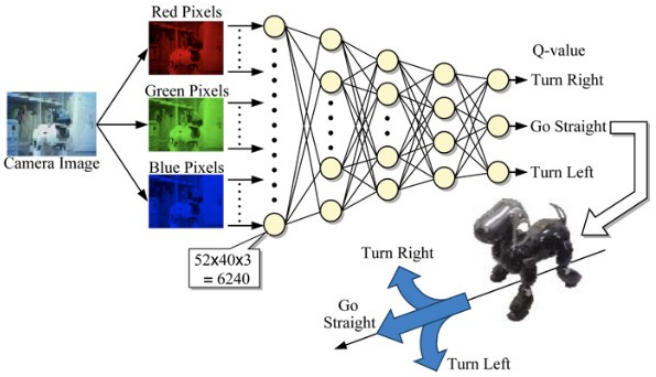
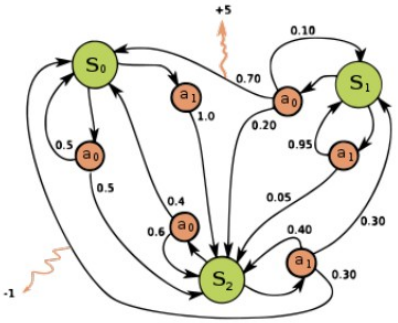
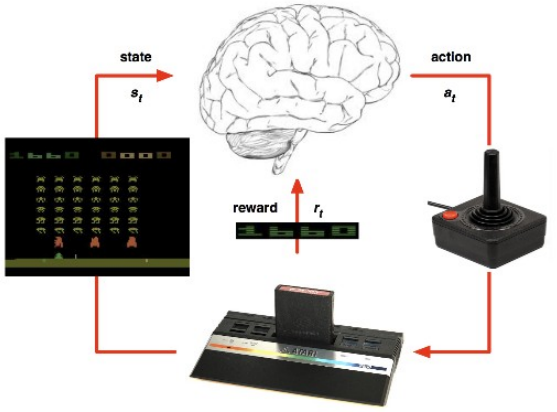
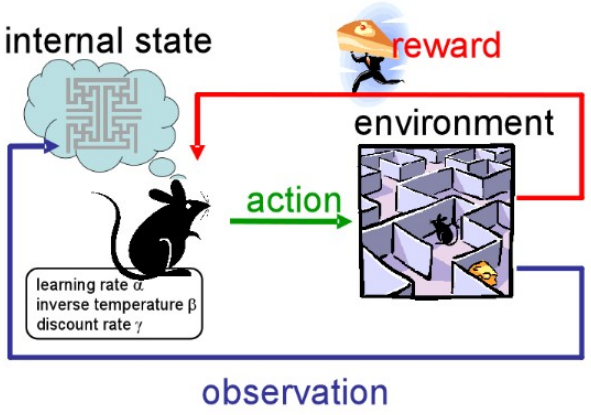
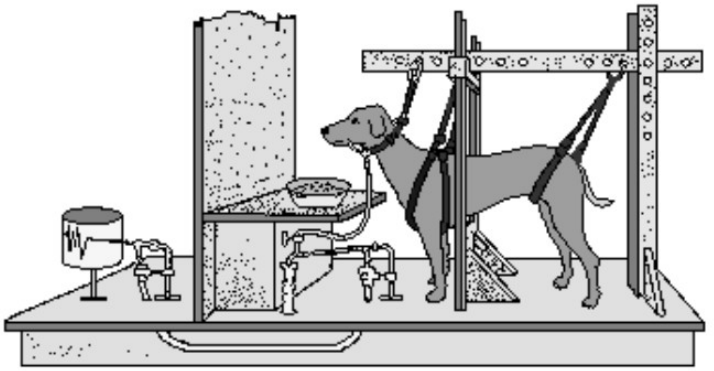
COMP 138: Reinforcement Learning











