Bellman Optimality Equations

Reading

- Sutton, Richard S., and Barto, Andrew G. Reinforcement learning: An introduction. MIT press, 2018.
  - http://www.incompleteideas.net/book/the-book-2nd.html
  - Chapter 4
- Puterman, Martin L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.
  - Chapter 4
- David Silver lecture on Dynamic Programming
  - https://www.youtube.com/watch?v=Nd1-UUMVfz4

Overview

- The RL problem boils down to finding the policy that maximizes the values of all states
  - i.e., the optimal policy
- The Bellman optimality equations provide a convenient property that the optimal policy must satisfy
- We will first prove the optimality equations
- We will also derive a tool for finding the optimal policy
  - The Policy Improvement Theorem

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
- Similarly, in the infinite-horizon case, a policy π is better than policy π' if

$$v_\pi(s) \geq v_{\pi'}(s), \forall s \in S$$

Policy Evaluation

- In order to find the optimal policy, we first need a way to evaluate policies
  - i.e., compute the state-value function $v_\pi(s)$ for each state s
  - Also compute the action-value function $q_\pi(s, a)$
- Every time we change the policy, we need to evaluate it (in order to check if we improved it)
- So far, we've seen one way to compute state values
  - How?
  - For a given policy π, compute the matrix form of the value function vector

Policy Evaluation: the infinite-horizon case

- In the infinite-horizon case, we can use the Bellman equation:

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

  – which can be rewritten in matrix form:

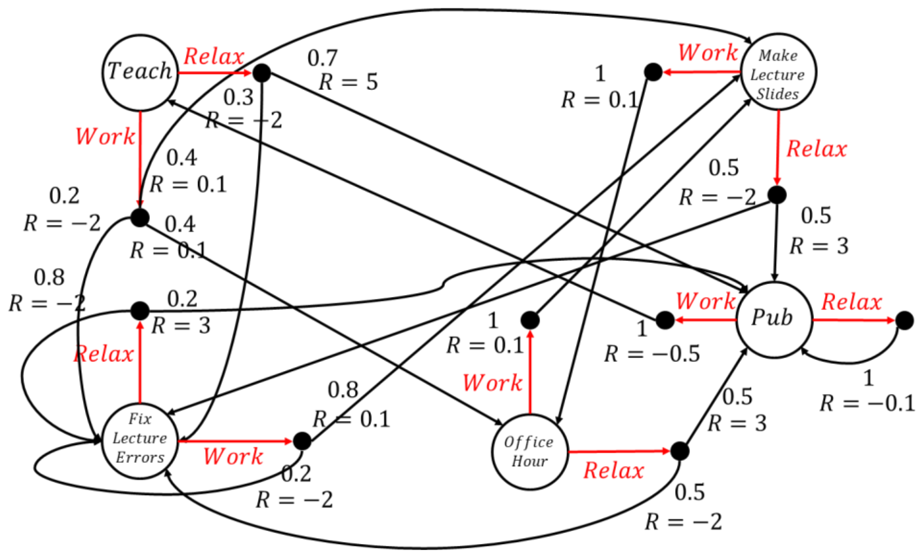
$$v_\pi(\boldsymbol{s}) = R_\pi(\boldsymbol{s}) + \gamma \boldsymbol{P}_\pi v_\pi(\boldsymbol{s})$$

- Thus, the state value vector is:

$$v_\pi(\boldsymbol{s}) = (\boldsymbol{I} - \gamma \boldsymbol{P}_\pi)^{-1} R_\pi(\boldsymbol{s})$$

