Markov Decision Processes

Reading

Rensselaer

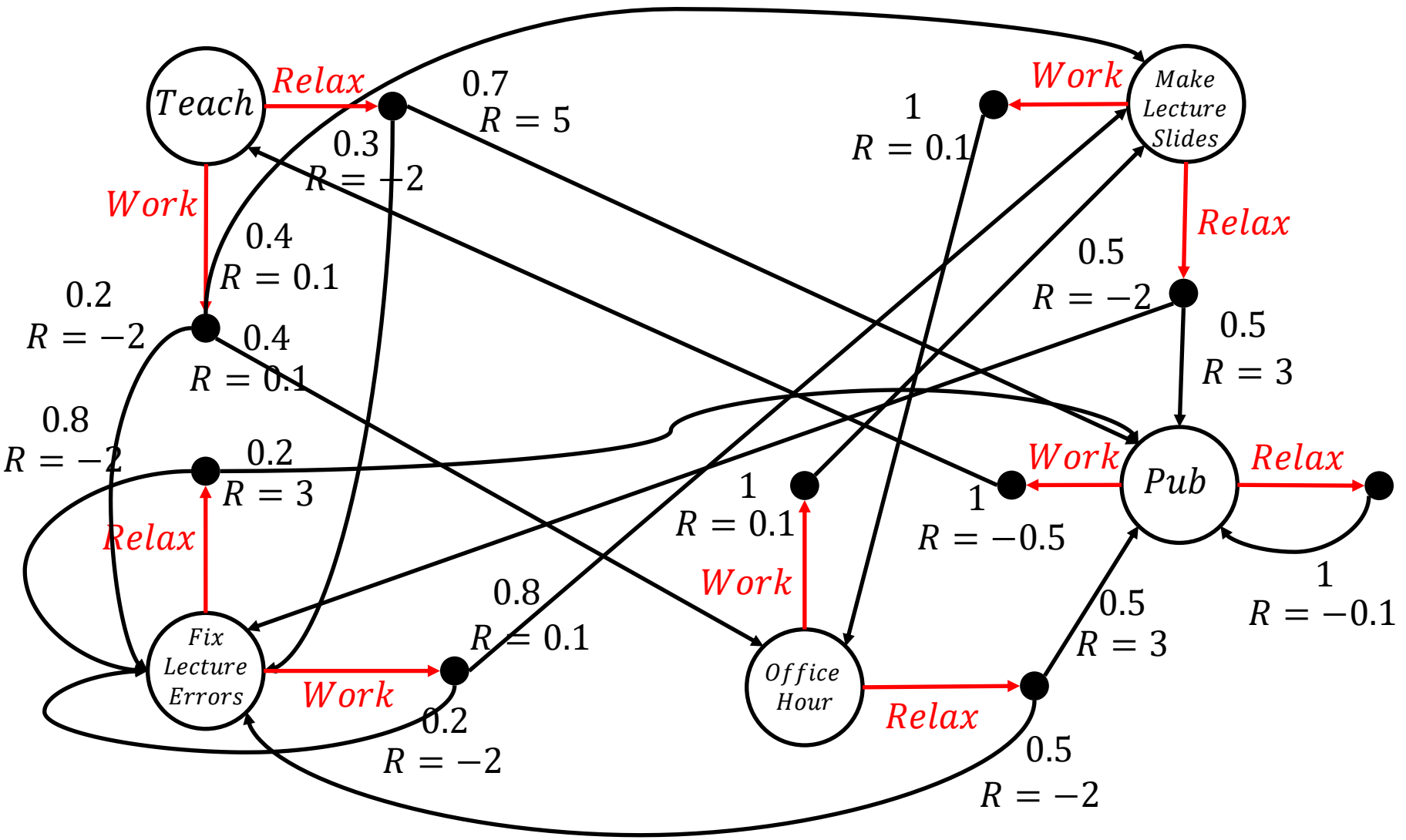
- Sutton, Richard S., and Barto, Andrew G. Reinforcement learning: An introduction. MIT press, 2018.
  - http://www.incompleteideas.net/book/the-book-2nd.html
  - Chapter 3
- Puterman, Martin L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.
  - Chapters 2, 3, 4
- David Silver lecture on Markov Decision Processes
  - https://www.youtube.com/watch?v=lfHX2hHRMVQ
  - Overall good, but with a bias for MDPs with a terminal state
- MDP formalization
  - We'll only talk about MDP in these slides

