Dynamic Programming

CMPS 4660/6660: Reinforcement Learning

Dynamic Programming

- Contractions and Banach’s fixed point theorem
- Policy Evaluation
- Policy Optimization
    - Value Iteration
    - Policy Iteration

Norms

- V: a vector space over the reals
- $f : v \rightarrow \mathbb{R}_0^+$ is a norm if
  - If $f(v) = 0$, then $v = 0$
  - For $u, v \in V$, $f(u + v) \leq f(u) + f(v)$

Examples of Norms

- $V = (R^d, +, \cdot)$
  - l^p norms: for $p \geq 1$, $\|v\|_\mathrm{p} = \left(\sum_{i=1}^{d} |v_i|^p\right)^{1/p}$
  - l^∞ norms: $\|v\|_\infty = \max_{1 \leq i \leq d} |v_i|$
- $V = (B(X), +, \cdot)$
  - $B(X) = \left\{f: X \to \mathbb{R}: \sup_{x \in X} |f(x)| < +\infty\right\}$ -- the vector space of uniformly bounded real functions over domain X
  - $\|f\|_\infty = \sup_{x \in X} |f(x)|$

Convergence in norm

- $(V, \ \|\cdot\|)$: a normed vector space
- $\{v_n\}_{n\geq 0}$ is said to converge to v in norm if $\lim_{n\to\infty} \|v_n - v\| = 0$, denoted by $v_n \to_{\|\cdot\|} v$.
- In a d-dimensional vector space, this is equivalent to $v_{n,i} \to v_i$
  - $v_{n,i}$ - i-th component of v_n

Cauchy Sequence

- $(V, \ ||\cdot||)$: a normed vector space
- $\{v_n\}_{n\geq 0}$ is called a Cauchy sequence if $\lim_{n\to\infty} \sup_{m\geq n} ||v_n - v_m|| = 0$
- $(V, \ ||\cdot||)$ is called complete if every Cauchy sequence is convergent in norm
- A complete, normed vector space is called a Banach space
- Theorem: $(B(X), ||\cdot||_\infty)$ is a Banach space for non-empty X

Contraction Mappings

- $(V, \ \|\cdot\|)$: a normed vector space
- A mapping T: $V \rightarrow V$ is called L-Lipschitz if for any $u, v \in V$,

$$\|Tu - Tv\| \leq L\|u - v\|$$

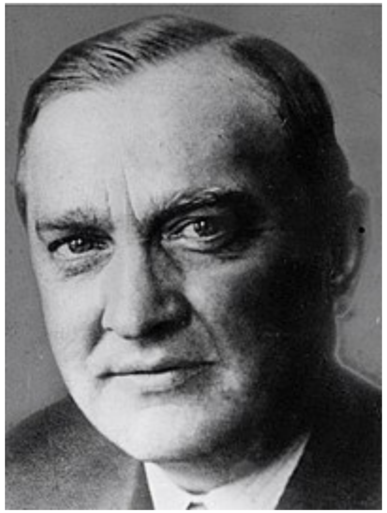
  - $L \leq 1$: T is called a non-expansion
  - $L < 1$: T called a L-contraction

Fixed Point

- $v \in V$ is called a fixed point of T if $Tv = v$
- $V = B(\mathcal{S})$: the vector space of bounded value functions over state space $\mathcal{S}$
- Bellman equation: $v_\pi = r^\pi + \gamma P^\pi v_\pi$
  - v_π is a fixed point $T^\pi: V \rightarrow V$, $T^\pi v = r + \gamma P v$
  - T^π is called the Bellman operator underlying π
- Bellman optimality equation: $v_*(s) = \max_a \left[r(s,a) + \gamma \sum_{s'} P_{ss'}(a) v_*(s') \right]$
  - v_* is a fixed point $T^*: V \rightarrow V$, $(T^* v)(s) = \max_a \left[r(s,a) + \gamma \sum_{s'} P_{ss'}(a) v(s') \right]$
  - T^* is called the Bellman optimality operator

