PATTERN RECOGNITION AND MACHINE LEARNING

CHAPTER 15: MARKOV DECISION PROCESS

Learning Objectives

1、What is the Markov decision process (MDP)?

2、What is the partial observable MDP?

3、What is the Bellman equation?

4、What are value iteration and policy iteration?

5、What are policy improvement and policy evaluation?

6、How to use observation and prediction to update belief?

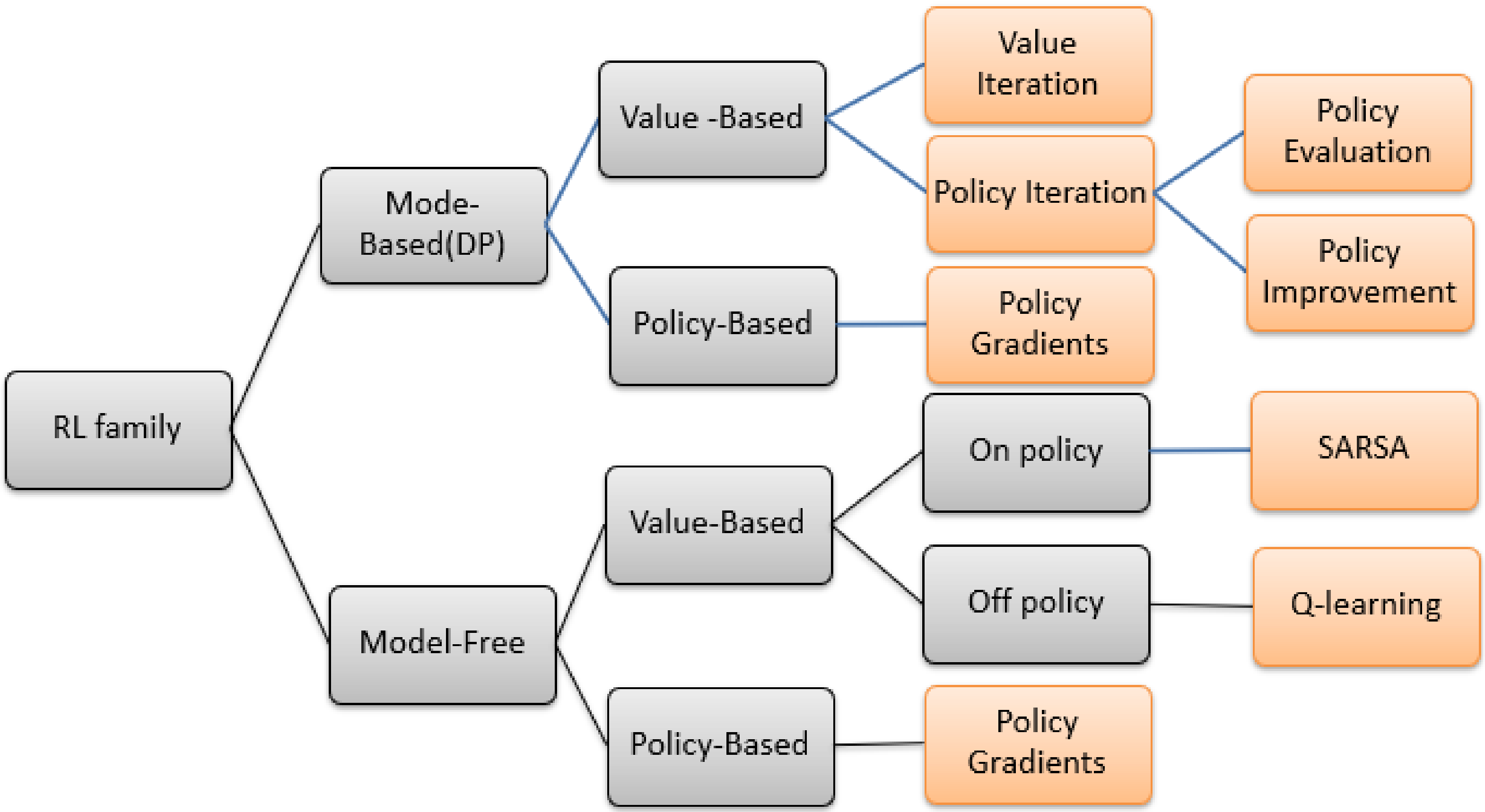
7、What is the max-sum algorithm?

8、How to reduce the computational complexity of POMDP?

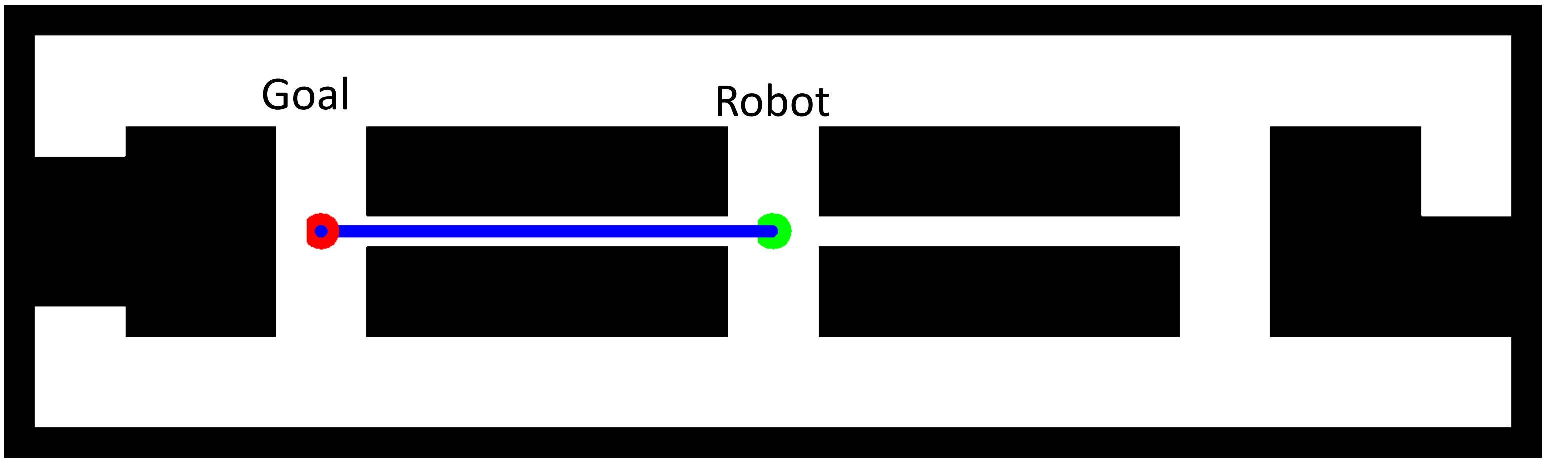
Outlines

- Markov Decision Process (MDP)
- Value Iteration and Policy Iteration
- Partially Observable MDP (POMDP)
- POMDP Observation and Prediction
- POMDP Approximation