Overview

Rensselaer

- Markov reward processes (MRPs) are an extension of Markov chains
  - You get a reward after each state transition
  - You can calculate your expected reward over time
- Markov decision processes (MDPs) are an extension of MRPs
  - Add actions to influence the transition probabilities
  - Model the control problem
- Both models lead to classical recursive equalities known as the Bellman equations

MDP for Workday Example



Questions

- What is the expected reward in $Teach$ if I choose to $Relax$

$$5 * 0.7 - 2 * 0.3 = 2.9$$

- In the infinite-horizon, undiscounted case, what is one strategy that gives infinite reward w.p. 1?
  - Alternate between MLE and OH by applying $Work$
  - If $\gamma = 0.9$, what is the discounted reward in this case?

$$\sum_{k=0}^{\infty} \gamma^k 0.1 = \frac{0.1}{1 - 0.9} = 1$$

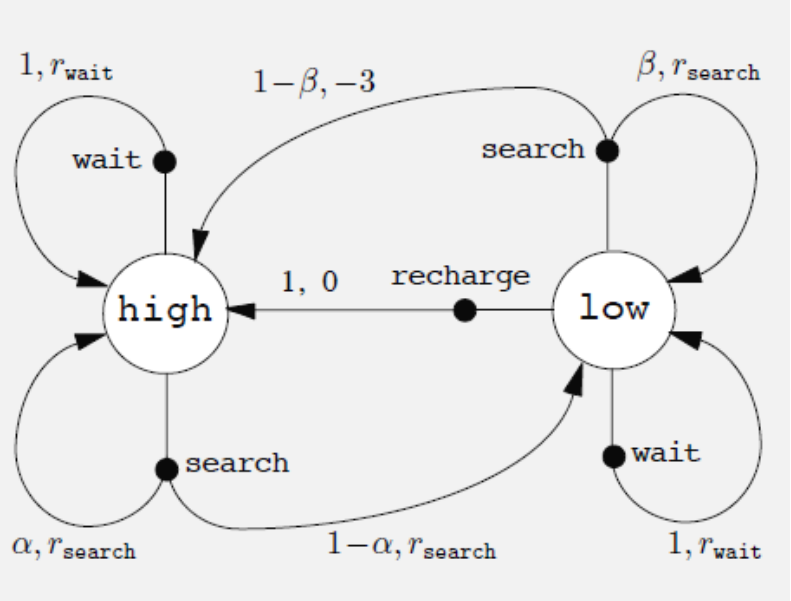
MDP Formalization

Rensselaer

- An MDP is a 5-tuple (S, A, P, R, η) where
  - S is the set of states (aka the state space)
  - A is the set of actions
  - $P: S \times A \times S \rightarrow \mathbb{R}$ is the probabilistic transition function
    - $\mathbb{P}[S_t | S_{t-1}, A_{t-1}] = P(S_{t-1}, A_{t-1}, S_t)$
  - $R: S \times A \times S \rightarrow \mathbb{R}$ is the reward function
    - One-step reward when applying action A_{t-1} from state S_{t-1} and landing in state S_t: $R(S_{t-1}, A_{t-1}, S_t)$
    - Can also derive expected reward from state s and action a: $R_e(s, a) = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$
    - By convention, the reward associated with some transition is actually received on the next step
    - The reward is typically determined by which state you land in
  - $\eta: S \rightarrow \mathbb{R}$ is the initial state distribution

