REINFORCEMENT LEARNING

CP8319/CPS824
Lecture 5

Instructor: Nariman Farsad

* Some of the slides in this deck are adopted from courses offered David Silver, Emma Brunskill, and Sergey Levine.

Today’s Agenda

1. **Last Lecture Review**

2. Policy Iteration

Markov Decision Process (MDP)

A Markov decision process (MDP) is a Markov reward process with decisions/actions

Definition

A *Markov Decision Process* is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathbf{P}, R, \gamma \rangle$

- $\mathcal{S}$ is a (finite) set of states
- $\mathcal{A}$ is a finite set of actions
- $\mathbf{P}$ is dynamics/transition model for each action,

$$P^{a}_{s,s'} = P\left(S_{t+1} = s' | S_t = s, A_t = a\right)$$

- R is the reward function, $R(s, a) = \mathbb{E}[r_t | S_t = s, A_t = a]$
- γ is a discount factor, $\gamma \in [0, 1]$

MDP: Policy

Definition

A *policy* π is a distribution over actions given states,

$$\pi(a|s) = P\ (A_t = a|S_t = s)$$

- Policy specifies what action to take in each state
  - Can be deterministic or stochastic
- A policy fully defines the behaviour of an agent
- MDP policies depend on the current state (not the history)
- i.e. Policies are *stationary* (time-independent),

$$A_t \sim \pi(\cdot\ |S_t), \forall t > 0$$

State Value Function

Definition

The *state-value function* $v^{\pi}(s)$ of an MDP is the expected return starting from state s, and then following policy π

$$v^{\pi}(s) = \mathbb{E}_{\pi}[G_t | S_t = s]$$

MDP Policy Evaluation: Iterative Algorithm

```
Initialize $v_0(s) = 0$ for all $s$
For $k = 1$ until convergence:
    For all $s \in \mathcal{S}$:
        $v^{\pi}_{k+1}(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left( R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P^{a}_{s,s'} v^{\pi}_{k}(s') \right)$
```

This is known as Bellman expectation backup

Action Value Function

Definition

The *action-value function* $Q^{\pi}(s, a)$ is the expected return starting from state s, taking action a, and then following policy π

$$Q^{\pi}(s, a) \;=\; \mathbb{E}_{\pi}[G_t | S_t, = s, A_t = a]$$

The action-value function can be decomposed using Bellman equation,

$$\begin{aligned} Q^{\pi}(s, a) &= \mathbb{E}_{\pi}[r_t + \gamma Q^{\pi}(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a] \\ &= R(s, a) + \gamma \sum_{s' \in \mathcal{S}} P^{a}_{s,s'} \underbrace{\left(\sum_{a' \in \mathcal{A}} \pi(a' | s') Q^{\pi}(s', a') \right)}_{v^{\pi}(s')} \end{aligned}$$

Optimal Value Function

Definition

The *optimal state-value function* $v^*(s)$ is the maximum value function over all policies

$$v^*(s) \;= \max_{\pi} v^{\pi}(s)$$

The *optimal action-value function* $Q^*(s,a)$ is the maximum action-value function over all policies

$$Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a)$$

- The optimal value function specifies the best possible performance in the MDP.
- An MDP is “solved” when we know the optimal value fn.

Optimal Policy

Define a partial ordering over policies

$$\pi \geq \pi' \, if \, v^{\pi}(s) \geq v^{\pi'}(s), \forall s$$

Theorem

For any Markov Decision Process

- *There exists an optimal policy π^* that is better than or equal to all other policies, $\pi^* \geq \pi, \forall \pi$*
- *All optimal policies achieve the optimal value function,* $v^{\pi^*}(s) = v^*(s)$, or $\pi^* = \underset{\pi}{\operatorname{argmax}}\, v^{\pi}(s)$
- *All optimal policies achieve the optimal action-value function,* $Q^{\pi^*}(s,a) = Q^*(s,a)$

Optimal Policy Using Optimal Action-Value Function

An optimal policy can be found by maximising over $Q^*(s, a)$,

$$\pi^*(a|s) = \begin{cases} 1, & \text{if } a = \max_{a \in \mathcal{A}} Q^*(s, a) \\ 0, & otherwise \end{cases}$$

- There is always a deterministic optimal policy for any MDP
- If we know $Q^*(s, a)$, we immediately have the optimal policy

