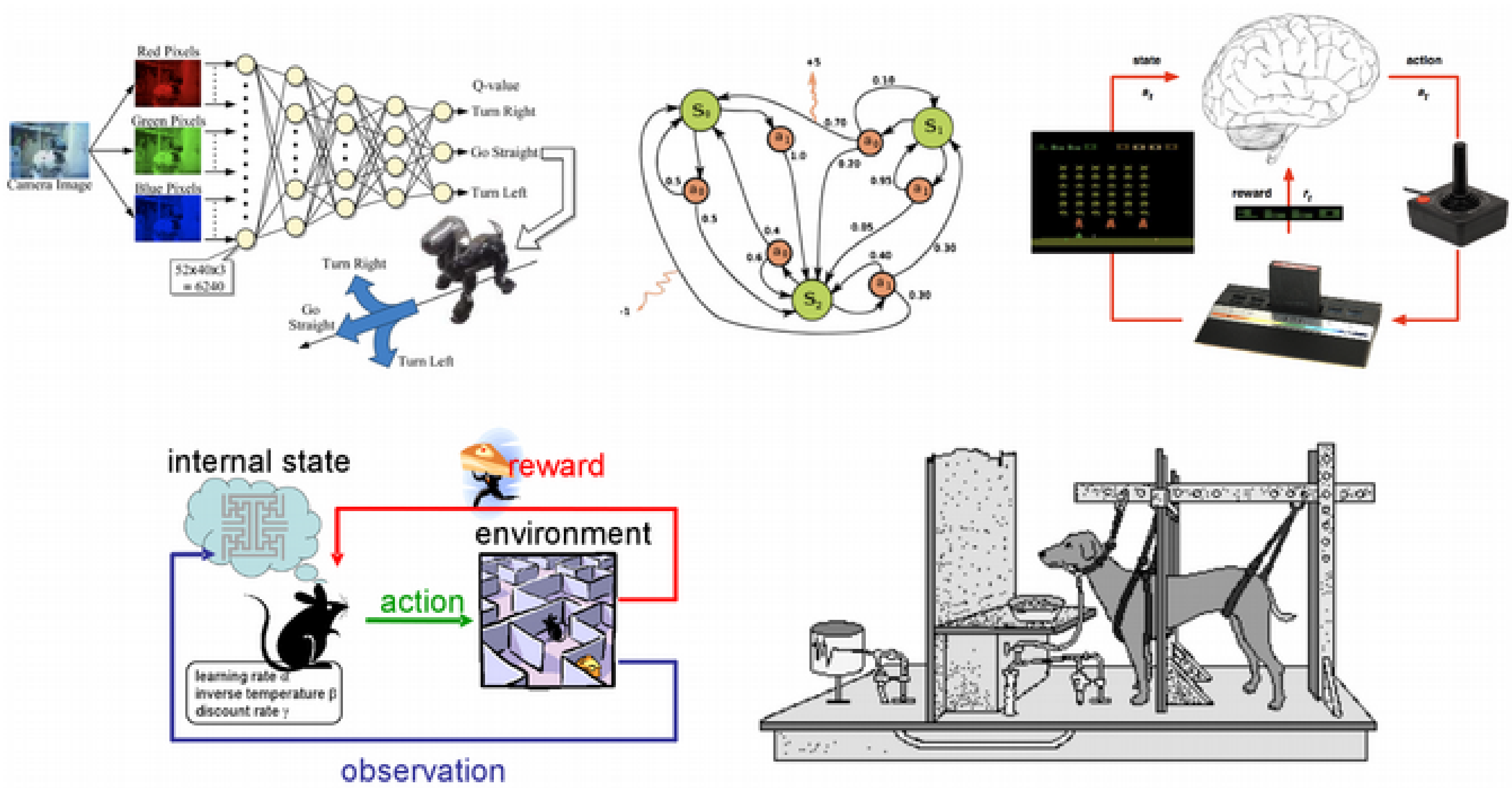
COMP 138: Reinforcement Learning



Instructor: Jivko Sinapov
Webpage: https://www.eecs.tufts.edu/~jsinapov/teaching/comp150_RL_Fall2020/

Announcements

Reading Assignment

- Chapter 6 of Sutton and Barto

Research Article Topics

- Transfer learning
- Learning with human demonstrations and/or advice
- Approximating q-functions with neural networks

Reading Assignment

- Chapter 6 of Sutton and Barto
- Matthew E. Taylor, Peter Stone, and Yaxin Liu. **Transfer Learning via Inter-Task Mappings for Temporal Difference Learning**. Journal of Machine Learning Research, 8(1):2125-2167, 2007.
- Responses should discuss both readings
- You get extra credit for answering others' questions!