Example: Recycling Robot

- Robot looking for soda cans to recycle
  - Two battery states: $high$ and low
  - Actions are $recharge$, $wait$ (for someone to bring a can) and $search$
  - While searching in $high$ mode, battery can become depleted with some probability $1 - \alpha$
  - While searching in low mode, robot may need to be rescued with probability $1 - \beta$



Policy

Rensselaer

- The policy (i.e., the controller) π is a function from observations/states to actions
  - Can be deterministic or stochastic
  - For stochastic policies, the notation $\pi(a|s)$ means
$$\mathbb{P}_\pi[A_t = a|S_t = s]$$
- When clear from context, we'll use $\pi(s)$ for deterministic policies
- In the fully observable case, there always exists an optimal deterministic policy
  - Any real-world controller is a deterministic system anyway
    - Modulo hardware failures, cosmic rays, etc.

MDP vs MRP

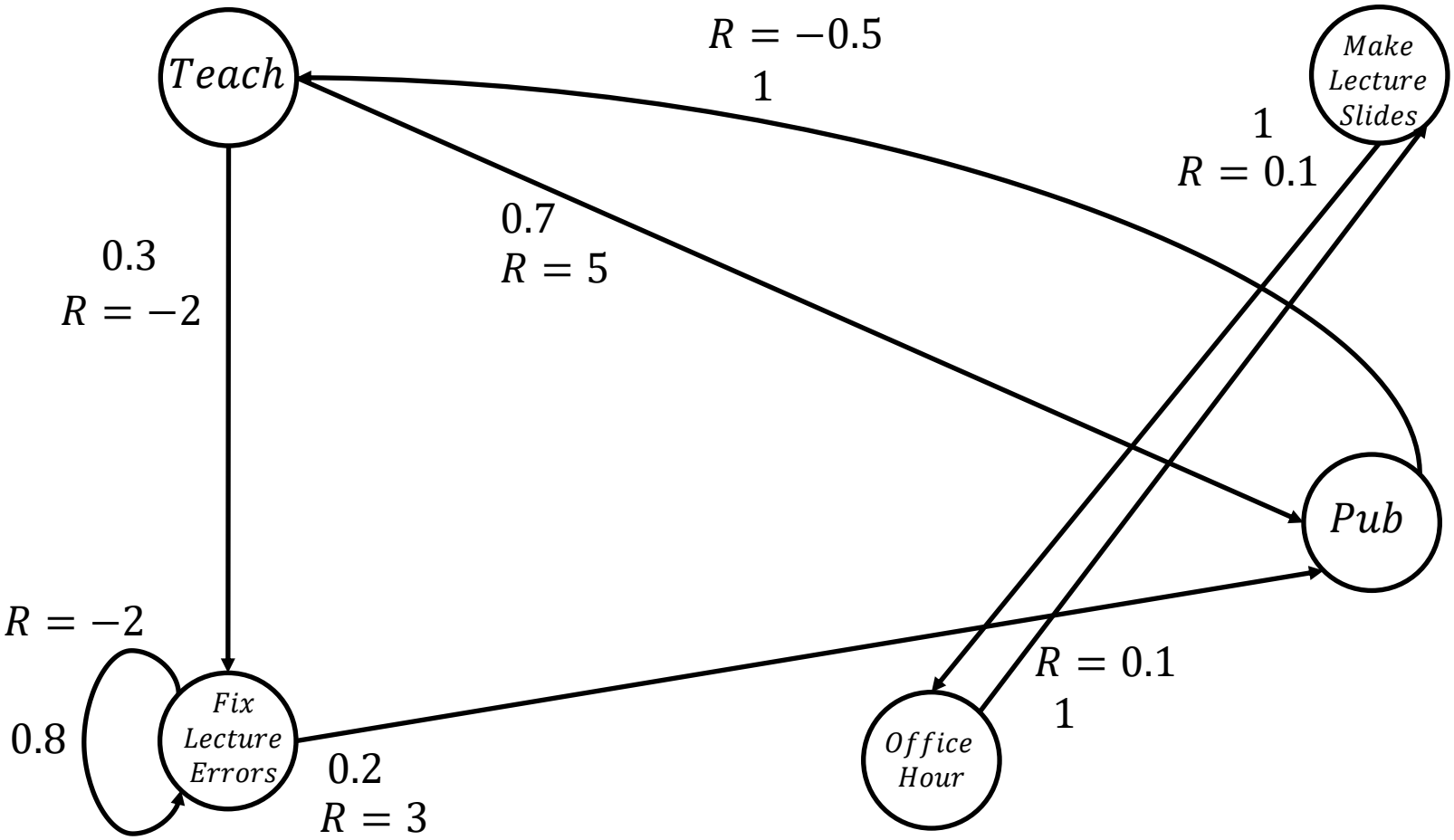
Rensselaer

- Given a policy π, the MDP is essentially an MRP
  - True for both stochastic and deterministic policies
- Everything we know about MRPs directly applies to MDPs
- Of course, the main challenge in MDPs is how to choose π
  - This is the RL problem

Converting an MDP to MRP, Workday Example

- Let's define π as follows:
  - $\pi(Teach) = Relax$
  - $\pi(OH) = Work$
  - $\pi(MLS) = Work$
  - $\pi(FLE) = Relax$
  - $\pi(Pub) = Work$

Converting an MDP to MRP, Workday Example



Workday Example MRP, Infinite Horizon

- New matrices are

