ORCS 4529: Reinforcement Learning

Shipra Agrawal

Columbia University
Industrial Engineering and Operations Research

Contents I

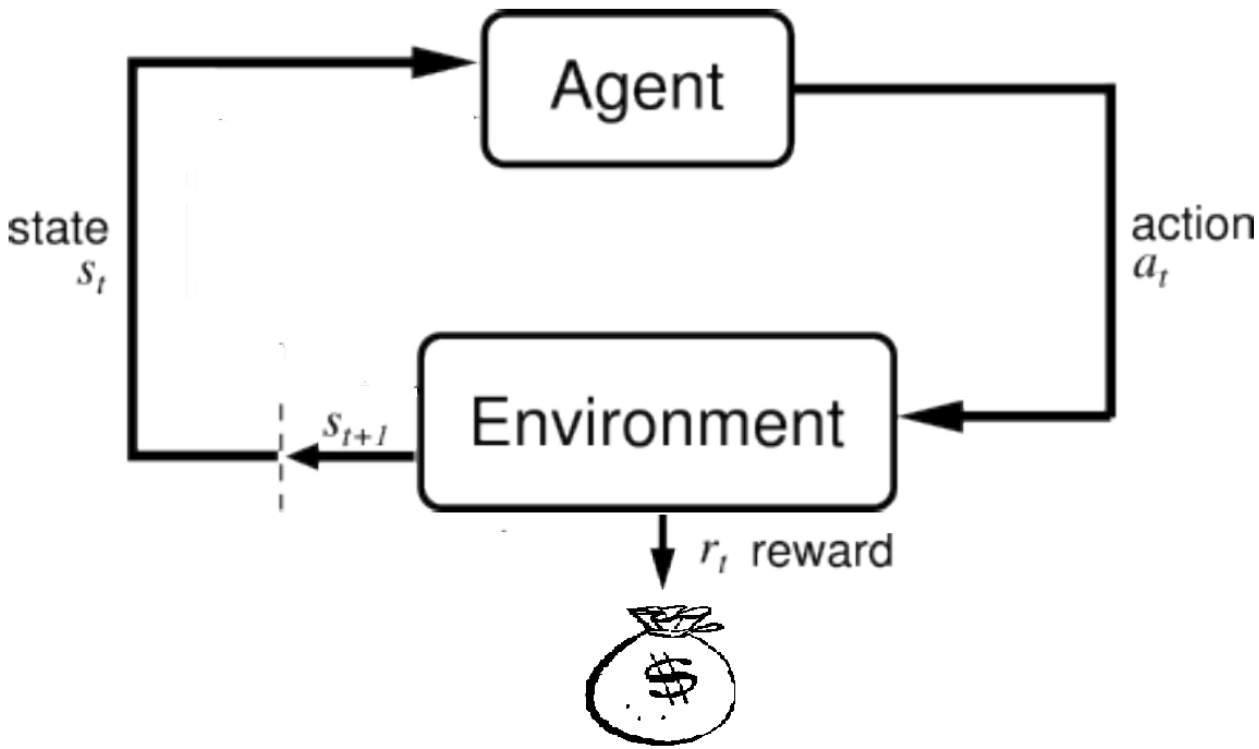
Markov Decision Process (MDP) definition

State space, Action space, Reward model, Transition model

$$(\mathcal{S}, \mathcal{A}, R, P)$$

Starting state s_1 or a distribution of starting state
Finite horizon H or inifinte horizon

- At $t = 1, 2, \ldots$, observe s_t, take action a_t, observe reward r_t and next state s_{t+1}.



Markov Property of MDP

- At $t = 1, 2, \ldots$, observe s_t, take action a_t, observe reward r_{t+1} and next state s_{t+1}.
- Markov Property:

1.

$$Pr[s_{t+1} = s' | a_t = a, s_t = s, \text{history till time } t] = E[s_{t+1} = s' | a_t = a, s_t = s] = P(s, a, s')$$

$$E[r_{t+1} | a_t = a, s_t = s, \text{history till time } t] = E[r_{t+1} | a_t = a, s_t = s] =: R(s, a)$$

Some times reward model is expressed as $R(s, a, s')$, so that $R(s, a) = \sum_{s'} R(s, a, s') P(s, a, s)$

Solution Concept: Policy

- Markovian Policy vs. History Dependent policy
- Deterministic Policy vs. Randomized Policy
- Stationary vs. Non-stationary policy

1. A policy specifies what action to take at any time step.
2. A Markovian policy is a mapping from state space to action action $\pi : S \to A$.
3. for any history dependent policy π, there exists a Markovian (randomized) policy π' such that $\Pr(s_t = s, a_t = a | s_1)$ is same for both the policies. This is formally proven in Theorem 5.5.1 of [**?**].
4. A deterministic policy $\pi : S \to A$ is mapping from any given state to an action. A randomized policy $\pi : S \to \Delta^A$ is a mapping from any given state to a distribution over actions.
5. In general (with some abuse of terminology), a non-stationary policy refers to a sequence of policies $(\pi_1, \pi_2, \ldots, \pi_t, \ldots)$. A stationary policy then refers to a static sequence $(\pi, \pi, \ldots, \pi)$, i.e., $\pi_t = \pi$ for all rounds $t = 1, 2, \ldots$.

Stationary Policy π

$$\pi : \mathcal{S} \to \mathcal{A} \text{ or } \pi : \mathcal{S} \to \Delta^{\mathcal{A}}$$

- Markov reward process (compare to Markov chains)
- Stationary distribution d^π of a stationary policy π

1. Draw the MDP with 5 and red and mutliple black arrows from each state for state transition (two actions red and black)
2. MRP: process obtained on fixing a stationary policy on each state. To show what happens on fixing a policy: remove all red arrows from one state, remove black from other etc. on one state just distribute the probability over all red and black
3. The transition probability vector and reward for this MRP in state s is given by $\Pr(s'|s) = P^\pi_s, \mathbb{E}[r_t|s] = r^\pi_s$, where P^π is an $S \times S$ matrix, and r^π is an S-dimensional vector defined as:

$$P^\pi_{s,s'} = \mathbb{E}_{a\sim\pi(s)}[P(s,a,s')], \forall s, s' \in S$$

$$r^\pi_s = \mathbb{E}_{a\sim\pi(s)}[R(s,a)]$$

4. The stationary distribution (if exists) of this Markov chain when starting from state s_1 is also referred to as the stationary distribution of the policy π, denoted by d^π:

$$d^\pi(s) = \lim \Pr(s_t = s|s_1, \pi)$$

Goal of an MDP: Finite Horizon

Find (possibly non-stationary) policy $\pi = (\pi_1, \ldots, \pi_H)$ that maximizes 'Value' starting from state s_1.

- Episodic or finite horizon setting.

$$V^{\pi}(s_1) = \mathbb{E}[\sum_{t=1}^{H} \gamma^{t-1} r_t | s_1; a_t = \pi_t(s_t)]$$

Optimal policy and value depend critically on s_1, H.
$0 \leq \gamma \leq 1$.

Goal of an MDP: Finite Horizon

Find (possibly non-stationary) policy $\pi = (\pi_1, \ldots, \pi_H)$ that maximizes 'Value' starting from state s_1.

- Episodic or finite horizon setting.

$$V^{\pi}(s_1) = \mathbb{E}[\sum_{t=1}^{H} \gamma^{t-1} r_t | s_1; a_t = \pi_t(s_t)]$$

Optimal policy and value depend critically on s_1, H. $0 \leq \gamma \leq 1$.

1. This is closest to the "Dynamic programming problem" you may have studied in your optimization or algorithm design and analysis class

Goal of an MDP: Infinite Horizon

Find (possibly non-stationary) policy $\pi = (\pi_1, \pi_2 \ldots, \pi_t, \ldots,)$ that maximizes 'Gain' or 'Value' starting from state s_1.

- Infinite horizon expected total reward (Value).

$$V^{\pi}(s_1) = \lim_{T \to \infty} \mathbb{E}[\sum_{t=1}^{T} \gamma^{t-1} r_t | s_1; a_t = \pi_t(s_t)]$$

- Infinite horizon discounted sum of rewards (Value).

$$V^{\pi}(s_1) = \lim_{T \to \infty} \mathbb{E}[\sum_{t=1}^{T} \gamma^{t-1} r_t | s_1; a_t = \pi_t(s_t)]$$

- Infinite horizon average reward (gain):

$$\rho^{\pi}(s_1) = \lim_{T \to \infty} \mathbb{E}[\frac{1}{T} \sum_{t=1}^{T} r_t | s_1; a_t = \pi_t(s_t)]$$

Optimal Policy

A policy that maximizes the gain or value starting from the starting state.

Is optimal policy Markovian? Stationary?

Assume finite or countable states and actions.

- If an optimal policy exists, there always exists a **Markovian policy** that is optimal.
- In all three **infinite horizon settings**, if an optimal policy exists, there always exists a **stationary policy** that is optimal.

Reference to proofs available in lecture notes.

Optimal Policy

A policy that maximizes the gain or value starting from the starting state.

Is optimal policy Markovian? Stationary?

Assume finite or countable states and actions.

- If an optimal policy exists, there always exists a **Markovian policy** that is optimal.
- In all three **infinite horizon settings**, if an optimal policy exists, there always exists a **stationary policy** that is optimal.

Reference to proofs available in lecture notes.

The results below assume finite and countable states and actions, and bounded rewards.

Theorem ([?], Theorem 6.2.7)

For any infinite horizon discounted MDP, there always exists a deterministic stationary policy π that is optimal.

Theorem ([?], Theorem 7.1.9)

For any infinite horizon expected total reward MDP, assuming for all policies either the upper or lower limit of the total reward objective exists, then there exists a deterministic stationary policy π that is optimal.

Theorem ([?], Theorem 8.1.2)

For infinite horizon average reward MDP, there always exist a stationary (possibly randomized) policy which is an optimal policy.

MDP formulation Example 1

Robot learning to move on a line

- Three actions: walk or run or stay.
- On walking: the robot to move one step without falling.
- On running: robot might move two steps forward (80% chance), or fall (20% chance). Once the robot falls, it cannot get up.
- Once a target position (say 5 steps away from the starting position) is reached, the robot stays there.
- Aim: move forward on the line, quickly and without falling, and reach the target position.

