TTIC 31170:
Robot Learning and Estimation

Spring 2019

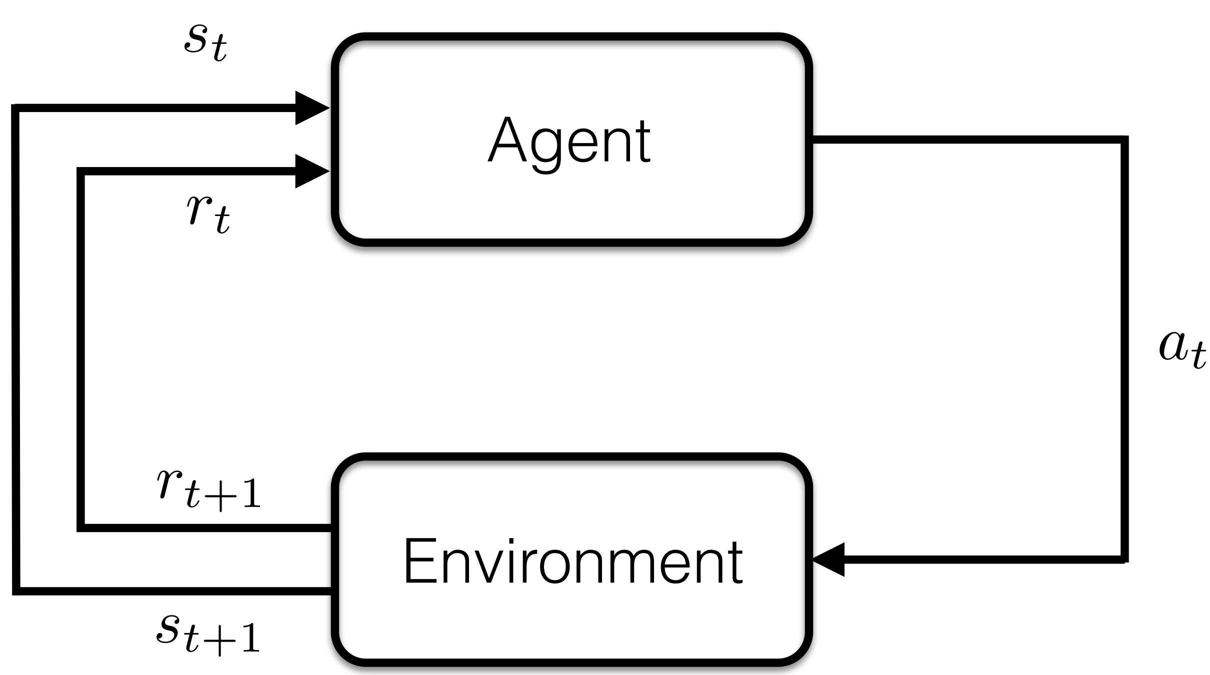
Matthew Walter
TTI-Chicago

Lecture 10: Markov Decision Processes

Some slides adapted from Pieter Abbeel & David Silver

Planning agent

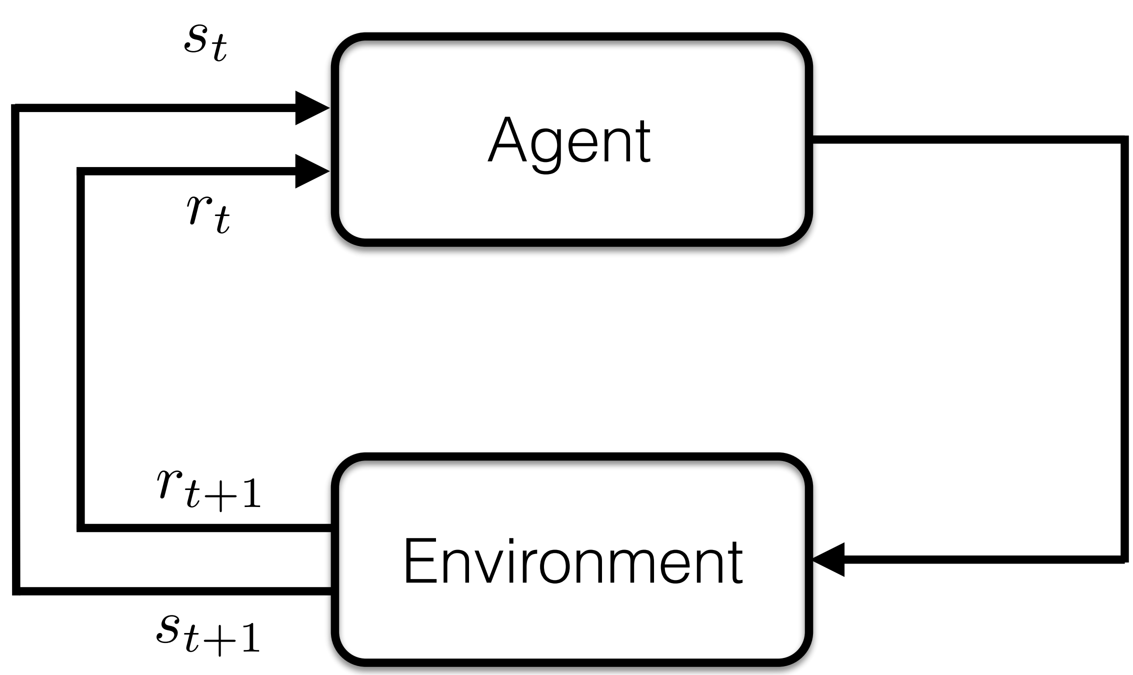
fully
vs.
partially
observable



deterministic
vs.
stochastic

Drawing adapted from Sutton & Barto, 1998

Classic planning

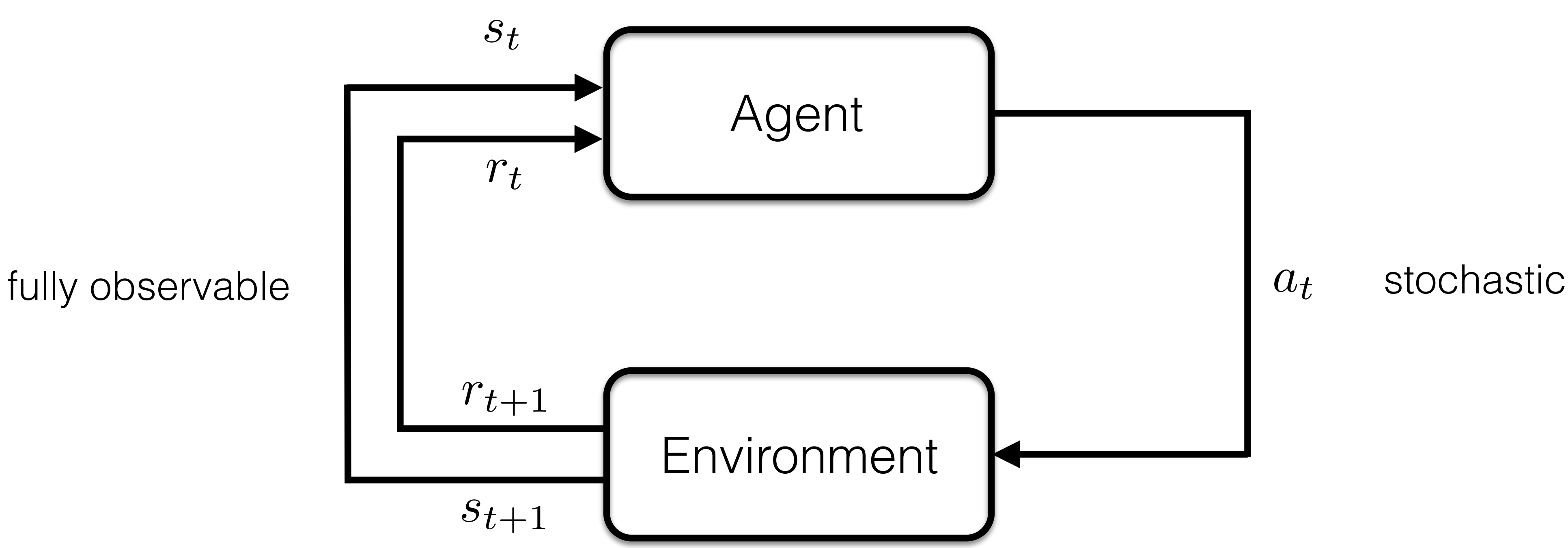


fully observable

deterministic

Drawing adapted from Sutton & Barto, 1998

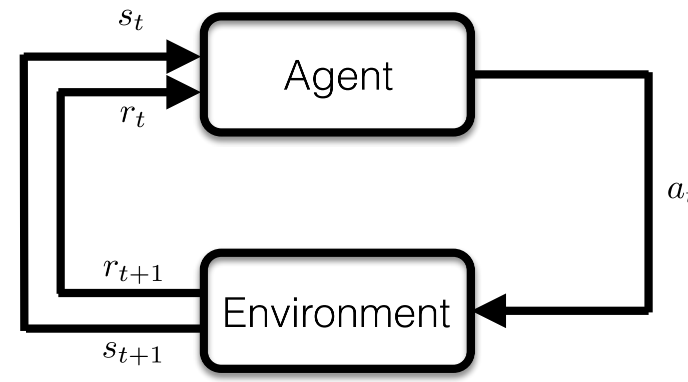
Markov Decision Process



Assumption: Agent gets to observe the state

Drawing adapted from Sutton & Barto, 1998

Markov Decision Process

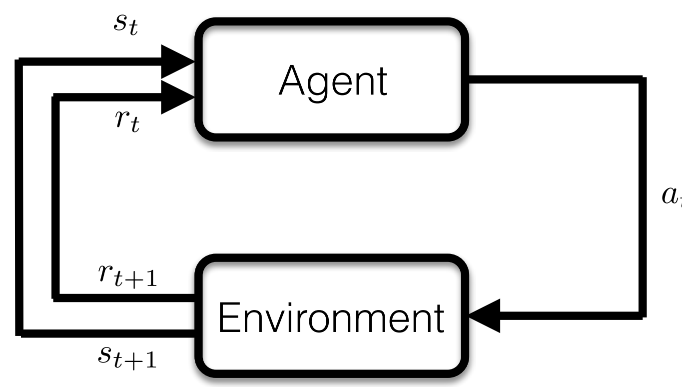


- Given: $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma, T \rangle$
  - $\mathcal{S}$: set of states (finite)
  - $\mathcal{A}$: set of actions (finite)
  - $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \times \{0, 1, 2, \ldots, T\} \to [0, 1]$
  - $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \times \{0, 1, 2, \ldots, T\} \to \mathbb{R}$
  - $\gamma \in (0, 1]$: discount factor
  - T : planning horizon

- Goal: Find ***policy*** $\pi : \mathcal{S} \times \{0, 1, 2, \ldots, T\} \to \mathcal{A}$ that maximizes expected sum of rewards

$$\pi^* = \arg\max_{\pi} E_{s^T} \left[\sum_{t=0}^{T} \gamma^t \mathcal{R}_t(s_t, a_t, s_{t+1}) | \pi \right]$$

Markov Decision Process

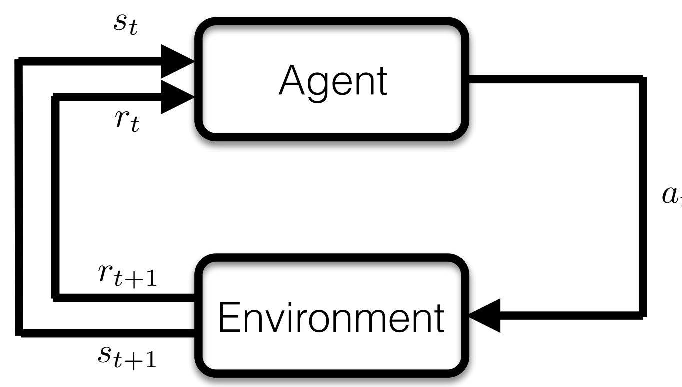


- Given: $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma, T \rangle$
  - $\mathcal{S}$: set of states (finite)
  - $\mathcal{A}$: set of actions (finite)
  - $\mathcal{T}_t(s, a, s') = P(s_{t+1} = s' | s_t = s, a_t = a)$
  - $\mathcal{R}_t(s, a, s') =$ reward for $(s_{t+1} = s', s_t = s, a_t = a)$
  - $\gamma \in (0, 1]$: discount factor
  - T : planning horizon

- Goal: Find ***policy*** $\pi : \mathcal{S} \times \{0, 1, 2, \ldots, T\} \rightarrow \mathcal{A}$ that maximizes expected sum of rewards

$$\pi^* = \arg\max_{\pi} E_{s^T} \left[\sum_{t=0}^{T} \gamma^t \mathcal{R}_t(s_t, a_t, s_{t+1}) | \pi \right]$$

Markov Decision Process



- Given: $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma, T \rangle$
  - $\mathcal{S}$: set of states (finite)
  - $\mathcal{A}$: set of actions (finite)
  - $\mathcal{T}_t(s, a, s') = P(s_{t+1} = s' | s_t = s, a_t = a)$
  - $\mathcal{R}_t(s, a, s') =$ reward for $(s_{t+1} = s', s_t = s, a_t = a)$
  - $\gamma \in (0, 1]$: discount factor