Workday example, MDP -> MRP

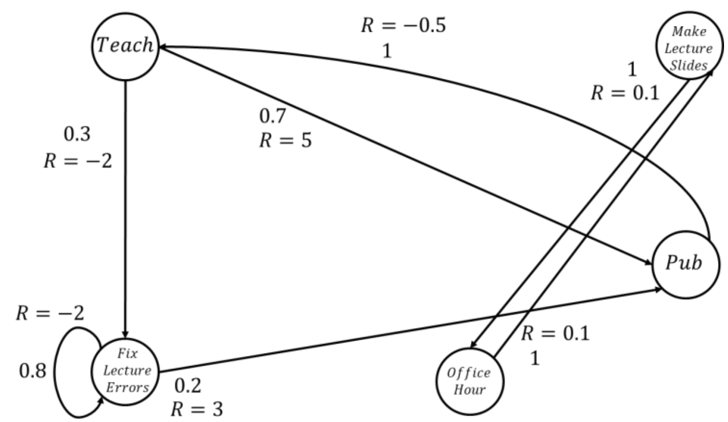
MDP



Policy

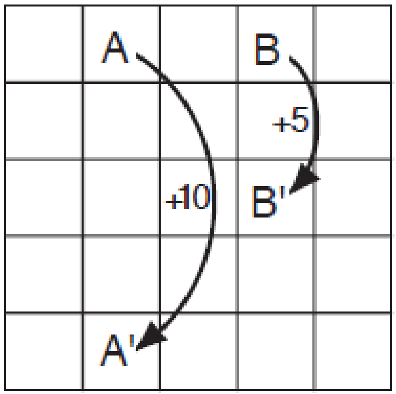
- Let's define π as follows:
  - $\pi(Teach) = Relax$
  - $\pi(OH) = Work$
  - $\pi(MLS) = Work$
  - $\pi(FLE) = Relax$
  - $\pi(Pub) = Work$

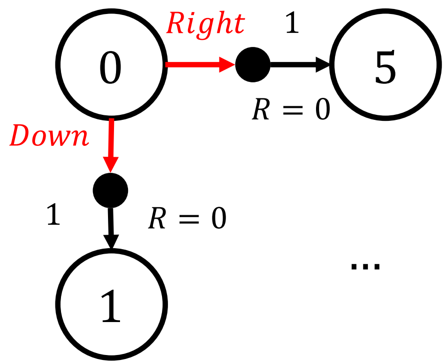
MRP



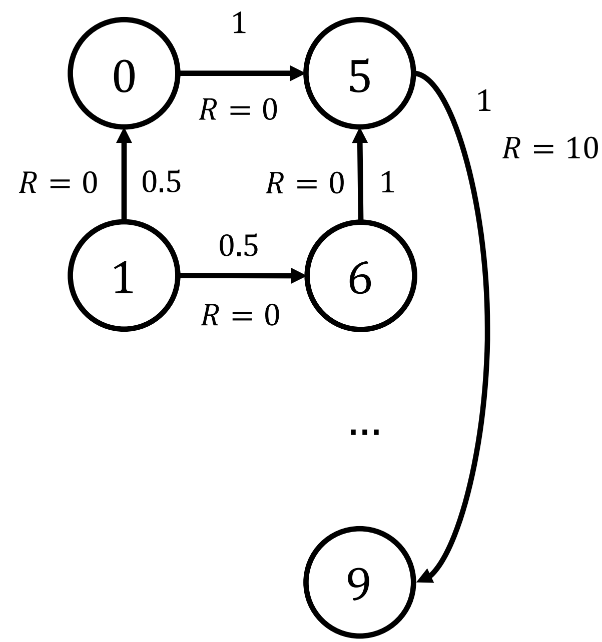
Gridworld example, MDP -> MRP

MDP

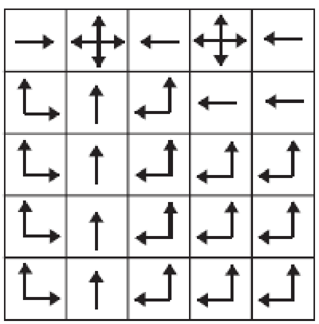




MRP



Policy



Iterative Policy Evaluation

- Inverting $\boldsymbol{I} - \gamma \boldsymbol{P}$ may be expensive if number of states is large
- Another approach is to use linear systems theory!
- Start from a random initialization for $v(\boldsymbol{s}) = v_0(\boldsymbol{s})$
- Look at linear system

$$v_k(\boldsymbol{s}) = R(\boldsymbol{s}) + \gamma \boldsymbol{P} v_{k-1}(\boldsymbol{s})$$