Programming Assignment #2

- Homework 2 is out

Class Project Discussion

- What makes a good project?
- What makes a good team?

Reading Responses

“What are some real word applications of DP?”
- Boriana

“Since there are at least four ways Monte Carlo methods are advantageous over DP mentioned, are there any problems in which using DP is more practical?”

- Catherine

Reading Responses

“How can we define the stopping conditions for value iteration or the Monte-Carlo method (how many iterations is enough)?”

– Tung

Reading Responses

“Are DP methods dependent on initial states?”

– Eric

Reading Responses

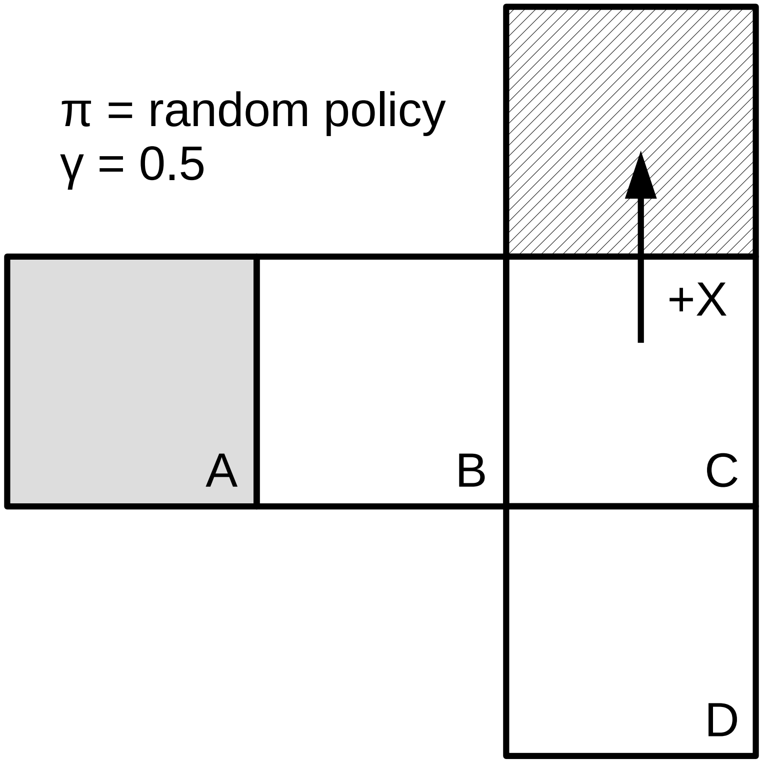
"In the Asynchronous Dynamic Programming method, according to what to choose which states should be updated more frequently?"

– Pandong

Any other questions about DP?