Optimal Policy Search

- One option is searching to compute best policy

- Number of deterministic policies is $|\mathcal{A}|^{|\mathcal{S}|}$

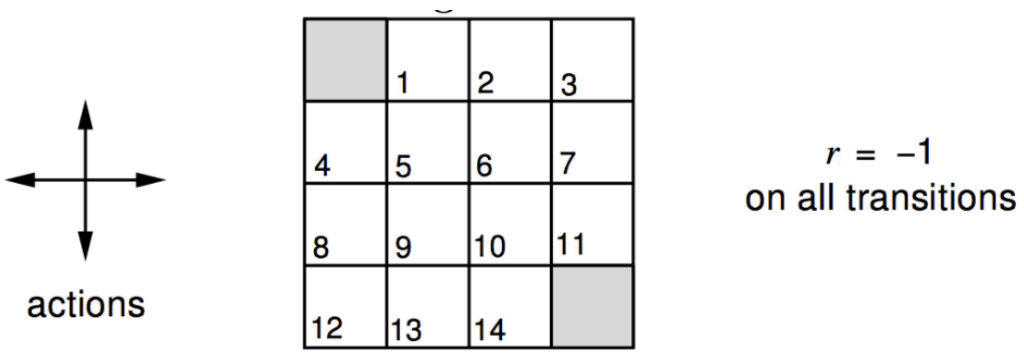
- Better Options:
  - Policy Iteration
  - Value Iteration

Today’s Agenda

1. Last Lecture Review

2. **Policy Iteration**

MDP Policy Evaluation: Example



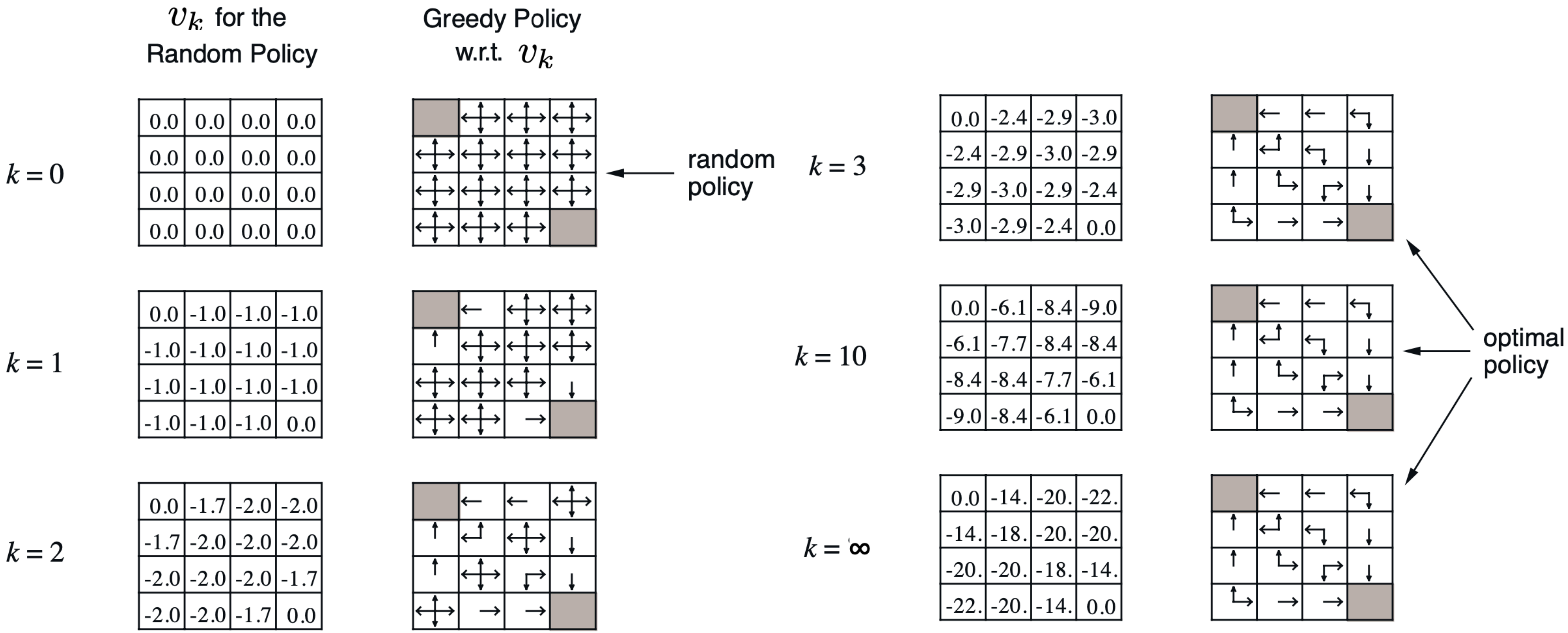
	1	2	3
4	5	6	7
8	9	10	11
12	13	14	

$r = -1$
on all transitions

- Undiscounted episodic MDP ($\gamma = 1$)
- Nonterminal states $1, \ldots, 14$
- One terminal state (shown twice as shaded squares)
- Actions leading out of the grid leave state unchanged
- Reward is -1 until the terminal state is reached
- Agent follows uniform random policy