MDP formulation Example 1

Robot learning to move on a line

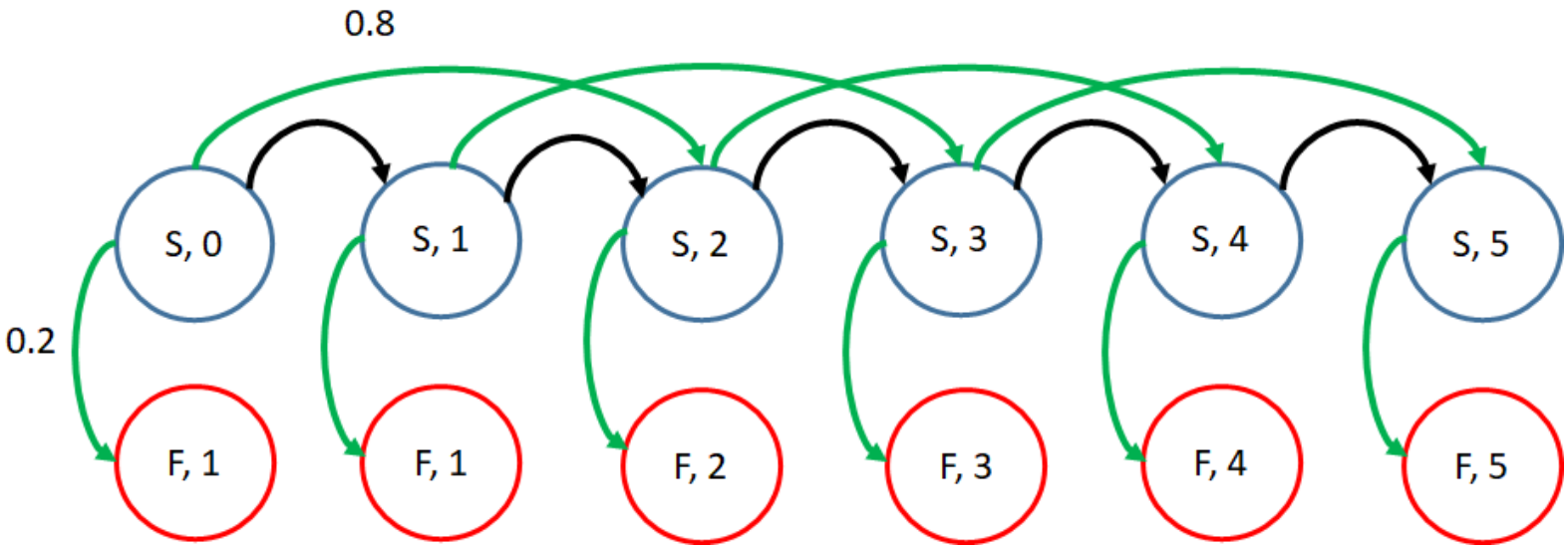
- Three actions: walk or run or stay.
- On walking: the robot to move one step without falling.
- On running: robot might move two steps forward (80% chance), or fall (20% chance). Once the robot falls, it cannot get up.
- Once a target position (say 5 steps away from the starting position) is reached, the robot stays there.
- Aim: move forward on the line, quickly and without falling, and reach the target position.

What is the state space, action space, reward model, transition model?

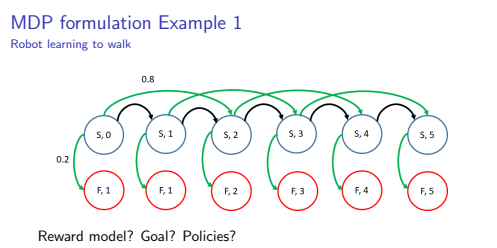
Easier to represent transition model using the diagram

MDP formulation Example 1

Robot learning to walk

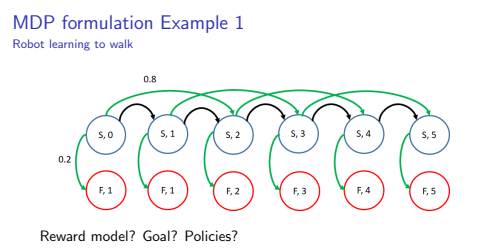


Reward model? Goal? Policies?



Discuss reward model and goal and its implications. How should we implement Aim 1 and Aim 2. Discuss stationary vs non-stationary policy.

- Formulation 1: let reward is the number of steps the agent moves. If the goal is set to be the total reward or average reward (infinite horizon), then the agent should just walk, should not take the risk of falling by running. The total optimal reward is infinite. average optimal reward is 1. But if the goal is set as discounted reward, so it may be useful to run (depending on how small γ is).
- Formulation 2: set reward 0 for all the states except (S,5), where the reward is 1. discounted sum of rewards is γ^τ, equivalent to minimize τ, the time to reach the end without falling. optimal policy might be walking initially and running later



discuss examples of Markovian or non-Markovian policies, deterministic and randomized policies. stationary policies, non-stationary policies

- Greedy policy: maximize immediate expected reward: run on every state. the 2 steps gain at 80% chance means that expected immediate reward is .8*2 = 1.6, which is more than the expected immediate reward of 1 step that can be earned by walking, but that greedy approach ignores all the reward you can make in future if you don't fall.
- Example of randomized policy?
- Consider the policy that takes action walk in all states (S, i) with i less than 4, and run on $i >= 4$. Is that a stationary or non-stationary policy?

Example 2: Inventory control MDP

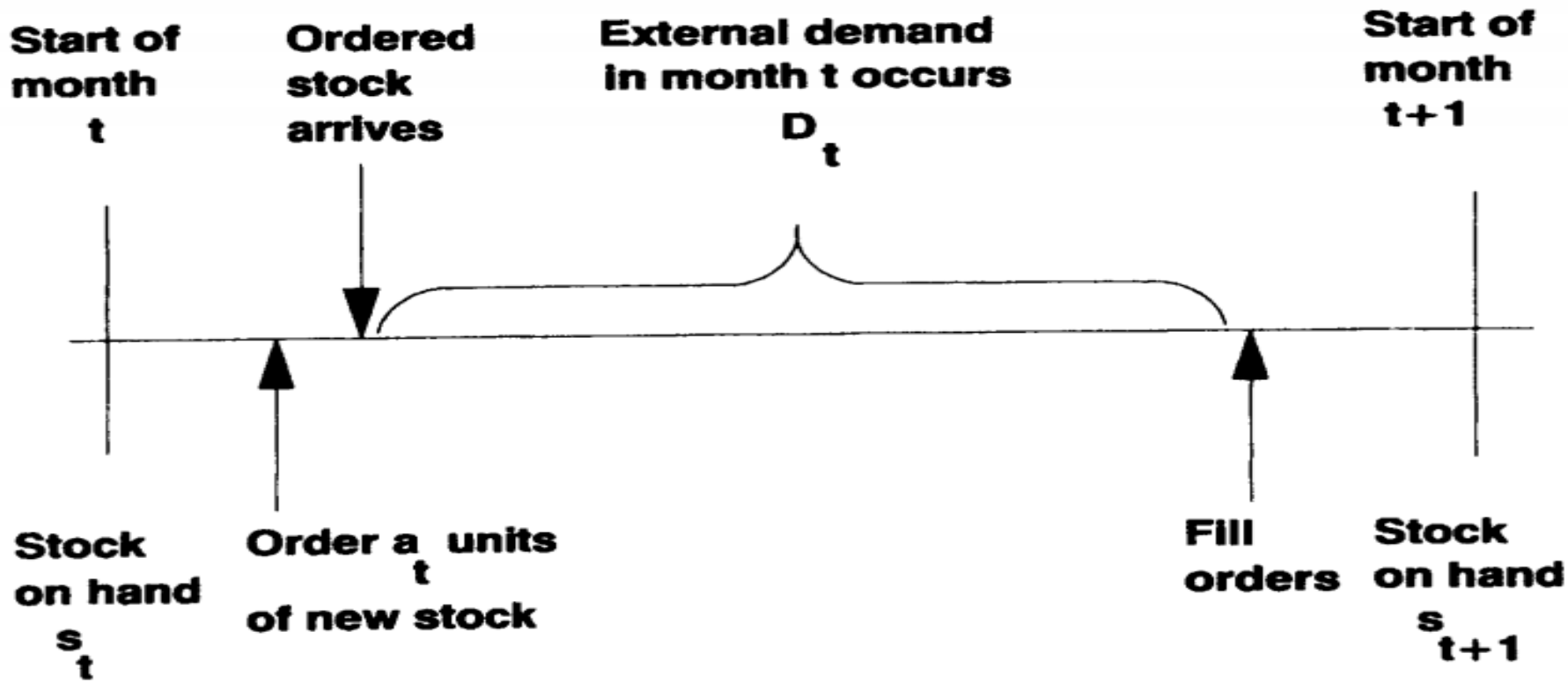


Figure: Timing of events in an inventory problem (Figure taken from Puterman:1994.)

MDP formulation

$$s_{t+1} = \max\{s_t + a_t - D_t, 0\} \equiv [s_t + a_t - D_t]^+.$$

$$O(u) = [K + c(u)]1_{\{u>0\}}.$$

$$r_t(s_t, a_t, s_{t+1}) = -O(a_t) - h(s_t + a_t) + f(s_t + a_t - s_{t+1}).$$

The goal of a inventory policy could be to maximize expected total reward in a finite horizon, or discounted reward if the firm cares more about near future.