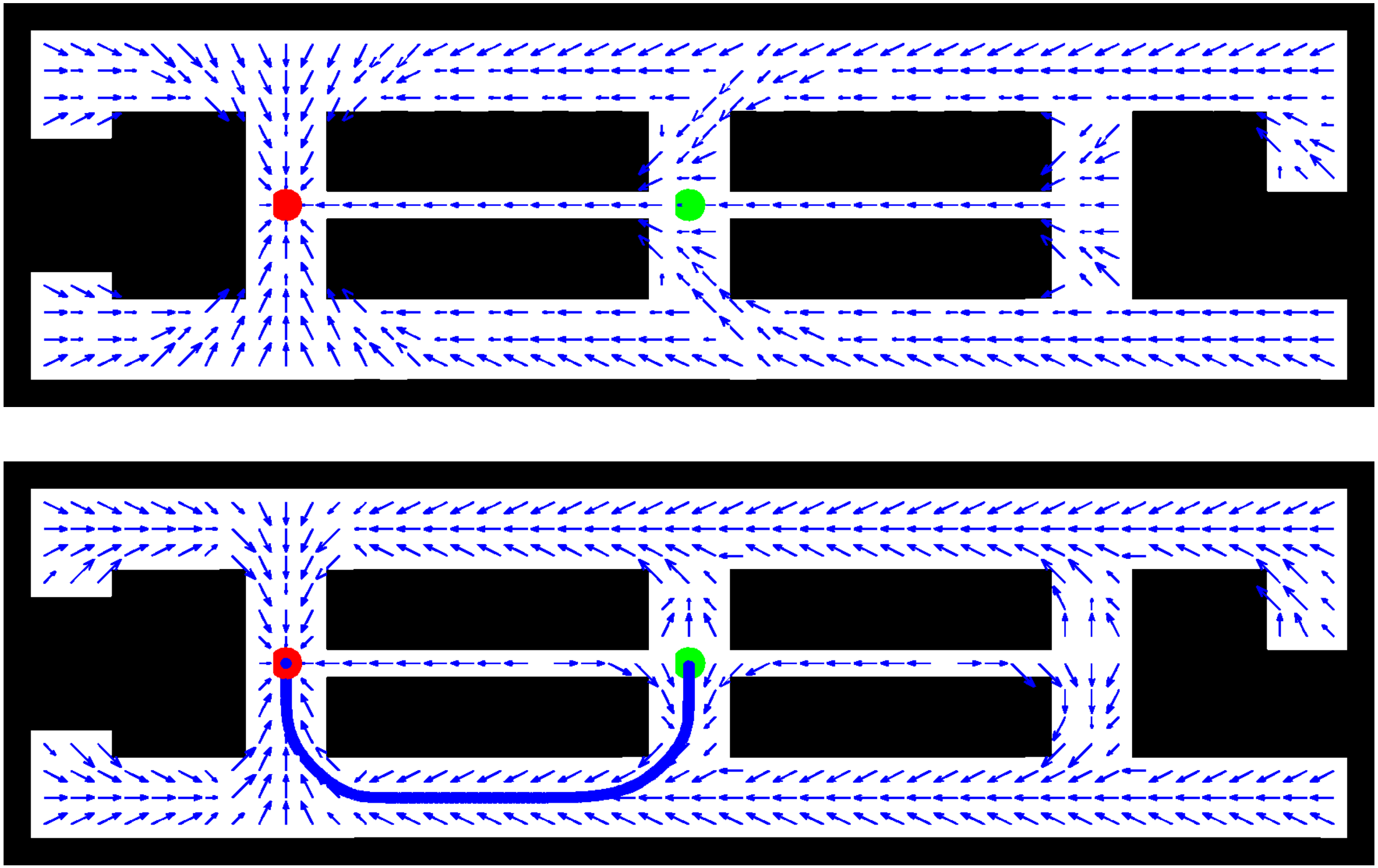
Reinforcement Learning



Robot Navigation Problem

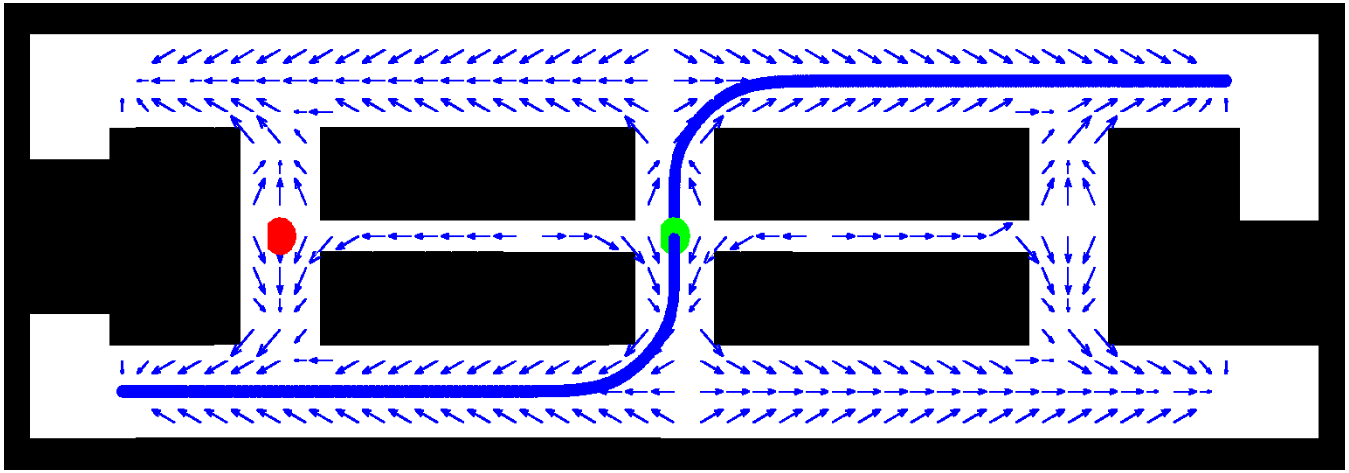


Uncertainty in Motion

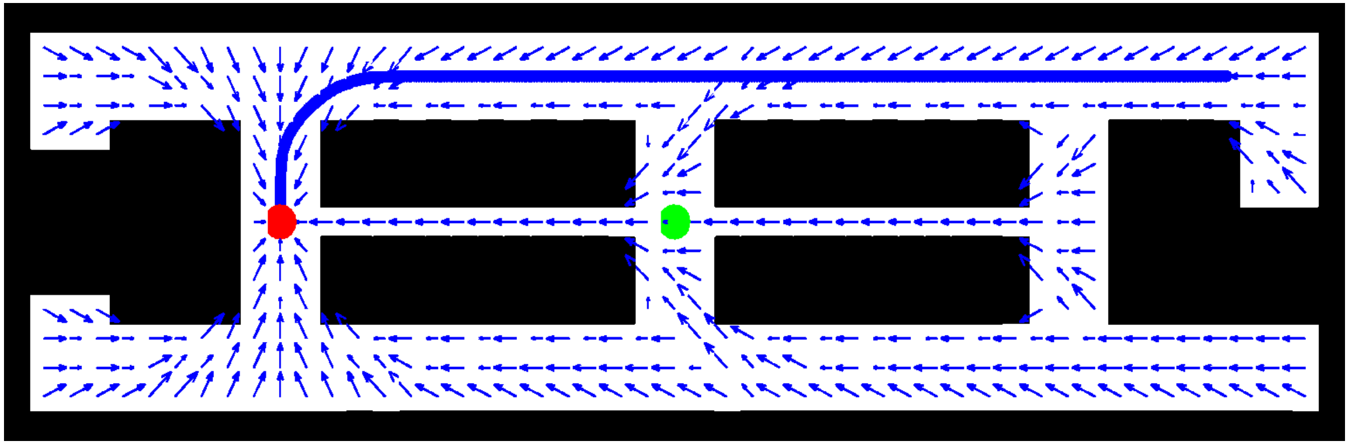


Uncertainty in Motion and Observation

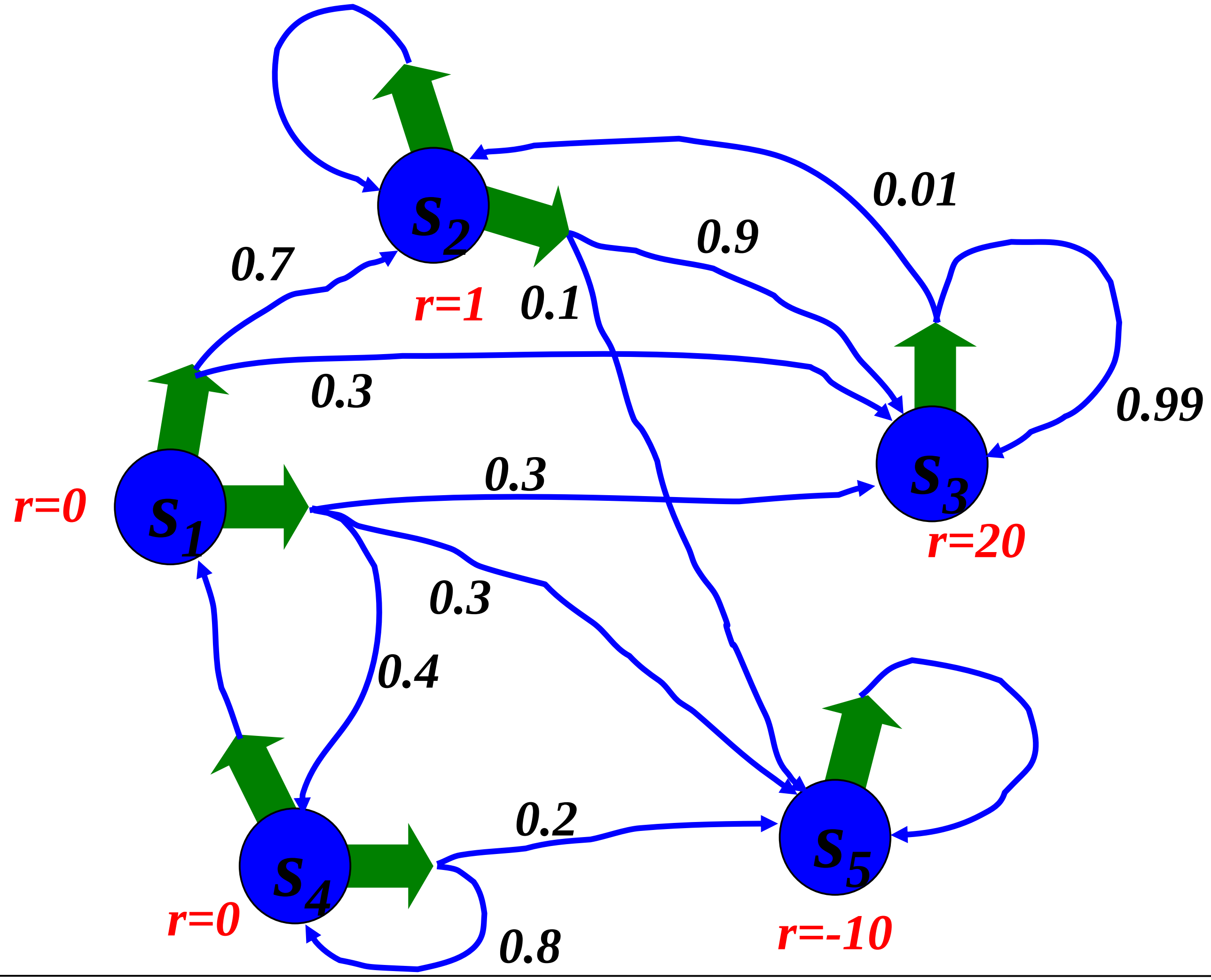
1st Step



2nd Step



Markov Decision Process



Markov Decision Process

			RIGHT GOAL
	OBSTACLE		WRONG GOAL
START POSITION			

Markov Decision Process Setup

- **Given:**

  States x, Actions u

  Transition probabilities $p(x'|u, x)$

  Reward function $r(x, u)$

- **Wanted:**

  Policy $\pi(x)$ that maximizes the future expected reward

Policy and Cumulative Reward

- Policy (fully observable case): $\pi : \quad x_t \to \quad u_t$

- Expected cumulative reward: $R_T \quad = \quad E\left[\sum_{\tau=0}^{T} \gamma^\tau r_{t+\tau}\right]$

$$\boxed{R_\infty \quad \leq \quad r_{\max} + \gamma r_{\max} + \gamma^2 r_{\max} + \gamma^3 r_{\max} + \ldots \quad = \quad \frac{r_{\max}}{1-\gamma}}$$