  - T : planning horizon

- Goa
expe

The policy is time-invariant in the infinite horizon case

π $t=0$

MDP: Objectives

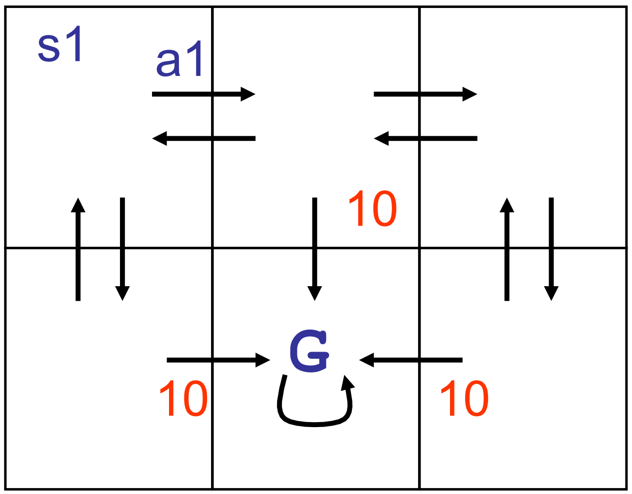
$$\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma, T \rangle$$

- Find a policy: $\pi : \mathcal{S} \times \{0, 1, 2, \ldots, T\} \rightarrow \mathcal{A}$

- that optimizes
  - minimizes (un)discounted expected cost to reach goal
  - maximizes (un)discounted expected reward

- given a finite/infinite/indefinite horizon

- assuming fully observable



Lifetime reward

- Optimal policy maximizes expected reward for time horizon
- Finite horizon:
  - Rewards accumulate for fixed period
  - \$100k + \$100k + \$100k = \$300k
- Infinite horizon:
  - Reward accumulates forever
  - \$100k + \$100k + …. = infinity
- Discounting:
  - Future rewards are worth less
  - \$100k + γ \$100k + γ^2 \$100k + …. = finite

Time horizon

- Greedy (myopic): $T = 1$
  - Only concerned with immediate reward
  - Easy to find solutions (polynomial time)

- Finite horizon: $1 < T < \infty$
  - Policy changes with time (different time horizons)
  - Harder to find optimal solution than infinite horizon case

- Infinite horizon: $T = \infty$
  - Policy doesn't change with time
  - Discount factor is critical ($\gamma < 1$ required in most cases)

Discounted rewards

- The return is the total discounted reward from time t

$$R_T = \mathcal{R}_0 + \gamma\mathcal{R}_1 + \gamma^2\mathcal{R}_2 + \ldots \gamma^T\mathcal{R}_T$$
$$= \sum_{k=0}^{T} \gamma^k \mathcal{R}_k$$

- The discount $\gamma \in (0, 1]$ is the present value of future rewards

- The value of receiving reward $\mathcal{R}$ after $k+1$ time steps is $\gamma^k\mathcal{R}$

- This values immediate reward over delayed reward
  - γ close to 0 leads to “myopic” evaluation
  - γ close to 1 leads to “far-sighted” evaluation

Why discount rewards?

- $\gamma < 1$ guarantees that optimal policy exists for infinite horizon case
- Exception: Undiscounted reward is ok if all sequences terminate
- Mathematically convenient (e.g., finite sum with infinite horizon)
- Avoids infinite returns in cyclic Markov processes
- Future may be uncertain
- For financial rewards, immediate rewards may earn more interest

Value function

- The value function $V_T^\pi(s)$ gives the long-term value of state s when executing policy π

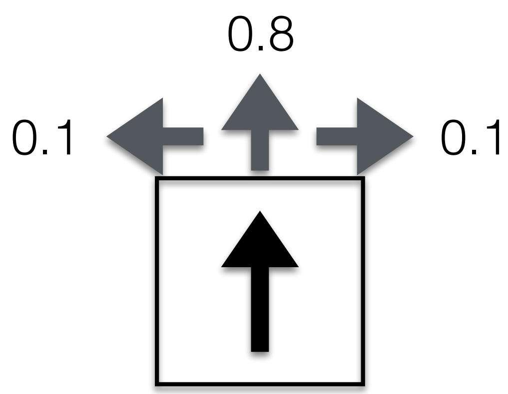
- The ***state-value function**** $V_T^\pi(s)$ is the expected return starting from state s and executing policy π