Example 3: Tabular MDP

Robot learning to walk

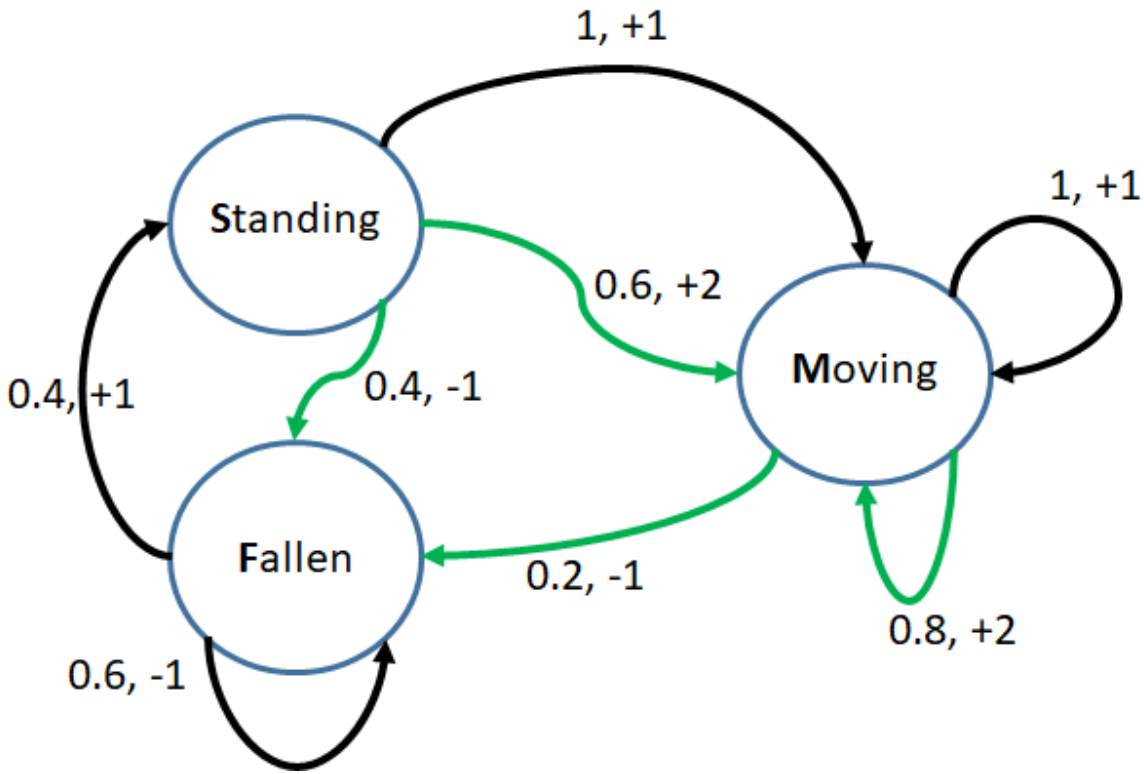


Figure: A simple MDP for the robot toy example

MDP formulation

Here, state space $S = \{'F', 'S', 'M'\}$, $A = \{'slow', 'fast'\}$. R is an $S \times A$ matrix and P is $S \times A \times S$ matrix.

$$R = \begin{bmatrix} (0.6 \times -1 + 0.4 \times 1) & 0 \\ 1 & (0.6 \times 2 + 0.4 \times -1) \\ 1 & (0.8 \times 2 + 0.2 \times -1) \end{bmatrix} = \begin{bmatrix} -0.2 & 0 \\ 1 & 0.8 \\ 1 & 1.4 \end{bmatrix}$$

$$P(s, \text{slow}, s') = \begin{bmatrix} 0.6 & 0.4 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad P(s, \text{fast}, s') = \begin{bmatrix} 1 & 0 & 0 \\ 0.4 & 0 & 0.6 \\ 0.2 & 0 & 0.8 \end{bmatrix}$$

Contents I

Solving an MDP: Finite horizon

$$\max_{\pi} \mathbb{E}[\sum_{t=1}^{H} \gamma^{t-1} r_t | s_1; a_t = \pi_t(s_t)]$$

where maximum is taken over all (non-stationary) policies $\pi = (\pi_1, \ldots, \pi_k)$

Solving an MDP: Finite horizon

$$\max_{\pi} \mathbb{E}[\sum_{t=1}^{H} \gamma^{t-1} r_t | s_1; a_t = \pi_t(s_t)]$$

where maximum is taken over all (non-stationary) policies $\pi = (\pi_1, \ldots, \pi_k)$

- Tree of decisions, 2H height: exponentially large; (s1: A actions: S states: A actions...)
- Optimal substructure property: discuss the idea of memoization: memorize value of state s at this level - do not repeat the calculation
- reduce from $(AS)^H$ computational nodes to HS computations

Solving an MDP: Finite horizon

Dynamic programming algorithm using optimal substructure property

Define for all $s \in \mathcal{S}$, $k = 1, \ldots, H$,

$$V_k^*(s) = \max_{\pi=\{\pi_t\}} \mathbb{E}[\sum_{t=1}^{k} \gamma^{t-1} r_t | s_1 = s]$$

Then, optimal substructure property:

$$V_k^*(s) = \max_a R(s,a) + \gamma \sum_{s'} P(s,a,s') V_{k-1}^*(s')$$

Dynamic programming uses this property backwards starting from $V_1^*(\cdot)$ to finally compute $V_H^*(\cdot)$, the optimal value for horizon H.

Proof

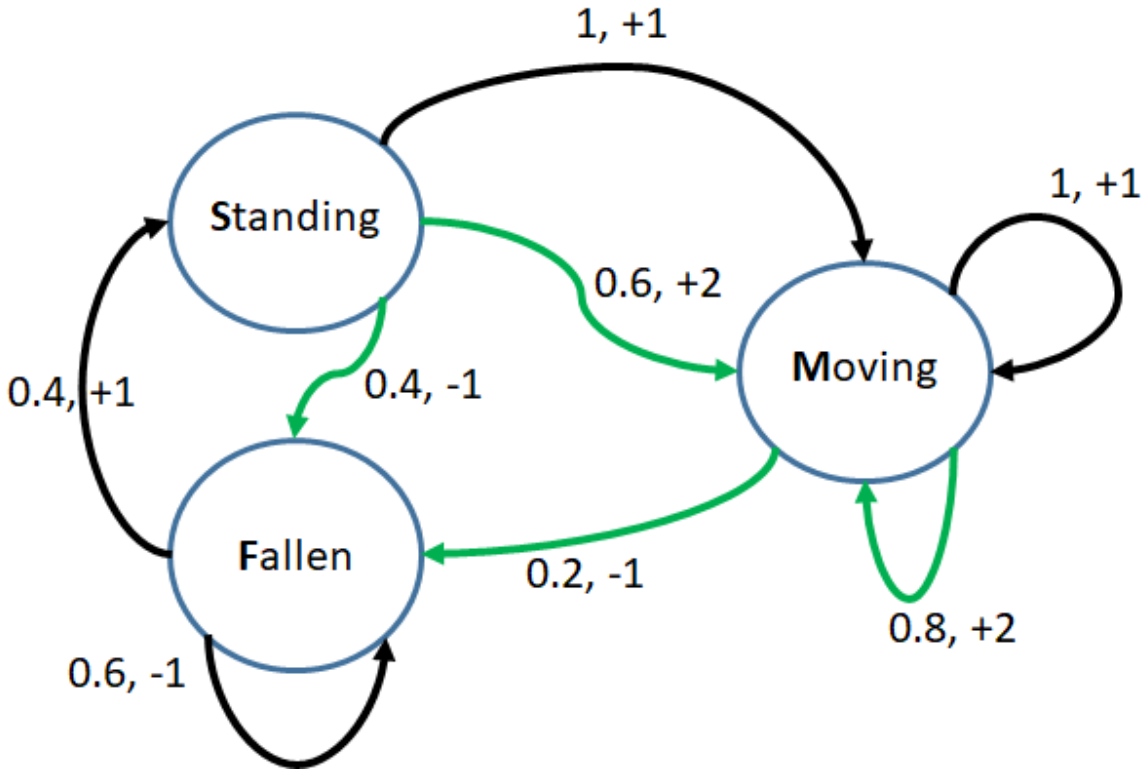
Proof

$$V_k^*(s) = \max_\pi \mathbb{E}[\sum_{t=1}^{k} \gamma^{t-1} r_t | s_1 = s]$$

where maximum is taken over all (non-stationary) policies $\pi = (\pi_1, \ldots, \pi_k)$, $a_t = \pi_t(s_t), \mathbb{E}[r_t|s_t] = R(s_t, a_t), \Pr(s_{t+1} = s'|s_t, a_t) = P(s_t, a_t, s')$. Then, we have the optimal substructure property:

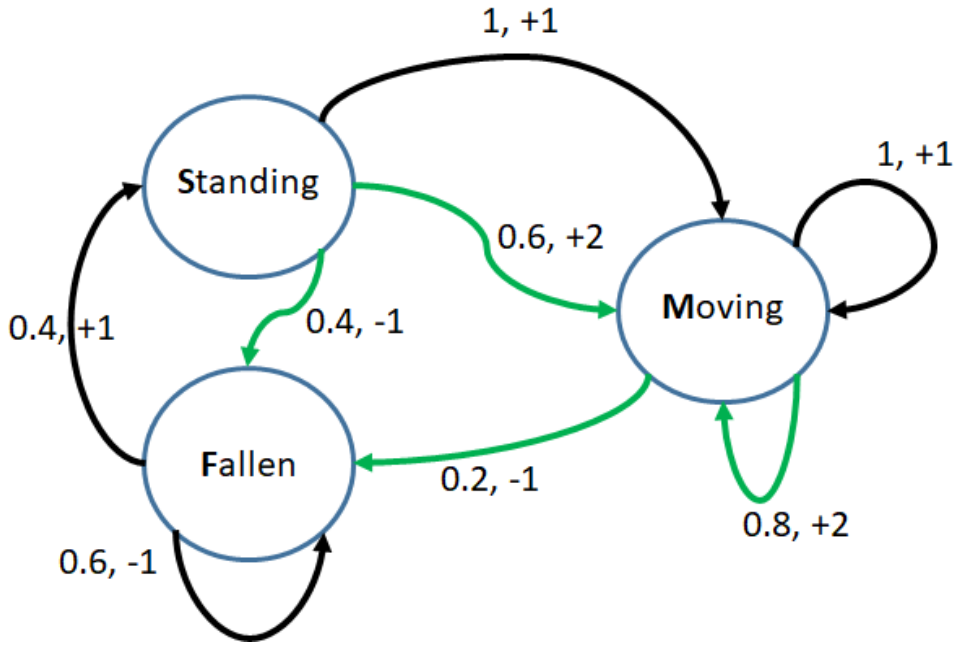
$$\begin{aligned}
V_k^*(s) &= \max_\pi \left\{ \mathbb{E}[r_1|s_1 = s] + \mathbb{E}[\mathbb{E}[\sum_{t=2}^{k} \gamma^{t-1} r_t | s_1 = s, s_2 = s']] \right\} \\
&= \max_a \left\{ R(s,a) + \max_{\pi_2, \ldots, \pi_k} \sum_{s'} P(s,a,s') \mathbb{E}[\sum_{t=2}^{k} \gamma^{t-1} r_t | s_2 = s'] \right\} \\
&= \max_a R(s,a) + \gamma \sum_{s'} P(s,a,s') \left\{ \max_{\pi_1, \ldots, \pi_{k-1}} \mathbb{E}[\sum_{t=1}^{k-1} \gamma^{t-1} r_t | s_1 = s'] \right\} \\
&= \max_a R(s,a) + \gamma \sum_{s'} P(s,a,s') V_{k-1}^*(s'), k = 1, \ldots, H
\end{aligned}$$

Solve the Robot MDP



Let's optimize the expected sum of rewards ($\gamma = 1$) for horizon $H = 4$.