$$\boldsymbol{P} = \begin{bmatrix} 0 & 0 & 0 & 0.3 & 0.7 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8 & 0.2 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}, R_e(\boldsymbol{s}) = \begin{bmatrix} 2.9 \\ 0.1 \\ 0.1 \\ -1 \\ -0.5 \end{bmatrix}$$

- For $\gamma = 0.9$, the state values are

$$(\boldsymbol{I} - \gamma \boldsymbol{P})^{-1} R_e(\boldsymbol{s}) = [5.54 \quad 1 \quad 1 \quad -0.69 \quad 4.49]^T$$

- For $\gamma = 0.5$, the state values are

$$(\boldsymbol{I} - \gamma \boldsymbol{P})^{-1} R_e(\boldsymbol{s}) = [3.03 \quad 0.2 \quad 0.2 \quad -1.50 \quad 1.02]^T$$

- State values are higher than in the previous MRP case. Why?
  - Control actions altered the MRP probabilities
  - The policy π has done something right

Converting an MDP to MRP, General Procedure

Rensselaer

- To calculate MRP probabilities in case of deterministic policy
  - For each pair of states s_1 and s_2:

$$P_{MRP}(s_1, s_2) = P_{MDP}(s_1, \pi(s_1), s_2)$$

- Similarly, for the reward:

$$R_{MRP}(s_1, s_2) = R_{MDP}(s_1, \pi(s_1), s_2)$$

- In case of stochastic policy π:

$$P_{MRP}(s_1, s_2) = \sum_a P_{MDP}(s_1, a, s_2)\pi(a|s_1)$$

$$R_{MRP}(s_1, s_2) = \sum_a R_{MDP}(s_1, a, s_2)\pi(a|s_1)$$

Policy Notation

- We use π subscripts to indicate probabilities with respect to π
  - E.g., $\mathbb{P}_\pi$ and $\mathbb{E}_\pi$
- For example:

$$\mathbb{P}_\pi[R_t = r | S_{t-1} = s] =$$

$$= \sum_a \mathbb{P}[R_t = r, A_{t-1} = a | S_{t-1} = s]$$

$$= \sum_a \mathbb{P}[R_t = r | S_{t-1} = s, A_{t-1} = a] \, \pi(a|s)$$

$$= \sum_a \sum_{s'} \mathbb{P}[R_t = r, S_t = s' | S_{t-1} = s, A_{t-1} = a] \, \pi(a|s)$$

$$= \sum_a \sum_{s'} \mathbb{P}[R_t = r | S_t = s', S_{t-1} = s, A_{t-1} = a] P(s, a, s') \, \pi(a|s)$$

