7.26

$\boxed{\text{Review}}$

* Markov decision process $MDP = \{\mathcal{S}, \mathcal{A}, P(s'|s,a), R(s)\}$

  state, actions, transition probs, rewards

* Policy $\pi: \mathcal{S} \to \mathcal{A}$ maps states into actions
* Value functions $V^\pi(s) = E^\pi[\sum_{t=0}^\infty \gamma^t R(s) \mid s_0 = s]$ (state)

  $Q^\pi(s,a) = E^\pi[\sum_{t=0}^\infty \gamma^t R(s) \mid s_0 = s, a_0 = a]$ (action)

  $0 \le \gamma < 1$ discount factor

* Bellman equation

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

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

* Policy evaluation — how to compute $V^\pi(s)$?

  Solve linear equations

  Takes $O(n^3)$ for MDP with n states

HW9: what to do if n is very large ?

* Policy improvement

  greedy policy $\pi'(s) = \arg\max_a Q^\pi(s,a)$

  Thm: $V^{\pi'}(s) \ge V^\pi(s)$ for all states s.

Today:

- two algorithms to compute π^*
- learning w/o model of MDP
- extensions

Policy Iteration

— How to compute π^*?

Algorithm:

(1) initialize policy at random

(2) repeat until convergence $O(n^3)$

— compute $V^\pi(s)$ and $Q^\pi(s,a)$ for current policy

— derive greedy policy $\pi(s) = \underset{a}{\mathrm{argmax}}\, Q^\pi(s,a)$

$$\pi_0 \xrightarrow{\text{evaluate}} \begin{matrix} V^{\pi_0}(s) \\ Q^{\pi_0}(s,a) \end{matrix} \xrightarrow{\text{improve}} \pi_1 \xrightarrow{\text{evaluate}} \cdots$$

* Is this guaranteed to converge?
  - Cannot cycle because $V^{\pi'}(s) \geq V^\pi(s)$ for all states s.
  - Cannot go on forever because # policies is finite: $|A|^{|S|}$
  - Policy can't be indefinitely improved.

  Typically converges in far fewer steps than $|A|^{|S|}$

* Does it always converge to an optimal policy π^*? Yes.
* Thm:

  suppose $\pi'(s) = \pi(s)$ for all states s, or rather that $V^{\pi'}(s) = V^\pi(s)$

  Then $V^\pi(s) = V^*(s)$

  Note: optimal value function is unique, even if there are many optimal policies

* Proof strategy

(1) Derive "Bellman optimality equation" satisfied by $V^{\pi}(s)$ when $V^{\pi}(s) = V^{\pi'}(s)$

(2) Show that $V^{\pi}(s) \geq V^{\tilde{\pi}}(s)$ for all policies $\tilde{\pi}$ and states s of MDP.

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

Step 1: From Bellman equation for π'

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

By assumption: $V^{\pi'}(s) = V^{\pi}(s)$

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

By assumption, π' is greedy with respect to $V^{\pi}(s)$

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

"Bellman optimal equation"

(set of n nonlinear eqs for $V^{\pi}(s)$ where $s = 1, 2, \dots, n$)

$\hookrightarrow$ because max operation is nonlinear.

($\therefore$ different than Bellman equation)

Step 2:

Iterate right side:

$$V^{\pi}(s) = R(s) + \gamma \max_a \sum_{s'} P(s'|s, a) \Big[\overbrace{R(s') + \gamma \max_{a'} \sum_{s''} P(s''|s, a) V^{\pi}(s'')}^{V^{\pi}(s')} \Big]$$

Iterate again and again ...

$\boxed{B}$

Now show that this iterated expression (taken out to infinity) implies optimality

Let $\tilde{\pi}(s)$ be any other (non-optimal) policy

From Bellman eqn:

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

greedy

$$\leq R(s) + \gamma \max_a \sum_{s'} P(s'|s, a) V^{\tilde{\pi}}(s')$$

iterate

$\boxed{A}$

Iterate

$$\leq R(s) + \gamma \max_a \sum_{s'} P(s'|s,a)\Big[R(s') + \gamma \max_{a'} \sum_{s''} P(s''|s',a) V^{\tilde{\pi}}(s'')\Big]$$