Instructor: Jivko Sinapov

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

Homework 1 is due today

Programming Assignment #2

- Homework 2 is out

Moderated Discussion

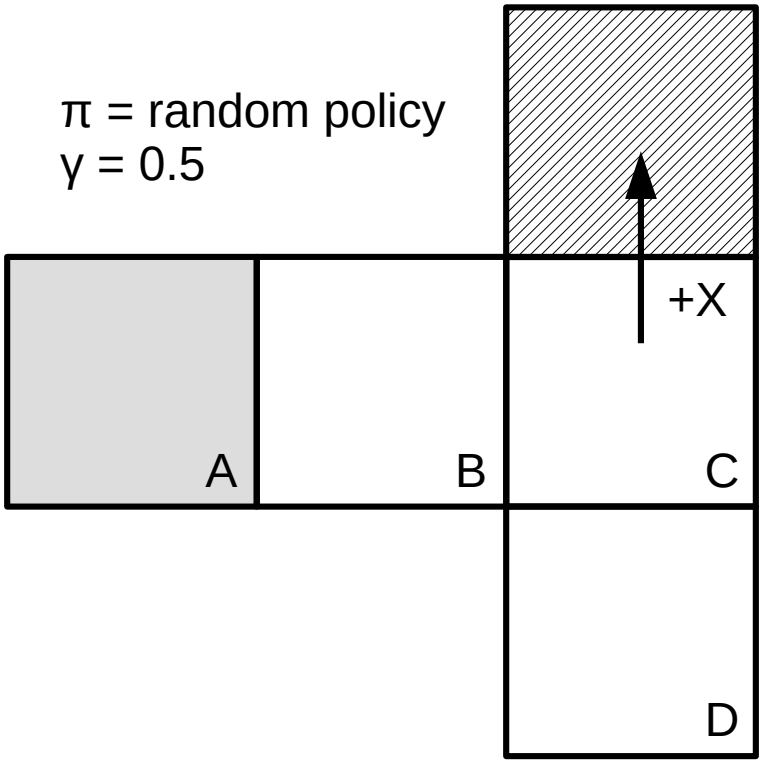
- Chenxuan Liu & Yuanfei Wang

Class Project Discussion

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

Dynamic Programming

π = random policy
γ = 0.5

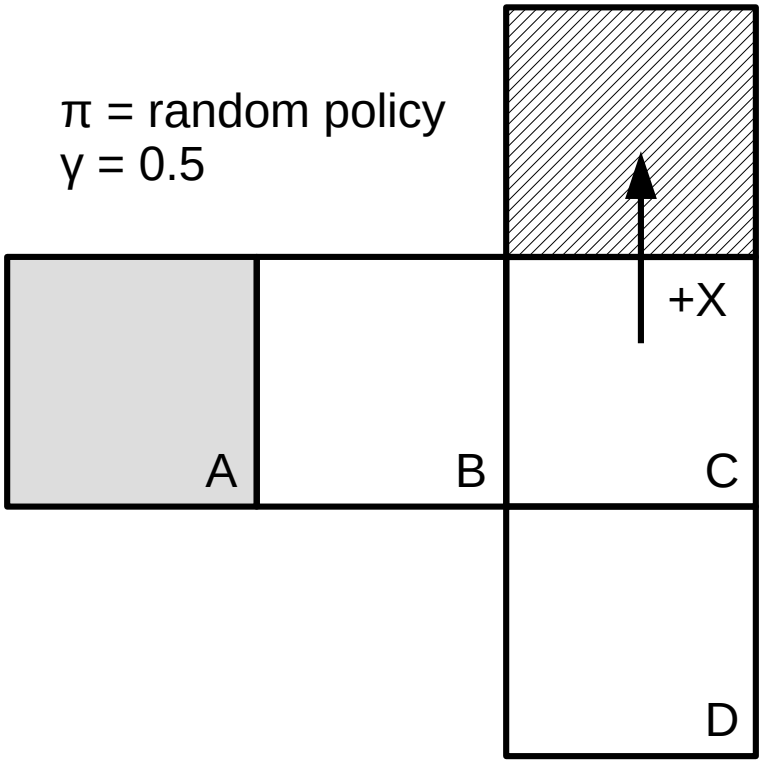


	V_0	V_1	V_2	V_3	V_4	V_5
A	0	0	0			
B	0	0	X/12			
C	0	X/3				
D	0	X/6				

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

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

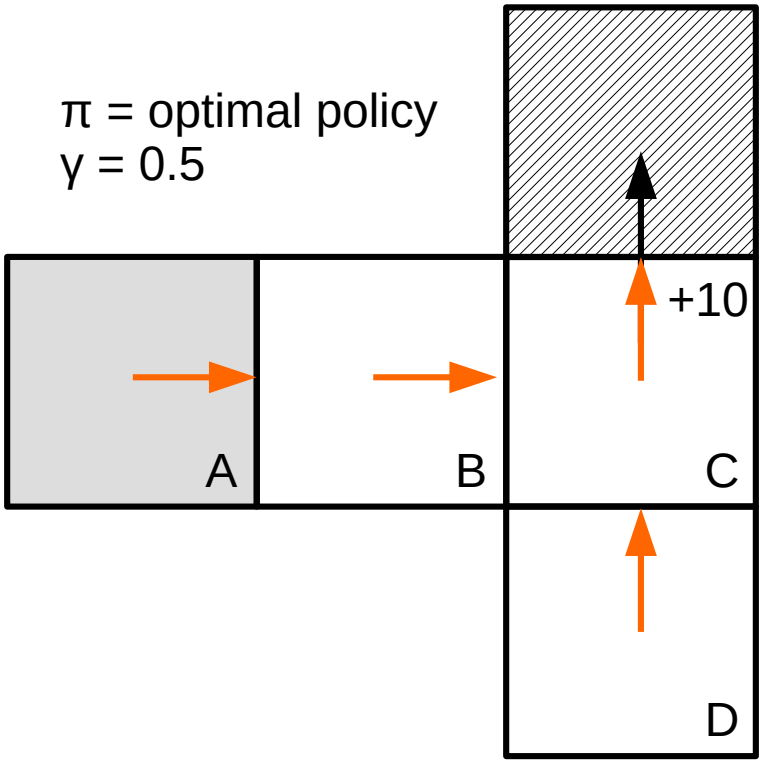
π = random policy
γ = 0.5



	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



	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

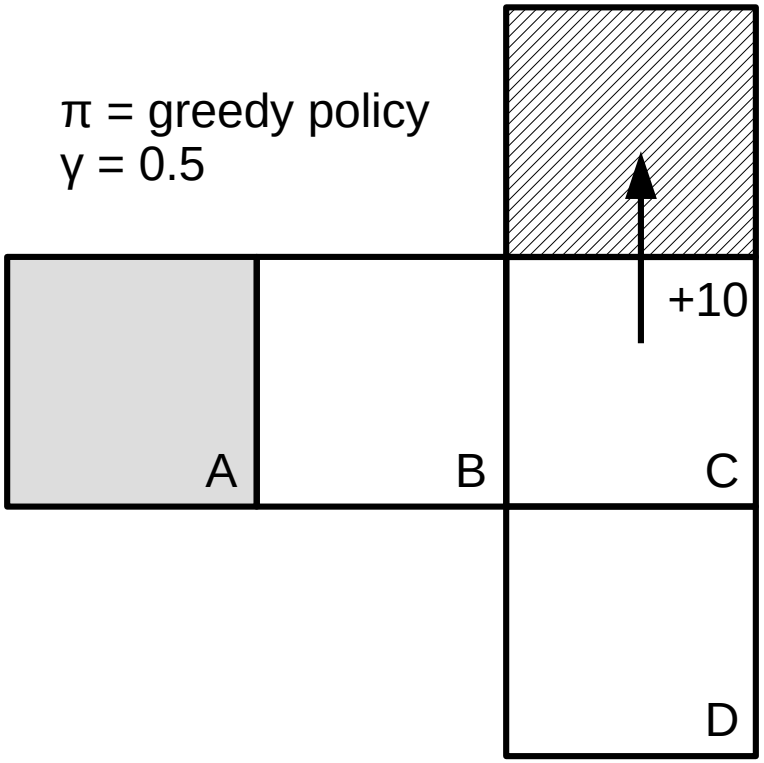
- Main idea: if for a particular state *s*, we can do better than following the current policy by taking a different action, then the current policy is not optimal and changing it to follow the different action at state *s* improves it

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)



	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:

Policy iteration (using iterative policy evaluation)

1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Repeat
   $\Delta \leftarrow 0$
   For each $s \in \mathcal{S}$:
   $v \leftarrow V(s)$
   $V(s) \leftarrow \sum_{s',r} p(s', r \,|\, s, \pi(s)) \big[r + \gamma V(s') \big]$
   $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
   old-action $\leftarrow \pi(s)$
   $\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r \,|\, s, a) \big[r + \gamma V(s') \big]$
   If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

Reading Responses

“The text uses the notation p(s’,r| s,a). Could this be clarified a bit more? Does it represent the transition probability?”

Reading Responses

- “The stopping condition θ is mentioned but not elaborated upon. What are the practical implications of choosing a larger or smaller θ?”