  - where $\mathbb{P}[R_t = r | S_t = s', S_{t-1} = s, A_{t-1} = a] = 1$ if $R(s, a, s') = r$ (and 0, otherwise)

State Value Function

- In the finite-horizon case, the value function is

$$v_\pi^t(s) \coloneqq \mathbb{E}_\pi\left[R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-t+1} R_T \middle| S_t = s\right]$$
$$= \mathbb{E}_\pi[G_t | S_t = s]$$

  - where v_π is the state-value function for policy π
  - where $\mathbb{E}_\pi$ is the expected value when policy π is used, i.e.,

$$\mathbb{E}_\pi[R_{t+1}|S_t = s] = \sum_r r\mathbb{P}_\pi[R_{t+1} = r|S_t = s]$$
$$= \sum_r r \sum_a \mathbb{P}_\pi[R_{t+1} = r, A_t = a|S_t = s]$$
$$= \sum_r r \sum_a \mathbb{P}[R_{t+1} = r|S_t = s, A_t = a]\,\pi(a|s)$$
$$= \sum_a \pi(a|s) \sum_r r\mathbb{P}[R_{t+1} = r|S_t = s, A_t = a]$$
$$= \sum_a \pi(a|s) R_e(s, a)$$

State Value Function, cont'd

- In the infinite-horizon case, it is

$$v_\pi(s) \coloneqq \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \cdots | S_t = s]$$
$$= \mathbb{E}_\pi[G_t | S_t = s]$$

- This is exactly the same as the MRP case, except we now also have a policy π
  - Already saw Workday example values

Action Value Function

- Similar to the state value, but for a specific action
- In the finite-horizon case, the value function is

$$q_\pi^t(s,a) \coloneqq \mathbb{E}_\pi\left[R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-t+1} R_T \middle| S_t = s, A_t = a\right]$$
$$= \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$$

- In the infinite-horizon case:

$$q_\pi(s,a) \coloneqq \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \cdots | S_t = s, A_t = a]$$
$$= \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$$

- Intuitively, how good is a given action for a specific state
  - In other words, how good is a given action-state pair

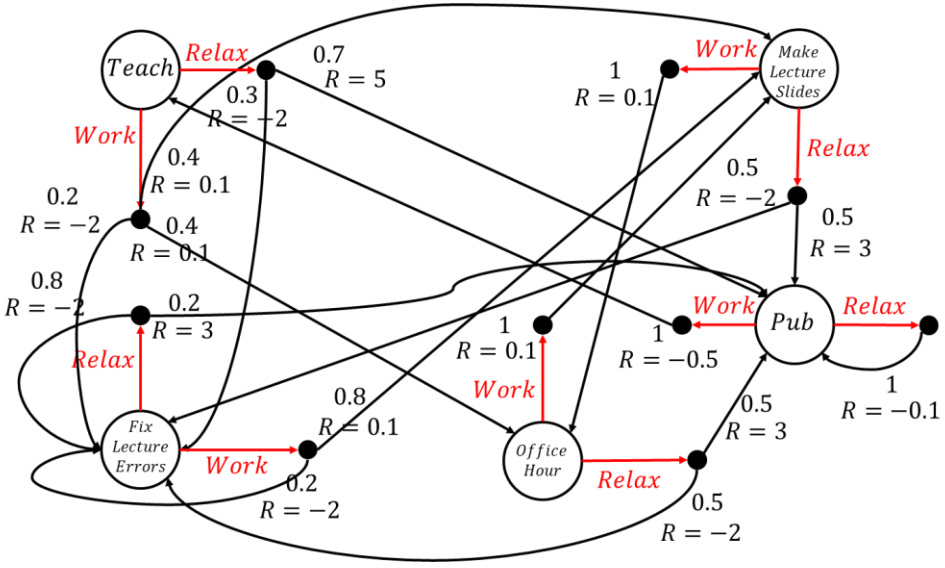
Workday Example, Action Values

- What is the 1-step value of applying $Work$ from $Teach$

$$-2 * 0.2 + 0.1 * 0.4 + 0.1 * 0.4 = -0.32$$

- What is the 1-step value of applying $Work$ from OH

$$0.1$$



Bellman Equation

- For a given policy π, the Bellman equation for MDPs is similar to the MRP one
  - Not surprising given that for any fixed policy, an MDP is an MRP
  - In the finite-horizon case, policies may also be **time-dependent**
    - Technically, need to write π^t, but will ignore t to avoid clutter
- The recursion is similar to the MRP case