$$\pi(n|\cdot) = \pi(e|\cdot) = \pi(s|\cdot) = \pi(w|\cdot) = 0.25$$

MDP Policy Evaluation: Example



v_k for the Random Policy

Greedy Policy w.r.t. v_k

random policy

optimal policy

$k = 0$

0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0

$k = 1$

0.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	0.0

$k = 2$

0.0	-1.7	-2.0	-2.0
-1.7	-2.0	-2.0	-2.0
-2.0	-2.0	-2.0	-1.7
-2.0	-2.0	-1.7	0.0

$k = 3$

0.0	-2.4	-2.9	-3.0
-2.4	-2.9	-3.0	-2.9
-2.9	-3.0	-2.9	-2.4
-3.0	-2.9	-2.4	0.0

$k = 10$

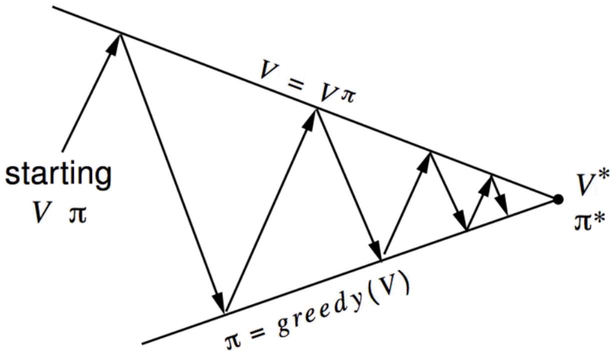
0.0	-6.1	-8.4	-9.0
-6.1	-7.7	-8.4	-8.4
-8.4	-8.4	-7.7	-6.1
-9.0	-8.4	-6.1	0.0

$k = \infty$

0.0	-14.	-20.	-22.
-14.	-18.	-20.	-20.
-20.	-20.	-18.	-14.
-22.	-20.	-14.	0.0

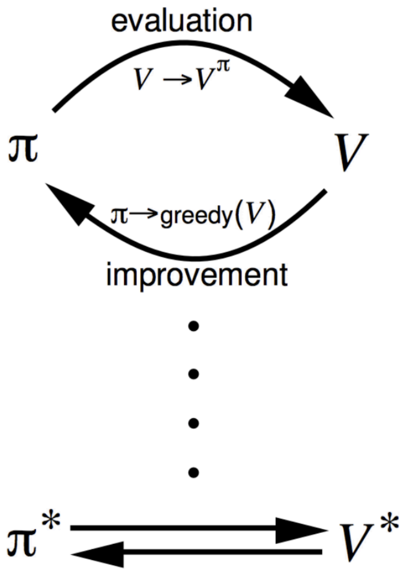
Policy Iteration

```
Set $i = 0$
Initialize $\pi_0(s)$ randomly for all states $s$
While $i == 0$ or $\| \pi_i - \pi_{i-1} \|_1 > 0$ (L1-norm, measures if the policy changed for any state):
    • $v^{\pi_i} \leftarrow$ MDP value function policy evaluation of $\pi_i$ (see slide 6 for formula)
    • $\pi_{i+1} \leftarrow$ Policy improvement
    • $i = i + 1$
```



Policy evaluation Estimate v_π
Iterative policy evaluation

Policy improvement Generate $\pi' \geq \pi$
Greedy policy improvement



Policy Improvement

Compute state-action value of a policy π_i
For $s \in \mathcal{S}$ and $a \in \mathcal{A}$:

- $Q^{\pi_i}(s,a) = R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P_{s,s'}^{a} v^{\pi_i}(s')$

Compute new policy π_{i+1}, for all $s \in \mathcal{S}$

- $\pi_{i+1}(s) = \arg\max_{a \in \mathcal{A}} Q^{\pi_i}(s,a)$

With probability 1 choose an action that maximizes Q (i.e., a deterministic policy)