- System is stable. Why?
  - All entries (and eigenvalues) of $\gamma \boldsymbol{P}$ are < 1 (for $\gamma < 1$)
- System converges to unique solution $(\boldsymbol{I} - \gamma \boldsymbol{P})^{-1} R(\boldsymbol{s})$
  - Why?
  - If $v_{k+1}(\boldsymbol{s}) = v_k(\boldsymbol{s}) = \boldsymbol{v}$, then $\boldsymbol{v} = R(\boldsymbol{s}) + \gamma \boldsymbol{P} \boldsymbol{v}$, i.e.,

$$\boldsymbol{v} = (\boldsymbol{I} - \gamma \boldsymbol{P})^{-1} R(\boldsymbol{s})$$

    - Keep in mind $\boldsymbol{I} - \gamma \boldsymbol{P}$ is not invertible when $\gamma = 1$

Workday example, Iterative Value evolution

- Recall state values are (for $\gamma = 0.9$)

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

- Using iterative evaluation
  - Starting with $\boldsymbol{v}_0 = [0 \quad 0 \quad 0 \quad 0 \quad 0]^T$

$$\boldsymbol{v}_{10} = [4.59 \quad 0.65 \quad 0.65 \quad -1.58 \quad 3.64]^T$$
$$\boldsymbol{v}_{30} = [5.43 \quad 0.96 \quad 0.96 \quad 0 - 0.80 \quad 4.38]^T$$
$$\boldsymbol{v}_{50} = [5.53 \quad 0.99 \quad 0.99 \quad 0 - 0.70 \quad 4.47]^T$$

- After 50 iterations, converged within 0.01 if the true values

Gridworld Example, iterative value evolution

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

- Recall state values are
- Using iterative evaluation
  - Starting from $\boldsymbol{v}_0 = \mathbf{0} \in \mathbb{R}^{25}$
  - v_{10} is

14.31	15.90	14.31	10.90	9.81
12.88	14.31	12.88	11.59	10.44
11.59	12.88	11.59	10.44	5.90
10.44	11.59	10.44	5.90	5.31
5.90	10.44	5.90	5.31	4.78

  - v_{50} is

21.86	24.29	21.86	19.29	17.36
19.68	21.86	19.68	17.71	15.94
17.71	19.68	17.71	15.94	14.29
15.94	17.71	15.94	14.29	12.86
14.29	15.94	14.29	12.86	11.58

- After 50 iterations, converge within 0.15 of true values
- In general, can stop iterating when $||v_{k+1} - v_k|| \leq \epsilon$
  - Where ϵ is a hyperparameter

What about the finite-horizon case?

- Evaluate a policy recursively, starting from the last step T
  - Use the Bellman equation

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

- Remember, the policy and the value function may be time-dependent in the finite-horizon case!
- Will discuss this in more detail when we talk about Dynamic Programming

Bellman Optimality Equation

- A policy π is better than another policy π' if
$$v_\pi(s) \geq v_{\pi'}(s), \forall s \in S$$
- A policy π^* is optimal if there exists no better policy than π^*
- Turns out the optimal policy also has a nice recursive property
$$\pi_*(s) = arg \max_a q_{\pi_*}(s, a)$$
  - This is the Bellman Optimality Equation
  - Pick the action with the highest value
  - Obvious in a sense
    - But any policy that satisfies the Bellman optimality equation is optimal

Bellman Optimality Equation, proof

- Bellman optimality equation:

$$\pi_*(s) = arg\max_a q_{\pi_*}(s, a)$$
$$= arg\max_a \mathbb{E}_{\pi_*}[R_{t+1} + \gamma v_{\pi_*}(S_{t+1}) | S_t = s, A_t = a]$$

- Proof by induction backward in time (for finite horizon T):
  - Base case ($t = T - 1$):
    - Only one step to make
    - For any state s, the optimal action is

$$arg\max_a \mathbb{E}_{\pi_*}[R_T | S_{T-1} = s, A_{T-1} = a] =$$
$$= arg\max_a q_{\pi_*}(s, a)$$

    - So π_* is optimal by construction

Bellman Optimality Equation, proof

- Proof by induction backward in time (for finite horizon T):
  - Inductive case:
  - Assume π_* is optimal at time $t+1$
    - i.e., $v_{\pi_*}^{t+1}(s) \geq v_{\pi'}^{t+1}(s), \forall s, \pi'$
  - What do we need to show?