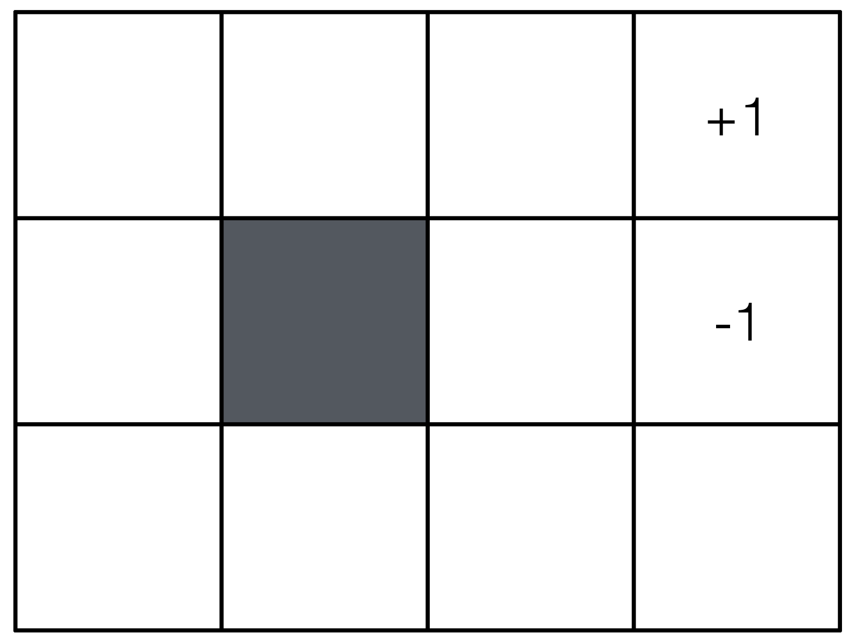
$$R_T^\pi(s) = \sum_{t=0}^{T} \gamma^t \mathcal{R}_t$$

$$V_T^\pi(s) = E_{s^T}\left[R_T^\pi(s)\right]$$

Example: Grid world

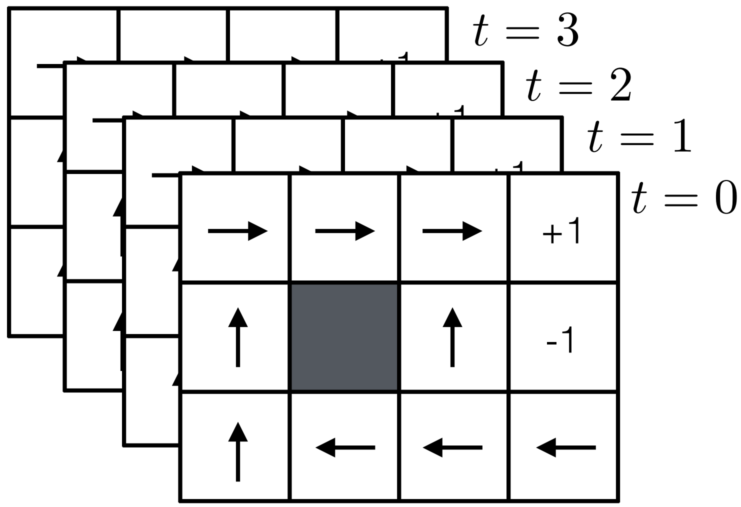
- Environment discretized into a set of cells
- Certain cells are unreachable (e.g., walls)
- State transitions are stochastic
- Big rewards come at goal





Solving MDPs

- MDP solver seeks an optimal policy $\pi^* : \mathcal{S} \times \{0, \ldots, T\} \rightarrow \mathcal{A}$
  - Policy π gives an action for each (state, time)



  - Optimal policy maximizes expected sum of rewards

- Contrast: If deterministic, we just need an optimal action sequence

Value function for a given policy

- $V_T^{\pi}(s)$ is the accumulated lifetime (T) reward resulting from starting in state s and executing policy π:

immediate reward

future accumulated lifetime reward

$$V_T^{\pi}(s) = \sum_{s'} P\left(s' | s, \pi_t(s)\right) \left(\mathcal{R}_t(s, \pi_t(s), s') + \gamma V_{T-1}^{\pi}(s')\right)$$

Value function for a given policy

- $V_T^\pi(s)$ is the accumulated lifetime (T) reward resulting from starting in state s and executing policy π:

$$\pi_1(s) = \arg\max_a \sum_{s'} P(s'|s,a)\mathcal{R}(s,a,s')$$

$$V_1(s) = \max_a \sum_{s'} P(s'|s,a)\mathcal{R}(s,a,s')$$

$$\pi^* = \arg\max_\pi E_{s^T}\left[\sum_{t=0}^{T} \gamma^t \mathcal{R}_t(s_t, a_t, s_{t+1})|\pi\right]$$

Value function for a given policy

- In the discounted, infinite horizon case, value function tends to equilibrium

$$V^{\pi}_{\infty}(s) = \sum_{s'} P\left(s'|s, \pi_t(s)\right)\left(\mathcal{R}_t(s, \pi_t(s), s') + \gamma V^{\pi}_{\infty}(s')\right)$$

Invariance known as the *Bellman equation*

Value function for a given policy

- In the discounted, infinite horizon case, value function tends to equilibrium

$$V_\infty^\pi(s) = \sum_{s'} P\left(s'|s, \pi_t(s)\right)\left(\mathcal{R}_t(s, \pi_t(s), s') + \gamma V_\infty^\pi(s')\right)$$

Invariance known as the *Bellman equation*

- Assumption: fixed-point convergence for finite horizon case

$$V^\pi(s) = \sum_{s'} P\left(s'|s, \pi_t(s)\right)\left(\mathcal{R}_t(s, \pi_t(s), s') + \gamma V^\pi(s')\right)$$

Bellman equation in matrix form

- The Bellman equation can be expressed as a matrix operation

$$\begin{bmatrix} V^\pi(1) \\ \vdots \\ V^\pi(n) \end{bmatrix} = \begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & \ldots & \vdots \\ P_{n1} & \cdots & P_{nn} \end{bmatrix} \left(\begin{bmatrix} R(1) \\ \vdots \\ R(n) \end{bmatrix} + \gamma \begin{bmatrix} V^\pi(1) \\ \vdots \\ V^\pi(n) \end{bmatrix} \right)$$

Bellman equation in matrix form

- The Bellman equation can be expressed as a matrix operation

$$\begin{bmatrix} V^\pi(1) \\ \vdots \\ V^\pi(n) \end{bmatrix} = \begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & \dots & \vdots \\ P_{n1} & \cdots & P_{nn} \end{bmatrix} \left(\begin{bmatrix} R(1) \\ \vdots \\ R(n) \end{bmatrix} + \gamma \begin{bmatrix} V^\pi(1) \\ \vdots \\ V^\pi(n) \end{bmatrix} \right)$$

- For small MDPs, the function can be solved directly ($\mathcal{O}(n^3)$)

$$\begin{aligned} \mathbb{V} &= \mathbb{P}(\mathbb{R} + \gamma\mathbb{V}) \\ (\mathbb{I} - \gamma\mathbb{P})\mathbb{V} &= \mathbb{P}\mathbb{R} \\ \mathbb{V} &= (\mathbb{I} - \gamma\mathbb{P})^{-1}\mathbb{P}\mathbb{R} \end{aligned}$$

Policy for given value function

A value function $V(s)$ defines the corresponding policy

$$\pi(s) = \arg\max_{a \in \mathcal{A}} \sum_{s'} P(s'|s,a) \left(\mathcal{R}(s,a,s') + \gamma V(s') \right)$$

Policy for given value function

A value function $V(s)$ defines the corresponding policy

$$\begin{aligned}\pi(s) &= \arg\max_{a\in\mathcal{A}} \sum_{s'} P(s'|s,a)\left(\mathcal{R}(s,a,s') + \gamma V(s')\right) \\ &= \arg\max_{a\in\mathcal{A}} Q(s,a)\end{aligned}$$

$$Q(s,a) = \sum_{s'} P(s'|s,a)\left(\mathcal{R}(s,a,s') + \gamma V(s')\right)$$ *Action-value function*

$$V(s) = \max_{a\in\mathcal{A}} Q(s,a)$$ *State-value function*

Optimal value function

- The *optimal value function* is the maximum value function over all policies

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

- The *optimal action-value function* 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 of the MDP
- An MDP is “solved” when we know the optimal value function

Optimal policy

- Define a partial ordering over policies

$$\pi \geq \pi' \text{ if } V^{\pi}(s) \geq V^{\pi'}(s) \; \forall s$$

- 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 $V^{\pi^*}(s) = V^*(s) \; \forall s$
  - All optimal policies achieve the optimal action-value function $Q^{\pi^*}(s,a) = Q^*(s,a) \; \forall s$

Value function to optimal policy

For a given state $s_t = s$, suppose that we knew the optimal value function for all other states

1. Consider all actions available at $s_t = s$

2. Select action a_i with greatest lifetime reward:

$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} \sum_{s'} P\left(s' | s, a\right) \left(\mathcal{R}_t(s, a, s') + \gamma V^*(s')\right)$$

Value function to optimal policy

For a given state $s_t = s$, suppose that we knew the optimal value function for all other states

1. Consider all actions available at $s_t = s$

2. Select action a_i with greatest lifetime reward:

$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} \sum_{s'} P\left(s'|s,a\right)\left(\mathcal{R}_t(s,a,s') + \gamma V^*(s')\right)$$

Determine optimal policy by finding optimal value function

Action-value function to optimal policy

- One can find the optimal policy by maximizing over $Q^*(s,a)$

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

- Every MDP has a deterministic optimal policy

- Knowing $Q^*(s,a)$ directly gives us the optimal policy

Solving MDPs

- Optimal Control: Given an MDP
  - Given an MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma, T \rangle$
  - Find the optimal policy π^*

