Introduction to Reinforcement Learning

Lecture 3a

Dr. Apurva Narayan

http://anarayan.com

Value Iteration

- Value when no time left:

$$V(s_h) = \max_{a_h} R(s_h, a_h)$$

- Value with one time step left:

$$V(s_{h-1}) = \max_{a_{h-1}} R(s_{h-1}, a_{h-1}) + \gamma \sum_{s_h} \Pr(s_h | s_{h-1}, a_{h-1}) V(s_h)$$

- Value with two time steps left:

$$V(s_{h-2}) = \max_{a_{h-2}} R(s_{h-2}, a_{h-2}) + \gamma \sum_{s_{h-1}} \Pr(s_{h-1} | s_{h-2}, a_{h-2}) V(s_{h-1})$$

- ...
- Bellman's equation:

$$V(s_t) = \max_{a_t} R(s_t, a_t) + \gamma \sum_{s_{t+1}} \Pr(s_{t+1} | s_t, a_t) V(s_{t+1})$$

$$a_t^* = \operatorname*{argmax}_{a_t} R(s_t, a_t) + \gamma \sum_{s_{t+1}} \Pr(s_{t+1} | s_t, a_t) V(s_{t+1})$$

Solving MDPs: Value Iteration

- Bellman Equation gives us a recursive definition of the optimal value:

$$V^*(s) = \max_{a \in A} \sum_{s' \in S} P(s'|s,a)[R(s,a,s') + \gamma V^*(s')]$$

Key Idea: solve iteratively via dynamic programming

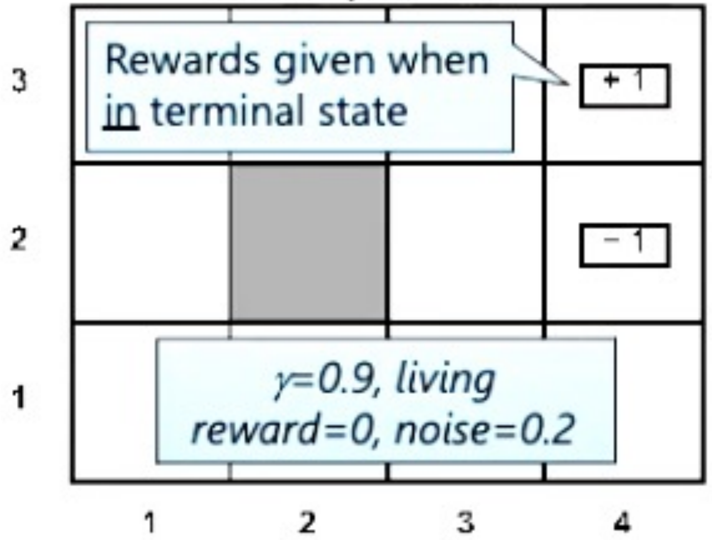
- Start with $V_0(s) = 0$ for all states s
- Iterate the Bellman update until convergence:

$$V_{i+1}(s) \leftarrow \max_{a \in A} \sum_{s' \in S} P(s'|s,a)[R(s,a,s') + \gamma V_i(s')]$$

Example: Value Iteration

Bellman Update Rule: $$V_{i+1}(s) \leftarrow \max_{a \in A} \sum_{s' \in S} P(s'|s,a)[R(s,a,s') + \gamma V_i(s')]$$