Consider upper bound $\boxed{A}$ on $V^{\tilde{\pi}}(s)$ from iterating this inequality t times, and compare to equality after t iterations for $\boxed{B}$

As $t \to \infty$

$$V^{\tilde{\pi}}(s) \leq \lim_{t\to\infty} \boxed{A} = \lim_{t\to\infty} \boxed{B} = V^{\pi}(s)$$

Thus for all policies $\hat{\pi}(s)$ and states s,

$$V^{\pi}(s) \geq V^{\hat{\pi}}(s)$$

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

To compute π^*:

$$\pi^*(s) = \arg\max_a Q^*(s,a) = \arg\max_a \Big[R(s) + \gamma \sum_{s'} P(s'|s,a) V^*(s')\Big]$$

Pros / Cons of policy iteration:

(+) converges very quickly

(−) each step requires policy evaluation $O(n^3)$

Transition

What to do if n is so large that policy evaluation is prohibitive?

Idea: look for approximate solution in $O(n^2)$, and refine solution as resources permit

$\boxed{\text{Value iteration}}$ — another (less direct) way to compute π^*

* How to compute $V^*(s)$ directly?

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

$$= \max_a \Big[R(s) + \gamma \sum_{s'} P(s'|s,a) V^*(s')\Big]$$

Bellman optimality eqn

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

n nonlinear eqns for $V^*(s)$, $s = 1, 2, \ldots, n$

$\boxed{\text{Value Iteration}}$

Bellman optimality eqn:

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

* n nonlinear eqns, for n unknowns $V^*(s)$, $s = 1, 2, \cdots, n$

How to solve?

* Algorithm (HW 9.2)

(1) initialize $V_0(s) = 0$ for all s (at 0^{th} iteration)

(2) iterate

every iteration is $O(n^2)$

$$\begin{cases} V_{k+1}(s) = R(s) + \gamma \max_a \left[\sum_{s'} P(s'|s,a)\, V_k(s') \right] \\ \quad \text{for all } s = 1, 2, \cdots, n \end{cases}$$

estimate at k^{th} iteration.

Note: this algorithm works directly on value functions; policies do not (seemingly) appear!

But incremental policies can be computed from

$$\pi_{k+1}(s) = \text{greedy}[V_k(s)] = \arg\max_a \sum_{s'} P(s'|s,a)\, V_k(s')$$

(3) suppose it converges:

$$\lim_{k \to \infty} V_k(s) = V^*(s)$$

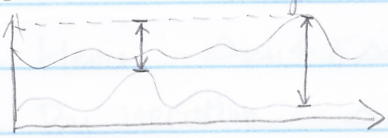
Then $\pi^*(s) = \arg\max_a Q^*(s,a)$

* Does this algorithm converge?

Clearly, $V^*(s)$ is a fixed point of iteration.

But are there other fixed points? No.

Does it always reach $V^*(s)$? Yes.



Lemma: for any functions $f(a)$ and $g(a)$

$$\left| \max_a f(a) - \max_a g(a) \right| \le \max_a |f(a) - g(a)|$$

Proof of lemma:

For all a: $f(a) - \max_{a'} g(a') \leq f(a) - g(a)$

Max over a: $\max_a [f(a) - \max_{a'} g(a')] \leq \max_a [f(a) - g(a)]$

$$\leq \max_a |f(a) - g(a)|$$

$\therefore$ $\boxed{\max_a f(a) - \max_a g(a) \leq \max_a |f(a) - g(a)|}$

By symmetry, exchange $g \leftrightarrow f$

$\therefore$ $\boxed{\max_a g(a) - \max_a f(a) \leq \max_a |g(a) - f(a)|}$

combining these gives the lemma.

Thm: value iteration converges

$$\lim_{k\to\infty} [V_k(s)] \to V^*(s) \text{ for all states } s$$

Proof: let $\Delta_k = \max_s |V_k(s) - V^*(s)|$ error of k^{th} iteration of algorithm

$$\Delta_{k+1} = \max_s |V_{k+1}(s) - V^*(s)|$$

$$= \max_s \Big| \underbrace{[R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V_k(s')]}_{\text{definition of } V_{k+1}(s)} - \underbrace{[R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V^*(s')]}_{\text{Bellman optimality eqn}} \Big|$$

$$\Delta_{k+1} = \gamma \max_s \Big| \max_a \underbrace{\sum_{s'} P(s'|s,a) V_k(s')}_{f(a)} - \max_a \underbrace{\sum_{s'} P(s'|s,a) V^*(s')}_{g(a)} \Big|$$