Dynamic Programming

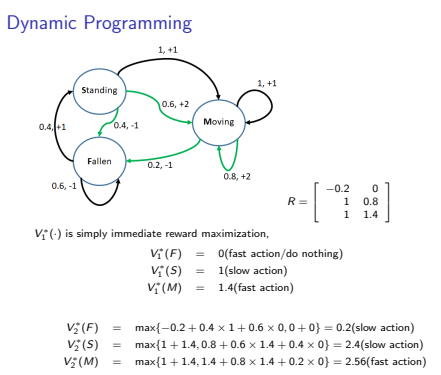


$$R = \begin{bmatrix} -0.2 & 0 \\ 1 & 0.8 \\ 1 & 1.4 \end{bmatrix}$$

$V_1^*(\cdot)$ is simply immediate reward maximization,

$$\begin{aligned} V_1^*(F) &= 0(\text{fast action/do nothing}) \\ V_1^*(S) &= 1(\text{slow action}) \\ V_1^*(M) &= 1.4(\text{fast action}) \end{aligned}$$

$$\begin{aligned} V_2^*(F) &= \max\{-0.2 + 0.4 \times 1 + 0.6 \times 0, 0 + 0\} = 0.2(\text{slow action}) \\ V_2^*(S) &= \max\{1 + 1.4, 0.8 + 0.6 \times 1.4 + 0.4 \times 0\} = 2.4(\text{slow action}) \\ V_2^*(M) &= \max\{1 + 1.4, 1.4 + 0.8 \times 1.4 + 0.2 \times 0\} = 2.56(\text{fast action}) \end{aligned}$$



1. if time horizon is 1, the robot should not try to get up from fallen state.
2. remember this is the policy at the last step in the horizon (0 steps are remaining) we are computing the decisions backwards.
3. (If you use $\gamma < 1$, it might take more time steps for the action in state M to become slow action, depending on how small γ is. Intuitively, if horizon is short or future is either discounted heavily you might want to be more aggressive).
4. In the next iteration, the policy remains the same: remember this is the policy at the first step in the sequential decision making (3 steps are remaining) we are computing the decision backwards.

Contents I

Infinite horizon settings: Bellman optimality equations

We still use the memoization idea but fixed point equations instead of recursive equations.

- Memoization idea in finite horizon: Given remaining horizon k, the optimal value from a state s is fixed irrespective of how you arrived there.
- Memoization in infinite horizon: ~~Given remaining horizon k,~~ the optimal value from a state s is fixed irrespective of how you arrived there.

Bellman equations for value of a stationary policy

Infinite horizon discounted reward setting

Value of stationary policy π from state s (discount factor $\gamma < 1$):

$$V_\gamma^\pi(s) := \lim_{T\to\infty} \mathbb{E}[\sum_{t=1}^{T} \gamma^{t-1} r_t | s_1 = s; a_t = \pi(s_t)]$$

Bellman equations

$$V_\gamma^\pi(s) = \mathbb{E}_{a\sim\pi(s), s'\sim P(s,a)} \left[R(s,a,s') + \gamma V_\gamma^\pi(s') \right]$$

Vector form for finite state space

$$V_\gamma^\pi = \mathbf{R}^\pi + \gamma P^\pi V_\gamma^\pi$$

Value of stationary policy π from state s (discount factor $\gamma < 1$):

$$V_\gamma^\pi(s) := \lim_{T\to\infty} \mathbb{E}[\sum_{t=1}^{T} \gamma^{t-1} r_t | s_1 = s; a_t = \pi(s_t)]$$

Bellman equations

$$V_\gamma^\pi(s) = \mathbb{E}_{a\sim\pi(s), s'\sim P(s,a)}\left[R(s,a,s') + \gamma V_\gamma^\pi(s')\right]$$

Vector form for finite state space

$$V_\gamma^\pi = \mathbf{R}^\pi + \gamma P^\pi V_\gamma^\pi$$

1. Other ways to write

$$V_\gamma^\pi(s) = \mathbb{E}_{a\sim\pi(s)}\left[R(s,a) + \gamma \sum_{s'} P(s,a,s') V^\pi(s')\right],$$

$$V_\gamma^\pi(s) = \mathbb{E}_{a\sim\pi(s)}\left[R(s,a) + \gamma \mathbb{E}_{s'\sim P(s,a)}[V^\pi(s')]\right],$$

Define $R^\pi(s) := \mathbb{E}_{a\sim\pi(s)}[R(s,a)]$.

Proof

$$\begin{aligned} V_\gamma^\pi(s) &= \mathbb{E}[r_1 + \gamma r_2 + \gamma^2 r_3 + \gamma^3 r_4 + \ldots | s_1 = s] \\ &= E[r_1 | s_1 = s] + \gamma \mathbb{E}[\mathbb{E}[r_2 + \gamma r_3 + \gamma^2 r_4 + \ldots | s_2] | s_1 = s] \end{aligned}$$

The first term here is simply the expected reward in state s when (possibly randomized) action is given by $\pi(s)$. The second term is γ times the value function at $s_2 \sim P(s, \pi(s), \cdot)$. We have $R^\pi(s) := \mathbb{E}_{a \sim \pi(s)}[R(s, a)]$. Then,

$$\begin{aligned} V_\gamma^\pi(s) &= \mathbb{E}[R^\pi(s_1) + \gamma V_\gamma^\pi(s_2) | s_1 = s] \\ &= R(s, \pi(s)) + \gamma \sum_{s_2 \in S} P(s, \pi(s), s_2) V_\gamma^\pi(s_2) \\ &= R^\pi(s) + \gamma [P^\pi V_\gamma^\pi](s) \end{aligned}$$

Bellman Optimality Equations

Infinite horizon discounted reward setting

Define optimal value:

$$V_\gamma^*(s) = \sup_{\pi=\{\pi_t\}} \mathbb{E}[r_1 + \gamma r_2 + \gamma^2 r_3 + \gamma^3 r_4 + \ldots | s_1 = s]$$

Bellman optimality equations:

$$V_\gamma^*(s) = \max_a R(s,a) + \gamma \sum_{s'} P(s,a,s') V_\gamma^*(s')$$

More generally,

$$V_\gamma^*(s) = \max_a R(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)}[V_\gamma^*(s')]$$

Proof

For all s, by definition

$$\begin{aligned}
V^*_\gamma(s) &= \sup_{\pi=\{\pi_t\}} \mathbb{E}[r_1 + \gamma r_2 + \gamma^2 r_3 + \gamma^3 r_4 + \ldots | s_1 = s] \\
&= \sup_\pi \mathbb{E}[r_1 | s_1 = s] \\
&\quad + \gamma \sup_\pi \mathbb{E}[\mathbb{E}[r_2 + \gamma r_3 + \gamma^2 r_4 + \ldots | s_1 = s, s_2 = s']] \\
&= \sup_{\pi_1} \mathbb{E}_{a\sim\pi_1(s)} \left[R(s,a) + \gamma \mathbb{E}_{s'\sim P(s,a)} \left[\sup_{\pi_2,\pi_3,\ldots,} \mathbb{E}[r_2 + \gamma r_3 + \ldots | s_2 = s'] \right] \right] \\
&= \max_a R(s,a) + \gamma \mathbb{E}_{s'\sim P(s,a)}[V^*_\gamma(s')]
\end{aligned}$$

Technically, above only shows that V^*_γ satisfies the Bellman equations. Theorem 6.2.2 (c) in [**?**] shows that V^* is in fact unique solution of above equations. Therefore, satisfying these equations is sufficient to guarantee optimality, so that it is not difficult to see that the deterministic (stationary) policy

$$\pi^*_\gamma(s) := \arg\max_a R(s,a) + \gamma \mathbb{E}_{s'\sim P(s,a)}[V^*_\gamma(s')]$$

is optimal (see [**?**] Theorem 6.2.7 for formal proof).

Optimal Policy

Infinite horizon discounted reward setting

$$\pi^*(s) := \arg\max_a R(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)}[V^*_\gamma(s')]$$

Value of optimal policy can also be computed as

$$V^*_\gamma = (I - \gamma P^{\pi^*})^{-1} R^{\pi^*}$$

Inverse exists for $\gamma < 1$

$$\pi^*(s) := \arg\max_a R(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)}[V_\gamma^*(s')]$$

Value of optimal policy can also be computed as

$$V_\gamma^* = (I - \gamma P^{\pi^*})^{-1} R^{\pi^*}$$

Inverse exists for $\gamma < 1$

1. And, by Bellman equations for value of a stationary policy, we have , $V_\gamma^* = V_\gamma^{\pi^*} = R^{\pi^*} + \gamma P^{\pi^*} V_\gamma^{\pi^*}$, i.e., $V_\gamma^* = (I - \gamma P^{\pi^*})^{-1} R^{\pi^*}$, where the inverse exists for $\gamma < 1$.

Finding an optimal policy

Solving the Bellman equations

Assume finite state space and action space (aka 'tabular' setting).

- Linear programming
- Iterative algorithms

Assume finite state space and action space (aka 'tabular' setting).

