Chapter 4: Dynamic programming

SB: Chap 4

4.1 Policy evaluation

- Goal: Compute v_π for given policy π
- Direct method: Solve linear equation

$$v_\pi = \rho_\pi + \gamma P_\pi v_\pi$$

$$\left(v_\pi\right) = \left(\rho_\pi\right) + \gamma \left(P_\pi\right)\left(v_\pi\right)$$

$$\Rightarrow \quad v_\pi = (I - \gamma P_\pi)^{-1} \rho_\pi$$

- Complexity: $O(|S|^3)$

- Iterative method: $v_0 \to v_1 \to \cdots \to v_\pi$

$$v_{k+1}(s) = E_\pi\left[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s\right]$$

$$v_{k+1} = \rho_\pi + \gamma P_\pi v_k$$

Bellman backup

matrix product

$$= T_\pi(v_k)$$

$$T_\pi(v) = \rho_\pi + \gamma P_\pi v$$

Bellman operator

- Initial value v_0 arbitrary, except $v_0(\text{terminal state}) = 0$
- v_π fixed point of T_π
- Convergence: $v_k \to v_\pi$ as $k \to \infty$
- T_π is a contraction
- Complexity: $|S|^2 \times M$

matrix product # iterations

- Not exact v_π for finite # iterations

4.2. Policy iteration

- Consider deterministic policies $\pi(s)$

V_π, q_π

- Choose $a = \pi(s)$ — on policy — $V_\pi(s) = q_\pi(s, \pi(s))$
- Choose $a \neq \pi(s)$ — off policy — $V_\pi(s) \neq q_\pi(s, a)$

- Policy improvement theorem: Let π and π' such that

$$q_\pi(s, \pi'(s)) \geq q_\pi(s, \pi(s)) = V_\pi(s)$$

(π': off policy; π: on policy)

for all $s \in \mathcal{S}$. Then π' is as good or better than π in the sense that

$$V_{\pi'}(s) \geq V_\pi(s)$$

($V_{\pi'}$: better value under π')

for all $s \in \mathcal{S}$.

- Greedy policy selection:

$$\pi'(s) = \arg\max_a q_\pi(s, a)$$

  - Equation for updating π
  - Fixed point: π_* for q_* (optimal policy)
  - Strict improvement unless optimal

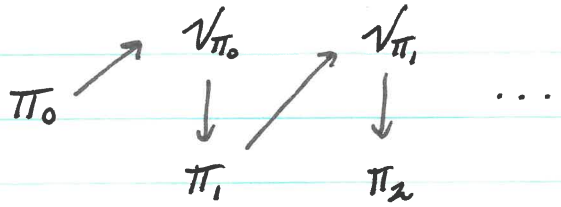
- ε-greedy policy selection:

$$\pi'(s) = \begin{cases} \arg\max_a q_\pi(s, a) & \text{Prob } 1-\varepsilon \\ u\{\mathcal{S}\} & \text{Prob } \varepsilon \end{cases}$$

($u\{\mathcal{S}\}$: exploration)

  - No strict improvement

· Policy iteration algorithm



Policy evaluation (Direct or iterative)

Policy improvement

· Convergence: $\pi_k \to \pi_*$, $v_k \to v_*$, $q_k \to q_*$ as $k \to \infty$

· Note: Policy evaluation for q_π

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

$$= \rho(s,a) + \gamma \sum_{s'} q_\pi(s',\pi(s'))\,p(s'|s,a) \quad \text{(on policy)}$$

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

$$\to \quad q_\pi = \rho + \gamma P_a v_\pi$$

· Note:

Planning	Learning
· MDP model known	· MDP model unknown
· Solve to get v_π	· Estimate v_π by experience/sampling
· Solve to get v_*, π_*	· Converge to v_*, π_*
v_π, v_*	q_π, q_*

4.3. Value iteration

· Bellman optimality equation:

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

$q_*(s,a)$

$$v_*(s) = \max_a \left[\rho(s,a) + \gamma (P_a v_*)(s)\right]$$

· Bellman optimality operator:

$$T_*(\cdot) = \max_a \left[\rho + \gamma (P_a \cdot)\right]$$

↑ $T_*(\cdot)$ ↑ $(P_a \cdot)$

· Value iteration algorithm:

$$v_0 \longrightarrow v_1 = T_*(v_0) \longrightarrow v_2 = T_*(v_1) \longrightarrow \cdots$$

$$v_{k+1}(s) = \max_a \mathbb{E}\left[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s, A_t = a\right]$$

↑ $k+1$ ↑ k

$$v_{k+1}(s) = \max_a \left[\rho(s,a) + \gamma \sum_{s'} v_k(s') p(s'|s,a)\right]$$

$$= \max_a \left[\rho(s,a) + \gamma (P_a v_k)(s)\right]$$

- One step of iterative policy evaluation
- Max defines "optimal" policy at ~~stage~~ k+1 (policy improvement)
- Greedy policy improvement (can also use ε-greedy)
- Convergence: $v_k \to v_*$ as $k \to \infty$

4.4. Action value iteration

time $t+1$

$$q_{k+1}(s,a) = E\left[R_{t+1} + \gamma \max_{a'} q_k(S_{t+1}, a') \mid S_t = s, A_t = a \right]$$

update

previous estimate, $v_k(S_{t+1})$

$$q_{k+1}(s,a) = \rho(s,a) + \gamma \max_{a'} \sum_{s'} q_k(s',a') \, p(s'|s,a)$$

- Max defines policy improvement at stage $k+1$
- Fixed point: q_* Bellman optimality equation
- Convergence: $q_k \to q_*$ as $k \to \infty$

- Policy improvement: $\pi_{k+1}(s) = \arg\max_a q_k(s,a)$

- Variant:

$$q_k(s,a) = E\left[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s, A_t = a \right]$$

$q_k \leftarrow v_k$

$$q_k(s,a) = \rho(s,a) + \sum_{s'} v_k(s') \, p(s'|s,a)$$

$$= \rho(s,a) + \gamma \left(P_a v_k \right)(s)$$

action a

- Max included in v_k