$$v_{\pi_*}^{t}(s) \geq v_{\pi'}^{t}(s), \forall s, \pi'$$

  - Using the Bellman equation for π_*:

$$\begin{aligned} v_{\pi_*}^{t}(s) &= q_{\pi_*}^{t}\big(s, \pi_*(s)\big) \\ &= \max_a \big[q_{\pi_*}^{t}(s,a)\big] \\ &= \max_a \Big[\mathbb{E}_{\pi_*}\big[R_{t+1} + \gamma v_{\pi_*}^{t+1}(S_{t+1}) \big| S_t = s, A_t = a\big]\Big] \end{aligned}$$

Bellman Optimality Equation, proof

- Proof by induction backward in time (for finite horizon T):
  - Assume π_* is optimal at time $t+1$
    - i.e., $v_{\pi_*}^{t+1}(s) \geq v_{\pi'}^{t+1}(s), \forall s, \pi'$
  - Consider any other policy π'

$$
\begin{aligned}
v_{\pi_*}^{t}(s) &= \max_a \left[\mathbb{E}_{\pi_*}\left[R_{t+1} + \gamma v_{\pi_*}^{t+1}(S_{t+1}) \middle| S_t = s, A_t = a\right]\right] \\
&= \max_a \left[R_e(s,a) + \sum_{s'} \gamma v_{\pi_*}^{t+1}(s') P(s,a,s')\right] \\
&\geq \max_a \left[R_e(s,a) + \sum_{s'} \gamma v_{\pi'}^{t+1}(s') P(s,a,s')\right]
\end{aligned}
$$

  - Inequality true for any a, so true for max also

Bellman Optimality Equations, proof

- Proof by induction backward in time(for finite horizon T):

$$
\begin{aligned}
v_{\pi_*}^{t}(s) &= \max_a \left[\mathbb{E}_{\pi_*}\left[R_{t+1} + \gamma v_{\pi_*}^{t+1}(S_{t+1}) \middle| S_t = s, A_t = a\right]\right] \\
&= \max_a \left[R(s,a) + \sum_{s'} \gamma v_{\pi_*}^{t+1}(s') P(s,a,s') \right] \\
&\geq \max_a \left[R(s,a) + \sum_{s'} \gamma v_{\pi'}^{t+1}(s') P(s,a,s') \right] \\
&\geq R\big(s,\pi'(s)\big) + \sum_{s'} \gamma v_{\pi'}^{t+1}(s') P(s,\pi'(s),s') \\
&= \mathbb{E}_{\pi'}\left[R_{t+1} + \gamma v_{\pi'}^{t+1}(S_{t+1}) \middle| S_t = s, A_t = \pi'(s)\right] \\
&= \mathbb{E}_{\pi'}\left[R_{t+1} + \gamma v_{\pi'}^{t+1}(S_{t+1}) \middle| S_t = s\right] \\
&= v_{\pi'}^{t}(s)
\end{aligned}
$$

Deterministic policy version

Bellman Optimality Equations, proof

- Proof by induction backward in time(for finite horizon T):

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

$$= \max_a \left[R(s,a) + \sum_{s'} \gamma v_{\pi_*}^{t+1}(s') P(s,a,s')\right]$$

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

Stochastic policy version

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

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

$$= v_{\pi'}^t(s)$$

where $a^* = \operatorname*{argmax}_a \left[R(s,a) + \sum_{s'} \gamma v_{\pi'}^{t+1}(s') P(s,a,s')\right]$

Policy Improvement Theorem

- The Bellman optimality equation tells us what properties the optimal policy must satisfy
  - But it doesn't tell us how to find that policy
- Suppose we have a current policy π, potentially not optimal
  - Suppose we know $v_\pi(s), q_\pi(s,a), \forall s, a$
  - How can we improve the policy for a given s?
  - Pick an action that has a higher q value
- We know $v_\pi(s) = q_\pi(s, \pi(s))$
  - What if there existed an action a' s.t.
$$q_\pi(s, a') \geq q_\pi(s, \pi(s))$$
  - Turns out the policy that selects a' is better