Banach's fixed point theorem

- Let V be a Banach space and T a L-contraction mapping. Then
  - T has a unique fixed point v
  - For any $v_0 \in V$, if $v_{n+1} = Tv_n$, then
    - $\lim_{n \to \infty} \|v_n - v\| = 0$
    - $\|v_n - v\| \leq L^n \|v_0 - v\|$ (geometric convergence)



Stefan Banach
(1892-1945)

Proof of Banach's fixed point theorem

Pick $v_0 \in V$ and define $v_{n+1} = Tv_n$

Step 1: sequence $\{v_n\}$ is convergent

It suffices to show that $\{v_n\}$ is a Cauchy sequence (since V is a Banach space)

$$
\begin{aligned}
\|v_{n+k} - v_n\| &= \|Tv_{n-1+k} - Tv_{n-1}\| \\
&\leq L\|v_{n-1+k} - v_{n-1}\| \\
&\leq L^2\|v_{n-2+k} - v_{n-2}\| \\
&\vdots \\
&\leq L^n\|v_k - v_0\| \\
&\leq L^n(\|v_k\| + \|v_0\|)
\end{aligned}
$$

Since $\|v_k\| \leq \|v_k - v_{k-1}\| + \|v_{k-1} - v_{k-2}\| + \ldots + \|v_1 - v_0\|$

$$
\begin{aligned}
\|v_k\| &\leq (L^{k-1} + L^{k-2} + \cdots + 1)\|v_1 - v_0\| \\
&\leq \frac{1}{1-L}\|v_1 - v_0\| \quad \text{since } L < 1
\end{aligned}
$$

Thus, $\|v_{n+k} - v_n\| \leq L^n\left(\frac{1}{1-L}\|v_1 - v_0\| + \|v_0\|\right)$

and so, $\lim_{n\to\infty} \sup_{k\geq 0}\|v_{n+k} - v_n\| = 0$ since $L < 1$

Proof of Banach's fixed point theorem

Step 2: let v be the limit of $\{v_n\}$. We show that $Tv = v$.

Take limits of both sides in $v_{n+1} = Tv_n$.

The left side converges to v, and the right side converges to Tv_n (T is a contraction, hence it is continuous.) Thus, we must have $v = Tv$.

Step 3: uniqueness of the fixed point of T

Assume $Tv = v$ and $Tv' = v'$. Then, $\|v - v'\| = \|Tv - Tv'\| \leq L\|v - v'\|$. Since $L < 1$, we must have $\|v - v'\| = 0$, which implies $v = v'$.

Proof of Banach's fixed point theorem

Step 4: geometric convergence

$$\begin{aligned}\|v_n - v\| &= \|Tv_{n-1} - Tv\| \\ &\leq L\|v_{n-1} - v\| \\ &\leq L^2\|v_{n-2} - v\| \\ &\vdots \\ &\leq L^n\|v_0 - v\|\end{aligned}$$

Dynamic Programming

- Contractions and Banach's fixed point theorem
- Policy Evaluation
- Policy Optimization
  - Value Iteration
  - Policy Iteration

Prediction (Policy Evaluation)

- Bellman equation: $v_\pi = r^\pi + \gamma P^\pi v_\pi$
- $V = (B(\mathcal{S}), \|\cdot\|_\infty)$
- $T^\pi: V \to V$ where $T^\pi v = r^\pi + \gamma P^\pi v$

Fact 1: T^π is a γ-contraction with respect to $\|\cdot\|_\infty$ ⇒ v_π is the unique fixed point of the Bellman equation underlying π

Fact 2: T^π is monotone, i.e., if $u \le v$, then $T^\pi u \le T^\pi v$

⇒ If $v_0 \le T v_0$, then $v_0 \le v_1 \le v_2 \le v_3 \le \cdots$

If $v_0 \ge T v_0$, then $v_0 \ge v_1 \ge v_2 \ge v_3 \ge \cdots$

Prediction (Policy Evaluation)

