Project Task

▷ What is optimality?

↦ It means choosing actions on which you will be able to maximise your rewards.

In a word it is about making the best decisions at every step.

Optimality

Policy optimality

→ A policy π^* will be optimal if it provides the highest expected return.

→ $\pi^* > \pi$

→ $\boxed{\pi^* = \arg\max_{\pi} V^{\pi}(s)}$

⊙ π → policy

⊙ V^{π} → value func^{c} under policy π

Value Func optimality

↦ $V^*(s)$ - The max possible expected return

↦ $Q^*(s,a)$ - The max expected return from taking action a in state s & acting optimally.

$$\pi^* = \begin{cases} 1, & \text{if } a = \arg\max_{a} Q^{\pi}(s) \\ 0, & \text{otherwise} \end{cases}$$

2) Expectation

↦ In a word we can say, it is a way to handle uncertainity

It gives us the average outcomes we can expect from a random event.

So,

$$E[X] = \sum_x x \cdot P(x)$$

$P(x) \rightarrow$ Probability distrn
$x \rightarrow$ Random variable

Bellman Expectation Eqn

$$V^{\pi}(s) = E_{\pi}\left[R_{t+1} + \gamma V^{\pi}(s_{t+1}) \mid s_t = s\right]$$

- $\gamma \rightarrow$ Distribution factor
- $E_{\pi}[\cdot] \rightarrow$ Expected value

Bellman Optimal Eqn

$$V^{*}(s) = \max_a E\left[R_{t+1} + \gamma V^{*}(s_{t+1}) \mid s_t = s, A_t = a\right]$$

Here for every state s, calculate the expected return for each possible action a, Then the action which will give maximum expected return is chosen.

Now,

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

$$q_\pi^*(s,a) = r(s,a) + \gamma \sum_{s'} p(s'|a,s) \underbrace{\sum_{a' \in A} \pi^*(a'|s') q_\pi^*(s',a')}_{\max_{a'} q^*(s',a')}$$

$$= r(s,a) + \gamma \sum_{s'} p(s'|a,s) \max_{a' \in A} q^*(s',a')$$

Exercise 3.25 | Page no → 67

Equation of $v^*(s,a)$ in terms of $q^*(s,a)$ is

$$\boxed{v^*(s,a) = \max q^*(s,a)}$$

$v^*(s,a)$ → Max^m return

$q^*(s,a)$ → Expected return

So, from here we can say that

$v^*(s)$ = max return form (s,a)

$= \max q^*(s,a)$

3.26

When starting in state s, taking action a & following optimal policy π^*,

$$q^*(s,a) = E\left[R_{t+1} + \gamma V^*(S_{t+1}) \,\middle|\, S_t = s, A_t = a\right]$$

The four-argument transition func

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

$$q^*(s,a) = \sum p(s', r \mid s, a) \cdot [r + \gamma V^*(s')]$$

3.27

① We know,

$$v^*(s) = \max q^*(s,a)$$

A/c to the optimal policy π^*

$\pi^*(s)$ = action a that gives $q^*(s,a) = v^*(s)$

$$\pi^*(s) = \arg\max q^*(s,a)$$

3.28

We know, $q^*(s,a) = \sum_{s'} p(s' \mid s,a)\left[r(s,a,s') + \gamma v^*(s')\right]$

&

$$\pi^*(s) = \arg\max q^*(s,a)$$

(iii)

So, we can say

$$\pi^*(s) = \arg\max_{s'} \sum p(s'|s,a)\left[r(s,a,s') + \gamma v^*(s')\right]$$

3.28 3.29

Bellman Expectⁿ for $v^\pi(s)$:-

$$v^\pi(s) = \sum \pi(a|s) \sum p(s',r|s,a)\left[r + \gamma v^\pi(s')\right]$$

Bellman Optimalit

3.29

3-argument transition probⁿ

$p(s'|s,a)$: the probⁿ of transitioning $s' \rightarrow s$ ↓ action

So,

(i) Bellman expectⁿ for v^π

$$v^\pi(s) = \sum \pi(a|s)\left[r(s,a) + \gamma \sum p(s'|s,a) v^\pi(s')\right]$$

(ii) Bellman Optimality eqn for v^*

$$v^*(s) = \max\left[r(s,a) + \gamma \sum_{s'} p(s'|s,a) v^*(s')\right]$$

(iii) ~~Bellman Expect~~

Bellman Expectation Eqn $q^\pi(s,a)$

$$q^\pi(s,a) = r(s,a) + \gamma \sum_{s'} p(s'|s,a) \sum_{a'} \pi(a'|s') q^\pi(s',a')$$

(iv) Bellman Optimality Eqn $q^*(s,a)$

$$q^*(s,a) = r(s,a) + \gamma \sum_{s'} p(s'|s,a) \max_{a'} q^*(s',a')$$

Part 2

OK, now I am ~~Ok~~ considering that, I am ~~curren~~ currently in a ~~jungle~~ cave and try to find ~~to find~~ out my ~~en~~ ancestral treasure. which is ~~under a tree~~ in a room

So in a ~~jungle~~ cave there ~~is~~ are lots of ~~trees~~ rooms and to reach ~~their~~ there, I can go in different directions. Some paths lead to dead end some to the treasure. Now I have a map of the ~~jungle~~ cave but I don't know which

direction or action lead to the best outcome overall.

So for me the challenges are to the value of the rooms of the cave that will decide my future rewards.

↦ Value Iteration

Here I start with guess. For each room, I consider every possible move. For each move I will calculate that if I take that action, I will get an immediate reward $R(s,a)$ & the discounted value of the room I land in next, $\gamma \sum_{s'} P(s'|s,a) V(s')$.