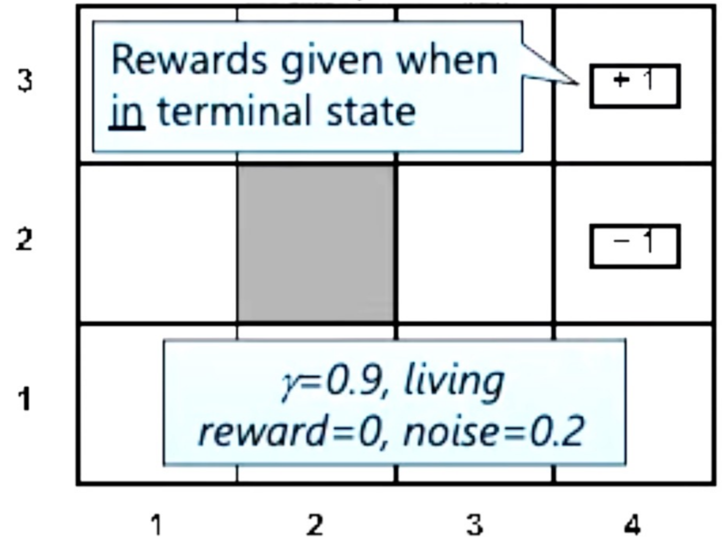
Example MDP



Example: Value Iteration

Bellman Update Rule: $$V_{i+1}(s) \leftarrow \max_{a \in A} \sum_{s' \in S} P(s'|s,a)[R(s,a,s') + \gamma V_i(s')]$$

Example MDP



V_0

	1	2	3	4
3	0	0	0	0
2	0		0	0
1	0	0	0	0

Start with $V_0(s) = 0$

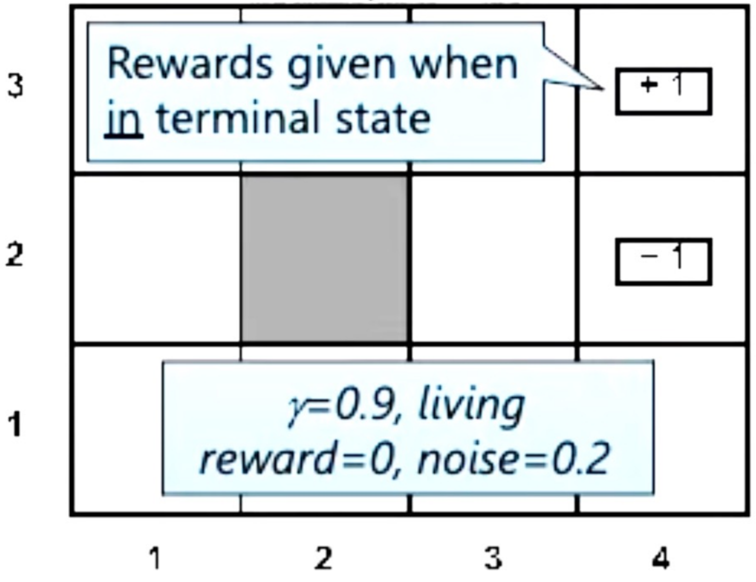
V_1

	1	2	3	4
3	0	0	0	+1
2	0		0	-1
1	0	0	0	0

Example: Value Iteration

Bellman Update Rule: $$V_{i+1}(s) \leftarrow \max_{a \in A} \sum_{s' \in S} P(s'|s,a)[r(s,a,s') + \gamma V_i(s')]$$