- Linear programming
- Iterative algorithms

1. Cannot do backward recursion of DP
2. Solving a system of S fixed point equations.

LP for solving Bellman Equations

The fixed point of Bellman optimality equations can be found by solving the following linear program.

$$\begin{aligned} \min_{\mathbf{v} \in \mathbb{R}^S} \quad & \sum_s v_s \\ \text{subject to} \quad & v_s \geq R(s,a) + \gamma P(s,a)^\top \mathbf{v} \quad \forall a, s \end{aligned}$$

Proof

We prove that any optimal solution $\mathbf{v}^*$ of the above LP, if exists, satisfies Bellman equation, and any solution to the Bellman equation forms a (feasible and) optimal solution to the above LP. Consider any optimal solution $\mathbf{v}^*$ of the above LP, if exists. Then, we show that it satisfies the Bellman equations. It is easy to see from the constraints that it satisfies for all s,

$$v_s^* \geq \max_a R(s,a) + \gamma P(s,a)^T \mathbf{v}^*$$

We prove that the above must indeed be satisfied by equality by v_s^* for all s. Assume for contradiction that some s above is not satisfied by equality, then $v_s^* = \delta + \max_a R(s,a) + \gamma P(s,a)^T \mathbf{v}^*$ for some $\delta \geq 0$. Then, we argue that a better solution to the LP can be obtained by decreasing v_s^* by δ, which is a contradiction to the optimality of v^*. Specifically, if we obtain a new vector $\hat{v}$ as $\hat{v}_s = v_s^* - \delta, \hat{v}_{-s} = v_{-s}^*$, then the new solution is still feasible, because

$$\hat{v}_s = v_s^* - \delta = \max_a R(s,a) + \gamma P(s,a)^T \mathbf{v}^* \geq \max_a R(s,a) + \gamma P(s,a)^T \hat{v}$$

and for any $s' \neq s$,

$$\hat{v}_{s'} = v_{s'}^* \geq \max_a R(s,a) + \gamma P(s,a)^T \mathbf{v}^* \geq \max_a R(s,a) + \gamma P(s,a)^T \hat{v}$$

Also, $\sum_s \hat{v}_s = \sum_s v_s^* - \delta$. Now since the objective minimizes $\sum_s v_s$, this means $\hat{v}$ strictly improves upon the optimal value, which is a contradiction. Therefore, it must

For the other direction, let V^* satisfies the Bellman optimality equations. Then we show that V^* will be a feasible and optimal solution of the above LP. V^* clearly satisfies the constraints of the above LP. Next, we show that V^* minimizes the objective of the LP. Consider any feasible $\mathbf{v}$. Then,
$v_s \geq R(s,a) + P(s,a)^\top \mathbf{v}, \forall s, a$ implies that

$$v_s \geq R(s, \pi^*(s)) + \gamma P(s, \pi^*(s))^\top \mathbf{v}, \forall s$$

(Above is written assuming optimal policy π^* is deterministic, which is in fact true in the infinite horizon discounted reward case.) Or,

$$(I - \gamma P^{\pi^*})\mathbf{v} \geq R^{\pi^*}$$

Because $\gamma < 1$, $(I - \gamma P^\pi)^{-1}$ exists for all π, and for any $u \geq 0$

$$(I - \gamma P^\pi)^{-1} u = (I + \gamma P^\pi + \gamma^2 (P^\pi)^2 + \cdots) u \geq 0$$

Therefore, from above

$$(I - \gamma P^{\pi^*})^{-1}((I - \gamma P^{\pi^*})\mathbf{v} - R^{\pi^*}) \geq 0$$

Or,

$$\mathbf{v} \geq (I - \gamma P^{\pi^*})^{-1} R^{\pi^*} = V^*$$

Therefore, $\mathbf{1}^\top \mathbf{v} \geq \mathbf{1}^\top V^*$ for all feasible $\mathbf{v}$. This proves that V^* is an optimal solution

Iterative algorithms

Finite state space and action space

- Value iteration: Iteratively improve estimate of optimal value vector $[V^*(1), \ldots, V^*(S)]$.
- Q-value iteration: Iteratively improve the estimate of Q-values $[Q^*(s, a), s \in \mathcal{S}, a \in \mathcal{A}]$. (to be defined)
- Policy iteration: Iteratively improve estimate of optimal policy $[\pi^*(1), \ldots, \pi^*(S)]$.

Iterative algorithms: Value Iteration

Infinite horizon discounted reward setting

Estimate the optimal value vector

1. Start with an arbitrary initialization $\mathbf{v}^0$. Specify $\epsilon > 0$
2. **Repeat** for $k = 1, 2, \ldots$ **until** $\|\mathbf{v}^k - \mathbf{v}^{k-1}\|_\infty \leq \epsilon \frac{(1-\gamma)}{2\gamma}$:
   - Update value vector estimate $\mathbf{v}^k$ for every $s \in S$, improve the value vector as:

$$\mathbf{v}^k(s) = \max_{a \in A} R(s,a) + \gamma \sum_{s'} P(s,a,s') \mathbf{v}^{k-1}(s'), \qquad (1)$$

3. Output a near-optimal policy

$$\pi(s) \in \arg\max_a R(s,a) + \gamma P(s,a)^\top \mathbf{v}^k \qquad (2)$$

Bellman operator

$L, L^\pi : \mathbb{R}^S \to \mathbb{R}^S$.

$$
\begin{aligned}
[LV](s) &:= \max_{a \in A} R(s,a) + \gamma \sum_{s'} P(s,a,s') V(s') \\
[L^\pi V](s) &:= \mathbb{E}_{a \in \pi(s)} [R(s,a) + \gamma \sum_{s'} P(s,a,s') V^\pi(s')]
\end{aligned}
$$

Iterative algorithms: Value Iteration

Infinite horizon discounted reward setting

Estimate the optimal value vector

1. Start with an arbitrary initialization $\mathbf{v}^0$. Specify $\epsilon > 0$
2. **Repeat** for $k = 1, 2, \ldots$ **until** $\|\mathbf{v}^k - \mathbf{v}^{k-1}\|_\infty \leq \epsilon \frac{(1-\gamma)}{2\gamma}$:

$$\mathbf{v}^k = L\mathbf{v}^{k-1}$$

3. Output a near-optimal policy

$$\pi(s) \in \arg\max_a R(s,a) + \gamma P(s,a)^\top \mathbf{v}^k$$

Analysis

Theorem (Theorem 6.3.3, Section 6.3.2 in Puterman:1994)

The convergence rate of the above algorithm is linear at rate γ. Specifically,

$$\|\mathbf{v}^k - V^*\|_\infty \leq \frac{\gamma^k}{1-\gamma}\|v^1 - v^0\|_\infty$$

Further, let π^k be the arg max policy defined by v^k. Then,

$$\|V^{\pi^k} - V^*\|_\infty \leq \frac{2\gamma^k}{1-\gamma}\|v^1 - v^0\|_\infty$$

Contraction property of L-operator

$$\|Lv - Lu\|_\infty \leq \gamma \|v - u\|_\infty.$$

$$\|L^\pi v - L^\pi u\|_\infty \leq \gamma \|v - u\|_\infty.$$

Proof.

First assume $Lv(s) \geq Lu(s)$. Let
$a_s^* = \arg\max_{a \in A} R(s,a) + \gamma \sum_{s'} P(s,a,s')v(s')$

$$\begin{aligned}
0 &\leq Lv(s) - Lu(s) \\
&\leq R(s,a_s^*) + \gamma \sum_{s'} P(s,a_s^*,s')v(s') - R(s,a_s^*) - \gamma \sum_{s'} P(s,a_s^*,s')u(s') \\
&= \gamma P(s,a_s^*)^\top (v - u) \\
&\leq \gamma \|v - u\|_\infty
\end{aligned}$$

Repeating a symmetric argument for the case $Lu(s) \geq Lv(s)$ gives the lemma statement. Similar proof holds for L^π. □

Proof of the value iteration convergence theorem

By Bellman equations $V^* = LV^*$.

$$
\begin{aligned}
\|V^* - v^k\|_\infty &= \|LV^* - v^k\|_\infty \\
&\leq \|LV^* - Lv^k\|_\infty + \|Lv^k - v^k\|_\infty \\
&= \|LV^* - Lv^k\|_\infty + \|Lv^k - Lv^{k-1}\|_\infty \\
&\leq \gamma\|V^* - v^k\| + \gamma\|v^k - v^{k-1}\| \\
&\leq \gamma\|V^* - v^k\| + \gamma^k\|v^1 - v^0\| \\
\|V^* - v^k\|_\infty &\leq \frac{\gamma^k}{1-\gamma}\|v^1 - v^0\|
\end{aligned}
$$

Let $\pi = \pi^k$ be the policy at the end of k iterations. Then, $V^\pi = L^\pi V^\pi$ by Bellman equations. Further, by definition of $\pi = \pi^k$,

$$L^\pi v^k(s) = \max_a R(s,a) + \gamma \sum_{s'} P(s,a,s') v^k(s') = L v^k(s).$$

Therefore,

$$\begin{aligned}
\|V^\pi - v^k\|_\infty &= \|L^\pi V^\pi - v^k\|_\infty \\
&\leq \|L^\pi V^\pi - L^\pi v^k\|_\infty + \|L^\pi v^k - v^k\|_\infty \\
&= \|L^\pi V^\pi - L^\pi v^k\|_\infty + \|L v^k - L v^{k-1}\|_\infty \\
&\leq \gamma \|V^\pi - v^k\| + \gamma \|v^k - v^{k-1}\| \\
\|V^\pi - v^k\|_\infty &\leq \frac{\gamma}{1-\gamma} \|v^k - v^{k-1}\| \\
&\leq \frac{\gamma^k}{1-\gamma} \|v^1 - v^0\|
\end{aligned}$$

Adding the two results we get the theorem statement.

Q-values and Q-value-iteration

Q-values are defined as values after fixing the first action

- $Q^*(s, a)$ is defined the expected utility on taking action a in state s, and thereafter acting optimally.

$$Q^*(s, a) := R(s, a) + \sum_{s' \in \mathcal{S}} P(s, a, s') V^*(s')$$

$$V^*(s) := \max_a Q^*(s, a)$$

- Bellman Optimality Equation

$$Q^*(s, a) = R(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s, a, s') \left(\max_{a'} Q^*(s', a') \right)$$

- $Q^\pi(s, a)$ is defined the expected utility on taking action a in state s, and thereafter playing policy π.

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

$$V^\pi(s) = \mathbb{E}_{a \in \pi(s)}[Q^\pi(s, a)]$$

Q-values and Q-value iteration

Q-value iteration estimates $Q^*(s, a)$ s for all s, a.
(instead of $V^*(s)$ for all s in value iteration)

Why?
No need to know the MDP model to compute optimal policy:

$$\pi^*(s) = \arg\max_a Q^*(s, a)$$

Important in the learning setting when we don't know the model.

Q-value iteration estimates $Q^*(s, a)$ s for all s, a.
(instead of $V^*(s)$ for all s in value iteration)

Why?
No need to know the MDP model to compute optimal policy:

$$\pi^*(s) = \arg\max_a Q^*(s, a)$$

Important in the learning setting when we don't know the model.

In value iteration once we have $\hat{v}(s)$ (a good estimate of $V^*(s)$) for all s, we compute a near-optimal policy as

$$\pi(s) = \arg\max_a R(s, a) + \gamma P(s, a)^T \hat{v}$$

Even if we have good estimates of V^* from samples, we would need the reward and transition model just to compute the optimal policy.

Q-value iteration

1. Start with an arbitrary initialization $\mathbf{Q}^0 \in \mathbb{R}^{S \times A}$.
2. In every iteration k, improve the Q-value vector as:

$$\mathbf{Q}^k(s,a) = R(s,a) + \gamma \sum_{s'} P(s,a,s') \left(\max_{a'} Q^{k-1}(s',a') \right), \forall s, a$$

3. Stop if $\|Q^k - Q^{k-1}\|_\infty$ is small.
4. Output policy π^k defined as $\pi^k(s) = \arg\max_a Q^k(s,a)$

Convergence proof follows from value iteration convergence.