T^{π} is a γ-contraction with respect to $\|\cdot\|_{\infty}$

Proof:

$$\|T^{\pi}u - T^{\pi}v\|_{\infty} = \sup_{s\in\mathcal{S}}\left|\left[r^{\pi}(s) + \gamma \textstyle\sum_{s'} P^{\pi}_{ss'}u(s')\right] - \left[r^{\pi}(s) + \gamma \textstyle\sum_{s'} P^{\pi}_{ss'}v(s')\right]\right|$$

$$= \gamma \sup_{s\in\mathcal{S}}\left|\textstyle\sum_{s'\in\mathcal{S}} P^{\pi}_{ss'}\left(u(s') - v(s')\right)\right|$$

$$\leq \gamma \sup_{s\in\mathcal{S}} \sum_{s'\in\mathcal{S}} P^{\pi}_{ss'}\left|\left(u(s') - v(s')\right)\right|$$

$$\leq \gamma \sup_{s\in\mathcal{S}} \sum_{s'\in\mathcal{S}} P^{\pi}_{ss'}\|u - v\|_{\infty} = \gamma\|u - v\|_{\infty}$$

Iterative policy evaluation

Input: π (policy to be evaluated), $\theta > 0$ (threshold)

Initialize $V(s)$ for $s \in \mathcal{S}^+$, arbitrarily except $V(s^*) = 0$

Loop:

$\Delta \leftarrow 0$

Loop for each $s \in \mathcal{S}$:

$$V'(s) \leftarrow \sum_a \pi(a|s)\left(r(s,a) + \gamma \sum_{s'} P_{ss'}(a)\, V(s')\right)$$

$$\boxed{\Delta \leftarrow \max(\Delta, |V'(s) - V(s)|)}$$

$V \leftarrow V'$

until $\Delta < \theta$

Each iteration updates the values of all states

To reduce complexity, precompute

$$r^\pi(s) = \sum_{a \in \mathcal{A}(s)} \pi(a|s) r(s,a)$$

$$P^\pi_{s,s'} = \sum_{a \in \mathcal{A}(s)} \pi(a|s)\, P_{ss'}(a)$$

In-place iterative policy evaluation

Input: π (policy to be evaluated), $\theta > 0$ (threshold)

Initialize $V(s)$ for $s \in \mathcal{S}^+$, arbitrarily except $V(s^*) = 0$

Loop:

$\Delta \leftarrow 0$

Loop for each $s \in \mathcal{S}$:

$v \leftarrow V(s)$

$V(s) \leftarrow \sum_a \pi(a|s) \left(r(s,a) + \gamma \sum_{s'} P_{ss'}(a) V(s') \right)$

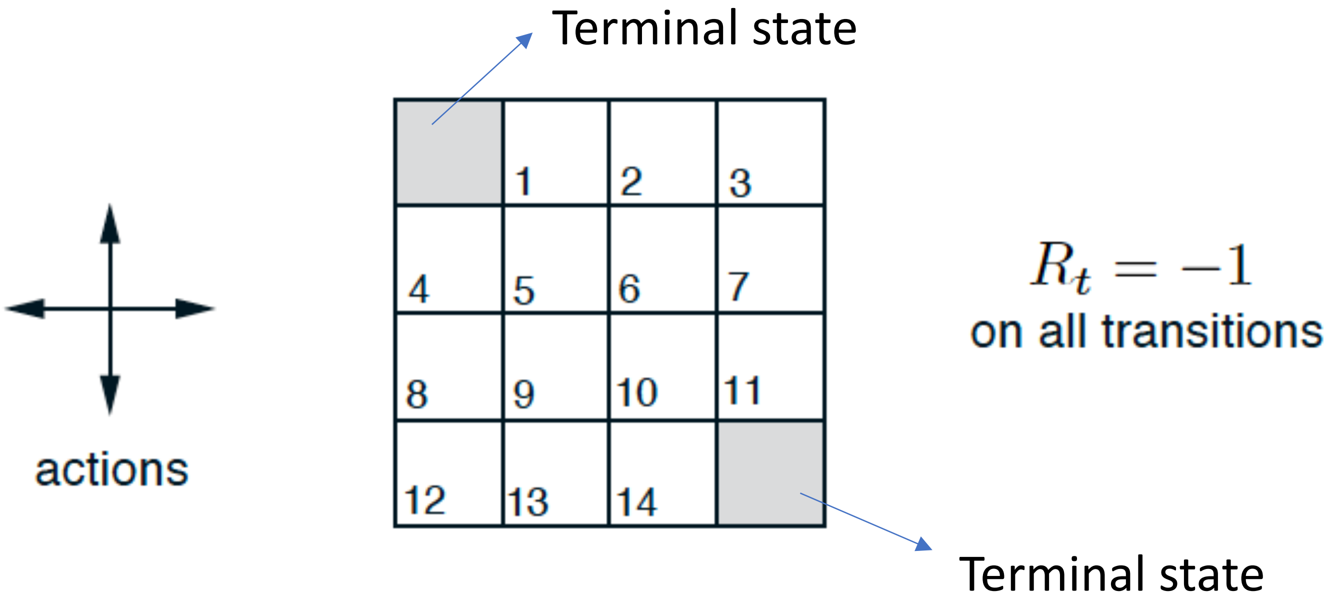
$\Delta \leftarrow \max(\Delta, |v - V(s)|)$

