AIL 722: Reinforcement Learning

Lecture 10: Policy Iteration and Value Iteration

Raunak Bhattacharyya

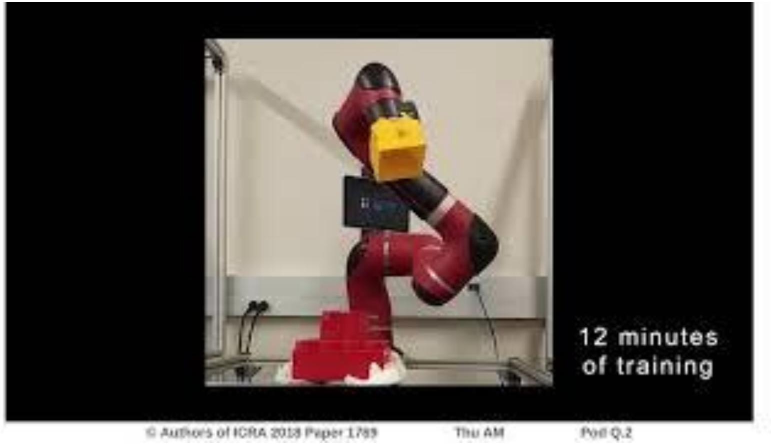
ScAI YARDI SCHOOL OF ARTIFICIAL INTELLIGENCE
INDIAN INSTITUTE OF TECHNOLOGY DELHI

Outline

- Defining Optimality
- Policy Iteration
- Value Iteration

Discounting

Episodic



[Source: Youtube]

Infinite horizon

[Source: Youtube]

$$\theta^* = \arg\max_\theta \mathbb{E}_{p_\theta(\tau)}\left[\sum_{t=0}^{T} \gamma^t \cdot r(s_t, a_t)\right]$$

Bellman Equation

$$V^\pi(s) = r\Big(s, \pi(s)\Big) + \gamma \cdot \mathbb{E}_{p(s' \mid s, \pi(s))}\Big[V^\pi(s')\Big]$$

Discounting

Deterministic policies

Optimality

- Goal: Finding a policy that achieves a lot of reward over the long run

- Notion of betterness: A policy is better than or equal to another policy if its expected return is greater than or equal to that of the other policy for all states

$$\pi \geq \pi' \text{ if and only if } V^{\pi}(s) \geq V^{\pi'}(s) \ \forall s \in \mathcal{S}$$

- There is always at least one policy that is better than or equal to all the other policies. This is called an optimal policy

Optimality

- All the optimal policies share the same state value function as well as the same state-action value function

$$V^*(s) = \max_\pi V^\pi(s)\ \forall s \in \mathcal{S}$$

$$Q^*(s,a) = \max_\pi Q^\pi(s,a)\ \forall s \in \mathcal{S} \text{ and } \forall a \in \mathcal{A}$$

$$J(\theta) = \mathbb{E}_{p(s_1)}\left[V^\pi(s_1)\right]$$

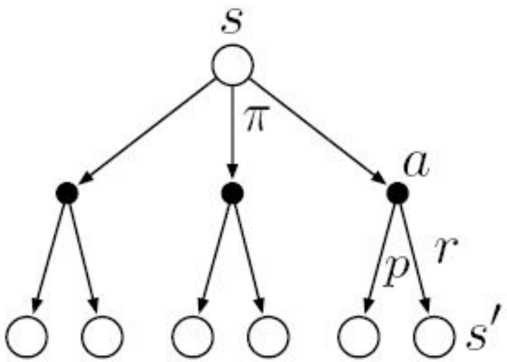
Policy Iteration

1. Evaluate $V^\pi(s)$
2. Set $\pi \longleftarrow \pi_{\text{new}}$

$$\pi_{\text{new}} = \begin{cases} 1 & \text{if } a = \arg\max_a A^\pi(s,a) \\ 0 & \text{otherwise} \end{cases}$$

Policy Evaluation

$$V^{\pi}(s) = r\Big(s, \pi(s)\Big) + \gamma \cdot \mathbb{E}_{p(s'|s,\pi(s))}\Big[V^{\pi}(s')\Big]$$