Policy iteration

Directly estimate the optimal policy π^*.
Use that at π^*

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

Greedy update should give back π^*

Policy iteration using values

1. Initialize policy π^0.
2. In every iteration $k = 0, 1, \ldots,$,
   - (Policy evaluation) Compute value $V^{\pi^k}(s)$, the value of policy π^k for every state s.
   - (Greedy Policy improvement) Compute new policy

$$\pi^{k+1}(s) := \arg\max_a R(s,a) + \gamma \sum_{s'} P(s,a,s') V^{\pi^k}(s'), \forall s$$

3. Stop when $\pi^{k+1} = \pi^k$.

Relaxed stopping criteria: not much change in policy or its value.

Policy iteration using Q-values

1. Initialize policy π^0.
2. In every iteration $k = 0, 1, 2, \ldots,$
   - (Policy evaluation) Compute value $Q^{\pi^k}(s, a)$, the Q-values of policy π^k for all s, a.
   - (Greedy Policy improvement) Compute new policy

$$\pi^{k+1}(s) := \arg\max_a Q^{\pi_k}(s, a), \forall s$$

3. Stop when $\pi^{k+1} = \pi^k$.

Relaxed stopping criteria: not much change in policy or its value.

Policy-iteration vs. Value-iteration

+ separate the policy evaluation (learning) and the improvement (optimization) steps
+ can actually be faster if estimating the performance of a fixed policy is much easier than finding an optimal policy
+ can warm start if there is a known good initial policy to start from and improve upon
+ always maintains a good policy
- parameterizing a policy (function) can be more difficult/complex than parameterizing a value (vector)
- If policy evaluation is done by an iterative method like value iteration then policy iteration is slower.

Policy iteration convergence proof

Theorem

For the policy π^k computed in the k^{th} iteration of policy iteration, we have

$$\|V^{\pi^k} - V^*\|_\infty \leq \gamma^k \|V^{\pi_0} - V^*\|_\infty$$

- Compare it to the value iteration convergence.
- Why does this look much better? Is it really much better?

Policy iteration convergence proof

Theorem

For the policy π^k computed in the k^{th} iteration of policy iteration, we have

$$\|V^{\pi^k} - V^*\|_\infty \leq \gamma^k \|V^{\pi_0} - V^*\|_\infty$$

- Compare it to the value iteration convergence.
- Why does this look much better? Is it really much better?

1. Each iteration is longer in policy iteration. But can actually be faster if the evaluation of a fixed policy is easy using sampling or other methods like $V^\pi = (I - \gamma P^\pi)^{-1} R^\pi$.

Policy iteration converence

How much does the policy improve in one step?

Lemma

$$V^{\pi^{k+1}} \geq LV^{\pi^k}$$

Once we can prove the above, the proof is similar to value iteration.

Why is this not trivial (unlike value iteration)?

Proof: We prove

$$V^{\pi^k} \leq LV^{\pi^k} \leq V^{\pi^{k+1}}$$

Let π be any policy, and π' be the policy obtained by the greedy step, i.e.,

$$\pi'(s) = \arg\max_a R(s,a) + \gamma \sum_{s'} P(s,a,s')V^{\pi}(s'), \forall s$$

Then, by definition of L and $L^{\pi'}$, we have,

$$LV^{\pi} = L^{\pi'}V^{\pi}$$

Also, by Bellman equations,

$$V^{\pi} = L^{\pi}V^{\pi} \leq LV^{\pi}$$

Combining the last two inequalities, we have

$$V^{\pi} \leq LV^{\pi} = L^{\pi'}V^{\pi} \tag{3}$$

Using above as base case, we prove by induction that for all $i = 1, 2, 3, \ldots,$

$$V^{\pi} \leq LV^{\pi} \leq (L^{\pi'})^i V^{\pi} \tag{4}$$

Suppose that the above statement is true for i, i.e.,

$$V^{\pi} \leq (L^{\pi'})^i V^{\pi}$$

Then, using monotonicity of $L^{\pi'}$, If we apply $L^{\pi'}$ on both sides:

$$L^{\pi'} V^{\pi} \leq (L^{\pi'})^{i+1} V^{\pi}$$

Plugging this in (3),

$$V^{\pi} \leq LV^{\pi} = L^{\pi'} V^{\pi} \leq (L^{\pi'})^{i+1} V^{\pi}$$

which proves statement (4) by induction.
Now, taking i to ∞ in (4), since $(L^{\pi'})^i V^{\pi}$ will converge to $V^{\pi'}$ (this follows from convergence of value iteration for evaluating a policy)

$$V^{\pi} \leq LV^{\pi} \leq V^{\pi'}$$

Proof of Theorem From the previous lemma, we have

$$V^{\pi^k} \geq LV^{\pi^{k-1}}$$

Subtracting from V^* and using $V^* = LV^*$,

$$V^* - V^{\pi^k} \leq LV^* - LV^{\pi^{k-1}}$$

so that

$$\|V^* - V^{\pi^k}\|_\infty \leq \|LV^* - LV^{\pi^{k-1}}\|_\infty \leq \gamma \|V^* - V^{\pi^{k-1}}\|_\infty$$

Applying this repeatedly for $k-1, \ldots, 1$, we get

$$\|V^* - V^{\pi^k}\|_\infty \leq \gamma^k \|V^* - V^{\pi^0}\|_\infty$$

Contents I

Infinite horizon Average reward goal

Find a policy π that maximizes the average reward under that policy

$$\rho^{\pi}(s_1) = \lim_{T \to \infty} \frac{1}{T} \mathbb{E}[\sum_{t=1}^{T} r_t | s_1 = s; a_t = \pi(s)]$$

Connections to finite horizon reward

$$\rho^{\pi}(s) = \lim_{T \to \infty} \frac{1}{T} V_T^{\pi}(s)$$

Connections to discounted reward

$$\rho^{\pi}(s) = \lim_{\gamma \to 1} (1 - \gamma) V_{\gamma}^{\pi}(s)$$

Bias of a policy

For average reward case, an important quantity is bias of a policy π from state s is defined as

$$h^{\pi}(s) = \lim_{T \to \infty} \mathbb{E}[\sum_{t=1}^{T} (r_t - \rho^{\pi}(s_t)) | s_1 = s; a_t = \pi(s_t)]$$

The limit in above is Cesaro limit which exists. More details in Section 8.2 of Puterman:1994.

Bias of a policy

For average reward case, an important quantity is bias of a policy π from state s is defined as

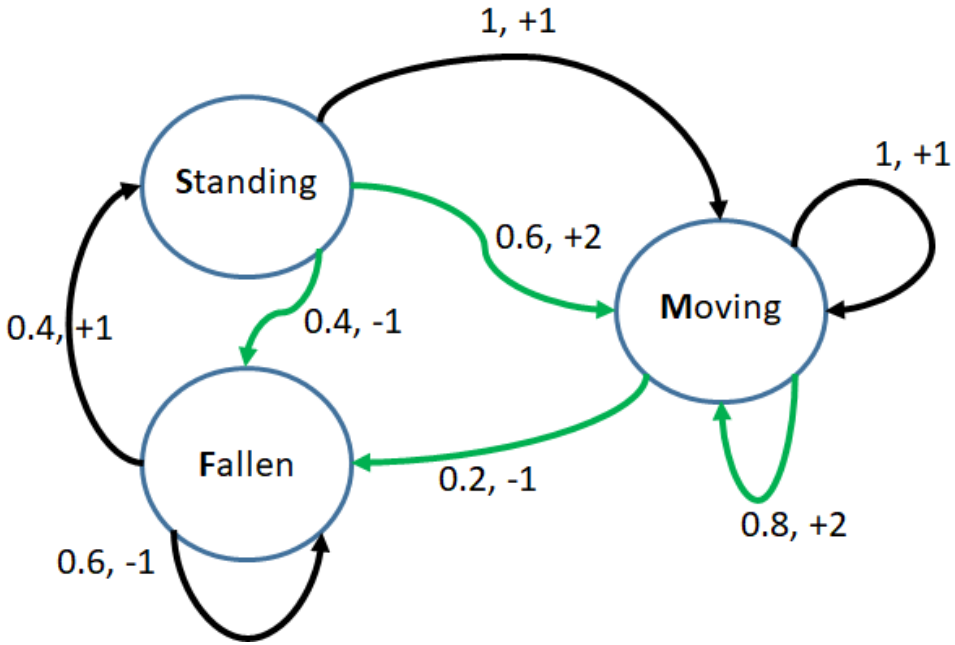
$$h^{\pi}(s) = \lim_{T \to \infty} \mathbb{E}[\sum_{t=1}^{T} (r_t - \rho^{\pi}(s_t)) | s_1 = s; a_t = \pi(s_t)]$$

The limit in above is Cesaro limit which exists. More details in Section 8.2 of Puterman:1994.

Cesaro limit: partial sums may not converge but, average of partial sum converges. For example consider the series $(1, -1, 1, -1, \ldots)$. Partial sums are $1, 0, 1, 0$..., does not converge but average of partial sums converges to 0.

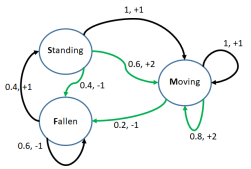
Significance of Bias: Bias represents initial (finite time) deviation from the infinite horizon average reward. It is not important for computing the performance of a given policy, but it plays a key role in finding the optimal policy. bias at the current policy will tell us how to improve the policy

Example



For each state, compute bias of the policy that plays slow action in all states.

Example



For each state, compute bias of the policy that plays slow action in all states.

For MDP in the Figure, consider the policy (say π) that takes the slow action in all states. The gain of this policy is $\rho^\pi(s) = 1$ for all states $s \in \{F, S, M\}$. The bias of this policy is $h^\pi(F) = (-0.2 - 1) \times (1/0.4) = -3, h^\pi(S) = 0, h^\pi(M) = 0$. (For calculating $h^\pi(F)$, note that the expected number of steps spent in Fallen state when taking slow actions is $1/0.4$, after that the reward of the given policy is 1).