From here I will pick the action that gives me the highest combined value

So, V.I eqn is

$$V_{K+1}(s) = \max \left[R(s,a) + \gamma \sum_{s'} P(s'|s,a) V_K(s') \right]$$

Now I repeat this over & over, updating my estimate of how good each room is. Eventually, I will be able to find the true best values V^*

So,

$$\pi^*(s) = \arg\max \left[R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^*(s') \right]$$

Policy Iteration

(a) Evalution of my current Policy

So, I follow my current plan using

$$V^{\pi}(s) = \sum \pi(a|s) \left[R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^{\pi}(s') \right]$$

So, by solving this, I will get a reliable estimate of how good my current behavior is.

(b) Some Improvement

So, now I am thinking that it will be quite better if look at each room & check ~~that my~~ if my action will lead to the highest expected return right now or would that be different from my current plan.

So after all that stuff, I have just updated my policy by:

$$\pi_{new}(s) = \arg\max \left[R(s,a) + \gamma \sum P(s'|s,a) V^{\pi}(s') \right]$$

Why my ideas will work.

Here each update uses the best possible action at that step.

Evaluating my current policy gives you precise knowledge of how good it is. since there are only finitely many policies, so I eventually I will be able to reach the best one

---

→ GPI is basically Generalized Policy Iteration. So it is a framework how policy evaluation & Policy improvement work.

So now π be my policy & v^{π} be my value function.

Here in GPI I will evalute current Policy & improve my policy based on my current value estimates.

Value Iteration

A/c to Bellman optimality operator

$$V_{k+1}(s) = \max \left[R(s,a) + \gamma \sum P(s'/s,a)\, V_k(s') \right]$$

Now, let

$$V_{k+1} = T^* V_k$$

$T^* \rightarrow$ operator

- $\gamma \rightarrow$ Discounted factor
- $R(s,a) \rightarrow$ Reward func
- $P(s'|s,a) \rightarrow$ Transition func
- Finite action $\rightarrow A$.

So,

$$(T^*V)(s) = \max \left[R(s,a) + \gamma \sum_{s'} P(s'|s,a)\, V(s') \right]$$

Now, let V and W be two value func

We have to show,

$$\| T^*V - T^*W \|_\infty \leq \gamma \| V - W \|_\infty$$

Let's look at state s/-

$$|T^*V(s) - T^*W(s)| = \left| \max_a Q_V(s,a) - \max_a Q_W(s,a) \right| \leq \max_a \left| Q_V(s,a) - Q_W(s,a) \right|$$

where,

$$Q_V(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a)\, V(s')$$

$$Q_W(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a)\, W(s')$$

So :

$$|Q_V(s,a) - Q_W(s,a)| = \gamma \left| \sum_{s'} P(s'|s,a)(V(s') - W(s')) \right| \leq \gamma \|V - W\|_\infty$$

Hence

$$\|T^*V - T^*W\|_\infty \leq \gamma \|V - W\|_\infty$$

So, Here $\gamma < 1$,

Hence contraction is proved

Since here T^* is contracting on the complete metric space ~~(S, d)~~.

Now, here a unique fixed pt V^* such that $T^*V^* = V^*$ & for any initial value V_0 the iteration $V_{k+1} = T^*V_k \to V^*$

So,

$$\pi^* = \arg\max \left[R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^*(s') \right]$$

Policy It

So

Policy

Now

⊙

⊙

So,

Then,

And

⊙-

Policy Iteration

So let initial policy is π_0. Policy Evalution is V^{π_k} & Policy improvement is π_{k+1}

Now let,

- V^{π_k}: the value func under policy π_k
- π_{k+1}: the greedy improvement of π_k

So,

$$Q^{\pi}(s, \pi'(s)) \geq V^{\pi}(s)$$

Then,

$$V^{\pi'}(s) \geq V^{\pi}(s), \forall_s$$

And if inequality is strict in any state, then:

$$V^{\pi'}(s) > V^{\pi}(s) \text{ for at least one } s$$

This shows :- Improving the policy makes it better or equa in value

- The seqn V^{π_k} is Monotonic & Bounded

$$V^{\pi_{k+1}} \geq V^{\pi_k}$$

Because policy improvement is

greedy & value func^{n} are well-defined

Also,

$V^{\pi_k} \leq V^*$ for all k,

$\therefore$ The seqn is monotonic increasing and bounded above, so it must converge

There are only finitely many policies:

$$|\Pi| = |A|^{|S|}$$

$\pi_{k+1} = \pi_k$

$\therefore$ Policy iteration $\rightarrow \pi^*$ in finite steps

---

$$V_{k+1}(s) = \max_a \left[R(s,a) + \gamma \sum P(s'|s,a) V_k(s) \right]$$

- n = number of states
- $|A| = m$ = number of actions per state
- $\gamma \in (0,1)$ = discount factor

Time CX

For each state s, loop over:

- Each action a: m times
- Each next s': n transitions.

$\mapsto$ $O(n \cdot m \cdot n) = O(n^2 m)$

$\mapsto$ Total iterations needed for ε-optimal value.

$$O\left(\frac{\log(1/\varepsilon)}{1-\gamma}\right)$$

So overall time cx:

$$O\left(\frac{n^2 m}{1-\gamma} \log \frac{1}{\varepsilon}\right)$$

Policy Evaluation

$$V^\pi = R^\pi + \gamma P^\pi V^\pi$$

$$\Rightarrow V^\pi = (I - \gamma P^\pi)^{-1} R^\pi$$

- V^π : $O(n)$
- P^π : $O(n^2)$
- Tr^n model : $O(n^2 m)$

Total : $O(n^2 m)$

Total Iteration

The number of possible deterministic policies is

$$|\Pi| = m^n$$

But in practice, PI converges very fast

Overall time

$O(K \cdot (n^3 + nm))$ [usually small K]