until $\Delta < \theta$

sweeps through the state space

usually converges faster

Example: Gridworld



$\mathcal{S} = \{1,2,\ldots,14\}$

$\mathcal{A} = \{\text{up, down, right, left}\}$

- Actions that would take the agent off the grid leave its location unchanged

Example: Gridworld

$\{v_k\}$ from iterative policy evaluation under equiprobable random policy

0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0

$k = 0$

0.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	0.0

$k = 1$

0.0	-1.7	-2.0	-2.0
-1.7	-2.0	-2.0	-2.0
-2.0	-2.0	-2.0	-1.7
-2.0	-2.0	-1.7	0.0

$k = 2$

0.0	-2.4	-2.9	-3.0
-2.4	-2.9	-3.0	-2.9
-2.9	-3.0	-2.9	-2.4
-3.0	-2.9	-2.4	0.0

$k = 3$

0.0	-6.1	-8.4	-9.0
-6.1	-7.7	-8.4	-8.4
-8.4	-8.4	-7.7	-6.1
-9.0	-8.4	-6.1	0.0

$k = 10$

0.0	-14.	-20.	-22.
-14.	-18.	-20.	-20.
-20.	-20.	-18.	-14.
-22.	-20.	-14.	0.0

$k = \infty$

Dynamic Programming

- Contractions and Banach's fixed point theorem
- Policy Evaluation
- Policy Optimization
    - Value Iteration
    - Policy Iteration

Control (Policy Optimization)

- Bellman optimality equation: $v_*(s) = \max_a \left[r(s,a) + \gamma \sum_{s'} P_{ss'}(a) v_*(s') \right]$
- $V = (B(\mathcal{S}), \|\cdot\|_\infty)$
- v_* is a fixed point of $T^*: V \to V$ where $(T^* v)(s) = \max_a \left[r(s,a) + \gamma \sum_{s'} P_{ss'}(a) v(s') \right]$

Fact 1: T^* is a γ-contraction with respect to $\|\cdot\|_\infty$

Fact 2: T^* is monotone, i.e., if $u \le v$, then $T^* u \le T^* v$

⟹ v_* is the unique solution to the Bellman optimality equation.

From Optimal Value to Optimal Policy

Theorem

Let π be the deterministic stationary policy such that

$$\pi(s) = \operatorname*{argmax}_{a \in \mathcal{A}(s)} \left[r(s,a) + \gamma \sum_{s'} P_{ss'}(a) v_*(s') \right], \forall s \in \mathcal{S}$$

Then $v_\pi = v_*$. Hence, π is optimal.