T=1 : greedy policy
T>1 : finite horizon case, typically no discount
T=infinity: infinite-horizon case, finite reward if discount < 1

Optimal Policy

- Expected cumulative reward of policy:

$$R_T^\pi(x_t) = E\left[\sum_{\tau=0}^{T} \gamma^\tau r_{t+\tau} \mid u_{t+\tau} = \pi(x_{t+\tau})\right]$$

- Optimal policy:

$$\pi^* = \underset{\pi}{\operatorname{argmax}} \quad R_T^\pi(x_t)$$

Return from Rewards

- **episodic** (vs. continuing) tasks
  “game over” after N steps
  optimal policy depends on N; harder to analyze

- **additive rewards**
  $R(x_t, x_{t+1}, \ldots) = r(x_t) + r(x_{t+1}) + r(x_{t+2}) + \ldots$
  infinite value for continuing tasks

- **discounted rewards**
  $R(x_t, x_{t+1}, \ldots) = r(x_t) + \gamma * r(x_{t+1}) + \gamma^2 * r(x_{t+2}) + \ldots$
  value bounded if rewards bounded

State Value Function

- Expected return when starting from x_t and following policy π:

$$V^{\pi}(x_t) = R_{\infty}^{\pi}(x_t)$$

- Bellman equation for policy π:

$$V^{\pi}(x) = \sum_{u} \pi(x,u) \left[r(x,u) + \gamma \int V^{\pi}(x') p(x'|x,u) dx' \right]$$

Optimal Value Function

- Optimal return for all possible policies:

$$V(x) = \max_{\pi} V^{\pi}(x)$$

- Bellman equation for optimal value function:

$$V(x) = \sum_{u} \pi(x,u) \left[r(x,u) + \gamma \int V(x') p(x'|x,u) dx' \right]$$

1-Step Optimal Policy and Value Function

- 1-step optimal policy:

$$\pi_1(x) = \underset{u}{\operatorname{argmax}} \quad r(x,u)$$

- Optimal value function of 1-step optimal policy:

$$V_1(x) = \quad \max_u \; r(x,u)$$

2-Step Optimal Policy and Value Function

- 2-step optimal policy:

$$\pi_2(x) \quad = \quad \operatorname*{argmax}_u \left[\underset{\boxed{\text{Current Reward}}}{r(x,u)} + \gamma \underbrace{\int V_1(x')\, p(x' \mid u, x)\, dx'}_{\text{Predicted Value}} \right]$$

- 2-step optimal value function:

$$V_2(x) \quad = \quad \max_u \left[\underset{\boxed{\text{Current Reward}}}{r(x,u)} + \gamma \underbrace{\int V_1(x')\, p(x' \mid u, x)\, dx'}_{\text{Predicted Value}} \right]$$

T-Step Optimal Policy and Value Function

- T-step optimal policy:

$$\pi_T(x) \quad = \quad \underset{u}{\operatorname{argmax}} \left[\underbrace{r(x,u)}_{\boxed{\text{Current Reward}}} + \gamma \int \underbrace{V_{T-1}(x')\, p(x' \mid u, x)\, dx'}_{\text{Predicted Value}} \right]$$

- T-step optimal value function:

$$V_T(x) \quad = \quad \max_{u} \left[\underbrace{r(x,u)}_{\boxed{\text{Current Reward}}} + \gamma \int \underbrace{V_{T-1}(x')\, p(x' \mid u, x)\, dx'}_{\text{Predicted Value}} \right]$$

Infinite Horizon

- Optimal value function:

$$V_\infty(x) = \max_u \Big[\underbrace{r(x,u)}_{\boxed{\text{Current Reward}}} + \gamma \underbrace{\int V_\infty(x')\, p(x' \mid u, x)\, dx'}_{\text{Predicted Value}} \Big]$$

- Bellman equation
  - ✓ Fixed point is optimal policy
  - ✓ Necessary and sufficient condition

Outlines

- Markov Decision Process (MDP)
- Value Iteration and Policy Iteration
- Partially Observable MDP (POMDP)
- POMDP Observation and Prediction
- POMDP Approximation

Value Iteration

for all *x* do

$$\hat{V} \longleftarrow r_{\min}$$

endfor

repeat until convergence

for all *x* do

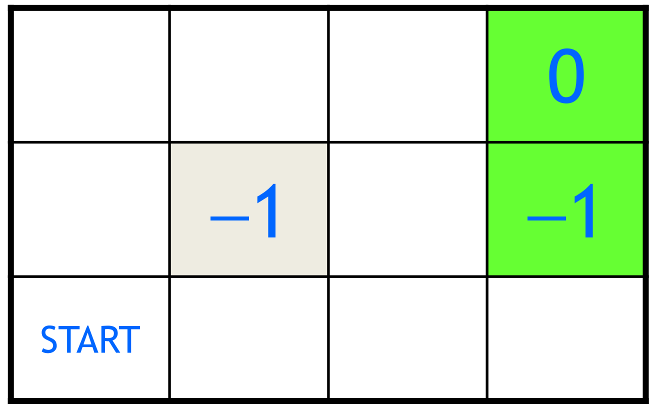
$$\hat{V}(x) \longleftarrow \max_u \left[r(x,u) + \gamma \int \hat{V}(x')\, p(x' \mid u, x)\, dx' \right]$$

endfor

endrepeat

$$\pi(x) = \operatorname*{argmax}_u \left[r(x,u) + \gamma \int \hat{V}(x')\, p(x' \mid u, x)\, dx' \right]$$