Reading Responses

“Both DP methods (Chapter 4) and Monte Carlo methods (Chapter 5) have their unique advantages and limitations. Can you provide scenarios where it's more appropriate to apply DP methods, and conversely, when should one opt for Monte Carlo methods, considering real-world challenges and requirements?”

Reading Responses

“If value iteration doesn’t reduce overall computation, but just improves intermediate policies quickly, when is it ever useful in the real world? Do we draw a line for when a suboptimal policy is “good enough”?”

“I wondered how DP actually is being used practically in modern reinforcement learning. Is it actually functioning as a foundation or theory as the book suggests, or is there any practical usage available?”

Reading Responses

“Why is it true that the worst case runtime of DP methods is polynomial with respect to the number of states and actions? (p 71)”

Reading Responses

“For the policy improvement theorem, it seems we are always selecting new policies greedily. And only when all submaximal actions are given zero probability, we may choose any apportioning scheme. But is there any possibility for us to select the new policies epsilon greedily?”

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 $Returns(s)$

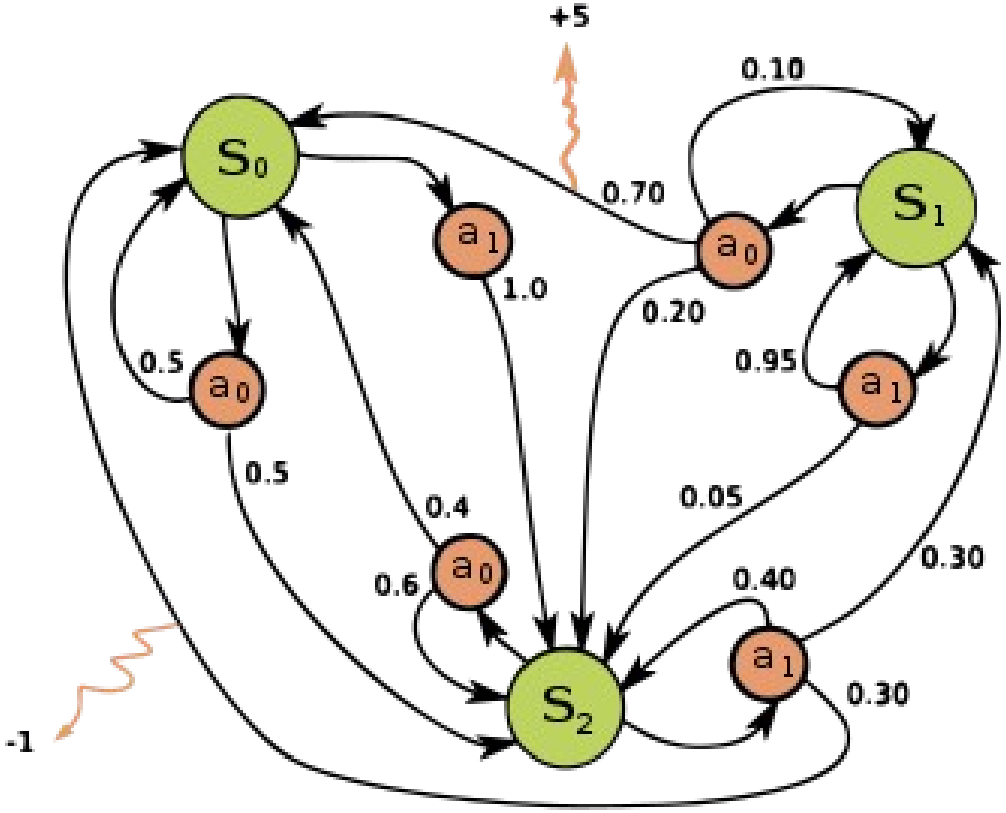
$V(s) \leftarrow \text{average}(Returns(s))$

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 $Returns(s, a)$
        $Q(s, a) \leftarrow \text{average}(Returns(s, a))$
    For each s in the episode:
        $\pi(s) \leftarrow \arg\max_a Q(s, a)$

Code Demo


+5
0.10
S0
0.70
a1
S1
a0
1.0
0.20
0.5
a0
0.95
a1
0.5
0.4
0.05
0.6
a0
0.40
0.30
a1
S2
0.30
-1

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 $Returns(s,a)$
$Q(s,a) \leftarrow \text{average}(Returns(s,a))$
(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

THE END