Dynamic Programming

π = random policy
γ = 0.5

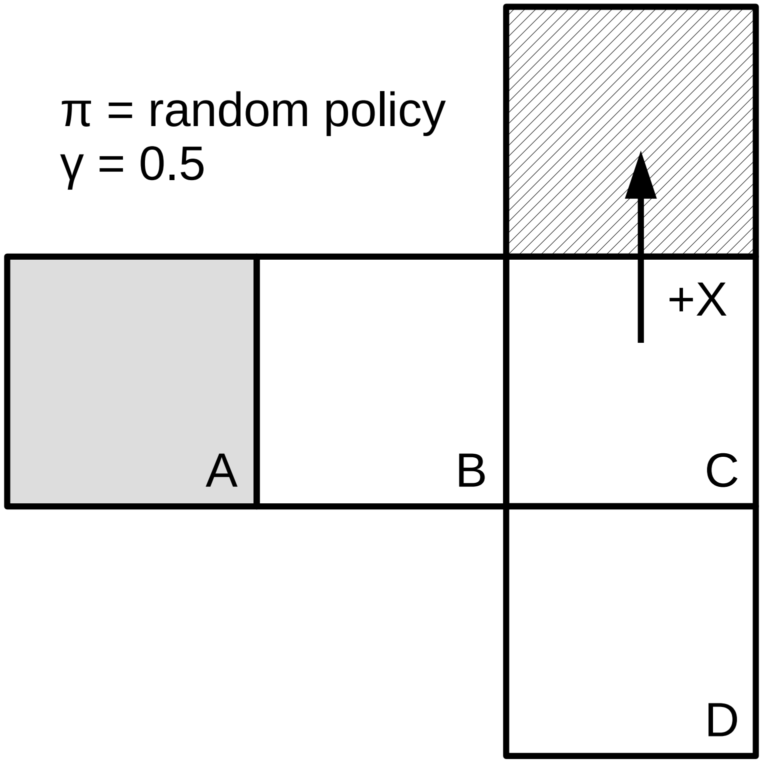


	V_0	V_1	V_2	V_3	V_4	V_5
A	0	0	0			
B	0	0	X/12			
C	0	X/3				
D	0	X/6				

$$\begin{aligned} v_{k+1}(s) &\doteq \mathbb{E}_\pi[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s] \\ &= \sum_a \pi(a|s) \sum_{s',r} p(s', r | s, a)\Big[r + \gamma v_k(s')\Big] \end{aligned}$$

1/3 * 0.5 * x/12 + 1/3 * 0.5 * x/6 + x/3

π = random policy
γ = 0.5

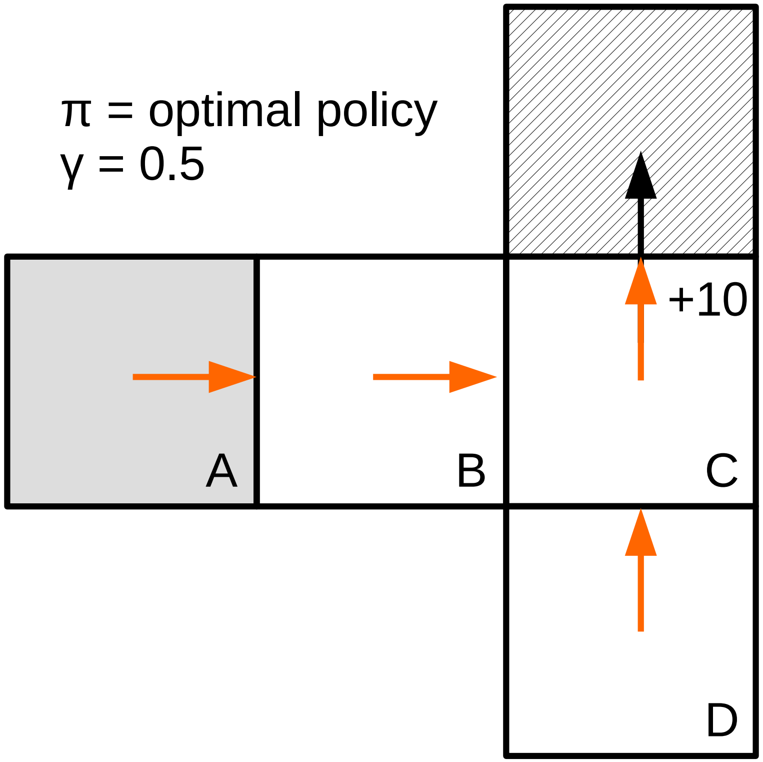


	V_0	V_1	V_2	V_3	V_4	V_5
A	0					
B	0	2nd				
C	0	1st				
D	0					

$$
\begin{aligned}
v_{k+1}(s) &\doteq \mathbb{E}_\pi[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s] \\
&= \sum_a \pi(a|s) \sum_{s',r} p(s', r | s, a)\Big[r + \gamma v_k(s')\Big]
\end{aligned}
$$

1/3 * 0.5 * x/12 + 1/3 * 0.5 * x/6 + x/3

π = optimal policy
γ = 0.5



	V_0	V_1	V_2	V_3	V_4	V_5
A	0					
B	0					
C	0					
D	0					

At each state, the agent has 1 or more actions allowing it to move to neighboring states. Moving in the direction of a wall is not allowed

$$\begin{aligned} v_{k+1}(s) &\doteq \mathbb{E}_\pi[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s] \\ &= \sum_a \pi(a|s) \sum_{s',r} p(s', r \mid s, a)\Big[r + \gamma v_k(s')\Big] \end{aligned}$$

WORKING TEXT AREA:

Policy Improvement

- Main idea: if for a particular state *s*, we can do better than following the current policy by taking a different action, then the current policy is not optimal and changing it to follow the different action at state *s* improves it