Proof: $T^\pi v_* = T^* v_* = v_* \Rightarrow v_\pi = v_*$

Value Iteration

Input: $\theta > 0$ (threshold)

Initialize $V(s)$ for $s \in \mathcal{S}^+$, arbitrarily except $V(s^*) = 0$

Loop:

$\Delta \leftarrow 0$

Loop for each $s \in \mathcal{S}$:

$v \leftarrow V(s)$

$$V(s) \leftarrow \max_{a \in \mathcal{A}(s)} \left(r(s,a) + \gamma \sum_{s'} P_{ss'}(a)\, V(s') \right)$$

$\Delta \leftarrow \max(\Delta, |v - V(s)|)$

until $\Delta < \theta$

Output the deterministic policy π such that

$$\pi(s) = \operatorname*{argmax}_{a \in \mathcal{A}(s)} \left(r(s,a) + \gamma \sum_{s'} P_{ss'}(a)\, V(s') \right)$$

Value Iteration

Theorem

Let v be a state-value function such that $|v(s) - v_*(s)| \leq \theta'$ for all $s \in S$, and π a greedy policy for v. Then for all $s \in S$,

$$|v_\pi(s) - v_*(s)| \leq \frac{2\gamma\theta'}{1-\gamma}$$

Proof: see Singh and Yee, "An Upper Bound on the Loss from Approximate Optimal-Value Functions", 1994.

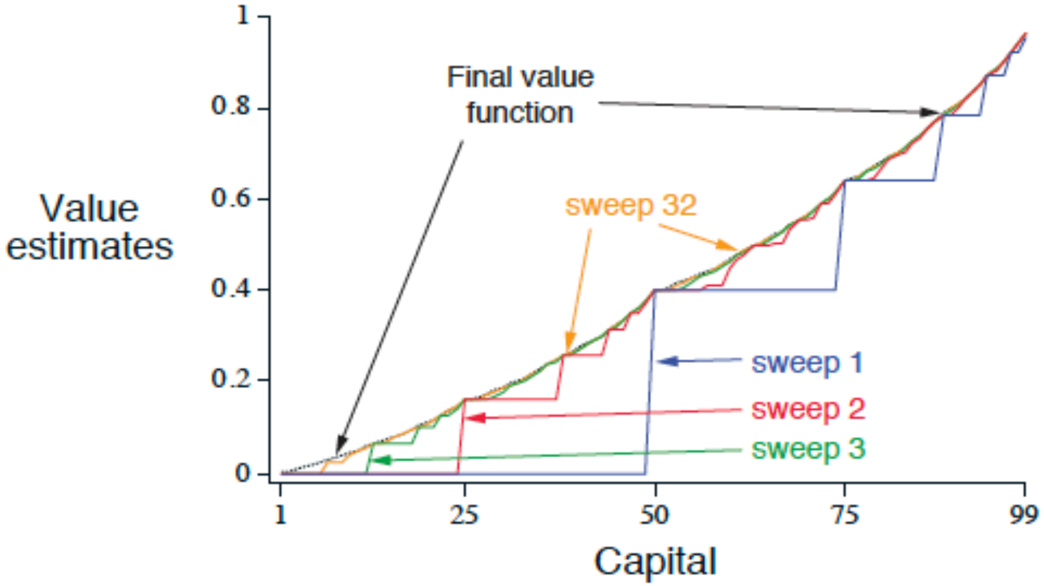
Gambler's Problem

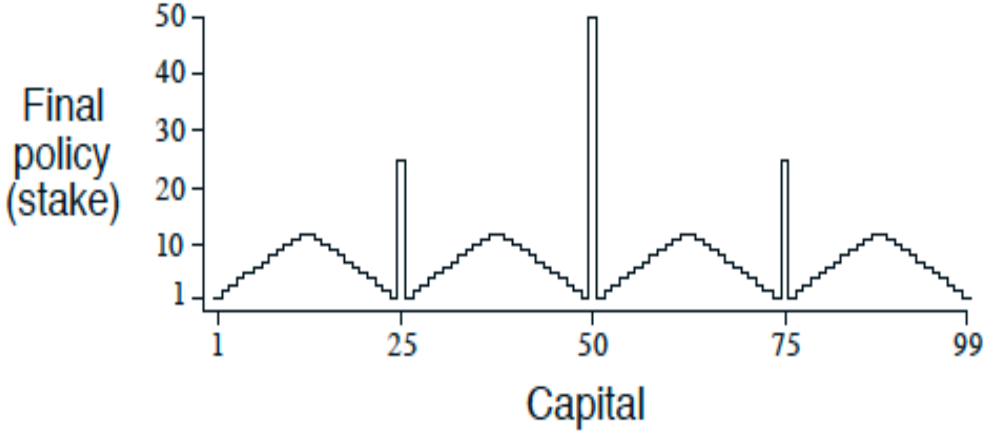
- A gambler has the opportunity to make bets on the outcomes of a sequence of coin flips.
  - If the coin comes up heads, he wins as many dollars as he has staked on that flip; if it is tails, he loses his stake.
  - The game ends when the gambler wins by reaching his goal of $100, or loses by running out of money.
- On each flip, the gambler must decide what portion of his capital to stake, in integer numbers of dollars.
- This problem can be formulated as an undiscounted, finite (non-deterministic) MDP.