- Exact methods:
  - Value iteration
  - Policy iteration
  - Q-learning
  - Sarsa

Value iteration

- Algorithm:
  - Initialize value function: $V_0^*(s) = 0 \text{ for all } s$
  - For $i = 1, 2, \ldots, T$:
    - For all states $s \in \mathcal{S}$:

value update
Bellman update/backup

$$V_{i+1}^*(s) \leftarrow \max_a \sum_{s'} P(s'|s,a)\left(\mathcal{R}(s,a,s') + \gamma V_i^*(s')\right)$$

$$\pi_{i+1}^*(s) \leftarrow \arg\max_a \sum_{s'} P(s'|s,a)\left(\mathcal{R}(s,a,s') + \gamma V_i^*(s')\right)$$

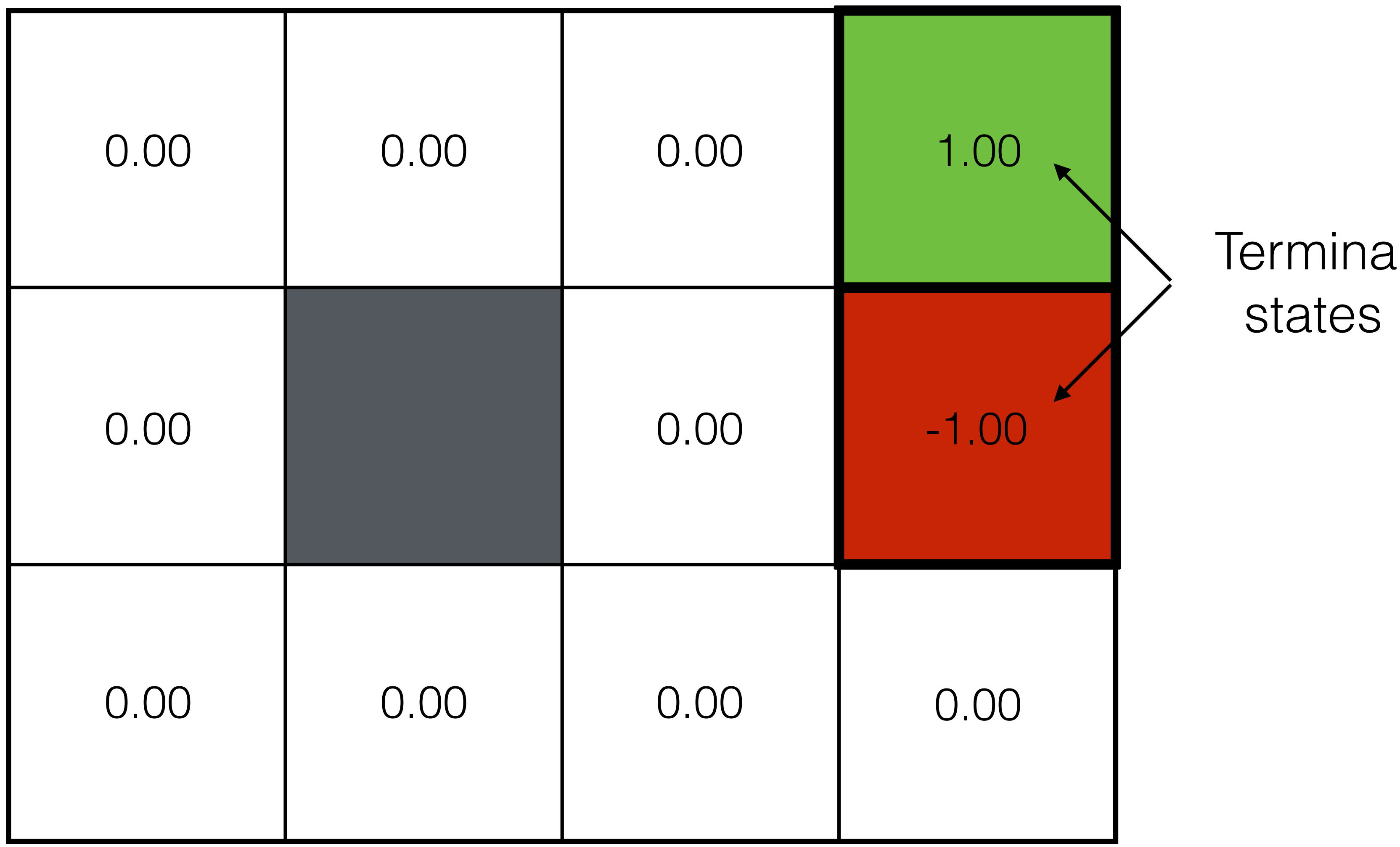
- $V_i^*(s)$: Expected sum of rewards accumulated when starting from state s and acting optimally for horizon of i steps