Example MDP



V_1

	1	2	3	4
3	0	0	0	+1
2	0		0	-1
1	0	0	0	0

V_2

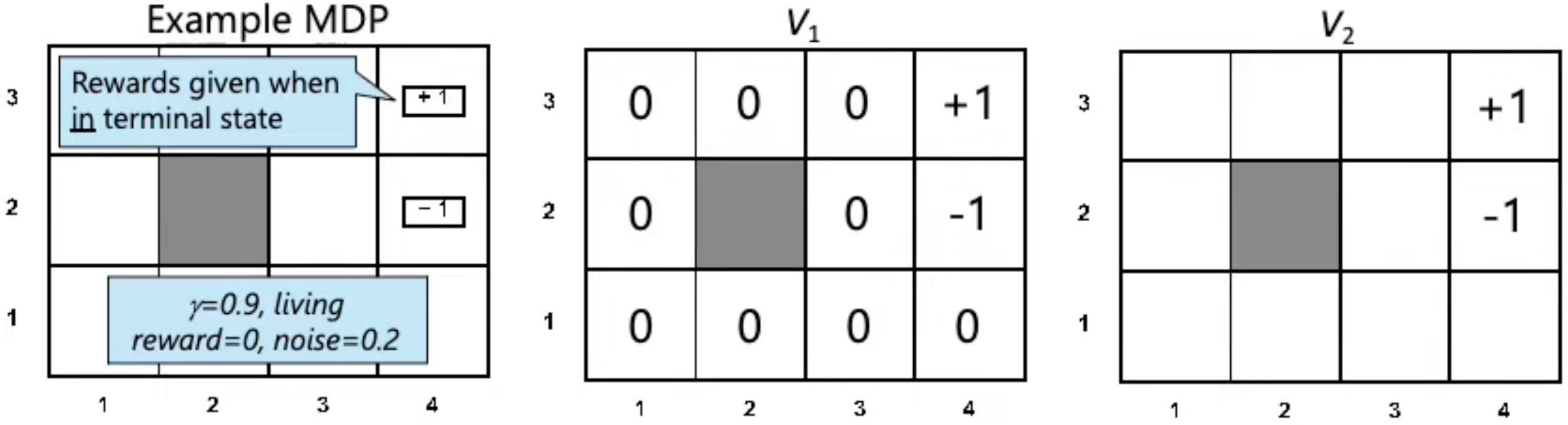
	1	2	3	4
3				
2				
1				

$V_2(\langle 3,4 \rangle) \leftarrow 1$

$V_2(\langle 2,4 \rangle) \leftarrow -1$

Example: Value Iteration

Bellman Update Rule: $V_{i+1}(s) \leftarrow \max_{a \in A} \sum_{s' \in S} P(s'|s,a)[r(s,a,s') + \gamma V_i(s')]$

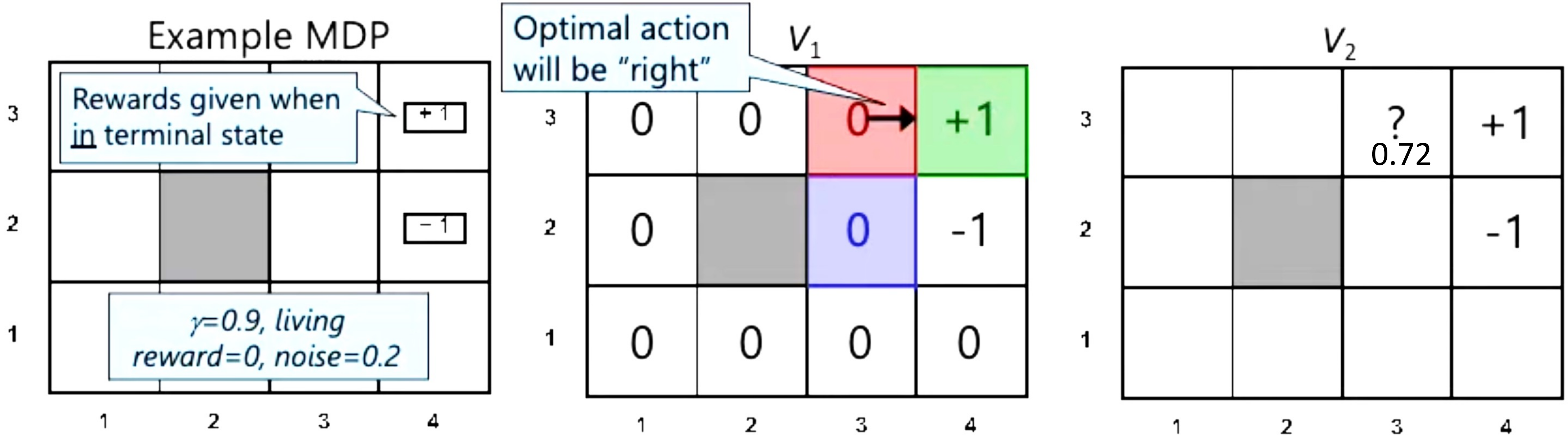


$V_2(\langle 3,4 \rangle) \leftarrow 1$

$V_2(\langle 2,4 \rangle) \leftarrow -1$

Example: Value Iteration

Bellman Update Rule: $V_{i+1}(s) \leftarrow \max_{a \in A} \sum_{s' \in S} P(s'|s,a)[R(s,a,s') + \gamma V_i(s')]$



