A=a] = \frac{N_a}{N}$
    - Then we use $\hat{P}$ to estimate $E[A]$: $\hat{E}[A] = \sum_a \hat{P}[A=a] \cdot a$
  - Model-Free approach:
    - Use the samples to estimate $E[A]$: $\frac{\sum_i a_i}{N}$

Basic Idea Behind Model-Free Learning

- In RL, our ultimate goal is to find an estimate of expected return in each state.
  - Model-based Learning: Estimate the probability distribution of return from the samples. then use it to calculate the expected return.
  - Model-Free: estimate the expected return directly from samples.

Passive RL

- Simplified Passive RL model
  - Input: stream of transitions produced by following some fixed policy $\pi(s)$
    - E.g., we are given the following episodes:

      $(s, \pi(s), s', r, \pi(s'), r', \ldots, end)$

      $(s, \pi(s), s'', r, \pi(s''), r'', \ldots, end)$

      $\vdots$
  - Output: estimate of the state values $\hat{V}_{\pi}(s)$
  - Note: we don't know T and R

Direct Evaluation

- Consider the passive learning model described before, with the goal of estimating $V_\pi(s)$, i.e. expected total discounted reward from s onward.
- Direct Evaluation: It uses returns, the actual sums of discounted reward from s onward.
  - Average over multiple trials and visits to "s." Everytime you visit "S", find the sum of discounted reward from "S" to the end.

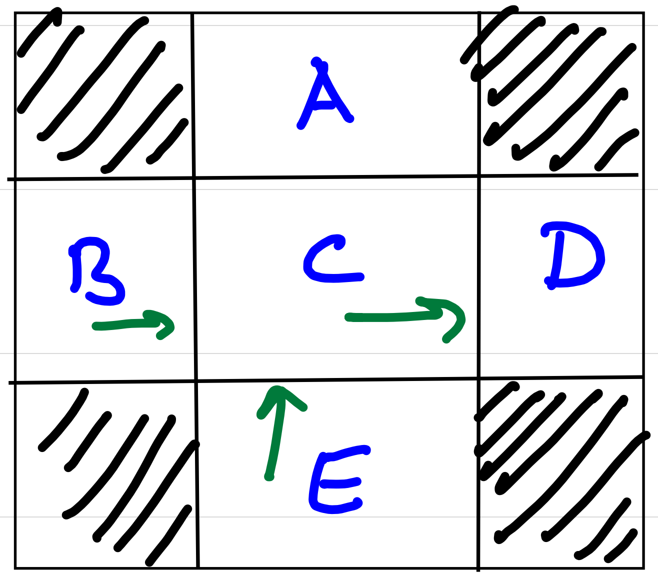
$$U_i(s) = R(s, \pi(s), s') + \gamma R(s', \pi(s'), s'') + \gamma^2 \dots$$

the i-th time you saw "S".

$$\hat{V}_\pi(s) = \frac{1}{N} \sum_i U_i(s)$$

This is also known as "direct utility estimation" or "Monte-Carlo evaluation"

Example: Direct Evaluation



Input policy π
(Assume $\gamma = 1$)

Episode 1:

B, east, C, -1 ,
C, east, D, -1 ,
D, exit, "end", +10

Episode 2

B, east, C, -1 ,
C, east, D, -1 ,
D, exit, "end", +10

Episode 3

E, east, C, -1 ,
C, east, D, -1 ,
D, exit, "end", +10

Episode 4

E, east, C, -1 ,
C, east, A, -1 ,
A, exit, "end", -10

Output Values

	-10	
+8	+4	10
	-2	

$$V_\pi(E) = \frac{U_1(E) + U_2(E)}{2} = \frac{\left(-1 + \gamma(-1) + \gamma^2(10)\right) + \left((-1) + \gamma(-1) + \gamma^2(-10)\right)}{2}$$

How come $\hat{V}_\pi(E) \neq \hat{V}_\pi(B)$?

Direct Evaluation Pros and Cons

- Pros:
  - It does not require any knowledge of T and R
  - It converges to the right answer in the limit.
- Cons:
  - It ignores information about state connections.
  - Each state must be learned separately
  - So, slow to learn.

How Can We Incorporate Information About state Connections?

- Direct Evaluation waits for each episode to end
  - Then it finds the sum of discounted return of state s for that episode
  - Then, it updates $\hat{V}_{\pi}(s)$
- why don't we use a similar approach to Policy Evaluatio?
  - Policy Evaluation uses the state Connection (by using Bellman equations):

$$V_{\pi,K+1}(s) \leftarrow \sum_{s'} T(s,\pi(s),s')\left[R(s,\pi(s),s') + \gamma V_{\pi,K}(s')\right]$$

  - Sadly, we don't have $R(\cdot)$ and $T(\cdot)$:(
    - Temporal Difference (TD) Learning to the rescue

Before TD, Let's See Some Naive Ideas

- Before we present TD, let's study some naive ideas to exploit state connections
  - Idea 1: Use actual samples to estimate the expectation
  - Idea 2: Update value of s after each transition s, a, s', r
- These two naive ideas help us better understand the design principle behind TD

Idea 1

- Let's take a second look at the state value recursion relation:

$$V_\pi(s) = \sum_{s'} T(s,\pi(s),s')\left[R(s,\pi(s),s') + \gamma V_\pi(s')\right]$$

$$= \mathbb{E}_{s' \mid s,\pi(s)}\left[R(s,\pi(s),s') + \gamma V_\pi(s')\right]$$

- Hence, to estimate $V_\pi(s)$, we must estimate the expectation above.
  - Just like what we saw earlier, estimate the expected value of a random variable by finding average of its realizations
- Next Lecture: Idea 2, TD, Q-Learning, ...