Dynamic Programming (DP)

compute optimal policies given a perfect model of the environment as a MDP.

Environment: finite MDP

$p(s', r \mid s, a)$

key idea: use of value functions to organize and structure the search for good policies.

Bellman Optimality Equations:

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

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

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

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

Bellman Equations $\xrightarrow{DP}$ Assignments

(update rules for improving approximations of the desired value function)

Policy Evaluation (Prediction)

iterative computation of the value functions for a given policy

$$
\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s] \\
&= \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\left[r + \gamma v_\pi(s')\right]
\end{aligned}
$$

iterative policy evaluation:

$$
\begin{aligned}
v_{k+1}(s) &\doteq \mathbb{E}[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s] \\
&= \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\left[r + \gamma v_k(s')\right]
\end{aligned}
$$

(expected update) on each state

"based on an expectation over all possible next states rather than on a sample next state"

$$\begin{cases} v \leftarrow V(s) \\ \boxed{V(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\left[r + \gamma V(s')\right]} \\ \Delta \leftarrow \max(\Delta, |v - V(s)|) \end{cases}$$

"sweep" through the state space

Policy Improvements (Theorem)

"We know how good it is to follow the current policy from s, that is $V_\pi(s)$, but would it be better or worse to change to the new policy?"

Computation of an improved policy given the value function for that policy

$$q_\pi(s,a) \doteq \mathbb{E}\left[R_{t+1} + \gamma V_\pi(S_{t+1}) \,\middle|\, S_t = s, A_t = a\right]$$

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

$$\boxed{\begin{array}{l} q_\pi(s, \pi'(s)) \geq V_\pi(s) \\ \\ V_{\pi'}(s) \geq V_\pi(s) \end{array}}$$

$$V_\pi(s) \leq q_\pi(s, \pi'(s))$$

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

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

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

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

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

$$\leq \mathbb{E}_{\pi'}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 V_\pi(S_{t+3}) \mid S_t = s]$$

$$\leq \mathbb{E}_{\pi'}[R_{t+1} + \gamma R_{t+2} + \cdots \mid S_t = s]$$

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

greedy policy π' :

$$\pi'(s) \doteq \underset{a}{\operatorname{argmax}}\, q_\pi(s,a)$$

$$= \underset{a}{\operatorname{argmax}}\, \mathbb{E}\left[R_{t+1} + \gamma V_\pi(S_{t+1}) \mid S_t = s, A_t = a\right]$$

$$= \underset{a}{\operatorname{argmax}} \sum_{s',r} p(s',r \mid s,a)\left[r + \gamma V_\pi(s')\right]$$

$$V_\pi = V_{\pi'}$$

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

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

The process of making a new policy that improve on an original policy, by making it greedy w.r.t. the value function of the original policy.

Policy Iteration

$$\pi_0 \xrightarrow{E} V_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} V_{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \cdots$$

$$\xrightarrow{I} \pi_* \xrightarrow{E} V_*$$

E : policy evaluation

I : policy improvement

Value Iteration

Policy iteration : involve policy evaluation

("require multiple sweeps through the state set")

$$\boxed{\text{"}\underline{\text{policy improvement}} + \underline{\text{truncated policy evaluation}}\text{"}}$$

"special case" of policy evaluation

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

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

$$\begin{cases} v \leftarrow V(s) \\ V(s) \leftarrow \max_a \sum_{s',r} p(s', r \mid s, a)[r + \gamma V(s')] \\ \Delta \leftarrow \max(\Delta, |v - V(s)|) \end{cases}$$

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

Asynchronous DP

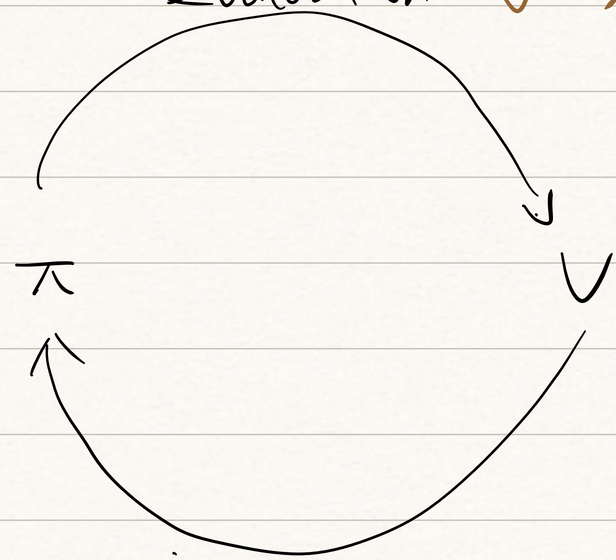
in-place iterative DP algorithm

Problem of policy iteration : <u>require sweeps of the state set</u>

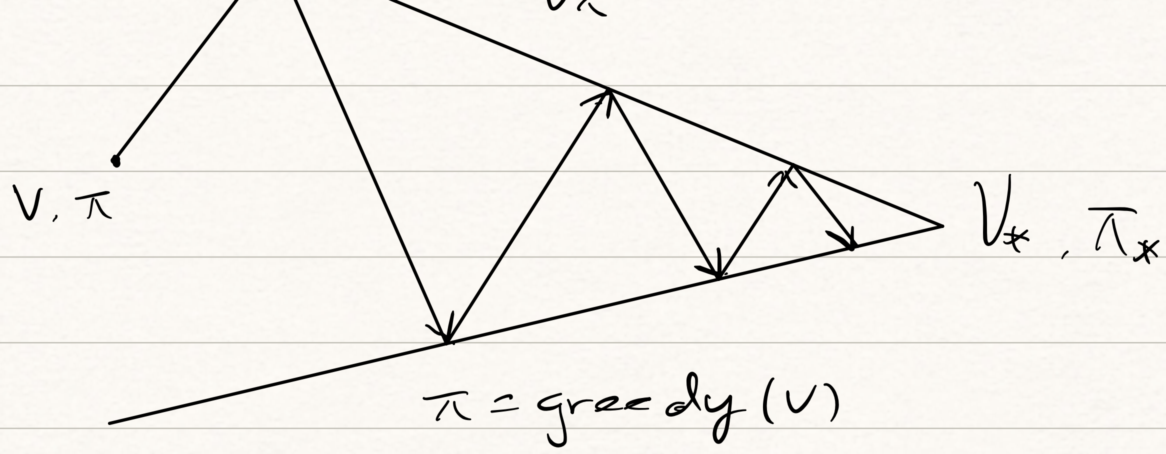
update the value of only one state, S_k, on each step k, using the value iteration update.

Generalized Policy Iteration (GPI)

evaluation $V \rightarrow V_\pi$



improvement $\pi \rightarrow \text{greedy}(V)$



Bootstrapping:

All of them update estimates of the values of states based on estimates of the values of successor states. That is, they <u>update estimates on the basis of other estimates</u>.