- $\pi_i^*(s)$: optimal action when in state s and acting for i steps

Value iteration: Example

noise = 0.2, γ = 0.9

One iteration



Value iteration: Example

noise = 0.2, γ = 0.9

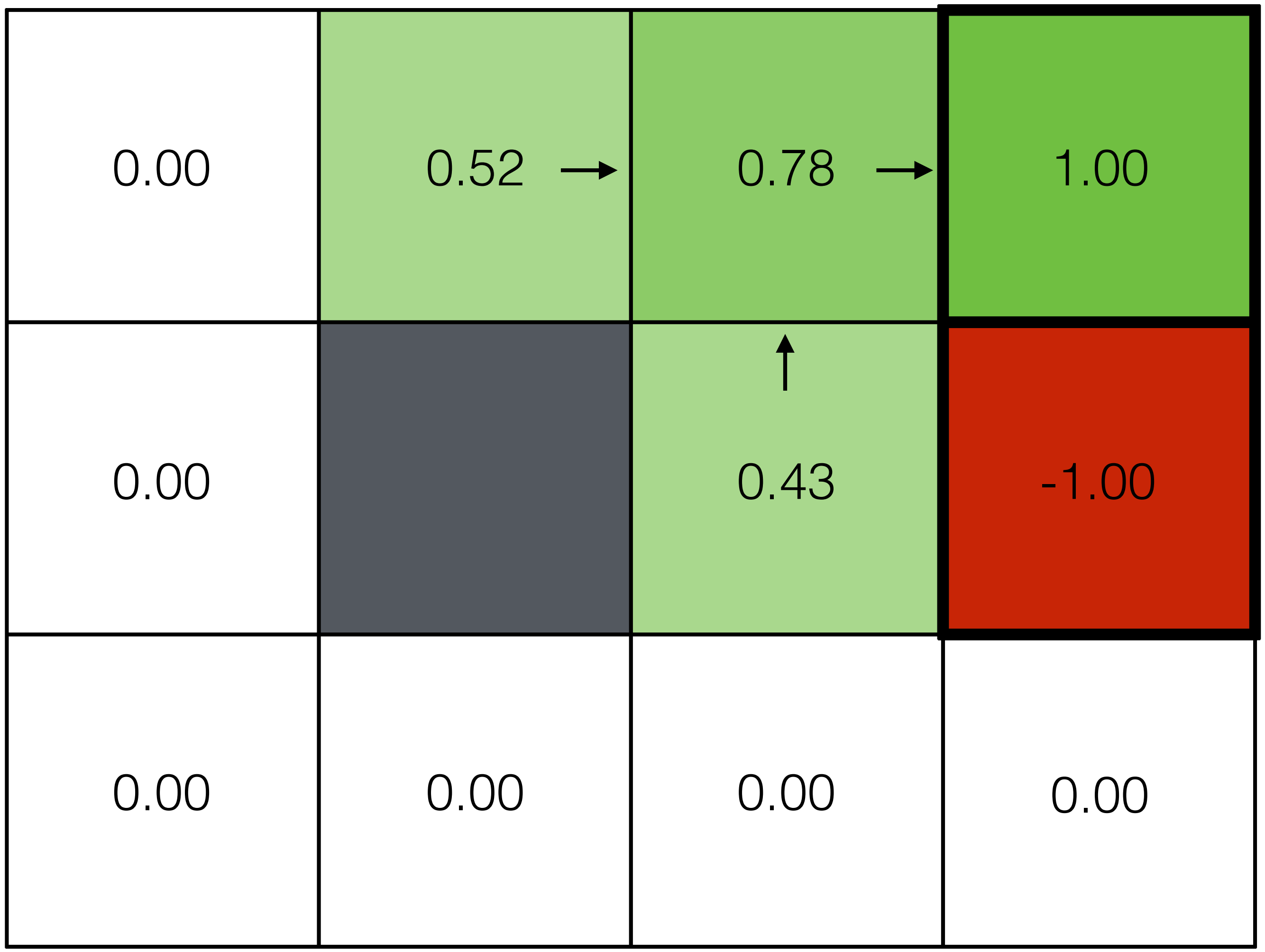
Two iterations

0.00	0.00	0.72 →	1.00
0.00		0.00	-1.00
0.00	0.00	0.00	0.00

Value iteration: Example

noise = 0.2, γ = 0.9

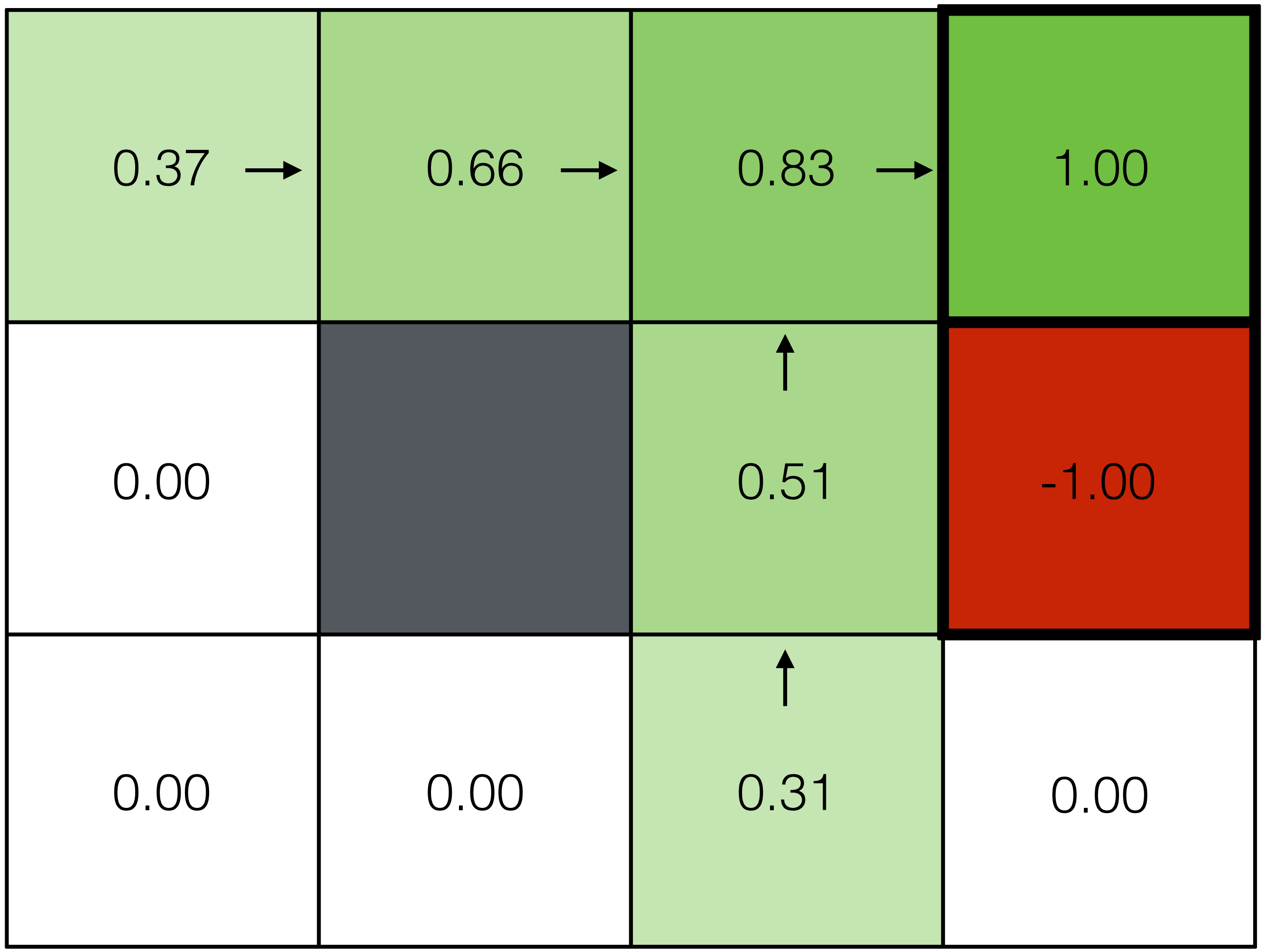
Three iterations



Value iteration: Example

noise = 0.2, γ = 0.9

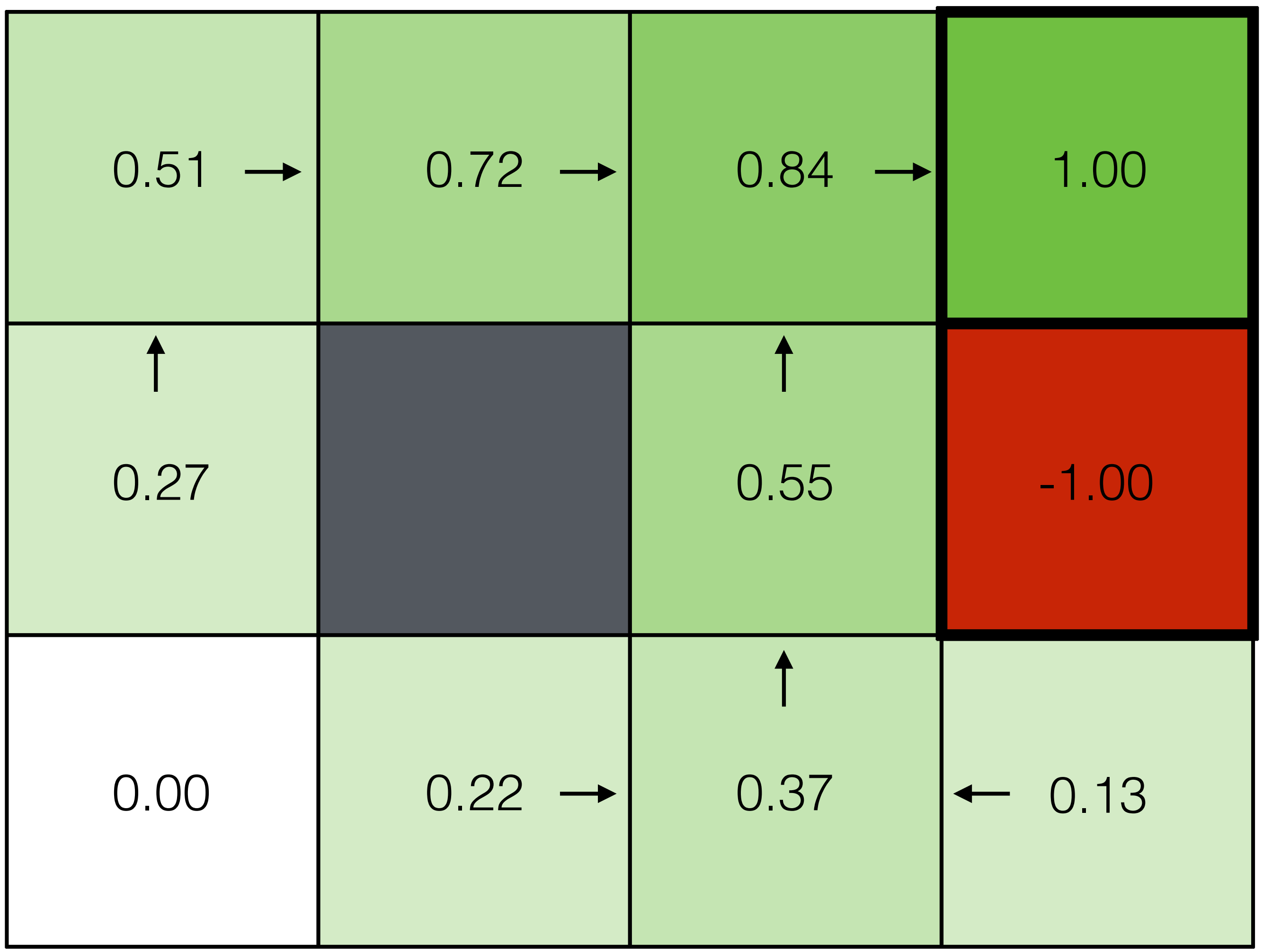
Four iterations



Value iteration: Example

noise = 0.2, γ = 0.9

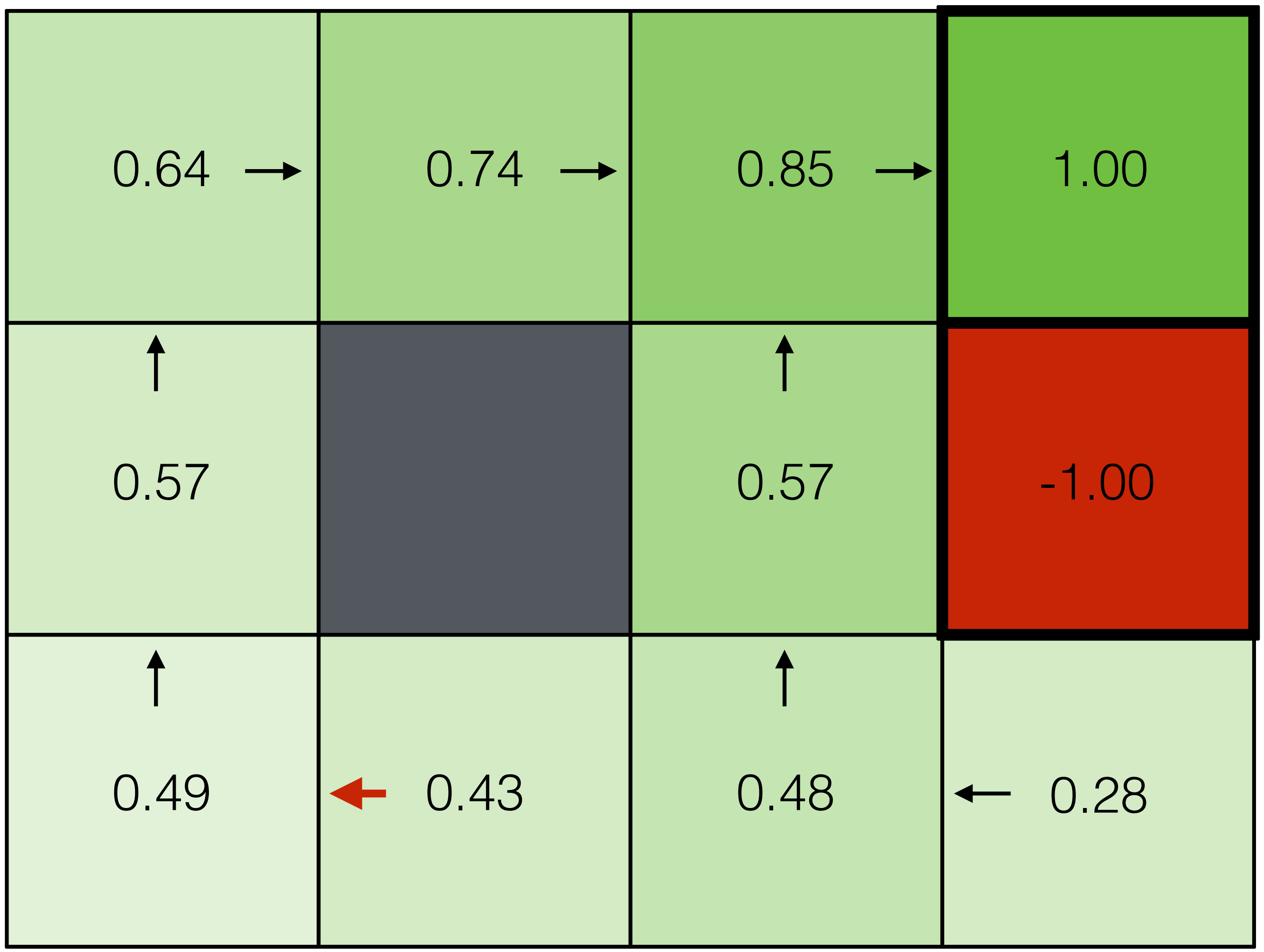
Four iterations



Value iteration: Example

noise = 0.2, γ = 0.9

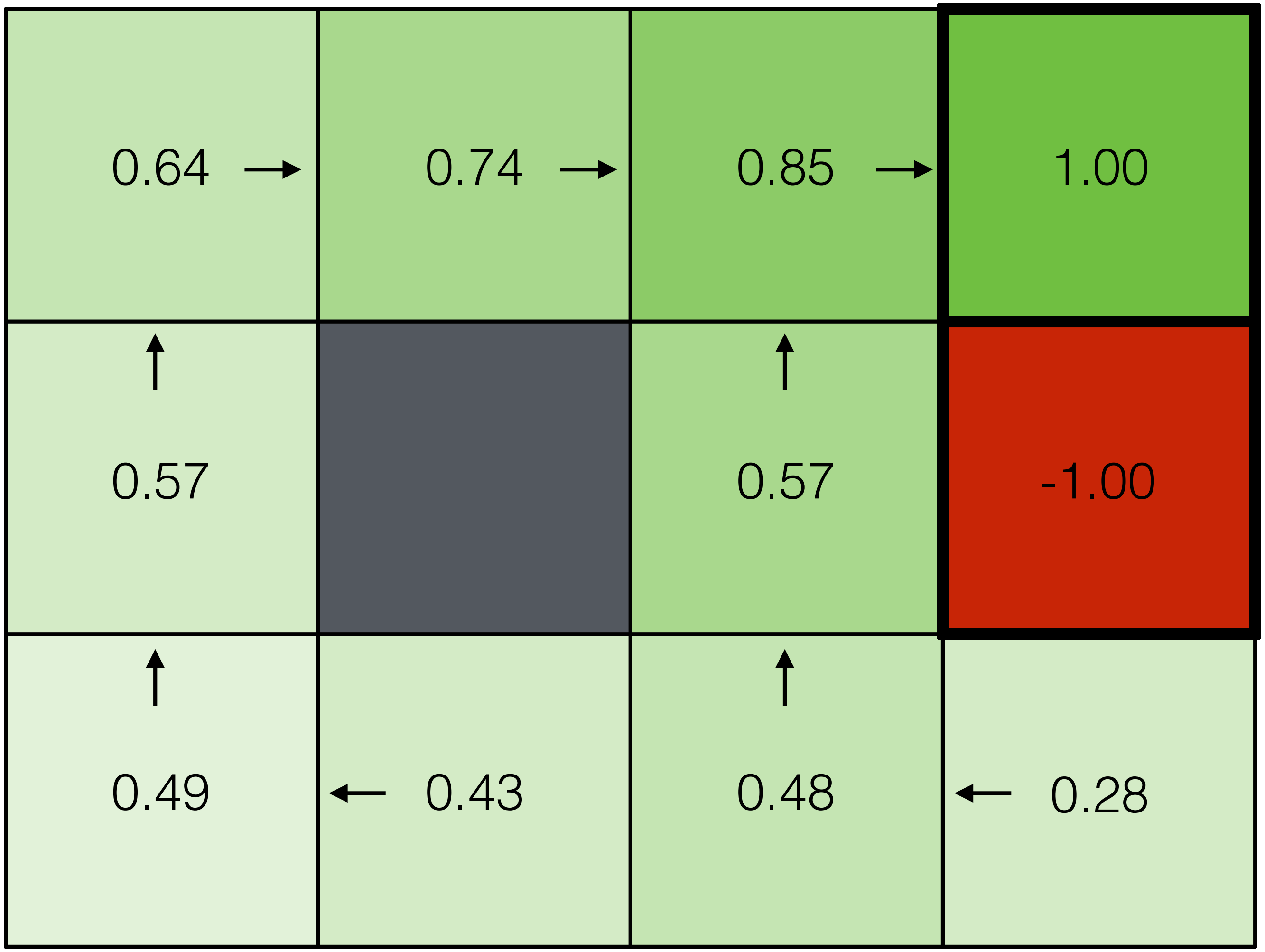
100 iterations



Value iteration: Example