MDP Model



Environment and reward:

a) Green rectangle: destination, reward = 0 for any action
b) Black rectangle : wall, reward = -1
c) reward = - 0.1 for each step in other states
d) action = {up, down, left, right}

MDP Model

0	1	2	3
4	5	6	7
8	9	10	11

a) Position 3: reward = 0 for any action
b) Positions 5 and 7: wall, reward = -1
c) reward = - 0.1 for each step in other states
d) action = {up/0, down/1, left/2, right/3}

transition probabilities:

$$\{x\text{: } \{u_1\text{: } (x', p(x'|x, u_1), r), u_2\text{: } (x', p(x'|x, u_2), r), \\ u_3\text{: } (x', p(x'|x, u_3), r), u_4\text{: } (x', p(x'|x, u_4), r)\ \}\}$$


```
{0: {0: (0, 1.0, -0.1), 1: (4, 1.0, -0.1), 3: (1, 1.0, -0.1), 2: (0, 1.0, -0.1)}, 1: {0: (1,
1.0, -0.1), 1: (1, 1.0, -1), 3: (2, 1.0, -0.1), 2: (0, 1.0, -0.1)}, 2: {0: (2, 1.0, -0.1), 1:
(6, 1.0, -0.1), 3: (3, 1.0, -0.1), 2: (1, 1.0, -0.1)}, 3: {0: (3, 1.0, 0), 1: (3, 1.0, 0), 3:
(3, 1.0, 0), 2: (3, 1.0, 0)}, 4: {0: (0, 1.0, -0.1), 1: (8, 1.0, -0.1), 3: (4, 1.0, -1), 2: (4,
1.0, -0.1)}, 5: {0: (1, 1.0, -0.1), 1: (9, 1.0, -0.1), 3: (6, 1.0, -0.1), 2: (4, 1.0, -0.1)}, 6:
{0: (2, 1.0, -0.1), 1: (10, 1.0, -0.1), 3: (6, 1.0, -1), 2: (6, 1.0, -1)}, 7: {0: (3, 1.0,
-0.1), 1: (11, 1.0, -0.1), 3: (7, 1.0, -1), 2: (6, 1.0, -0.1)}, 8: {0: (4, 1.0, -0.1), 1: (8,
1.0, -0.1), 3: (9, 1.0, -0.1), 2: (8, 1.0, -0.1)}, 9: {0: (9, 1.0, -1), 1: (9, 1.0, -0.1), 3:
(10, 1.0, -0.1), 2: (8, 1.0, -0.1)}, 10: {0: (6, 1.0, -0.1), 1: (10, 1.0, -0.1), 3: (11, 1.0,
-0.1), 2: (9, 1.0, -0.1)}, 11: {0: (11, 1.0, -1), 1: (11, 1.0, -0.1), 3: (11, 1.0, -0.1), 2:
(10, 1.0, -0.1)}}
```

Value Iteration (I)

Value Function V^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	1	2	3
4	5	6	7
8	9	10	11

$\boxed{V^0(0) = 0.0}$ $\boxed{V^1(0) = -0.1}$

$r(0, up) + V^0(0)*p(0|0,up) = -0.1 + (-0.0)*1 = -0.1$

$r(0, do) + V^0(4)*p(4|0,do) = -0.1 + (-0.0)*1 = -0.1$

$\boxed{r(0, rig) + V^0(1)*p(1|0,rig) = -0.1 + (-0.0)*1 = -0.1}$

$r(0, lef) + V^0(0)*p(0|0,lef) = -0.1 + (-0.0)*1 = -0.1$

$\boxed{V^0(1) = 0.0}$ $\boxed{V^1(1) = -0.1}$

$r(1, up) + V^0(1)*p(1|1,up) = -0.1 + (-0.0)*1 = -0.1$

$r(1, do) + V^0(1)*p(1|1,do) = -1.0 + (-0.0)*1 = -1.0$

$\boxed{r(1, rig) + V^0(2)*p(2|1,rig) = -0.1 + (-0.0)*1 = -0.1}$

$r(1, lef) + V^0(0)*p(0|1,lef) = -0.1 + (-0.0)*1 = -0.1$

Value Iteration (II)

Value Function V^1

– 0.1	– 0.1	– 0.1	0.0
– 0.1	0.0	– 0.1	0.0
– 0.1	– 0.1	– 0.1	– 0.1

0	1	2	3
4	5	6	7
8	9	10	11

$\boxed{V^1(0) = -0.1}$ $\boxed{V^2(0) = -0.2}$

$r(0, up) + V^1(0)*p(0|0,up) = -0.1 + (-0.1)*1 = -0.2$

$r(0, do) + V^1(4)*p(4|0,do) = -0.1 + (-0.1)*1 = -0.2$

$\boxed{r(0, rig) + V^1(1)*p(1|0,rig) = -0.1 + (-0.1)*1 = -0.2}$

$r(0, lef) + V^1(0)*p(0|0,lef) = -0.1 + (-0.1)*1 = -0.2$

$\boxed{V^1(1) = -0.1}$ $\boxed{V^2(1) = -0.2}$

$r(1, up) + V^1(1)*p(1|1,up) = -0.1 + (-0.1)*1 = -0.2$

$r(1, do) + V^1(1)*p(1|1,do) = -1.0 + (-0.1)*1 = -1.1$

$\boxed{r(1, rig) + V^1(2)*p(2|1,rig) = -0.1 + (-0.1)*1 = -0.2}$

$r(1, lef) + V^1(0)*p(0|1,lef) = -0.1 + (-0.1)*1 = -0.2$

Value Iteration (III)

Value Function V^2

– 0.2	– 0.2	– 0.1	0.0
– 0.2	0.0	– 0.2	0.0
– 0.2	– 0.2	– 0.2	– 0.2

$\boxed{V^2(0) = -0.2}$ $\boxed{V^3(0) = -0.3}$

$r(0, up) + V^2(0)*p(0|0,up) = -0.1+ (-0.2)*1= -0.3$

$r(0, do) + V^2(4)*p(4|0,do) = -0.1+ (-0.2)*1= -0.3$

$\boxed{r(0, rig) + V^2(1)*p(1|0,rig) = -0.1+ (-0.2)*1= -0.3}$

$r(0, lef) + V^2(0)*p(0|0,lef) = -0.1+ (-0.2)*1= -0.3$

0	1	2	3
4	5	6	7
8	9	10	11

$\boxed{V^2(1) = -0.2}$ $\boxed{V^3(1) = -0.2}$

$r(1, up) + V^2(1)*p(1|1,up) = -0.1+ (-0.2)*1= -0.3$

$r(1, do) + V^2(1)*p(1|1,do) = -1.0+ (-0.2)*1= -1.2$

$\boxed{r(1, rig) + V^2(2)*p(2|1,rig) = -0.1+ (-0.1)*1= -0.2}$

$r(1, lef) + V^2(0)*p(0|1,lef) = -0.1+ (-0.2)*1= -0.3$

Value Iteration (IV)