$$\begin{aligned} v_\pi^t(s) &= \mathbb{E}_\pi\left[R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-t+1} R_T \middle| S_t = s\right] \\ &= \mathbb{E}_\pi\left[R_{t+1} + \gamma(R_{t+2} + \cdots + \gamma^{T-t} R_T) \middle| S_t = s\right] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s] \end{aligned}$$

- Again, similar to the MRP case

$$\mathbb{E}_\pi[G_{t+1} | S_t = s] = \mathbb{E}\left[v_\pi^{t+1}(S_{t+1}) \middle| S_t\right]$$

Bellman Equation, cont'd

- Again, similar to the MRP case

$$\mathbb{E}_\pi[G_{t+1}|S_t = s] =$$

$$= \sum_g g\mathbb{P}_\pi[G_{t+1} = g\,|S_t = s]$$

$$= \sum_g g \sum_{s'} \mathbb{P}_\pi[G_{t+1} = g, S_{t+1} = s'|S_t = s]$$

$$= \sum_g g \sum_{s'} \mathbb{P}_\pi[G_{t+1} = g|S_{t+1} = s', S_t = s]\mathbb{P}_\pi[S_{t+1} = s'|S_t = s]$$

$$= \sum_{s'} \mathbb{P}_\pi[S_{t+1} = s'|S_t = s] \sum_g g\mathbb{P}_\pi[G_{t+1} = g|S_{t+1} = s']$$

$$= \sum_{s'} \mathbb{P}_\pi[S_{t+1} = s'|S_t = s]v^{t+t}(s')$$

$$= \mathbb{E}_\pi\left[v_\pi^{t+1}(S_{t+1})\middle|S_t = s\right]$$

Bellman Equation, cont'd

- So finally,

$$v_\pi^t(s) = \mathbb{E}_\pi\left[R_{t+1} + \gamma v_\pi^{t+1}(S_{t+1}) \middle| S_t = s\right]$$

- Exactly the same as in the MRP case
  - Book's expression is the same but notation a bit different
- What about the Bellman equation for the action value?
  - Turns out we can derive two similar recursive definitions

$$q_\pi^t(s,a) = \mathbb{E}_\pi\left[R_{t+1} + \gamma v_\pi^{t+1}(S_{t+1}) \middle| S_t = s, A_t = a\right]$$
$$q_\pi^t(s,a) = \mathbb{E}_\pi\left[R_{t+1} + \gamma q_\pi^{t+1}(S_{t+1}, A_{t+1}) \middle| S_t = s, A_t = a\right]$$

- A similar recursive definition exists for the action value function

$$q_\pi^t(s,a) = \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s, A_t = a]$$

Bellman Equation for Action Values

- To derive the first version, start from the definition

$$q_\pi^t(s,a) = \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s, A_t = a]$$

- As in the MRP case, note that

$$\begin{aligned}
\mathbb{E}_\pi[G_{t+1}|S_t = s, A_t = a] = \\
&= \sum_g g \sum_{s'} \mathbb{P}_\pi[G_{t+1} = g, S_{t+1} = s' | S_t = s, A_t = a] \\
&= \sum_g g \sum_{s'} \mathbb{P}_\pi[G_{t+1} = g | S_{t+1} = s', S_t = s, A_t = a] * \\
&\qquad * \mathbb{P}_\pi[S_{t+1} = s' | S_t = s, A_t = a] \\
&= \sum_g g \sum_{s'} \mathbb{P}_\pi[G_{t+1} = g | S_{t+1} = s'] \, \mathbb{P}_\pi[S_{t+1} = s' | S_t = s, A_t = a] \\
&= \sum_{s'} \mathbb{P}_\pi[S_{t+1} = s' | S_t = s, A_t = a] v^{t+1}(s') \\
&= \mathbb{E}_\pi\left[v^{t+1}(s') \middle| S_t = s, A_t = a\right]
\end{aligned}$$