Policy Improvement Theorem, cont'd

- **Policy Improvement Theorem**:
  - A policy π' is as good as, or better than, another policy π if for all $s \in S$

$$q_\pi\big(s, \pi'(s)\big) \geq v_\pi(s)$$

Policy Improvement Theorem Proof

- First recall that for a specific action a, the q value is:

$$q_\pi(s,a) = R_e(s,a) + \gamma \sum_{s'} P(s,a,s')v(s')$$
$$= R_e(s,a) + \gamma \boldsymbol{p}(s,a)^T v(\boldsymbol{s})$$

  – where $\boldsymbol{p}(s,a)^T = [P(s,a,s_1), \dots, P(s,a,s_N)]$

- Wlog, suppose π' is different from π only at s_1, i.e.,

$$q_\pi\big(s_1, \pi'(s_1)\big) \geq v_\pi(s_1)$$

- Using the Bellman equation:

$$q_\pi\big(s_1, \pi'(s_1)\big) = R_e\big(s_1, \pi'(s_1)\big) + \gamma \boldsymbol{p}\big(s_1, \pi'(s_1)\big)^T v_\pi(\boldsymbol{s})$$
$$v_\pi(s_1) = q_\pi\big(s_1, \pi(s_1)\big) = R_e\big(s_1, \pi(s_1)\big) + \gamma \boldsymbol{p}\big(s_1, \pi(s_1)\big)^T v_\pi(\boldsymbol{s})$$

- Then

$$R_e\big(s_1, \pi'(s_1)\big) + \gamma \boldsymbol{p}\big(s_1, \pi'(s_1)\big)^T v_\pi(\boldsymbol{s}) \geq R_e\big(s_1, \pi(s_1)\big) + \gamma \boldsymbol{p}\big(s_1, \pi(s_1)\big)^T v_\pi(\boldsymbol{s})$$

Policy Improvement Theorem Proof, cont'd

- Wlog, suppose π' is different from π only at s_1, i.e.,
$$q_\pi\big(s_1, \pi'(s_1)\big) \geq v_\pi(s_1)$$
- Using the Bellman equation:
$$q_\pi\big(s_1, \pi'(s_1)\big) = R_e\big(s_1, \pi'(s_1)\big) + \gamma \boldsymbol{p}\big(s_1, \pi'(s_1)\big)^T v_\pi(\boldsymbol{s})$$
$$v_\pi(s_1) = q_\pi\big(s_1, \pi(s_1)\big) = R_e\big(s_1, \pi(s_1)\big) + \gamma \boldsymbol{p}\big(s_1, \pi(s_1)\big)^T v_\pi(\boldsymbol{s})$$
- Then
$$R_e\big(s_1, \pi'(s_1)\big) + \gamma \boldsymbol{p}\big(s_1\ , \pi'(s_1)\big)^T v_\pi(\boldsymbol{s}) \geq R_e\big(s_1, \pi(s_1)\big) + \gamma \boldsymbol{p}\big(s_1, \pi(s_1)\big)^T v_\pi(\boldsymbol{s})$$
- Stack remaining values for π in a vector as follows:
$$\begin{bmatrix} R_e\big(s_1, \pi'(s_1)\big) + \gamma \boldsymbol{p}\big(s_1, \pi'(s_1)\big)^T v_\pi(\boldsymbol{s}) \\ R_e\big(s_2, \pi(s_2)\big) + \gamma \boldsymbol{p}\big(s_2, \pi(s_2)\big)^T v_\pi(\boldsymbol{s}) \\ \cdots \\ R_e\big(s_N, \pi(s_N)\big) + \gamma \boldsymbol{p}\big(s_N, \pi(s_N)\big)^T v_\pi(\boldsymbol{s}) \end{bmatrix} \geq \begin{bmatrix} R_e\big(s_1, \pi(s_1)\big) + \gamma \boldsymbol{p}\big(s_1, \pi(s_1)\big)^T v_\pi(\boldsymbol{s}) \\ R_e\big(s_2, \pi(s_2)\big) + \gamma \boldsymbol{p}\big(s_2, \pi(s_2)\big)^T v_\pi(\boldsymbol{s}) \\ \cdots \\ R_e\big(s_N, \pi(s_N)\big) + \gamma \boldsymbol{p}\big(s_N, \pi(s_N)\big)^T v_\pi(\boldsymbol{s}) \end{bmatrix}$$
  - where the inequality is interpreted element-wise