Gambler's Problem

- The state is the gambler's capital $s = \{0,1,2,3 \dots, 100\}$
- The actions are stakes $a \in \{1, 2, \dots, \min(s, 100 - s)\}$
- The reward is zero on all transitions except those on which the gambler reaches his goal, when it is +1.
- The state-value function then gives the probability of winning from each state.
- A policy is a mapping from levels of capital to stakes
  - The optimal policy maximizes the probability of reaching the goal.
  - Let p_h denote the probability of the coin coming up heads.
  - If p_h is known, then the entire problem space is known and can be solved

Gambler's Problem





$p_h = 0.4$

Asynchronous Value Iteration

- Synchronous VI
  - operates at all states simultaneously in every iteration
  - may stuck at bad states
- Asynchronous VI
  - $V(s)$ is updated for a subset of states in one iteration
  - Iteration orders can be deterministic or randomized
  - convergence is still guaranteed as long as all the states are visited infinitely number of times
- Advantage of asynchronous VI
  - Faster convergence
  - Parallel and distributed computation
  - Simulation-based/online implementation (see SB Ch.8)

Dynamic Programming

- Contractions and Banach's fixed point theorem
- Policy Evaluation
- Policy Optimization
    - Value Iteration
    - Policy Iteration

Policy Improvement

Theorem

Let π_0 be a stationary policy and let π be the greedy policy with respect to v_{π_0}. That is, $\pi(s) = \text{argmax}_a \left[r(s,a) + \gamma \sum_{s'} P_{ss'}(a) v_{\pi_0}(s') \right], \forall s \in \mathcal{S}$. Then we have

(1) $v_\pi \geq v_{\pi_0}$

(2) If $T^* v_{\pi_0}(s) > v_{\pi_0}(s)$ for some $s \in \mathcal{S}$, then $v_\pi > v_{\pi_0}$

(3) If $T^* v_{\pi_0}(s) = v_{\pi_0}(s)$ for all $s \in \mathcal{S}$, then π_0 is an optimal policy

Proof: Exercise

$$\pi_0 \xrightarrow{E} v_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} v_{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \cdots \xrightarrow{I} \pi_* \xrightarrow{E} v_*$$

Policy Improvement

Theorem

Let π_0 be a stationary policy and let π be the greedy policy with respect to v_{π_0}. That is, $\pi(s) = \mathrm{argmax}_a\left[r(s,a) + \gamma \sum_{s'} P_{ss'}(a) v_{\pi_0}(s')\right], \forall s \in \mathcal{S}$. Then we have

(1) $v_\pi \geq v_{\pi_0}$

(2) If $T^* v_{\pi_0}(s) > v_{\pi_0}(s)$ for some $s \in \mathcal{S}$, then $v_\pi > v_{\pi_0}$

(3) If $T^* v_{\pi_0}(s) = v_{\pi_0}(s)$ for all $s \in \mathcal{S}$, then π_0 is an optimal policy

Proof: See [CS] Appendix A.2 Theorem 3

- Note that $\pi(s) = \mathrm{argmax}_a\left[r(s,a) + \gamma \sum_{s'} P_{ss'}(a) v_{\pi_0}(s')\right]$

$$\nRightarrow v_\pi = \max_a\left[r(s,a) + \gamma \sum_{s'} P_{ss'}(a) v_{\pi_0}(s')\right]$$