Connection of Bias and value

Under a policy π, if two states s, s' are in the same irreducible class (i.e., can be reached from each other in finite expected time) then

- Finite time value:

$$h^\pi(s) - h^\pi(s') = \lim_{T\to\infty}(V_T^\pi(s) - V_T^\pi(s'))$$

  where $V_T^\pi(s) = \mathbb{E}[\sum_{t=1}^T r_t | s_1 = s]$

- Discounted value:

$$h^\pi(s) - h^\pi(s') = \lim_{\gamma\to 1}(V_\gamma^\pi(s) - V_\gamma^\pi(s'))$$

When comparing outcomes from two different states, bias behaves like the value function in the other settings.

Connection of Bias and value

Under a policy π, if two states s, s' are in the same irreducible class (i.e., can be reached from each other in finite expected time) then

- Finite time value:

$$h^\pi(s) - h^\pi(s') = \lim_{T\to\infty}(V_T^\pi(s) - V_T^\pi(s'))$$

  where $V_T^\pi(s) = \mathbb{E}[\sum_{t=1}^T r_t | s_1 = s]$
- Discounted value:

$$h^\pi(s) - h^\pi(s') = \lim_{\gamma\to 1}(V_\gamma^\pi(s) - V_\gamma^\pi(s'))$$

When comparing outcomes from two different states, bias behaves like the value function in the other settings.

To see why we need the two states to be in the same irreducible class, note that being in the same irreducible class ensures that $\rho^\pi(s_t) = \rho^\pi(s_t')$ for all the states visited from s vs s'.

Bellman equations in average reward case

Given a policy π such that all s, s' are reachable from each other in finite time.

$$\rho^\pi(s) + h^\pi(s) = \mathbb{E}_{a\sim\pi(s),s'\sim P(s,a)}\left[R(s,a,s') + h^\pi(s')\right], \forall s$$

Or, in compact notation:

$$\mathbf{h}^\pi + \rho^\pi = \mathbf{R}^\pi + P^\pi \mathbf{h}^\pi$$

Bellman equations in average reward case

Given a policy π such that all s, s' are reachable from each other in finite time.

$$\rho^\pi(s) + h^\pi(s) = \mathbb{E}_{a\sim\pi(s), s'\sim P(s,a)}\left[R(s,a,s') + h^\pi(s')\right], \forall s$$

Or, in compact notation:

$$\mathbf{h}^\pi + \rho^\pi = \mathbf{R}^\pi + P^\pi \mathbf{h}^\pi$$

Also, if we assume a policy π is such that all the states reached form a single recurrent class. Then, for all s, s',

$$\rho^\pi(s) = \rho^\pi(s')$$

And, for any $h \in \mathbb{R}^n, \rho \in \mathbb{R}$ that satisfy the equations,

$$\rho\mathbf{e} + h = \mathbf{R}^\pi + P^\pi h$$

we have that $\rho = \rho^\pi$ and $h = h^\pi + c\mathbf{e}$ for some constant c. Here $\mathbf{e}$ is the S-dimensional vector of all 1s.

Bellman equations in average reward case

Given a policy π such that all s, s' are reachable from each other in finite time.

$$\rho^\pi(s) + h^\pi(s) = \mathbb{E}_{a\sim\pi(s), s'\sim P(s,a)}\left[R(s,a,s') + h^\pi(s')\right], \forall s$$

Or, in compact notation:

$$\mathbf{h}^\pi + \rho^\pi = \mathbf{R}^\pi + P^\pi \mathbf{h}^\pi$$

(assumes finite or countable state space and policy space, and deterministic policy π for simplicity. Similar derivation can be done for randomized policy with notational changes.)

First we show that h^π, ρ^π satisfy the Bellman evaluation equations. We can use the Bellman equations for discounted value:

$$V_\gamma^\pi = \mathbf{R}_\pi + \gamma P^\pi V_\gamma^\pi$$

Subtract γV_γ^π from both sides:

$$V_\gamma^\pi(1-\gamma) = \mathbf{R}_\pi + \gamma P^\pi V_\gamma^\pi - \gamma V_\gamma^\pi$$

For state s:

$$\begin{aligned}
V_\gamma^\pi(s)(1-\gamma) &= \mathbf{R}_\pi(s) + \sum_{s'} \gamma P^\pi(s,s') V_\gamma^\pi(s') - \gamma V_\gamma^\pi(s) \\
&= \mathbf{R}_\pi(s) + \sum_{s'} \gamma P^\pi(s,s')(V_\gamma^\pi(s') - V_\gamma^\pi(s)) \\
\lim_{\gamma\to 1} V_\gamma^\pi(s)(1-\gamma) &= \mathbf{R}_\pi(s) + \sum_{s'} P^\pi(s,s') \lim_{\gamma\to 1} \gamma(V_\gamma^\pi(s') - V_\gamma^\pi(s)) \\
\rho^\pi(s) &= \mathbf{R}_\pi(s) + \sum_{s'} P^\pi(s,s')(h^\pi(s') - h^\pi(s))
\end{aligned}$$

Bellman equations in average reward case

Given a policy π such that all s, s' are reachable from each other in finite time.

$$\rho^\pi(s) + h^\pi(s) = \mathbb{E}_{a\sim\pi(s), s'\sim P(s,a)}\left[R(s,a,s') + h^\pi(s')\right], \forall s$$

Or, in compact notation:

$$\mathbf{h}^\pi + \rho^\pi = \mathbf{R}^\pi + P^\pi \mathbf{h}^\pi$$

This shows that ρ^π, h^π satisfies Bellman equations . For a proof of that second part, we want to show that any feasible ρ, h to the equations

$$\rho e + h = R^\pi + P^\pi h$$

gives gain, bias of policy π.
To see this (rough proof) multiply by $(P^\pi)^i$ on both sides and sum for $i = 0, 1, 2, \ldots, T-1$. Then,

$$T\rho e + \sum_{i=0}^{T-1}(P^\pi)^i h = \sum_{i=0}^{T-1}(P^\pi)^i R^\pi + \sum_{i=0}^{T-1}(P^\pi)^{i+1} h$$

which is equivalent to

$$T\rho e + h = \sum_{i=0}^{T-1}(P^\pi)^i R^\pi + (P^\pi)^T h$$

Dividing by T and taking $T \to \infty$, we get

$$\rho e = \lim_{T\to\infty} \frac{1}{T} \sum_{i=0}^{T-1}(P^\pi)^i R^\pi$$

That is ρ is the average reward of policy π

Bellman equations in average reward case

Given a policy π such that all s, s' are reachable from each other in finite time.

$$\rho^\pi(s) + h^\pi(s) = \mathbb{E}_{a\sim\pi(s), s'\sim P(s,a)}\left[R(s,a,s') + h^\pi(s')\right], \forall s$$

Or, in compact notation:

$$\mathbf{h}^\pi + \rho^\pi = \mathbf{R}^\pi + P^\pi \mathbf{h}^\pi$$

And rearranging the terms and taking $T \to \infty$, we get

$$h - \lim_{T\to\infty} (P^\pi)^T h = \lim_{T\to\infty} (\sum_{i=0}^{T-1} (P^\pi)^i R^\pi - \rho e)$$

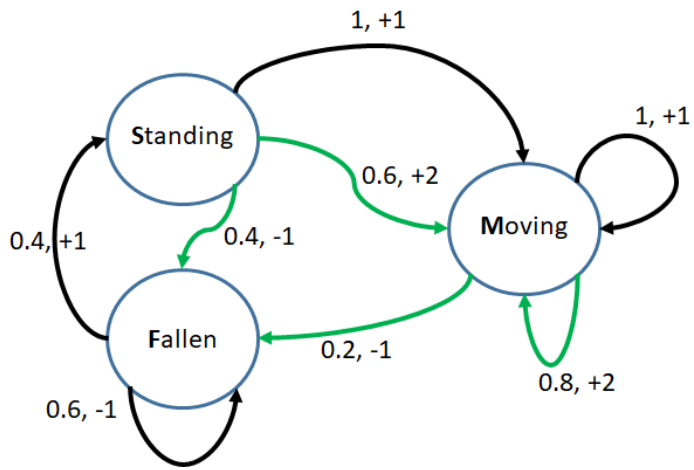
which (under certain technical conditions) gives

$$h + ce = \lim_{T\to\infty} \sum_{i=0}^{T-1} ((P^\pi)^i R^\pi - \rho e)$$

for a constant c, i.e., h is the bias of policy π within a constant c. For a more rigorous proof, refer to Theorem 8.2.6 in [**?**].

Example

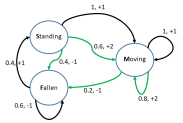
Consider the robot example. Check that the bias and average reward (aka gain) of the policy that always plays slow actions satisfy the Bellman equations stated above.



$$R^\pi = \begin{bmatrix} -0.2 \\ 1 \\ 1 \end{bmatrix}, P^\pi = \begin{bmatrix} 0.6 & 0.4 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Example

Consider the robot example. Check that the bias and average reward (aka gain) of the policy that always plays slow actions satisfy the Bellman equations stated above.



$R^\pi = \begin{bmatrix} -0.2 \\ 1 \\ 1 \end{bmatrix}$, $P^\pi = \begin{bmatrix} 0.6 & 0.4 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$

$$\underbrace{\begin{bmatrix} -3 \\ 0 \\ 0 \end{bmatrix}}_{h^\pi} + \underbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}}_{\rho^\pi \mathbf{e}} = \underbrace{\begin{bmatrix} -0.2 \\ 1 \\ 1 \end{bmatrix}}_{R^\pi} + \underbrace{\begin{bmatrix} 0.6 & 0.4 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}}_{P^\pi} \underbrace{\begin{bmatrix} -3 \\ 0 \\ 0 \end{bmatrix}}_{h^\pi}$$

Bellman optimality equations

Assume communicating MDP.

Definition

An MDP is called **communicating** if for any two states s, s', there exists a policy such that the expected number of steps to reach s' from s is finite.

Theorem

For communicating MDP, for optimal gain policy $\rho^*(s) = \rho^*(s') = \rho^*$, *i.e., optimal average infinite horizon reward does not depend on the starting state.*

Assume communicating MDP.

Definition

An MDP is called **communicating** if for any two states s, s', there exists a policy such that the expected number of steps to reach s' from s is finite.

Theorem

For communicating MDP, for optimal gain policy $\rho^*(s) = \rho^*(s') = \rho^*$, i.e., optimal average infinite horizon reward does not depend on the starting state.