Policy Improvement Theorem Proof, cont'd

- Stack remaining values for π in a vector as follows:

$$\begin{bmatrix} R_e(s_1,\pi'(s_1)) + \gamma \boldsymbol{p}(s_1,\pi'(s_1))^T v_\pi(\boldsymbol{s}) \\ R_e(s_2,\pi(s_2)) + \gamma \boldsymbol{p}(s_2,\pi(s_2))^T v_\pi(\boldsymbol{s}) \\ \dots \\ R_e(s_N,\pi(s_N)) + \gamma \boldsymbol{p}(s_N,\pi(s_N))^T v_\pi(\boldsymbol{s}) \end{bmatrix} \geq \begin{bmatrix} R_e(s_1,\pi(s_1)) + \gamma \boldsymbol{p}(s_1,\pi(s_1))^T v_\pi(\boldsymbol{s}) \\ R_e(s_2,\pi(s_2)) + \gamma \boldsymbol{p}(s_2,\pi(s_2))^T v_\pi(\boldsymbol{s}) \\ \dots \\ R_e(s_N,\pi(s_N)) + \gamma \boldsymbol{p}(s_N,\pi(s_N))^T v_\pi(\boldsymbol{s}) \end{bmatrix}$$

- In matrix form:

$$R_{\pi'}(\boldsymbol{s}) + \gamma \boldsymbol{P}_{\pi'} v_\pi(\boldsymbol{s}) \geq R_\pi(\boldsymbol{s}) + \gamma \boldsymbol{P}_\pi v_\pi(\boldsymbol{s})$$
$$R_{\pi'}(\boldsymbol{s}) + \gamma \boldsymbol{P}_{\pi'} v_\pi(\boldsymbol{s}) \geq v_\pi(\boldsymbol{s})$$
$$R_{\pi'}(\boldsymbol{s}) \geq v_\pi(\boldsymbol{s}) - \gamma \boldsymbol{P}_{\pi'} v_\pi(\boldsymbol{s})$$
$$R_{\pi'}(\boldsymbol{s}) \geq (\boldsymbol{I} - \gamma \boldsymbol{P}_{\pi'}) v_\pi(\boldsymbol{s})$$

- Pre-multiply both sides by $(\boldsymbol{I} - \gamma \boldsymbol{P}_{\pi'})^{-1}$
  - Inequalities don't switch sides (don't have time to prove)

$$(\boldsymbol{I} - \gamma \boldsymbol{P}_{\pi'})^{-1} R_{\pi'}(\boldsymbol{s}) \geq v_\pi(\boldsymbol{s})$$
$$v_{\pi'}(\boldsymbol{s}) \geq v_\pi(\boldsymbol{s})$$

– QED

Deterministic Policies: Greedy Policy Improvement

- Suppose we are given a deterministic policy π
- We can greedily improve π for each state

$$\begin{aligned}\pi'(s) &= arg\max_a q_\pi(s,a)\\ &= arg\max_a \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s, A_t = a]\end{aligned}$$

- By the policy improvement theorem, π' is better than or equal to π
- If $\pi' = \pi$, then $\pi' = \pi^*$:

$$\begin{aligned}v_{\pi'}(s) &= \max_a \left[\mathbb{E}_\pi[R_{t+1} + \gamma v_{\pi'}(S_{t+1}) | S_t = s, A_t = a]\right]\\ &= \max_a q_{\pi'}(s,a)\end{aligned}$$

  - Bellman optimality equation!

Summary

- The Bellman optimality equations provide a property that any optimal policy must satisfy
  - But don't tell us how to find that policy
- The Policy Improvement Theorem provides a tool for greedy policy search
  - Improve the policy iteratively and re-evaluate
- It also provides the foundation for a class of algorithms based on dynamic programming
  - Next lecture