Backup diagram for v_π

From Reinforcement Learning: An Introduction, Sutton & Barto

Iterative Policy Evaluation

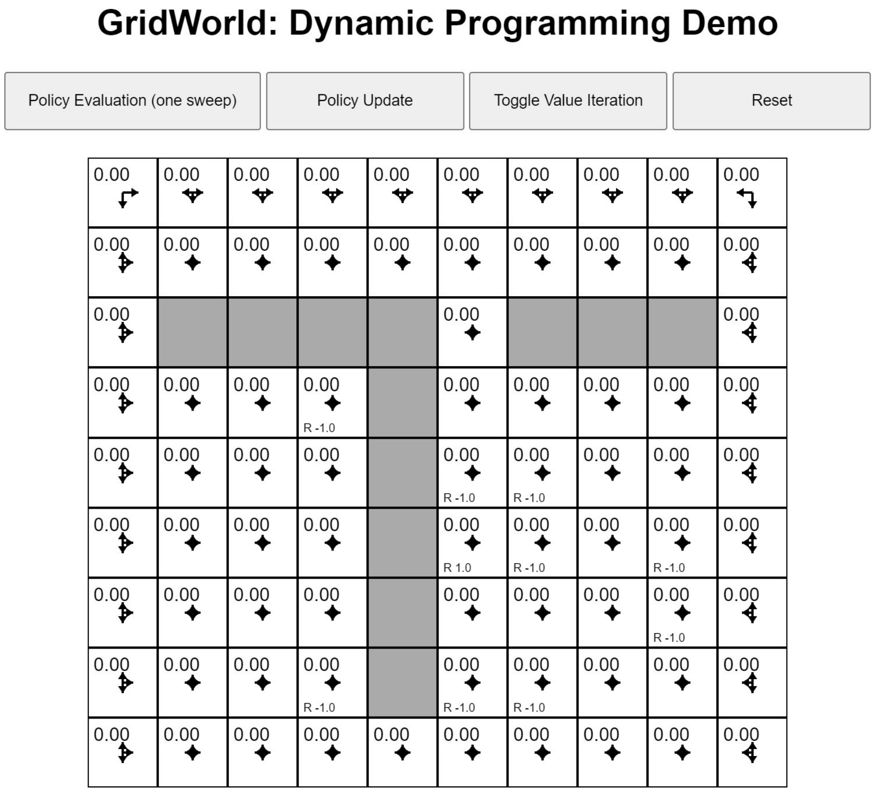
Iterative Policy Evaluation, for estimating $V \approx v_\pi$

Input π, the policy to be evaluated
Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
$\quad \Delta \leftarrow 0$
$\quad$ Loop for each $s \in \mathcal{S}$:
$\qquad v \leftarrow V(s)$
$\qquad V(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\big[r + \gamma V(s')\big]$
$\qquad \Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

From Reinforcement Learning: An Introduction, Sutton & Barto

Demo: Iterative Policy Evaluation



From ReinforceJS, Andrej Karpathy

Policy Iteration

$$1.\ V^\pi(s) = r\Big(s, \pi(s)\Big) + \gamma \cdot \mathbb{E}_{p(s'|s,\pi(s))}\Big[V^\pi(s')\Big]$$

$$2.\ \text{Set } \pi \longleftarrow \pi_{\text{new}}$$

$$\pi_{\text{new}} = \begin{cases} 1 & \text{if } a = \arg\max_a A^\pi(s,a) \\ 0 & \text{otherwise} \end{cases}$$

Finding the Policy

$$\pi_{\text{new}} = \begin{cases} 1 & \text{if } a = \arg\max_a A^\pi(s,a) \\ 0 & \text{otherwise} \end{cases}$$

$$A^\pi(s,a) = r(s,a) + \mathbb{E}_{p(s'|s,a)}\left[V^\pi(s')\right] - V^\pi(s)$$

Goal: Find the argmax

Finding the Policy

Goal: Find the argmax

$$\pi_{\text{new}} = \begin{cases} 1 & \text{if } a = \arg\max_a Q^\pi(s,a) \\ 0 & \text{otherwise} \end{cases}$$

$$Q^\pi(s,a) = r(s,a) + \gamma.\mathbb{E}_{p(s'|s,a)}\Big[V^\pi(s')\Big]$$

Extract the policy using Q-function (table in case of tabular setting)

Policy Iteration: Using Q-values

$$1.\ \ V^\pi(s) = r\Big(s, \pi(s)\Big) + \gamma \cdot \mathbb{E}_{p(s'|s,\pi(s))}\Big[V^\pi(s')\Big]$$

$$2.\ \ \text{Set } \pi \longleftarrow \pi_{\text{new}}$$

$$\pi_{\text{new}} = \begin{cases} 1 & \text{if } a = \arg\max_a Q^\pi(s,a) \\ 0 & \text{otherwise} \end{cases}$$