Bellman Equation for Action Values, cont'd

- To derive the second version, we need an extra marginalization

$$\mathbb{E}_\pi[G_{t+1}|S_t = s, A_t = a] =$$

$$= \sum_g g \sum_{s',a'} \mathbb{P}_\pi[G_{t+1} = g, S_{t+1} = s', A_{t+1} = a'|S_t = s, A_t = a]$$

$$= \sum_g g \sum_{s',a'} \mathbb{P}_\pi[G_{t+1} = g|S_{t+1} = s', A_{t+1} = a', S_t = s, A_t = a] *$$

$$* \mathbb{P}_\pi[S_{t+1} = s', A_{t+1} = a'|S_t = s, A_t = a]$$

$$= \sum_{s',a'} \mathbb{P}_\pi[S_{t+1} = s', A_{t+1} = a'|S_t = s, A_t = a] *$$

$$* \sum_g g\mathbb{P}_\pi[G_{t+1} = g|S_{t+1} = s', A_{t+1} = a']$$

$$= \sum_{s'} \sum_{a'} \mathbb{P}_\pi[S_{t+1} = s', A_{t+1} = a'|S_t = s, A_t = a]\, q^{t+1}(s', a')$$

$$= \mathbb{E}_\pi\left[q^{t+1}(S_{t+1}, A_{t+1})\middle|S_t = a, A_t = a\right]$$

Bellman Equation, cont'd

- So finally,

$$v_\pi^t(s) = \mathbb{E}_\pi\left[R_{t+1} + \gamma v_\pi^{t+1}(S_{t+1}) \middle| S_t = s\right]$$
$$q_\pi^t(s,a) = \mathbb{E}_\pi\left[R_{t+1} + \gamma v_\pi^{t+1}(S_{t+1}) \middle| S_t = s, A_t = a\right]$$
$$q_\pi^t(s,a) = \mathbb{E}_\pi\left[R_{t+1} + \gamma q_\pi^{t+1}(S_{t+1}, A_{t+1}) \middle| S_t = s, A_t = a\right]$$

- Note that one can express v_π^t in terms of q_π^t (how?)
  - Using marginalization:

$$v_\pi^t(s) = \sum_{s',a} P(s,a,s')\pi(a|s)\left[R(s,a,s') + \gamma v_\pi^{t+1}(s')\right]$$
$$= \sum_a \pi(a|s) q_\pi^t(s,a)$$

  - For deterministic policies

$$v_\pi^t(s) = q_\pi^t\big(s, \pi(s)\big)$$

Bellman Equation Matrix Form

- The Bellman equation in the infinite-horizon case is

$$v_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s]$$

- If we expand the expectation, we get:

$$v_\pi(s) = R_\pi(s) + \gamma \sum_{a,s'} \mathbb{P}_\pi[S_{t+1} = s', A_t = a | S_t = s] v_\pi(s')$$

$$= R_\pi(s) + \gamma \sum_{a,s'} P(s, a, s') \pi(a|s) v_\pi(s')$$

  - where $R_\pi(s) = \sum_a \pi(a|s)\, R_e(s, a)$

- We can once again write the Bellman equation in matrix form

$$v_\pi(\boldsymbol{s}) = R_\pi(\boldsymbol{s}) + \gamma \boldsymbol{P} v_\pi(\boldsymbol{s})$$

  - where $P_{ij} = \sum_a P(s_i, a, s_j) \pi(a|s_i)$

Bellman Equation Matrix Form, cont'd

Rensselaer

- The action-value Bellman equation is

$$q_\pi(s,a) = \mathbb{E}_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) | S_t = s, A_t = a]$$

