동적계획법과 강화학습 강의 노트(2주차)

작성자 : 이도륜(2022323103)

작성일시 : 2022.09.19

Lecture 2. Markov Decision Process(review)

1. Markov Decision Process

- S,A,R,T 로 모델을 정의할 수 있다.

  (S : States, A : Actions, R : Rewards, T: Transition Probability matrix)

- 목적 : State → Action optimal 찾는게 MDP의 목적이다.

- Bellman equation으로 optimal state value, optimal state action value를 찾고

  value/policy iteration으로 optimal policy를 찾는다.

- 문제 모델링이 중요하다.확률 과정(Stochastic process)은 시간의 진행에 대해 확률적인 변화를 가지는 구조를 의미함.

2. 다양한 문제 모델링 형태

2-1. 예1) 짚신, 우산 문제

- States = {S(sunny), R(rain)}

- Actions = {집신, 우산}

- R(s,a) ※Rewards는 State와 Action의 matrix이다.

	짚	우
S	5	−2
R	−1	10

- T = State size X state size X Action size

짚	S	R
S		
R		

우	S	R
S		
R		

2-2. 예2) 어부가 연어 잡는 문제

- States = {Empty, Low, med, high}

- Actions = {No Fishing, Fishing}

- R(s,a) ※Real data를 참고해야 좋은 모델

- T = State size X state size X Action size

3. Value Function

- Value를 계산하는 이유 : 평균적으로 얼마의 보상을 받을지 추정할 수 있다.

- Value Function 수식 : 현 State에서 Return의 평균으로 정의되어 있다.

(State-value function)

$$v(s) = \mathbb{E}[G_t \mid S_t = s]$$

- Return은 현시점에서 미래의 리워드를 합산한 형태이며, 이때 미래의 리워드는 패널티를 곱하여 덧셈하도록 되어 있다.

The *return* G_t is the total discounted reward from time-step t.

$$G_t = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- Action Value function : 현 State에서 Action이 주어졌을 때 Value function

$$q(s,a) = \mathbb{E}[G_t \mid S_t = s, A_t = a]$$

- argmax Q(s,a)로 optimal action을 구한다.

4. Policy

- given state에서 어떤 Action을 할지 결정

- Policy의 수는 Action의 State 승수만큼 가능

$$action^{state}$$

- 예) State 2, Action 2

		S	R
π^1	짚 only	짚	짚
π^2	우 only	우	우
π^3	똑똑이	짚	우
π^4	청개구리	우	짚

- Policy가 정해지면, MDP를 MRP로 바꿔서 Bellman eq.로 풀 수 있다.

Lecture 3. Value Evaluation and Policy Iteration

1. Bellman (expectation/Optimality) Equation

- Bellman expectation Equation 은 State 수만큼 Value Function이 있으므로,

inverse로 Direct로 풀수 있고, 복잡할 시엔 iterative한 방법으로 접근 가능하다.

- **Bellman Expectation Equation**
  - **A recursion for expected rewards**

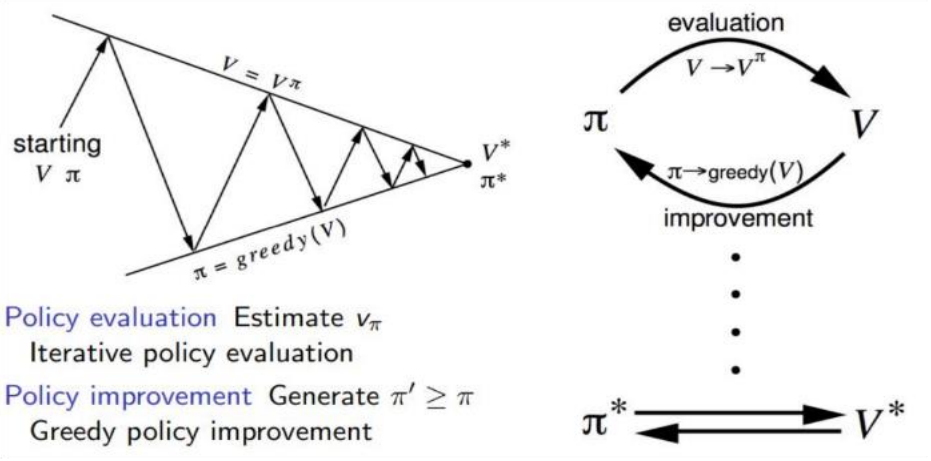
$$V^{\pi}(s) = R(s, \pi(s)) + \gamma \sum_{s'} P(s'|s, \pi(s)) V^{\pi}(s')$$

vs.

- **Bellman Optimality Equation**

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

- Bellman Optimality Equation 은 Value iteation, policy iteration, 강화학습 등으로 푼다.



- 강화학습 법 예) Q-learning, SARSA

- value iteration : 아래 수식과 같이 푼다.

(for k=1,~

$$v_{k+1}(s) = \max_{a} \{ \mathcal{R}_{s}^{\pi} + \gamma \sum \mathcal{P}_{ss'}^{\pi} v_k(s') \}$$

2. Policy evaluation : V_{Π}를 iterative하게 구함

(Π_1,V_{Π^1} → v 계산 후 update → V_{Π})

- 예)

짚신, 우산 문제

S R

Π_1 : 짚 짚

$P_{ss'} = \begin{bmatrix} 2/3 & 1/3 \\ 1/2 & 1/2 \end{bmatrix}$

$\gamma = 1/2$

$Start\ V(S) = 0, V(R) = 0$

$V^{k+1}(s) = R + \gamma \sum P_{ss'} V(s')$

first step

$V(S) = 5 + \frac{1}{2}(\frac{2}{3}(0) + \frac{1}{3}(0))$=5 → V(S)=5

$V(R) = -1 + \frac{1}{2}(\frac{1}{2}(0) + \frac{1}{2}(0))$=-1 → V(R)=-1

2nd step

$V(S) = 5 + \frac{1}{2}(\frac{2}{3}(5) + \frac{1}{3}(-1))$=5 → V(S)=update

$V(R) = -1 + \frac{1}{2}(\frac{1}{2}(5) + \frac{1}{2}(-1))$=-1 → V(R)=update

→ V(S), V(R) update 수렴할때까지 → $V^{\Pi^1}(S), V^{\Pi^1}(R)$ 정해짐

(※나머지 정책에 대해서도 동일하게 구할수 있음)