Policy Improvement

Proof of part (1)

$$\pi(s) = \text{argmax}_a \left[r(s,a) + \gamma \sum_{s'} P_{ss'}(a) v_{\pi_0}(s') \right], \forall s \in \mathcal{S}$$

$\Rightarrow T^{\pi} v_{\pi_0} \geq T^{\pi_0} v_{\pi_0} = v_{\pi_0}$

$\Rightarrow (T^{\pi})^2 v_{\pi_0} \geq T^{\pi} v_{\pi_0} \geq v_{\pi_0}$

…

$\Rightarrow (T^{\pi})^{\infty} v_{\pi_0} \geq v_{\pi_0}$

$\Rightarrow v_{\pi} \geq v_{\pi_0}$

Policy Iteration

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 $s \in \mathcal{S}$:

$v \leftarrow V(s)$

$V(s) \leftarrow \sum_a \pi(a|s)\left(r(s,a) + \gamma \sum_{s'} P_{ss'}(a)\, V(s')\right)$

$\Delta \leftarrow \max(\Delta, |v - V(s)|)$

until $\Delta < \theta$

3 Policy Improvement

policy-stable $\leftarrow$ true

For each $s \in \mathcal{S}$:

old-action $\leftarrow \pi(s)$

$\pi(s) \leftarrow \underset{a}{\operatorname{argmax}}[\, r(s,a) + \gamma \sum_{s'} P_{ss'}(a) V(s')]$

if *old-action* $\neq \pi(s)$, then *policy-stable*=false

If *policy-stable*, then stop and return V and π

else go to 2.

A subtle bug: policy continually switches between two or more policies that are equally good.

Policy Iteration for Action Values

1 Initialization

$Q(s,a) \in \mathbb{R}$ arbitrarily for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$

$\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2 Policy Evaluation

Loop:

$\Delta \leftarrow 0$

Loop for each $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$

$q \leftarrow Q(s,a)$

$Q(s,a) \leftarrow r(s,a) + \gamma \sum_{s'} P_{ss'}(a)\, Q(s', \pi(s'))$

$\Delta \leftarrow \max(\Delta, |q - Q(s,a)|)$

until $\Delta < \theta$

3 Policy Improvement

policy-stable ← true

For each $s \in \mathcal{S}$:

old-action ← $\pi(s)$

$\pi(s) \leftarrow \mathrm{argmax}_a\, Q(s,a)$

if *old-action* $\neq \pi(s)$, then *policy-stable*=false

If *policy-stable*, then stop and return Q and π

else go to 2.