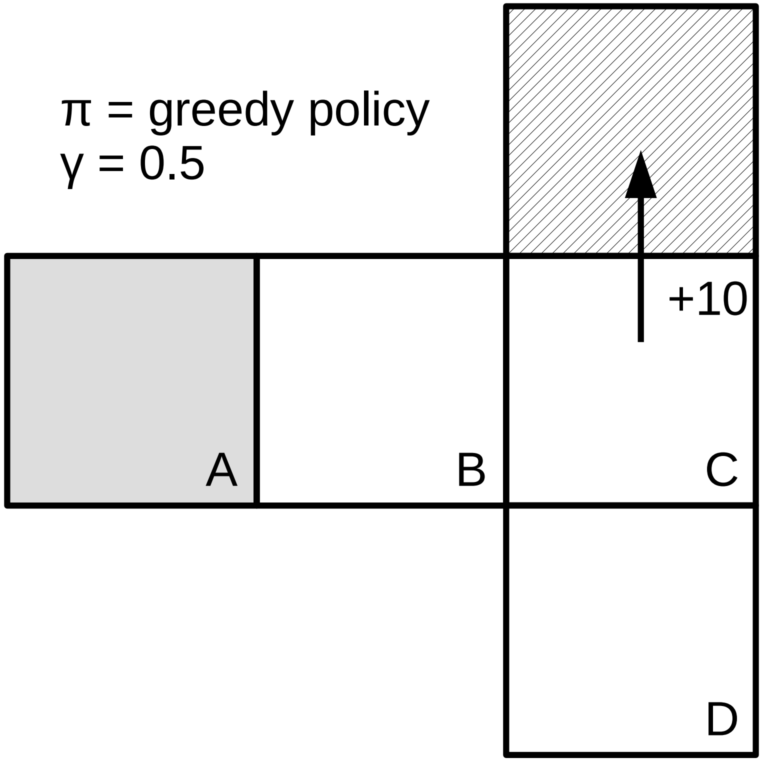
Policy Iteration

- evaluate → improve → evaluate → improve → …..

Value Iteration

- Main idea:
  - Do one sweep of policy evaluation under the current greedy policy
  - Repeat until values stop changing (relative to some small Δ)

π = greedy policy
γ = 0.5



	V_0	V_1	V_2	V_3	V_4	V_5
A	0					
B	0					
C	0					
D	0					

At each state, the agent has 1 or more actions allowing it to move to neighboring states. Moving in the direction of a wall is not allowed

$$\begin{aligned} v_{k+1}(s) &\doteq \max_a \mathbb{E}[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s, A_t = a] \\ &= \max_a \sum_{s',r} p(s', r \mid s, a)\Big[r + \gamma v_k(s')\Big], \end{aligned}$$

WORKING TEXT AREA:

Monte Carlo Methods

First-visit MC prediction, for estimating $V \approx v_\pi$

Initialize:

$\pi \leftarrow$ policy to be evaluated
$V \leftarrow$ an arbitrary state-value function
$Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Repeat forever:

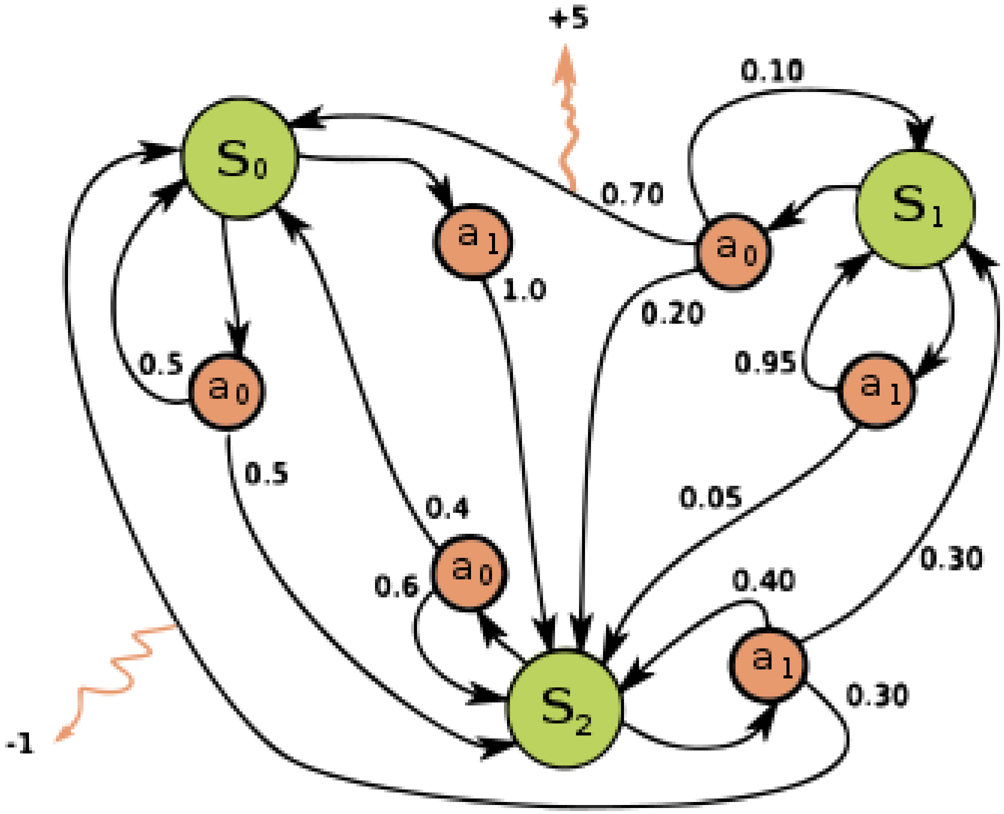
Generate an episode using π
For each state s appearing in the episode:
$G \leftarrow$ the return that follows the first occurrence of s
Append G to $Returns(s)$
$V(s) \leftarrow \text{average}(Returns(s))$