Why Policy Improvement Works? (1)

$$Q^{\pi_i}(s,a) = R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P_{s,s'}^{a} v^{\pi_i}(s')$$

$$\max_a Q^{\pi_i}(s,a) \geq R\big(s,\pi_i(s)\big) + \gamma \sum_{s' \in \mathcal{S}} P_{s,s'}^{\pi_i(s)} v^{\pi_i}(s') = v^{\pi_i}(s)$$

$$\pi_{i+1}(s) = \arg\max_a Q^{\pi_i}(s,a)$$

- Suppose we take $\pi_{i+1}(s)$ for one action, then follow π_i forever
  - Our expected sum of rewards is at least as good as if we had always followed π_i
- But new proposed policy is to always follow π_{i+1} ...

Why Policy Improvement Works? (2)

$$\pi_{i+1}(s) = \arg\max_a Q^{\pi_i}(s,a)$$

$$Q^{\pi_i}(s,\pi_{i+1}(s)) = \max_a Q^{\pi_i}(s,a) \geq Q^{\pi_i}(s,\pi_i(s)) = v^{\pi_i}(s)$$

$$\begin{aligned}
v^{\pi_i}(s) \leq Q^{\pi_i}\big(s,\pi_{i+1}(s)\big) &= R\big(s,\pi_{i+1}(s)\big) + \gamma \sum_{s'\in\mathcal{S}} P_{s,s'}^{\pi_{i+1}(s)} v^{\pi_i}(s') \\
&\leq R\big(s,\pi_{i+1}(s)\big) + \gamma \sum_{s'\in\mathcal{S}} P_{s,s'}^{\pi_{i+1}(s)} \left(\max_{a'} Q^{\pi_i}(s',a')\right) \\
&= R\big(s,\pi_{i+1}(s)\big) + \gamma \sum_{s'\in\mathcal{S}} P_{s,s'}^{\pi_{i+1}(s)} \left(R\big(s',\pi_{i+1}(s')\big) + \gamma \sum_{s''\in\mathcal{S}} P_{s',s''}^{\pi_{i+1}(s')} v^{\pi_i}(s'')\right) \\
&\ \ \vdots \\
&= v^{\pi_{i+1}}(s)
\end{aligned}$$

Policy Iteration: Questions

```
Set $i = 0$
Initialize $\pi_0(s)$ randomly for all states $s$
While $i\ == \ 0$ or $\| \pi_i \ - \ \pi_{i-1} \|_1 > \ 0$ (L1-norm, measures if the policy changed for any state):
    • $v^{\pi_i}$ ← MDP value function **policy evaluation** of $\pi_i$ (see slide 6 for formula)
    • $\pi_{i+1}$← **Policy improvement**
    • $i = i + 1$
```

- If policy doesn't change, can it ever change again?

- Is there a maximum number of iterations of policy iteration?

Policy Iteration: Questions

```
Set $i = 0$
Initialize $\pi_0(s)$ randomly for all states $s$
While $i == 0$ or $\| \pi_i - \pi_{i-1} \|_1 > 0$ (L1-norm, measures if the policy changed for any state):
    • $v^{\pi_i} \leftarrow$ MDP value function policy evaluation of $\pi_i$ (see slide 6 for formula)
    • $\pi_{i+1} \leftarrow$ Policy improvement
    • $i = i + 1$
```

- If policy doesn't change, can it ever change again?

No

- Is there a maximum number of iterations of policy iteration?

$|\mathcal{A}|^{|\mathcal{S}|}$since that is the maximum number of policies, and as the policy improvement step is monotonically improving, each policy can only appear in one round of policy iteration unless it is an optimal policy.

Policy Iteration: When does it stop?

- Suppose for all $s \in \mathcal{S},\ \pi_{i+1}(s) = \pi_i(s)$
- Then for all $s \in \mathcal{S},\ Q^{\pi_{i+1}}(s,a) = Q^{\pi_i}(s,a)$
- Recall policy improvement step
  - $Q^{\pi_i}(s,a) = R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P_{s,s'}^{a} v^{\pi_i}(s')$
  - $\pi_{i+1}(s) = \arg\max_a Q^{\pi_i}(s,a)$
  - $\pi_{i+2}(s) = \arg\max_a Q^{\pi_{i+1}}(s,a) = \arg\max_a Q^{\pi_i}(s,a)$
- Therefore, policy cannot ever change again

Next Time Value Iteration