Workflow

Start with some policy

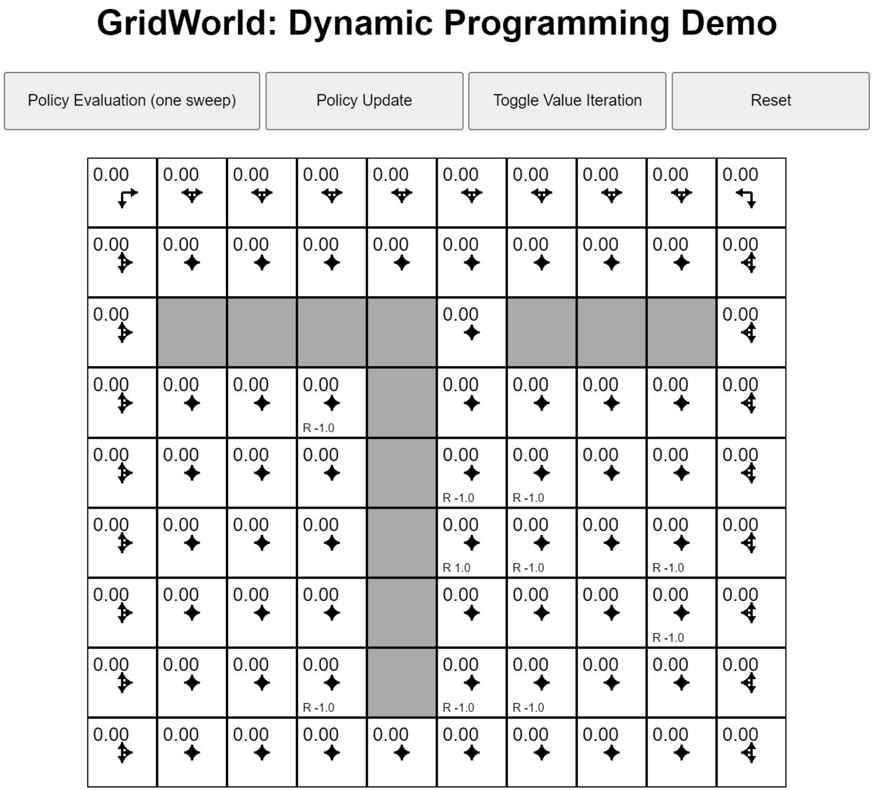
Find corresponding value (iterative policy evaluation until convergence)

From the converged V, Find corresponding Q-values for each action

Construct policy through argmax

Pseudocode?

Policy Iteration Demo



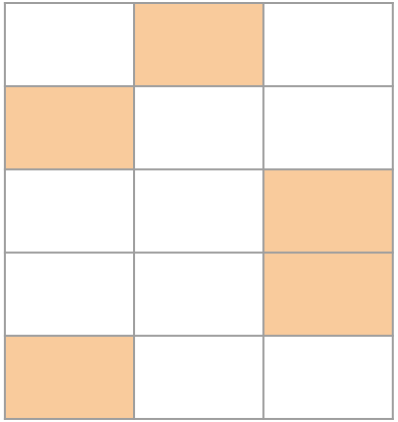
From ReinforceJS, Andrej Karpathy

Value Iteration

$a_?$
$a_?$
$a_?$
$a_?$
$a_?$

$V(s_1)$
$V(s_2)$
$V(s_3)$
$V(s_4)$
$V(s_5)$

$Q(s_1,a_1)$	$Q(s_1,a_2)$	$Q(s_1,a_3)$
$Q(s_2,a_1)$	$Q(s_2,a_2)$	$Q(s_2,a_3)$
$Q(s_3,a_1)$	$Q(s_3,a_2)$	$Q(s_3,a_3)$
$Q(s_4,a_1)$	$Q(s_4,a_2)$	$Q(s_4,a_3)$
$Q(s_5,a_1)$	$Q(s_5,a_2)$	$Q(s_5,a_3)$