noise = 0.2, γ = 0.9

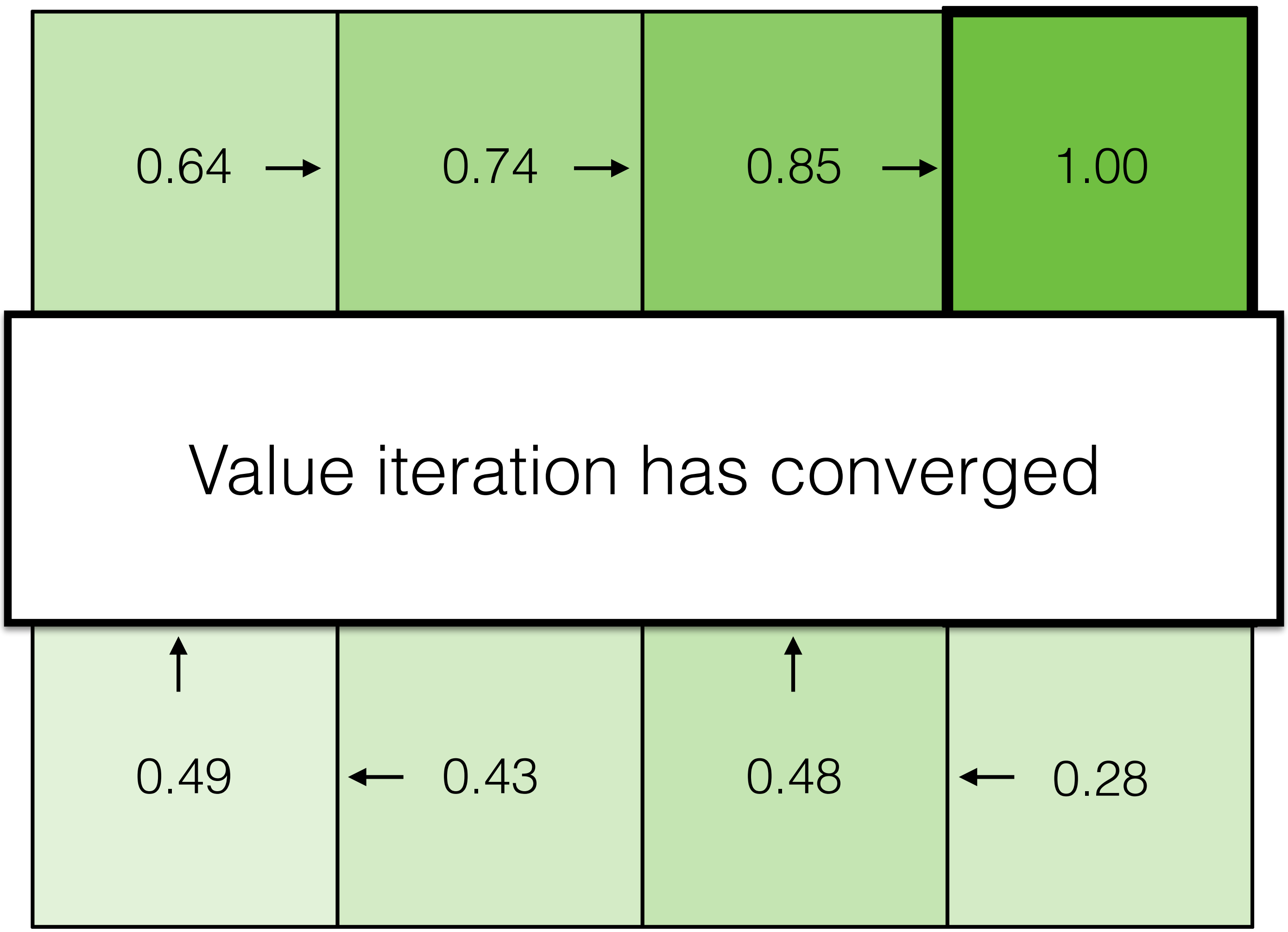
1,000 iterations



Value iteration: Example

noise = 0.2, γ = 0.9

1,000 iterations



Convergence of value iteration

- **Theorem**: Value iteration converges. At convergence, we have found the ***optimal value function*** $V^*(s)$ for the discounted infinite horizon problem that satisfies the Bellman equation:

$$V^*(s) = \max_a \sum_{s'} P(s'|s,a) \left(\mathcal{R}(s,a,s') + \gamma V^*(s')\right)$$

- Now we know how to act for infinite horizon w/ discounted reward
  - Run value iteration till convergence
  - Resulting value function yields optimal policy:

$$\pi^*(s) = \arg\max_a \sum_{s'} P(s'|s,a) \left(\mathcal{R}(s,a,s') + \gamma V^*(s')\right)$$

Convergence of value iteration