Value Function V^3

– 0.3	– 0.2	– 0.1	0.0
– 0.3	0.0	– 0.2	0.0
– 0.3	– 0.3	– 0.3	– 0.3

0	1	2	3
4	5	6	7
8	9	10	11

$\boxed{V^3(0) = -0.2}$ $\boxed{V^4(0) = -0.3}$

$r(0, up) + V^3(0)*p(0|0,up) = -0.1 + (-0.3)*1 = -0.4$

$r(0, do) + V^3(4)*p(4|0,do) = -0.1 + (-0.3)*1 = -0.4$

$\boxed{r(0, rig) + V^3(1)*p(1|0,rig) = -0.1 + (-0.2)*1 = -0.3}$

$r(0, lef) + V^3(0)*p(0|0,lef) = -0.1 + (-0.3)*1 = -0.4$

$\boxed{V^3(1) = -0.2}$ $\boxed{V^4(1) = -0.2}$

$r(1, up) + V^3(1)*p(1|1,up) = -0.1 + (-0.2)*1 = -0.3$

$r(1, do) + V^3(1)*p(1|1,do) = -1.0 + (-0.2)*1 = -1.2$

$\boxed{r(1, rig) + V^3(2)*p(2|1,rig) = -0.1 + (-0.1)*1 = -0.2}$

$r(1, lef) + V^3(0)*p(0|1,lef) = -0.1 + (-0.3)*1 = -0.4$

Value Iteration (V)

Value Function V^4

– 0.3	– 0.2	– 0.1	0.0
– 0.4	0.0	– 0.2	0.0
– 0.4	– 0.4	– 0.3	– 0.4

0	1	2	3
4	5	6	7
8	9	10	11

$\boxed{V^4(0) = -0.2}$ $\boxed{V^5(0) = -0.3}$

$r(0, up) + V^1(0)*p(0|0,up) = -0.1+ (-0.3)*1= -0.4$

$r(0, do) + V^1(4)*p(4|0,do) = -0.1+ (-0.4)*1= -0.5$

$\boxed{r(0, rig) + V^1(1)*p(1|0,rig) = -0.1+ (-0.2)*1= -0.3}$

$r(0, lef) + V^1(0)*p(0|0,lef) = -0.1+ (-0.3)*1= -0.4$

$\boxed{V^4(1) = -0.2}$ $\boxed{V^5(1) = -0.2}$

$r(1, up) + V^1(1)*p(1|1,up) = -0.1+ (-0.2)*1= -0.3$

$r(1, do) + V^1(1)*p(1|1,do) = -1.0+ (-0.2)*1= -1.2$

$\boxed{r(1, rig) + V^1(2)*p(2|1,rig) = -0.1+ (-0.1)*1= -0.2}$

$r(1, lef) + V^1(0)*p(0|1,lef) = -0.1+ (-0.3)*1= -0.4$

Stationary Value Function

Stationary Value Function

– 0.3	– 0.2	– 0.1	0.0
– 0.4	0.0	– 0.2	0.0
– 0.5	– 0.4	– 0.3	– 0.4

0	1	2	3
4	5	6	7
8	9	10	11

$\boxed{V(0) = -0.3}$

r(0, up) + V(0)*p(0|0,up) = – 0.1+ (– 0.3)*1= – 0.4

r(0, do) + V(4)*p(4|0,do) = – 0.1+ (– 0.4)*1= – 0.5

$\boxed{r(0, rig) + V(1)*p(1|0,rig) = -0.1+ (-0.2)*1= -0.3}$

r(0, lef) + V(0)*p(0|0,lef) = – 0.1+ (– 0.3)*1= – 0.4

$\boxed{V(1) = -0.2}$

r(1, up) + V(1)*p(1|1,up) = – 0.1+ (– 0.2)*1= – 0.3

r(1, do) + V(1)*p(1|1,do) = – 1.0+ (– 0.2)*1= – 1.0

$\boxed{r(1, rig) + V(2)*p(2|1,rig) = -0.1+ (-0.1)*1= -0.2}$

r(1, lef) + V(0)*p(0|1,lef) = – 0.1+ (– 0.3)*1= – 0.4

Optimal Policy for Value Iteration

Stationary Value Function

–0.3	–0.2	–0.1	0.0
–0.4	–0.0	–0.2	–0.0
–0.5	–0.4	–0.3	–0.4

V(0) = – 0.3

Optimal Action: right →

r(0, up) + V(0)*p(0|0,up) = – 0.1+ (– 0.3)*1= – 0.4

r(0, do) + V(4)*p(4|0,do) = – 0.1+ (– 0.4)*1= – 0.5

r(0, rig) + V(1)*p(1|0,rig) = – 0.1+ (– 0.2)*1= – 0.3

r(0, lef) + V(0)*p(1|0,lef) = – 0.1+ (– 0.3)*1= – 0.4

Optimal Policy

→	→	→	●
↑	□	↑	□
↑ →	→	↑	←

V(1) = – 0.2

Optimal Action: right →

r(1, up) + V(1)*p(1|1,up) = – 0.1+ (– 0.2)*1= – 0.3

r(1, do) + V(1)*p(1|1,do) = – 1.0+ (– 0.0)*1= – 1.0

r(1, rig) + V(2)*p(2|1,rig) = – 0.1+ (– 0.1)*1= – 0.2

r(1, lef) + V(0)*p(0|1,lef) = – 0.1+ (– 0.3)*1= – 0.4

Policy Iteration

- Often the optimal policy has been reached long before the value function has converged.
- Policy iteration (1) calculates a new policy based on the current value function and (2) then calculates a new value function based on this policy.