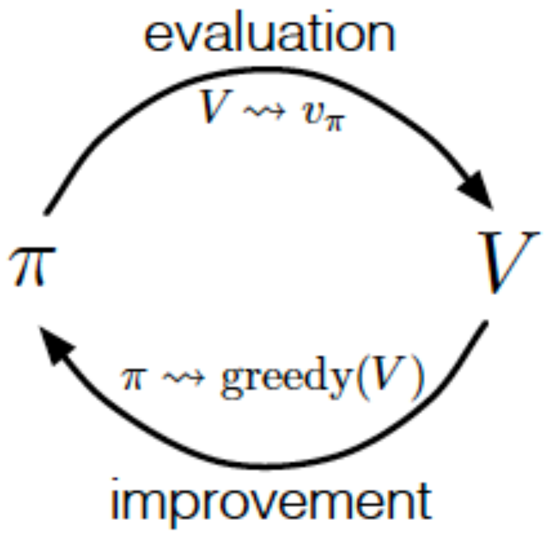
Policy Iteration

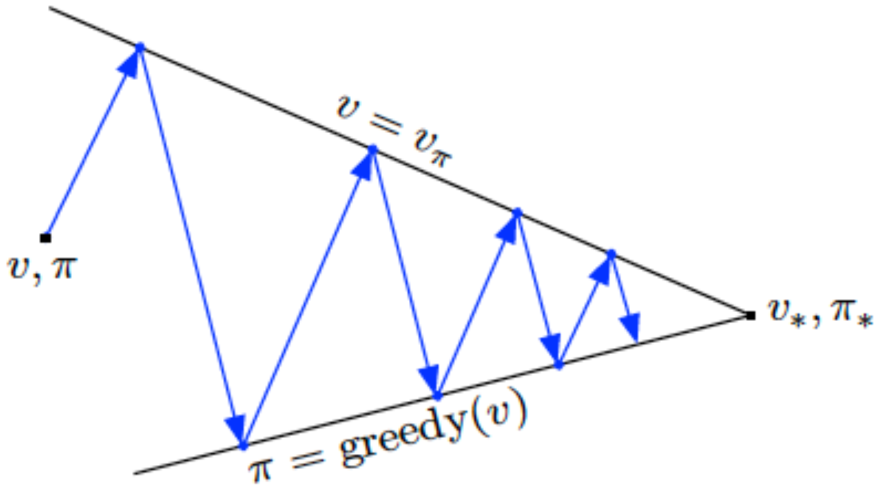
$$\pi_0 \xrightarrow{E} v_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} v_{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \cdots \xrightarrow{I} \pi_* \xrightarrow{E} v_*$$

- Each policy is a strict improvement over the previous one (unless it’s already optimal).
- A finite MDP only has a finite number of (deterministic stationary) policies => the process converges in a finite number of iterations.
- PI vs. VI
  - PI converges in fewer iterations than VI
  - But the computational cost of a single step in PI is much higher

Generalized Policy Iteration

- Generalized policy iteration (GPI) - letting policy-evaluation and policy-improvement processes interact, independent of the granularity and other details of the two processes.
- If both processes stabilize with respect to each other, the value function and policy must be optimal.





Linear Programming Method for MDP

- Policy Evaluation

$$v_\pi = r^\pi + \gamma P^\pi v_\pi \;\; \Rightarrow v_\pi = (I - \gamma P^\pi)^{-1} r^\pi$$

- Policy Optimization

$$\min_v \sum_{s \in \mathcal{S}} v(s)$$

subject to $v(s) \geq r(s,a) + \gamma \sum_{s'} P_{ss'}(a) v(s')\,, \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$

- The correctness of the LP is based on the following fact:

If $v \geq T^* v$, then $v \geq v_*$ (Exercise)

Partially Observable MDP

- A *Partially Observable Markov Decision Process* is a tuple $\langle X, \mathcal{A}, O, p, \gamma \rangle$
  - $X = \{1,2, \dots, d\}$ is a finite set of hidden states
  - $\mathcal{A}$ is a finite set of actions
  - O is a finite set of observations (including rewards)
  - $p(x', o|x, a) = \Pr\{X_t = x', O_t = o|X_{t-1} = x, A_{t-1} = a\}$
  - γ is a discount factor, $\gamma \in [0,1]$

Belief States

- A history H_t is a sequence of actions, observations and rewards,

$$H_t = O_0, A_0, O_1, A_1, \dots, O_{t-1}, A_{t-1}, O_t$$

- A *belief state* $S_t = \mathbf{s}_t \in \mathbb{R}^d$ is a probability distribution over states, conditioned on the history H_t

$$\mathbf{s}_t = (\Pr[X_t = i | H_t = h], \dots, \Pr[X_t = d | H_t = h])$$

POMDP to Belief MDP

- Belief update:

$$\mathbf{s}_{t+1}[i] = \frac{\sum_{j=1}^{d} \mathbf{s}_t[j] p(i, o | j, a)}{\sum_{j=1}^{d} \sum_{k=1}^{d} \mathbf{s}_t[j] p(k, o | j, a)}$$

- The belief state is Markov, i.e.,

$$\Pr(S_{t+1} = \mathbf{s}' \mid S_t = \mathbf{s}, A_t = a, S_{t-1} = \mathbf{s}_{t-1}, A_{t-1} = a_{t-1}, \dots, S_0 = \mathbf{s}_0)$$

$$= \Pr(S_{t+1} = \mathbf{s}' | S_t = \mathbf{s}, A_t = a)$$

- We thus obtain a continuous state MDP