- If we expand the expectation, we get:

$$q_\pi(s,a) = R_e(s,a) + \gamma \sum_{a',s'} \mathbb{P}[S_{t+1} = s', A_{t+1} = a' | S_t = s, A_t = a] q_\pi(s',a')$$

$$= R_e(s,a) + \gamma \sum_{a',s'} P(s,a,s')\pi(a'|s') q_\pi(s',a')$$

- We can once again write the Bellman equation in matrix form

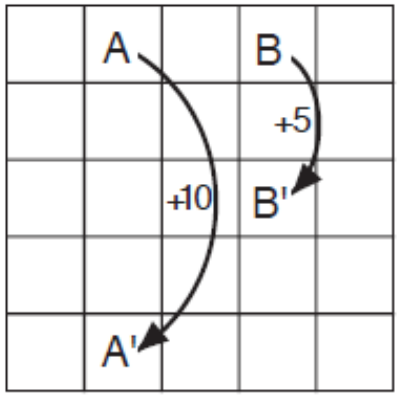
$$q_\pi(\boldsymbol{s},\boldsymbol{a}) = R_e(\boldsymbol{s},\boldsymbol{a}) + \gamma \boldsymbol{P} q_\pi(\boldsymbol{s},\boldsymbol{a})$$

  - What is the dimension of the q vector?
    - Number of states × number of actions
  - This is non-standard, so rarely used

Optimal Policy

- A policy π is better than another policy π' if
$$v_\pi^t(s) \geq v_{\pi'}^t(s), \forall s \in S, \forall t \in [1, T]$$
- A policy π^* is optimal if there exists no better policy than π^*
- The state-value function corresponding to π^* is denoted by v_*:
$$v_*(s) = \max_\pi v_\pi(s)$$
  - Similarly for q_*
- For (finite) MDPs, v_* has a unique solution
  - We will discuss several approaches to find v_*

Example



- Robot has 4 actions: $up, down, left, right$
- Each action results in a deterministic cell change
  - No change if agent tries to leave environment
  - All actions from A lead to A' and all actions from B lead to B'
- Get a reward of 10 from A, 5 from B, -1 if you hit a wall, and 0 otherwise
  - Reward discount is 0.9
- What is the optimal policy from each cell?
  - What are corresponding value functions?

Example, cont'd

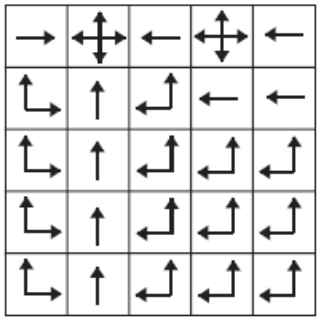
- Loop between A and A' takes 5 moves for a reward of 10
- Loop between B and B' takes 3 moves for a reward of 5
  - Looping between A and A' is more efficient
  - The value of A is then

$$v(A) = 10 + 0.9^5 * 10 + 0.9^{10} * 10 + \cdots$$
$$= \sum_{k=0}^{\infty} 10\left(0.9^5\right)^k = \frac{10}{1 - 0.9^5} = 24.4$$

- How do we compute the values for all states?
  $(\boldsymbol{I} - \gamma \boldsymbol{P})^{-1} R(\boldsymbol{s})$
  - What is the dimension of $\boldsymbol{P}$?
    - 25×25
  - We'll look at other methods next time

22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

v_*



π_*

Example, cont'd

- Loop between A and A' takes 5 moves for a reward of 10
- Loop between B and B' takes 3 moves for a reward of 5
  - Looping between A and A' is more efficient
  - The value of B is then

$$v(B) = 5 + 0.9^3 * 5 + 0.9^6 * 5 + \cdots$$

$$= \sum_{k=0}^{\infty} 5\left(0.9^3\right)^k = \frac{5}{1 - 0.9^3} = 18.45$$