Apply lemma:

$$\Delta_{k+1} \leq \gamma \max_s \max_a \Big| \sum_{s'} P(s'|s,a) [V_k(s') - V^*(s')] \Big|$$

$$\leq \gamma \max_s \max_a \Big| \sum_{s'} P(s'|s,a) \max_{s''} |V_k(s'') - V^*(s'')|$$

$$= \gamma \max_s \max_a \max_{s''} |V_k(s') - V^*(s'')|$$

$$= \gamma \max_{s''} |V_k(s'') - V^*(s'')|$$

$$= \gamma \Delta_k$$

Hence: $\Delta_{k+1} \leq \gamma \Delta_k$

By iteration: $\Delta_k \leq \gamma^k \Delta_0 \xrightarrow{k\to\infty} 0$ for $\gamma < 1$

$\Delta_1 \leq \gamma \Delta_0$

$\Delta_2 \leq \gamma \Delta_1 \leq \gamma^2 \Delta_0$

Assume rewards are bounded:

$$\Delta_0 = \max_s |V_0(s) - V^*(s)| \quad \text{initial error}$$
$$= \max_s |V^*(s)|$$
$$\leq \max_s |R(s)| (1 + \gamma + \gamma^2 + \gamma^3 + \dots) \quad \text{maximum accumulated sum of discounted rewards.}$$
$$= \max_s |R(s)| \left(\frac{1}{1-\gamma}\right)$$

Finally:

$$\Delta_k \leq \left(\frac{\gamma^k}{1-\gamma}\right) \max_s |R(s)| \to 0 \quad \text{as } k \to \infty$$

Convergence rate depends on γ.

Suggests that more iterations are required as $\gamma \to 1$.

HW9.2, $\gamma = 0.9925$

$\boxed{\text{Reinforcement Learning}}$

* What if $P(s'|s,a)$ and $R(s)$ are not known?

Can we learn π^* or $V^*(s)$ or $Q^*(s,a)$ from experience?

(1) Model-based (indirect) approach

Explore world, estimate model

$\hat{P}(s'|s,a) \approx P(s'|s,a)$ compute $\hat{\pi}^*$ from $\hat{P}(s'|s,a)$

(e.g. ML estimation) hope $\hat{\pi}^* \approx \pi^*$

* Cons: to store $P(s'|s,a)$ is $O(n^2)$ for n states.

Only care about $\pi^*(s)$, $V^*(s)$ which are $O(n)$

Is it really necessary to build a model?

Pro: model useful for task transfer, where $P(s'|s,a)$ are same for many tasks, but only rewards $R(s)$ or discount factor γ changes.

(2) Direct approach:

learn $V^*(s)$, $\pi^*(s)$ w/o building model. How?

Simpler question — how to evaluate a policy w/o model? how to compute $V^\pi(s)$ w/o knowing $P(s'|s,\pi(s))$?

* Explore state space under policy π

$$
\begin{array}{llllll}
\text{actions} & & \pi(s_0) & & \pi(s_1) & & \pi(s_2) & \\
\text{states} & s_0 & \longrightarrow & s_1 & \longrightarrow & s_2 & \longrightarrow & \cdots \\
\text{rewards} & R(s_0) & & R(s_1) & & R(s_2) & &
\end{array}
$$

* Recall Bellman eqn

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

* Temporal difference prediction

Initialize $V_0(s)=0$ for all s (at time $t=0$).

Update $$\underbrace{V_{t+1}(s_t)}_{\text{new estimate of } V(s_t)} = \underbrace{V_t(s_t)}_{\text{previous estimate of } V(s_t)} + \alpha \underbrace{[\underbrace{R(s_t)+\gamma V_t(s_{t+1})}_{\text{simulated step of MDP}} - \underbrace{V_t(s_t)}_{\text{old estimate of } V(s_t)}]}_{\text{error signal}}$$

$\alpha \geq 0$ learning rate (decrease over time)

on average this will be very small if estimate is good.

TD learning

Asymptotically $\lim_{t\to\infty} V_t(s) \to V^\pi(s)$ under certain conditions.