Intuitive proof: Suppose that that there exists $s_1 \neq s_2$ such that $\rho^*(s_1) > \rho^*(s_2)$. Since the MDP is communicating there exists a policy π_0 using which we can go from s_2 to s_1 in time τ with finite expected value, say $\mathbb{E}[\tau] \leq D$. Then we can construct a (possibly non-stationary) policy, which first goes from s_2 to s_1 using π_0 in at most D steps in expectation, and then uses the optimal policy (say π_1) for s_1. Such a policy will have infinite horizon average reward $\rho^*(s_1)$ which is strictly greater than $\rho^*(s_2)$, thus violating the optimality of $\rho^*(s_2)$.

Bellman Optimality Equations for average reward case

Assuming communicating MDP, gain and bias ρ, h of optimal policy satisfies the following equations:

$$\rho + h(s) = \max_a R(s,a) + \sum_{s' \in \mathcal{S}} P(s,a,s')h(s'), \forall s$$

Also, for any feasible solution (ρ, h) to the above equations, we can get an optimal policy π^* defined as

$$\pi^*(s) \in \arg\max_a R(s,a) + \sum_{s'} P(s,a,s')h(s'),$$

with $\rho = \rho^{\pi^*}$ and $h = h^{\pi^*} + c\mathbf{e}$ for some constant c.

- Note that to compute the optimal policy we need to just know the bias vector h that satisfies Bellman equations.

Bellman Optimality Equations for average reward case

Assuming communicating MDP, gain and bias ρ, h of optimal policy satisfies the following equations:

$$\rho + h(s) = \max_a R(s,a) + \sum_{s' \in \mathcal{S}} P(s,a,s')h(s'), \forall s$$

Also, for any feasible solution (ρ, h) to the above equations, we can get an optimal policy π^* defined as

$$\pi^*(s) \in \arg\max_a R(s,a) + \sum_{s'} P(s,a,s')h(s'),$$

with $\rho = \rho^{\pi^*}$ and $h = h^{\pi^*} + c\mathbf{e}$ for some constant c.

- Note that to compute the optimal policy we need to just know the bias vector h that satisfies Bellman equations.

We prove the above lemma partially. We assume that a solution $(\rho^*, h^*)$to the above equations exist. Then, we show that $\rho^* \geq \rho^\pi$ for every policy $\pi = (\pi_1, \ldots, \pi_T, \ldots,)$ with equality achieved by the arg max policy. Proof of existence of such a solution is more intricate, and is shown by using relation with discounted model through Laurent series expansion (refer to Section 9.1.3 of [**?**]).
Now, using the equation for first step policy as π_1 we have:

$$\rho^* \mathbf{e} \geq R_{\pi_1} + (P_{\pi_1} - I)h^*$$

Using the equation for π_2, and multiplying by P_{π_1} on both sides

$$\rho^* \mathbf{e} \geq P_{\pi_1} R_{\pi_2} + P_{\pi_1}(P_{\pi_2} - I)h^*$$

Similarly, for any $t = 1, 2, \ldots,$ we can get

$$\rho^* \mathbf{e} \geq P_{\pi_1} P_{\pi_2} \cdots P_{\pi_{t-1}} R_{\pi_t} + P_{\pi_1} P_{\pi_2} \cdots P_{\pi_{t-1}}(P_{\pi_t} - I)h^*$$

On adding above equations for for $t = 1, \ldots, T$, the first term on the right hand side adds to value of policy in T rounds. The second term reduces to $(\prod_{i=1}^{T} P_{\pi_i} - I)h^*$. Therefore,

$$T\rho^* \mathbf{e} \geq V_T^\pi + (\prod_{i=1}^{T} P_{\pi_i} - I)h^*$$

Bellman Optimality Equations for average reward case

Assuming communicating MDP, gain and bias ρ, h of optimal policy satisfies the following equations:

$$\rho + h(s) = \max_a R(s,a) + \sum_{s' \in \mathcal{S}} P(s,a,s')h(s'), \forall s$$

Also, for any feasible solution (ρ, h) to the above equations, we can get an optimal policy π^* defined as

$$\pi^*(s) \in \arg\max_a R(s,a) + \sum_{s'} P(s,a,s')h(s'),$$

with $\rho = \rho^{\pi^*}$ and $h = h^{\pi^*} + c\mathbf{e}$ for some constant c.

- Note that to compute the optimal policy we need to just know the bias vector h that satisfies Bellman equations.

Dividing by T and taking limit $T \to \infty$:

$$\rho^* \mathbf{e} \geq \lim_{T \to \infty} \frac{1}{T} V_T^\pi + \frac{1}{T}(\prod_{i=1}^{T} P_{\pi_i} - I)h^*$$

we get

$$\rho^* \mathbf{e} \geq \rho^\pi$$

This proves the first part of the lemma.

Now, for the arg max policy π^*, we have

$$\rho^* \mathbf{e} = R^{\pi^*} + (P^{\pi^*} - I)h^*$$

Therefore, ρ^*, h^* satisfy the Bellman evaluation equations for policy π^*, therefore it follows from the previous subsection that $\rho^* = \rho^{\pi^*}$, and $h^* = h^{\pi^*} + ce$

Solving Bellman equations: Linear Program

$$\begin{aligned} \min_{\rho \in R, \mathbf{h} \in \mathbb{R}^S} \quad & \rho \\ \text{subject to} \quad & \rho \geq R(s,a) + \textstyle\sum_{s'} P(s,a,s') h_{s'} - h_s \quad \forall a, s \end{aligned}$$

Write the dual LP for better interpretation:

$$\begin{aligned} \max_{q} \quad & \textstyle\sum_{s,a} q(s,a) R(s,a) \\ & \textstyle\sum_{s,a} P(s,a,s') q(s,a) - \sum_{a} q(s',a) = 0 \quad \forall s' \\ & \textstyle\sum_{s,a} q(s,a) = 1 \\ & q(s,a) \geq 0 \qquad \forall s, a \end{aligned}$$

That is, find the stationary distribution $q(s,a)$ that maximizes the expected reward.

Solving Bellman equations: value iteration/policy iteration

- Same algorithm but with $\gamma = 1$.
- Instead of estimating the value vector, we are updating and estimating bias.
- linear convergence with rate γ is not guaranteed since $\gamma = 1$. The convergence rate depends on the properties of the transition matrix.
- a sufficient condition for linear convergence is

$$\alpha := \max_{s,s',a,a'} \sum_{j \in S} \min\{P(s,a,j), P(s',a',j)\} > 0$$

then linear convergence with rate $1 - \alpha$. .

Solving Bellman equations: value iteration/policy iteration

- Same algorithm but with $\gamma = 1$.
- Instead of estimating the value vector, we are updating and estimating bias.
- linear convergence with rate γ is not guaranteed since $\gamma = 1$. The convergence rate depends on the properties of the transition matrix.
- a sufficient condition for linear convergence is

$$\alpha := \max_{s,s',a,a'} \sum_{j \in S} \min\{P(s,a,j), P(s',a',j)\} > 0$$

then linear convergence with rate $1 - \alpha$. .

i.e., on taking two different actions in any two different states, there is a positive probability to reach an identical state. This condition ensures that the Bellman operator in this case: is still a contraction - although in terms of span, i.e., $span(Lv - Lu) \leq (1 - \alpha) span(v - u)$. For more details, refer to Section 8.5.2 in Puterman:1994

Contents I

Reinforcement Learning algorithms

RL == MDP + unknown model
== Value/Policy iteration + sampling
OR
Direct function optimization from samples

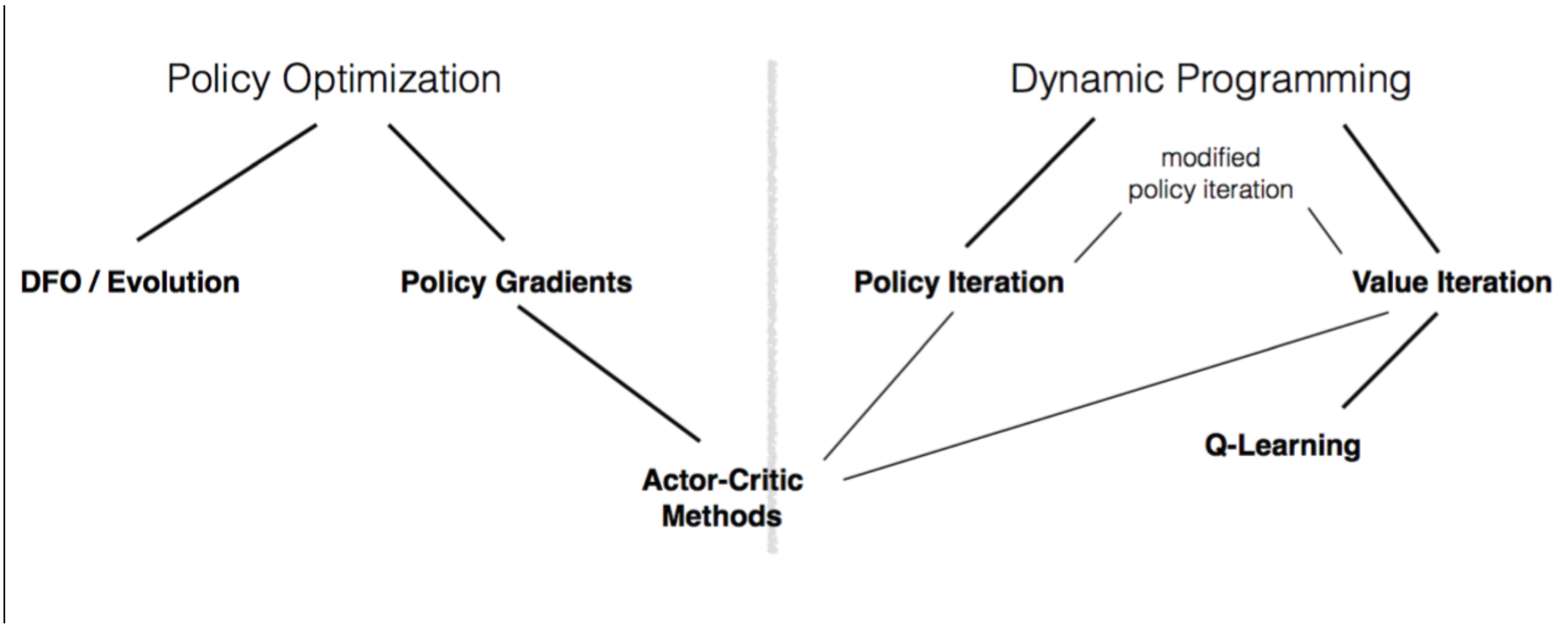


Figure: Algorithms for RL (Drawing taken from Pieter Abbeel's slides)