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

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
    $Q(s, a) \leftarrow$ arbitrary
    $\pi(s) \leftarrow$ arbitrary
    $Returns(s, a) \leftarrow$ empty list

Repeat forever:
    Choose $S_0 \in \mathcal{S}$ and $A_0 \in \mathcal{A}(S_0)$ s.t. all pairs have probability > 0
    Generate an episode starting from S_0, A_0, following π
    For each pair s, a appearing in the episode:
        $G \leftarrow$ the return that follows the first occurrence of s, a
        Append G to $Returns(s, a)$
        $Q(s, a) \leftarrow$ average$(Returns(s, a))$
    For each s in the episode:
        $\pi(s) \leftarrow \arg\max_a Q(s, a)$

Code Demo



+5
0.10
S_0
a_1
0.70
S_1
a_0
1.0
0.20
0.5
a_0
0.95
a_1
0.5
0.05
0.4
0.30
0.6
a_0
0.40
a_1
S_2
0.30
-1

Reading Responses

“- What is the advantage and disadvantages of model-free method? What is the advantage and disadvantages of model-based method?”

– Tung

Reading Responses

“In theory, both DP and Monte Carlo will find optimal policy, but since our implementation of the method won't iterate infinitely, will there be chances that the result is only local optimal value”

– Erli

Reading Responses

“Are there situations when on-policy methods are preferred over off-policy for reasons other than ease of implementation?”

– Eric

On-policy first-visit MC control (for ε-soft policies), estimates $\pi \approx \pi_*$

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:

$Q(s,a) \leftarrow$ arbitrary

$Returns(s,a) \leftarrow$ empty list

$\pi(a|s) \leftarrow$ an arbitrary ε-soft policy

Repeat forever:

(a) Generate an episode using π

(b) For each pair s, a appearing in the episode:

$G \leftarrow$ the return that follows the first occurrence of s, a

Append G to $Returns(s,a)$

$Q(s,a) \leftarrow$ average$(Returns(s,a))$

(c) For each s in the episode:

$A^* \leftarrow \arg\max_a Q(s,a)$ (with ties broken arbitrarily)

For all $a \in \mathcal{A}(s)$:

$$\pi(a|s) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(s)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(s)| & \text{if } a \neq A^* \end{cases}$$

Off-policy MC control, for estimating $\pi \approx \pi_*$

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:

$Q(s,a) \leftarrow$ arbitrary

$C(s,a) \leftarrow 0$

$\pi(s) \leftarrow \arg\max_a Q(S_t, a)$ (with ties broken consistently)

Repeat forever:

$b \leftarrow$ any soft policy

Generate an episode using b:

$S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T, S_T$

$G \leftarrow 0$

$W \leftarrow 1$

For $t = T-1, T-2, \ldots$ down to 0:

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

$C(S_t, A_t) \leftarrow C(S_t, A_t) + W$

$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)} [G - Q(S_t, A_t)]$

$\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$ (with ties broken consistently)

If $A_t \neq \pi(S_t)$ then exit For loop

$W \leftarrow W \frac{1}{b(A_t|S_t)}$

Finding Project Partner(s) Breakout

Monte Carlo Tree Search Video

THE END