(1) Policy improvement $$\pi^* = \underset{\pi}{\operatorname{argmax}} \quad R_T^\pi(x_t)$$

(2) Policy evaluation

$$R_T^\pi(x_t) \quad = \quad E\left[\sum_{\tau=0}^{T} \gamma^\tau r_{t+\tau} \mid u_{t+\tau} = \qquad \pi(x_{t+\tau})\right]$$

- Often converges faster to the optimal policy.

Policy Iteration

- Policy evaluation

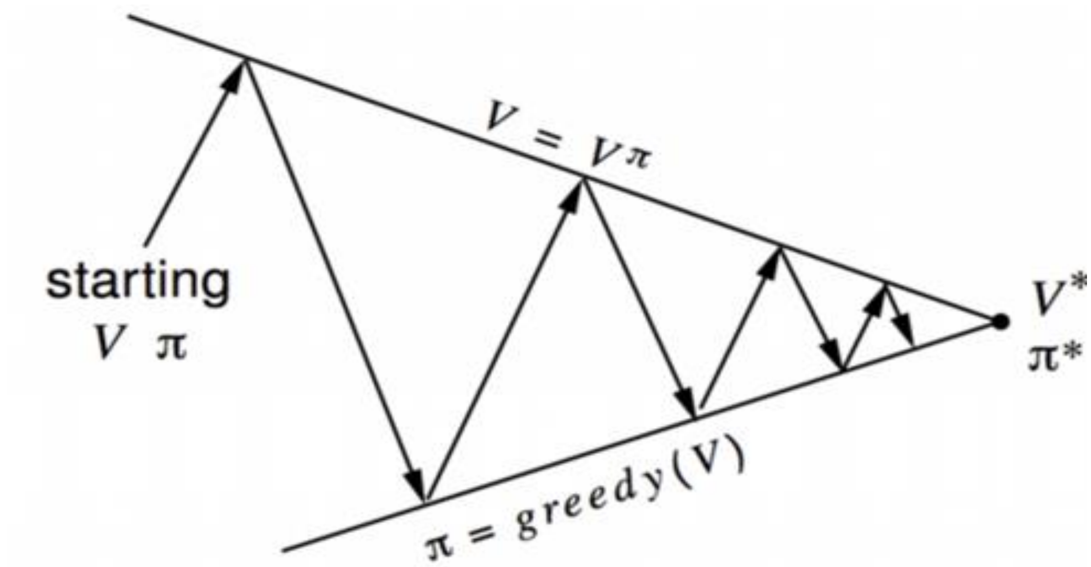
$$V_{k+1}^{\pi}(x) = \sum_{u} \pi(x,u) \left[r(x,u) + \gamma \int V_k^{\pi}(x') p(x'|x,u) dx' \right]$$

Until converged

- Policy improvement

$$\pi^*(x) = \arg\max_{u} \left[r(x,u) + \gamma \int V^{\pi}(x') p(x'|x,u) dx' \right]$$

Policy Iteration



Policy evaluation Estimate v_π

Iterative policy evaluation

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

Greedy policy improvement

Policy Evaluation Algorithm

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

Policy Iteration Algorithm

Policy Iteration (using iterative policy evaluation) for estimating $\pi \approx \pi_*$

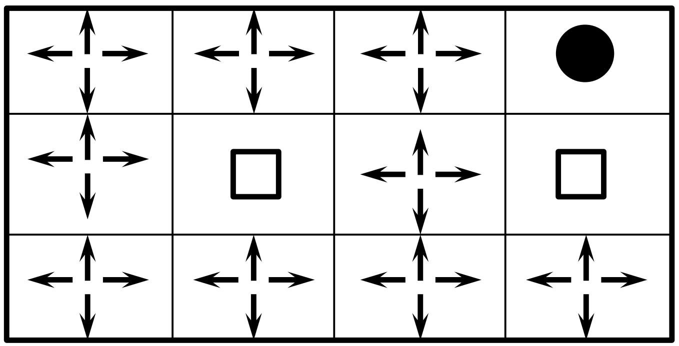
1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

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

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
   old-action $\leftarrow \pi(s)$
   $\pi(s) \leftarrow \operatorname{argmax}_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$
   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

Policy Iteration (I)

Policy π^0



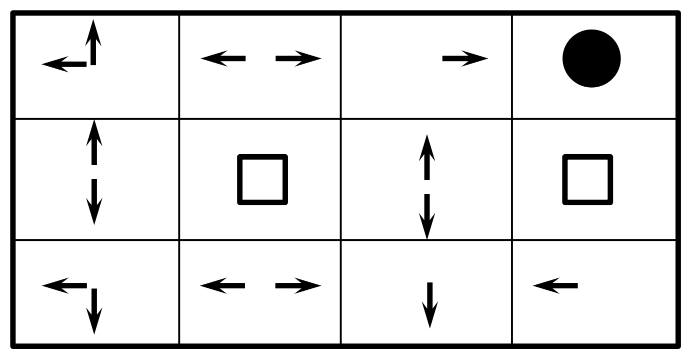
Initial Value Function

0.0	0.0	0.0	0.0
0.0		0.0	
0.0	0.0	0.0	0.0

Value Function V^{π^0}

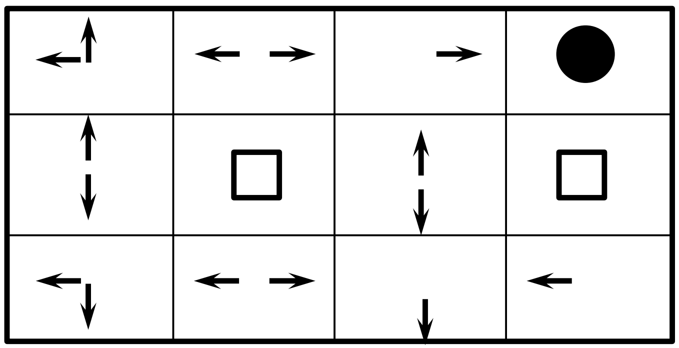
-0.1	-0.325	-0.1	0.0
-0.325		-0.55	
-0.1	-0.325	-0.1	-0.325

Policy π^1



Policy Iteration (II)