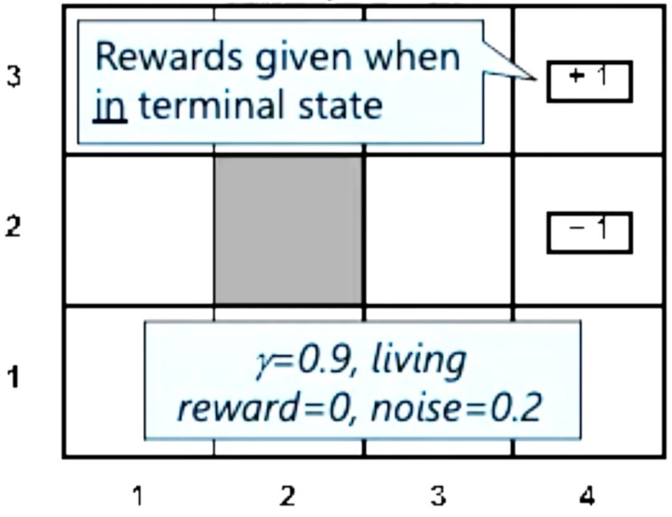
$V_2(\langle 3,3\rangle) \leftarrow \sum_{s' \in S} P(s'|\langle 3,3\rangle, right)[r(\langle 3,3\rangle, right, s') + 0.9V_i(s')]$

$\leftarrow 0.8[0 + 0.9 \times 1] + 0.1[0 + 0.9 \times 0] + 0.1[0 + 0.9 \times 0] = 0.72$

Example: Value Iteration

Bellman Update Rule: $$V_{i+1}(s) \leftarrow \max_{a \in A} \sum_{s' \in S} P(s'|s,a)[R(s,a,s') + \gamma V_i(s')]$$

Example MDP



V_2

	1	2	3	4
3	0	0	0.72	+1
2	0		0	-1
1	0	0	0	0

V_3

	1	2	3	4
3	0	0.52	0.78	+1
2	0		0.43	-1
1	0	0	0	0

- Information propagates outward from terminal states
- Eventually all states have correct value estimates

Solving MDPs: Value Iteration

Value Iteration:

- Start with $V_0(s) = 0$
- Iterate until convergence: $V_{i+1}(s) \leftarrow \max_{a \in A} \sum_{s' \in S} P(s'|s,a)[R(s,a,s') + \gamma V_i(s')]$

- Can prove that value iteration converges to the optimal value function

Policy Evaluation

- How do we calculate the V's for a fixed policy?

- Idea: Bellman updates for arbitrary policy:

$$V_0^{\pi}(s) = 0$$

$$V_{i+1}^{\pi}(s) \leftarrow \sum_{s'} P(s'|s, \pi(s))[R(s, \pi(s), s') + \gamma V_i^{\pi}(s')]$$

(value iteration update rule)

$$\boxed{V_{i+1}(s) \leftarrow \max_{a \in A} \sum_{s' \in S} P(s'|s,a)[R(s,a,s') + \gamma V_i(s')]}$$

Policy Iteration: An Alternative to Value Iteration

Repeat steps until convergence:

1. **Policy evaluation:** keep current policy π fixed, find value function $V^{\pi_k}(\cdot)$
   - Iterate simplified Bellman update until values converge:

$$V_{i+1}^{\pi_k}(s) \leftarrow \sum_{s'} P(s'|s,\pi_k(s))[R(s,\pi_k(s),s') + \gamma V_i^{\pi_k}(s')]$$

Chooses actions according to π

2. **Policy improvement:** find the best action for $V^{\pi_k}(\cdot)$ via one-step lookahead

$$\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma V^{\pi_k}(s')]$$

Policy iteration is optimal too!

- Faster than value iteration in terms of number of (outer) loops, but remember that step 1 has an inner loop too.

Comparison of Methods for Solving MDPs

Value Iteration	Each iteration updates both utilities (explicitly, based on current utilities) and the policy (possibly implicitly, based on current utilities)
Policy Iteration	• Several iterations to update utilities for a fixed policy • Occasional iterations to update policies
Hybrid Methods (asynchronous policy iteration)	Any sequences of partial updates to either policies or utilities will converge if every state is visited infinitely often