- In practice, terminate when values are “close enough”

$$\|V_{i+1}(s) - V_i(s)\| < \epsilon$$

$$\text{where } \|U\| = \max_s |U(s)|$$

- Theorem:

$$\|V_{i+1}(s) - V_i(s)\| < \epsilon \Rightarrow \|V_{i+1}(s) - V^*(s)\| < \frac{2\epsilon\gamma}{1-\gamma}$$

- Value iteration converges in polynomial time

Effects of discount and noise

noise = 0.0, γ = 0.1

0.00 →	0.00 →	0.01 ↓	0.01 →	0.10 ↓
0.00 ↓		0.10 ↓	0.10 →	1.00 ↓
0.00 ↓		1.00		10.00
0.00 →	0.01 →	↑ 0.10	0.10 →	↑ 1.00
-10.00	-10.00	-10.00	-10.00	-10.00

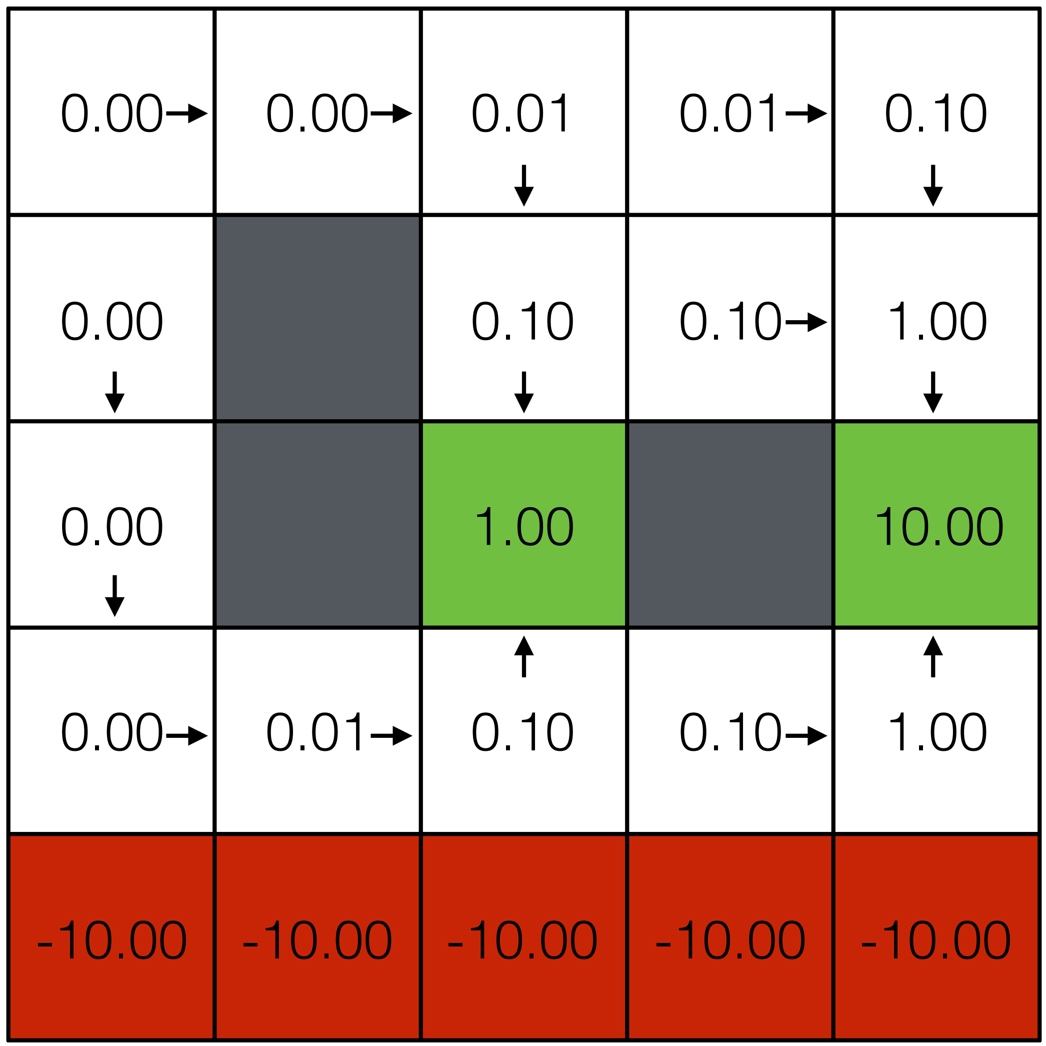
Effects of discount and noise

noise = 0.5, γ = 0.1

0.00 →	0.00 →	0.00 ↓	0.00 ↓	0.03 ↓
↑ 0.00		0.05 ↓	0.03 →	0.51 ↓
↑ 0.00		1.00		10.00
↑ 0.00	↑ 0.01	↑ 0.05	↑ 0.01	↑ 0.51
-10.00	-10.00	-10.00	-10.00	-10.00

