

Reinforcement Learning: Policy/value iteration

AI/ML Teaching

Goals

- Policy Iteration
- Generalized Policy Iteration (GPI)
- Value Iteration
- Frozen lake

Policy Iteration: Iterative Policy Evaluation

- Problem: Evaluate a given policy π
- Solution: Iterative application of Bellman expectation backup
- $v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_\pi$: converges to v_π
- At each iteration $k + 1$
 - For all states $s \in \mathcal{S}$
 - Update $v_{k+1}(s)$ from $v_k(s')$
 - s' : successor state of s

$$v_{k+1}(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_k(s') \right)$$
$$\mathbf{v}^{k+1} = \mathbf{R}^\pi + \gamma \mathbf{P}^\pi \mathbf{v}^k$$

v_k for the
Random Policy

$k = 0$

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	0.0

$k = 3$

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

$k = 1$

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

$k = 10$

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

$k = 2$

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

$k = \infty$

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

Policy Iteration: How to Improve a Policy

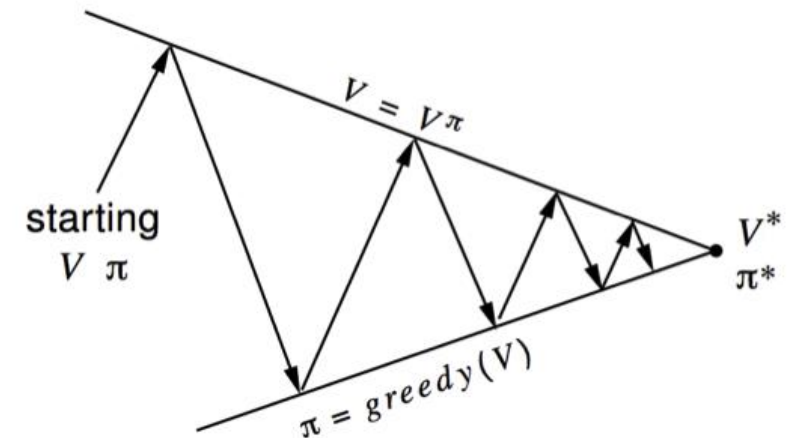
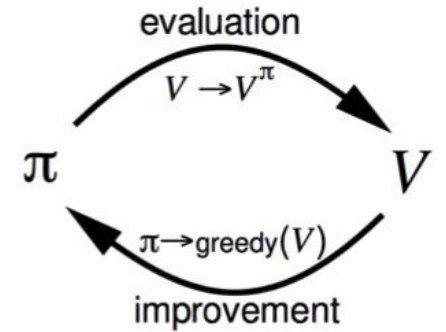
- Given a policy π
 - Policy evaluation: Evaluate the policy π

$$v_{\pi}(s) = \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \dots | S_t = s]$$

- Policy improvement: Improve the policy by greedy action w.r.t v_{π}

$$\pi' = \text{greedy}(v_{\pi})$$

- Policy iteration always converges to π^*



Can we stop before convergence?

\mathcal{V}_k for the Random Policy

















Greedy Policy
w.r.t. \mathcal{V}_k

 $k = 0$

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	0.0

















 $k = 3$

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

$$k = 1$$

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

 $k = 10$

















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

←	←	↙	
↑	↖	↙	↓
↑	↗	↘	↓
↘	→	→	

optimal
policy

















 $k = 2$

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

 $k = \infty$

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

Generalized Policy Iteration

- Policy evaluation: **Any** policy evaluation algorithm
 - Policy evaluation of $k = 1$ iteration \rightarrow value iteration (next slide)
 - Monte-Carlo/TD (next lecture)
- Policy improvement: **Any** policy improvement algorithm
 - Greedy policy improvement
 - ϵ -greedy improvement (next lecture)
- Contraction Mapping theorem
- Policy Improvement Theorem
- GLIE (greedy in the limit with infinite exploration)

Value Iteration

- Problem: Find optimal policy π
- Solution: Iterative application of Bellman optimality backup
- $v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_*$
- At each iteration $k + 1$
 - For all states $s \in \mathcal{S}$
 - Update $v_{k+1}(s)$ from $v_k(s')$
 - s' : successor state of s

Policy Iteration: Iterative Policy Evaluation

- Problem: Evaluate a given policy π
- Solution: Iterative application of Bellman expectation backup
- $v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_\pi$: converges to v_π

$$v_{k+1}(s) = \max_{a \in \mathcal{A}} \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_k(s') \right)$$
$$\mathbf{v}_{k+1} = \max_{a \in \mathcal{A}} \mathcal{R}^a + \gamma \mathcal{P}^a \mathbf{v}_k$$

Summary

- Prediction: value function v_π
- Control: optimal value function v_* and optimal policy π_*

Problem	Bellman Equation	Algorithm
Prediction	Bellman Expectation Equation	Iterative Policy Evaluation
Control	Bellman Expectation Equation + Greedy Policy Improvement	Policy Iteration
Control	Bellman Optimality Equation	Value Iteration

Practice

- Frozen Lake
 - Reward schedule
 - Reach goal: +1
 - Reach hole/frozen: 0
 - Action space
 - 0: left
 - 1: down
 - 2: right
 - 3: up



Reference

- David Silver, COMPM050/COMPGI13 Lecture Notes