Policy π^1



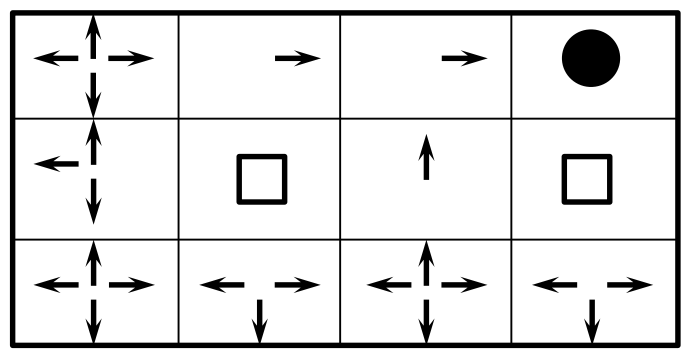
Value Function $V^{\pi 0}$

-0.1	-0.325	-0.1	0.0
-0.325		-0.55	
-0.1	-0.325	-0.1	-0.325

Value Function $V^{\pi 1}$

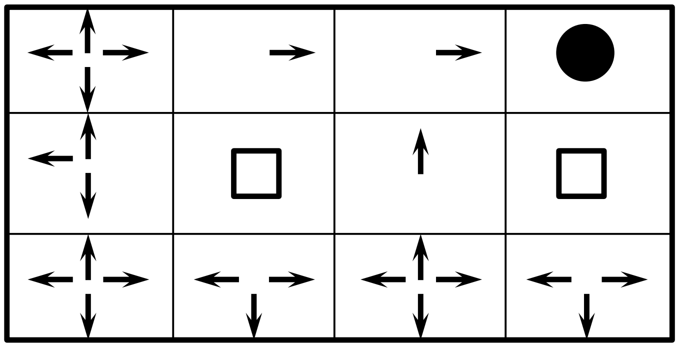
-0.2	-0.2	-0.1	0.0
-0.2		-0.2	
-0.2	-0.2	-0.2	-0.2

Policy π^2



Policy Iteration (III)

Policy π^2



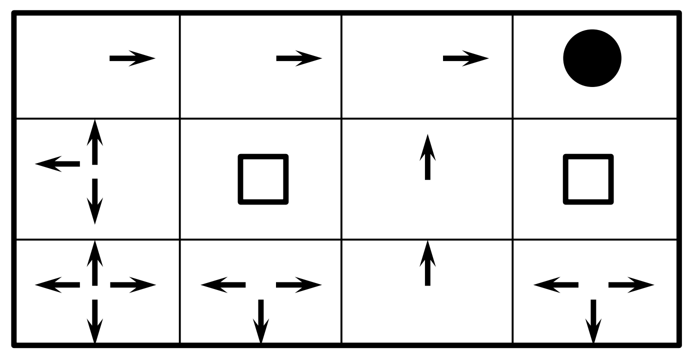
Value Function $V^{\pi 1}$

-0.2	-0.2	-0.1	0.0
-0.2		-0.2	
-0.2	-0.2	-0.2	-0.2

Value Function $V^{\pi 2}$

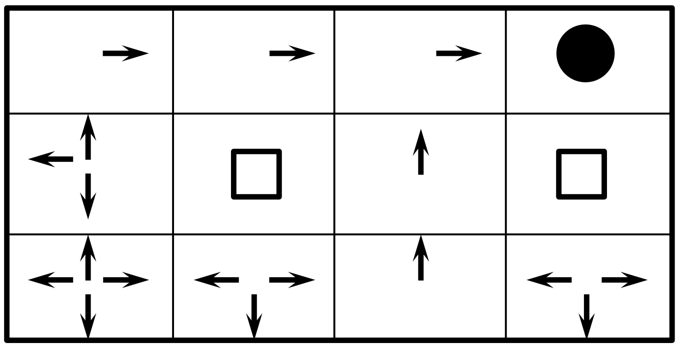
-0.3	-0.2	-0.1	0.0
-0.3		-0.2	
-0.3	-0.3	-0.3	-0.3

Policy π^3



Policy Iteration (IV)

Policy π^3



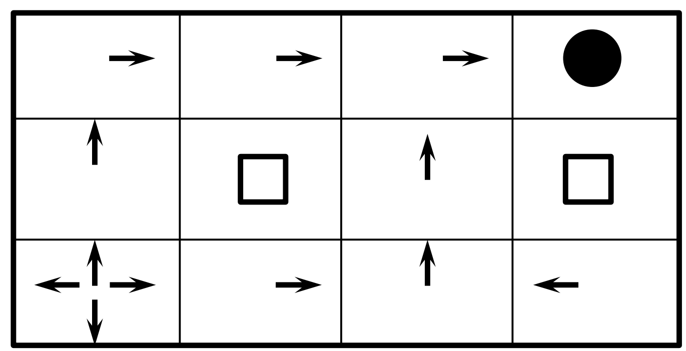
Value Function $V^{\pi 2}$

– 0.3	– 0.2	– 0.1	0.0
– 0.3		– 0.2	
– 0.3	– 0.3	– 0.3	– 0.3

Value Function $V^{\pi 3}$

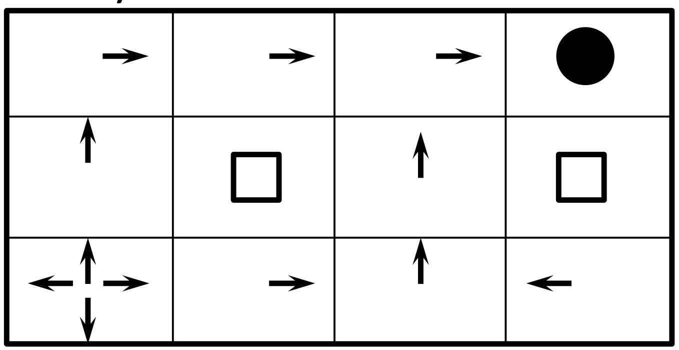
– 0.3	– 0.2	– 0.1	0.0
– 0.4	0.0	– 0.2	0.0
– 0.4	– 0.4	– 0.3	– 0.4

Policy π^4



Policy Iteration (V)

Policy π^4



Value Function $V^{\pi 3}$

– 0.3	– 0.2	– 0.1	0.0
– 0.4		– 0.2	
– 0.4	– 0.4	– 0.3	– 0.4

Value Function $V^{\pi 4}$

– 0.3	– 0.2	– 0.1	0.0
– 0.4	0.0	– 0.2	0.0
– 0.5	– 0.4	– 0.3	– 0.4

Policy π^4