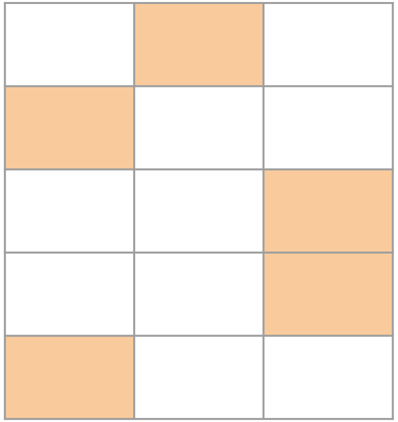


a_2
a_1
a_3
a_3
a_1

$a_?$
$a_?$
$a_?$
$a_?$
$a_?$

$V(s_1)$
$V(s_2)$
$V(s_3)$
$V(s_4)$
$V(s_5)$

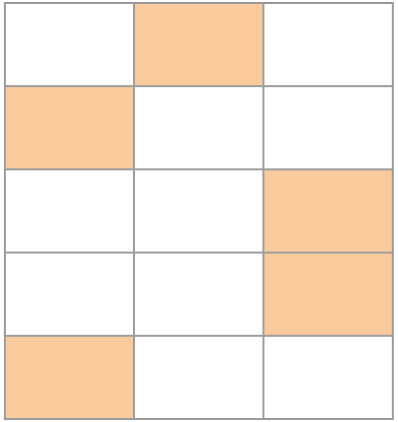
$Q(s_1,a_1)$	$Q(s_1,a_2)$	$Q(s_1,a_3)$
$Q(s_2,a_1)$	$Q(s_2,a_2)$	$Q(s_2,a_3)$
$Q(s_3,a_1)$	$Q(s_3,a_2)$	$Q(s_3,a_3)$
$Q(s_4,a_1)$	$Q(s_4,a_2)$	$Q(s_4,a_3)$
$Q(s_5,a_1)$	$Q(s_5,a_2)$	$Q(s_5,a_3)$



a_2
a_1
a_3
a_3
a_1

$V(s_1)$
$V(s_2)$
$V(s_3)$
$V(s_4)$
$V(s_5)$

$Q(s_1,a_1)$	$Q(s_1,a_2)$	$Q(s_1,a_3)$
$Q(s_2,a_1)$	$Q(s_2,a_2)$	$Q(s_2,a_3)$
$Q(s_3,a_1)$	$Q(s_3,a_2)$	$Q(s_3,a_3)$
$Q(s_4,a_1)$	$Q(s_4,a_2)$	$Q(s_4,a_3)$
$Q(s_5,a_1)$	$Q(s_5,a_2)$	$Q(s_5,a_3)$



$V(s_1)$
$V(s_2)$
$V(s_3)$
$V(s_4)$
$V(s_5)$

Value Iteration

Start with a random value function $V(s)$

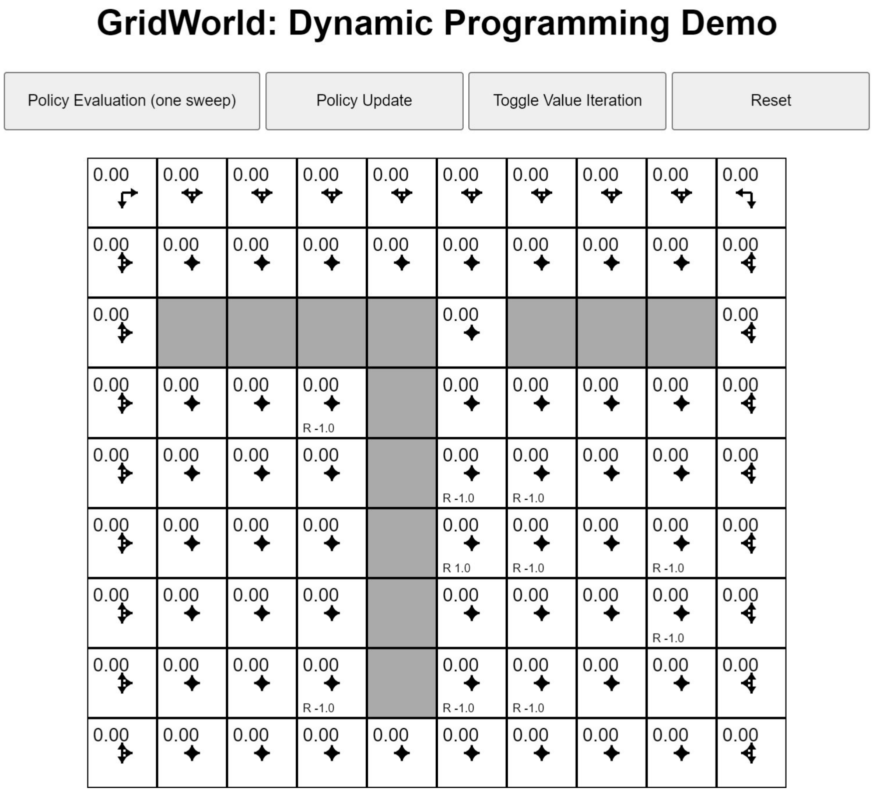
1. Set $Q(s,a) \longleftarrow r(s,a) + \gamma . \mathbb{E}_{p(s'|s,a)}\left[V^{\pi}(s')\right]$

2. Set $V(s) \longleftarrow \max_a Q(s,a)$

Pseudocode?

How do we find the policy?

Value Iteration Demo



From ReinforceJS, Andrej Karpathy