Dynamic Programming (DP)

- a collection of algorithms to compute optimal policies, given a model for the dynamics of the environment: $p(s', r | s, a)$

→ assume a finite MDP (episodic task)

- key idea: use value functions as a basis to search for policies

- recall Bellman eq's:

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

$$= \max_a \sum_{s', r} p(s', r | s, a)[r + \gamma v_*(s')]$$

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

$$= \sum_{s', r} p(s', r | s, a)[r + \gamma \max_{a'} q_*(s', a')]$$

DP algorithms: turn Bellman eqs into update rules for improved approximation to v_*, q_*

- Policy Evaluation:

compute value function $v_\pi(s)$ for a given fixed policy $\pi(a|s)$.

recall: $v_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s]$

$$= \sum_a \pi(a|s) \sum_{r,s'} p(s', r|s, a)[r + \gamma v_\pi(s')]$$

→ system of $|S|$ simultaneous linear equations for $|S|$ unknowns $v_\pi(s)$

→ alternative approach: solve iteratively, i.e. find a sequence $v_0, v_1, v_2, \ldots$ of value functions, s.t. $v_k(s) \xrightarrow{k\to\infty} v_\pi(s)$ $\forall s$.

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

$$= \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma v_k(s')]$$

has a fixed point at $v_k = v_\pi$

→ iterative policy evaluation:

- each iteration step updates $v_{k+1}(s)$ $\forall s \in S$ in-place

→ algorithm: iterative policy evaluation for estimating $V \approx v_\pi$

Iterative Policy Evaluation, for estimating $V \approx v_\pi$

Input π, the policy to be evaluated
Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop: (iterations)
  $\Delta \leftarrow 0$
  Loop for each $s \in \mathcal{S}$: (states)
    $v \leftarrow V(s)$
    $V(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)\big[r + \gamma V(s')\big]$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

HW: do Exercise 4.3

Policy Improvement

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

$$= \sum_{s',r} p(s', r|s, a)\big[r + \gamma V_\pi(s')\big]$$

Policy improvement theorem:
Let π, π' be deterministic policies, s.t.

$$q_\pi(s, \pi'(s)) \geq V_\pi(s) \quad \forall s \in S \quad (*)$$

Then: $V_{\pi'}(s) \geq V_\pi(s) \quad \forall s \in S$, i.e.
π' is as good as, or better than, π.

$$\pi'(a|s) \overset{\text{deterministic}}{=} \begin{cases} 1, & \text{for } a = \bar{a} \\ 0, & \text{otherwise} \end{cases} \iff \bar{a} = \pi'(s)$$

Proof:

$$v_\pi(s) \overset{(*)}{\leq} q_\pi(s, \pi'(s))$$

$$\overset{(def)}{=} \mathbb{E}\left[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = \pi'(s)\right]$$

π' determ.

$$= \sum_a \pi'(a|s) \sum_{r,s'} p(s', r \mid s, a)\left[r + \gamma v_\pi(s')\right]$$

$$= \mathbb{E}_{\pi'}\left[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s\right]$$

$$\overset{(*)}{\leq} \mathbb{E}_{\pi'}\left[R_{t+1} + \gamma q_\pi(S_{t+1}, \pi'(S_{t+1})) \mid S_t = s\right]$$

$$= \mathbb{E}_{\pi'}\left[R_{t+1} + \gamma \mathbb{E}\left[R_{t+2} + \gamma v_\pi(S_{t+2}) \mid S_{t+1}, A_{t+1} = \pi'(S_{t+1})\right] \mid S_t = s\right]$$

$$= \mathbb{E}_{\pi'}\left[R_{t+1} + \gamma \mathbb{E}_{\pi'}\left[R_{t+2} + \gamma v_\pi(S_{t+2}) \mid S_{t+1}\right] \mid S_t = s\right]$$

π' determ.

$$= \mathbb{E}_{\pi'}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 v_\pi(S_{t+2}) \mid S_t = s\right]$$

$$\leq \dots \leq$$

$$\leq \mathbb{E}_{\pi'}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \dots \mid S_t = s\right]$$

$$= v_{\pi'}(s)$$

→ given policy π, consider new greedy policy:

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

$$= \arg\max_a \mathbb{E}[R_{t+1} + \gamma V_\pi(S_{t+1}) \mid S_t = s, A_t = a]$$

$$= \arg\max_a \sum_{s',r} p(s',r \mid s,a)\,[r + \gamma V_\pi(s')]$$

- by construction, π' meets the requirements of the policy improvement theorem

$$\Rightarrow \pi' \geq \pi$$

→ suppose $\pi' = \pi$

then $V_{\pi'} = V_\pi$, and from def. of π' we have

$$V_{\pi'}(s) = \max_a \mathbb{E}[R_{t+1} + \gamma V_{\pi'}(S_{t+1}) \mid S_t = s, A_t = a]$$

$$= \max_a \sum_{s',r} p(s',r \mid s,a)\,[r + \gamma V_{\pi'}(s')]$$

note: policy improvement thm. holds also for stochastic policies:

if $\sum_a \pi'(a \mid s)\, q_\pi(s,a) \geq V_\pi(s) \quad \forall s$

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

Policy Iteration

$\pi_0 \xrightarrow{E} V_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} V_{\pi_1} \xrightarrow{I} \dots \pi_* \xrightarrow{E} V_{\pi_*}$

iterate between policy evaluation (E) & policy improvement (I)

. each operation E, I itself is iterative

Policy Iteration (using iterative policy evaluation) for estimating $\pi \approx \pi_*$

1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Loop:
   $\Delta \leftarrow 0$
   Loop for each $s \in \mathcal{S}$:
   $v \leftarrow V(s)$
   $V(s) \leftarrow \sum_{s',r} p(s', r \mid s, \pi(s)) \big[r + \gamma V(s')\big]$
   $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
   old-action $\leftarrow \pi(s)$
   $\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r \mid s, a) \big[r + \gamma V(s')\big]$
   If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

Value Iteration

→ do we need to run each E step until convergence? → NO!

- combine E & I steps:

$$v_{k+1}(s) = \max_a \mathbb{E}\left[R_{t+1} + \gamma V_k(S_{t+1}) \mid S_t = s, A_t = a\right]$$
$$= \max_a \sum_{s',r} p(s', r \mid s, a)\left[r + \gamma V_k(s')\right]$$

- turn Bellman eq. into update rule itself

Value Iteration, for estimating $\pi \approx \pi_*$

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

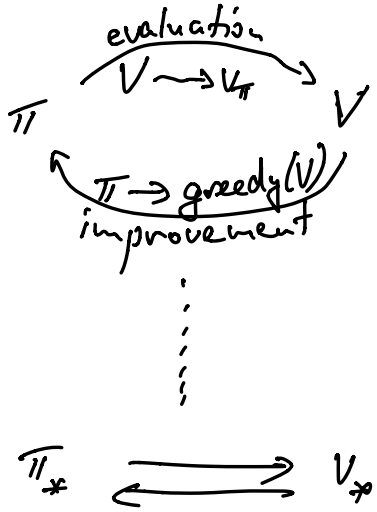
Loop:
| $\Delta \leftarrow 0$
| Loop for each $s \in \mathcal{S}$:
| $v \leftarrow V(s)$
| $V(s) \leftarrow \max_a \sum_{s',r} p(s', r \mid s, a)\left[r + \gamma V(s')\right]$
| $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

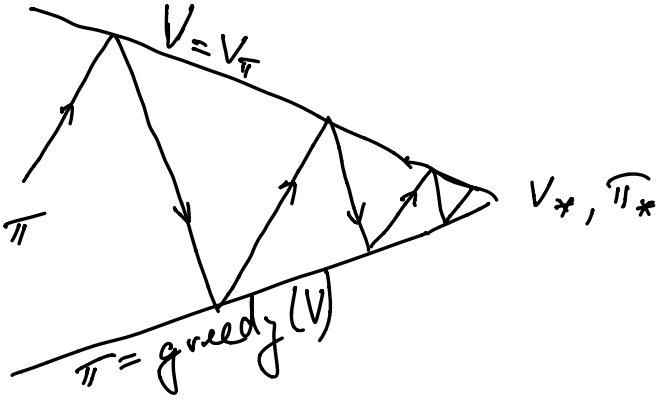
Output a deterministic policy, $\pi \approx \pi_*$, such that
$\pi(s) = \arg\max_a \sum_{s',r} p(s', r \mid s, a)\left[r + \gamma V(s')\right]$

Generalized Policy Iteration:

policy evaluation (E) & policy improvement (I)

- do we have to loop over all states until we switch from E → I, etc.?
- can iterate E arbitrary number of times





Efficiency & Drawbacks of DP

→ efficiency:
- DP guarantees to find optimal policy π_* in poly time in $|\mathcal{A}|$ (even though we may have $|\mathcal{A}|^{|\mathcal{S}|}$ i.e. exponentially many greedy policies)
- DP can handle larger state spaces than linear programming

→ drawbacks:
- requires knowledge of $p(s'|r|s,a)$
- involves operations over entire state space S ($|S|$ is exponentially large, or even continuous)