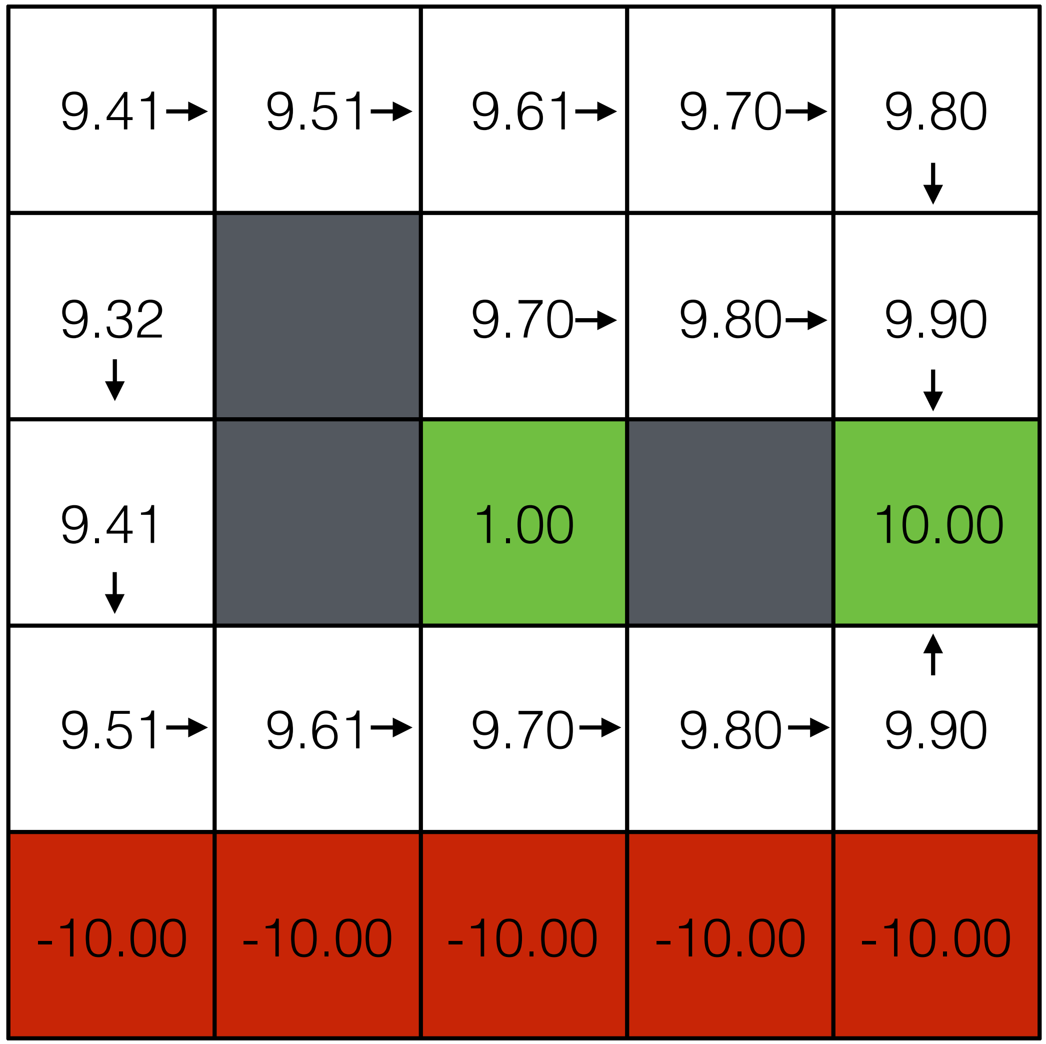
Effects of discount and noise

noise = 0.0, γ = 0.1



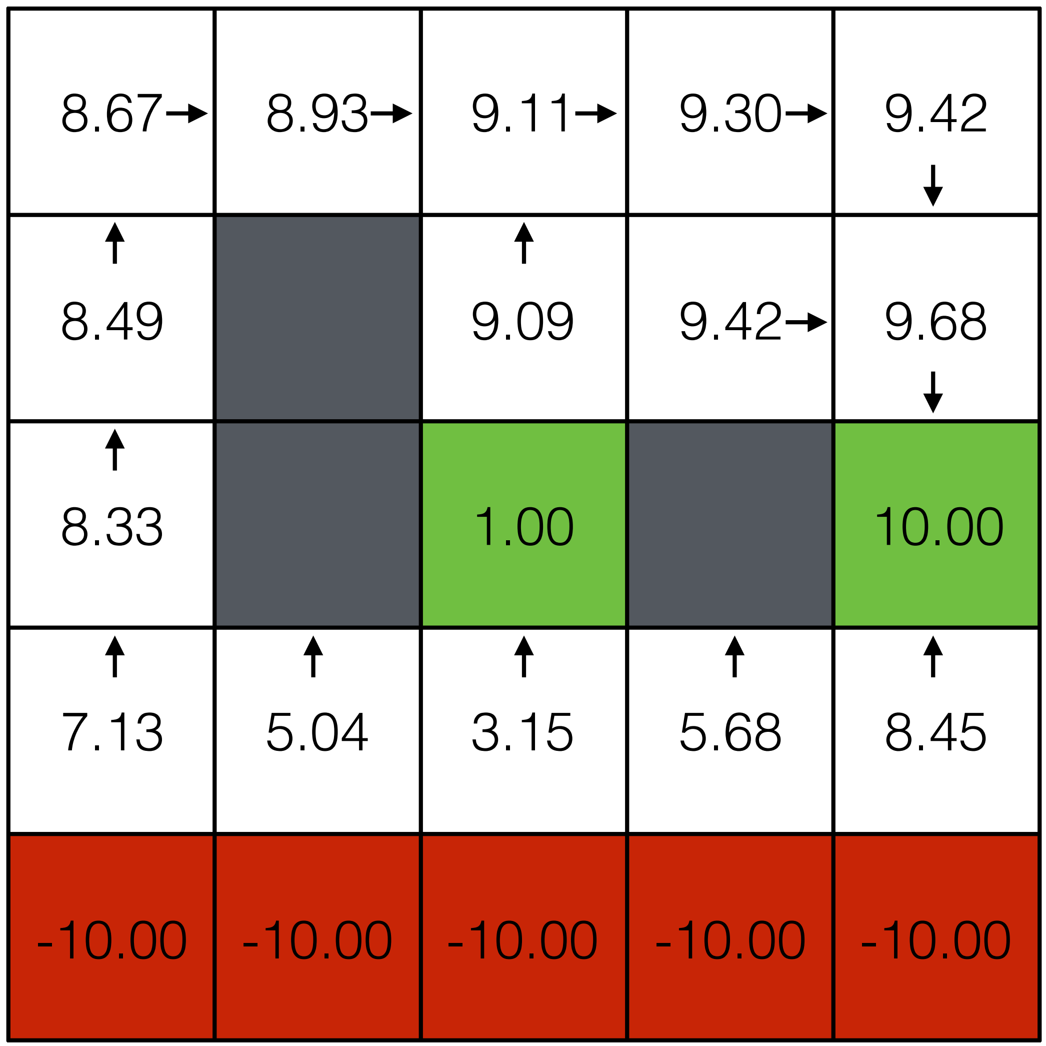
Effects of discount and noise

noise = 0.0, γ = 0.99



Effects of discount and noise

noise = 0.5, γ = 0.99



Solving MDPs

- Optimal Control: Given an MDP
  - Given an MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma, T \rangle$
  - Find the optimal policy π^*

- Exact methods:
  - Value iteration
  - Policy iteration
  - Q-learning
  - Sarsa

Policy improvement

- Suppose that we have determined the value function $V^{\pi}(s)$ for an arbitrary (deterministic) policy π

- Consider the corresponding action-value function

$$Q(s,a) = \sum_{s'} P(s'|s,a)(\mathcal{R}(s,a,s') + \gamma V^{\pi}(s'))$$

- Is it better to take $a \neq \pi(s)$ and subsequently follow π?
- Let π and π' be deterministic policies s.t.

$$Q^{\pi}(s, \pi'(s)) \geq V^{\pi}(s)\ \forall s \quad \Rightarrow \quad V^{\pi'}(s) \geq V^{\pi}(s)$$

Policy improvement

- More generally consider the greedy policy

$$\pi'(s) = \arg\max_{a \in \mathcal{A}} Q^{\pi}(s,a)$$
$$= \arg\max_{a \in \mathcal{A}} \sum_{s'} P(s'|s,a)(\mathcal{R}(s,a,s') + \gamma V^{\pi}(s'))$$

- $V^{\pi'}(s) = V^{\pi}(s)\ \forall s \quad \Rightarrow \quad V^{\pi'} = V^{\pi} = V^{*}$ and $\pi' = \pi = \pi^{*}$

- Gives rise to iterative procedure

$$\pi_0 \xrightarrow{\mathrm{E}} V^{\pi_0} \xrightarrow{\mathrm{I}} \pi_1 \xrightarrow{\mathrm{E}} V^{\pi_1} \xrightarrow{\mathrm{I}} \pi_2 \xrightarrow{\mathrm{E}} \cdots \xrightarrow{\mathrm{I}} \pi^{*} \xrightarrow{\mathrm{E}} V^{*}$$

Policy iteration

- **Step 1: Policy evaluation**: Calculate (non-optimal) utilities for some fixed policy until convergence
- **Step 2: Policy improvement**: Update policy using one step look-ahead with resulting converged (but not optimal) utilities as future values
- Repeat steps until policy converges
- Still optimal
- May converge faster than value iteration

Policy iteration

POLICY-EVALUATION$(\mathcal{S}, \pi, R, T, \gamma, \epsilon)$

$t \leftarrow 0$
for each state $s \in \mathcal{S}$
  do $V_0[s] \leftarrow 0$
repeat
  $change \leftarrow 0$
  $t \leftarrow t + 1$
  for each state $s \in \mathcal{S}$
    do $V_t[s] \leftarrow \left(\sum_{s'} T(s, \pi[s], s') \left[R(s, \pi[s], s') + \gamma \cdot V_{t-1}(s')\right]\right)$
      $change \leftarrow change + V_t[s] - V_{t-1}[s]$
until $change < \epsilon$
return V_t

Policy iteration

$\text{POLICY-ITERATION}(\mathcal{S}, \mathcal{A}, R, T, \gamma, \epsilon)$

for each state $s \in \mathcal{S}$
    do $i \leftarrow rand(0, |\mathcal{A}|)$
        $\pi[s] \leftarrow a_i$
$t \leftarrow 0$
$V_0 \leftarrow \text{POLICY-EVALUATION}(\mathcal{S}, \pi, R, T, \gamma, \epsilon)$
repeat
        $change \leftarrow 0$
        $t \leftarrow t + 1$
        for each state $s \in \mathcal{S}$
            do $\pi_t[s] \leftarrow max_a \left(\sum_{s'} T(s, a, s') \left[R(s, a, s') + \gamma \cdot V_{t-1}(s')\right]\right)$
        $V_t \leftarrow \text{POLICY-EVALUATION}(\mathcal{S}, \pi, R, T, \gamma, \epsilon)$
        for each state $s \in \mathcal{S}$
            do $change \leftarrow change + V_t[s] - V_{t-1}[s]$
    until $change < \epsilon$