강화학습 1 장 정리 **-김변민-**

강화학습이란 ?

- 기계학습의 한 종류로 순차적 의사결정 문제에서 "누적보상"을 최대화하기 위해 시행착오를 통해 행동을 교정하는 학습과정

Dynamic Programming

- MDP를 푸는데 이용됨
- 여기서 프로그래밍이란 흔히 말하는 프로그래밍이 아닌 계획 및 설계라는 의미로 사용됨

Markov process

Markov process 란 ?

- 미리 정의된 어떠한 확률 분포를 따라서 상태와 상태 사이를 이동해 다니는 과정 (ref. 바닥부터 배우는 강화학습)

구성 요소

- State
- Transition probability matrix

<우산장수 예제 in Markov process>

State = Sunny, Rainy

Transition probability matrix

	Sunny	Rainy
Sunny	α	$1-\alpha$
Rainy	$1-\beta$	β

Markov Reward Process

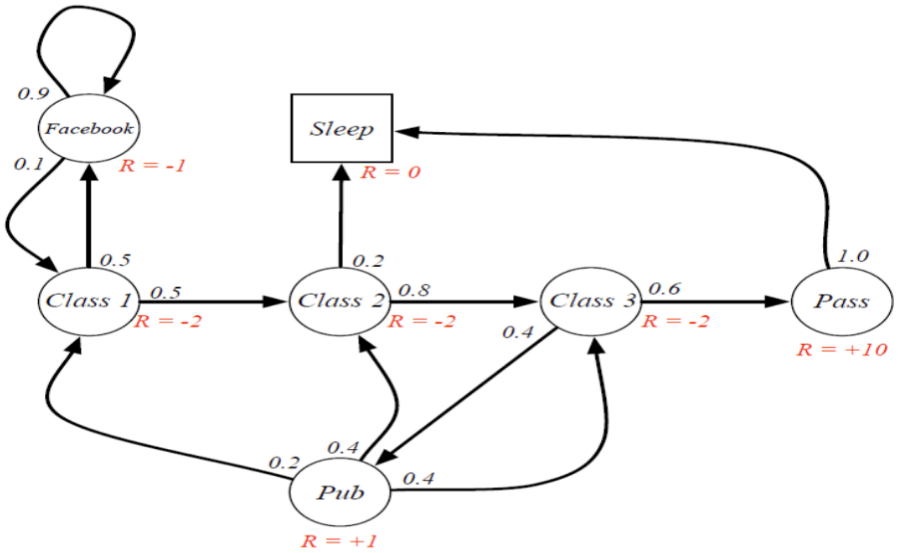
Markov Reward Process 란 ?

- Markov Process 에서 reward 와 discount factor 가 추가 된 과정

구성요소

- State
- Transition probability matrix
- Reward
- Discount Factor

<Student example in MRP>



S={Facebook, Class1, Class2, Class3, Pub, Pass, Sleep}

Discount factor 가 1 일 때 Class 3 의 Value 구하기

1. Class 3 의 value 를 구하기 위해서는 미래의 state 인 pass 의 value 부터 구해야 .

   V(pass) = 10 + 1*0 = 10

2. V(class 3) = -2(reward) + 0.6*10(v(pass)) + 0.4*0.8(v(pub)) = -4.3

   ➔ 이 상황은 v(Pub) 을 안다는 가정을 했지만 그게 아니라면 반복적으로 모든 state 의 value 값을 구해야 함

Return

Return 이란 ?

- 현재 t일 때 미래에 얻을 수 있는 감쇠 된 보상의 합

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

$\gamma \epsilon [0,1]$: 미래 reward에 대한 discount factor

현재 얻는 reward가 미래에 얻는 reward보다 valuable하기 때문에 0~1사이의 값 대입하여 먼 미래에 얻는 reward의 영향이 적도록 discount factor를 추가한다. 만일 discount factor 가 1이라면 미래의 return의 값이 감쇄되지 않는 것을 의미한다.

- $\gamma = 0$이면 myopic evaluation
- $\gamma = 1$이면 far-sighted evaluation

Value function

Value function 이란?

- 상태 s 로부터 시작하여 얻는 리턴의 기댓값

$$v(s) = E[G_t | S_t = s]$$

V(s)는 state에 대한 함수이기 때문에 어떤 state로 시작했는지에 따라 달라진다.

Bellman equation for MRP

-recursive relationship을 이용하여 미래에 발생하는 모든 return에 대해 infinite term을 계산하지 않아도 됨

$$\begin{aligned} v(s) &= E[G_t | S_t = s] \\ &= E[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots | S_t = s] \\ &= E[R_{t+1} + \gamma G_{t+1} | S_t = s] \\ &= E[R_{t+1} + \gamma v(s) | S_t = s] \end{aligned}$$

행렬로 표현하여 계산을 좀 더 용이하게 할 수 있다.

Matrices로 표현

$$v = R + \gamma P v$$
$$v = (I - \gamma P)^{-1} R$$

Markov Decision Process

MDP 는 MRP 에서 에이전트가 더해진 것이다. 그 말은 즉 MRP에서 액션을 더 해준다면 MDP 가 되는 것이다. 그러므로 <S, P, R, γ> 로 이루어진 MRP와 달리 MDP는 <S, A, P, R, γ> 로 구성된다. 여기서 S 는 state , A 는 action, P 는 transition probability, R 은 reward, γ 는 discount

factor 를 의미한다. MDP 에서는 MRP에 action 이라는 요소가 추가되고 엄밀히 말하면 transition probability 를 나타내는 P의 의미도 MRP 와는 다르다. MRP에서의 P 는 현재 state 가 다음 state 가 될 확률을 의미하지만 MDP 에서는 action이 추가되었기 때문에 MDP에서의 P는 현재 state 에서 어떠한 action을 취했을 때 다음 state가 될 확률을 의미한다. MRP와 MDP 의 가장 큰 차이점은 결국 action의 유무이다. 여기서 Action과 밀접하게 연관 된 개념 중 하나는 policy이다.

Policy

Policy 의 수식과 정의는 다음과 같다.

A policy π is a distribution over actions given states

π(a|s) = P[$A_t = a \mid S_t = s$]

파란 박스안에 있는 수식을 풀어서 해석하면 state s 에서 action a 를 선택할 확률이다. 즉 Policy 란 각 state 에서 어떤 action을 선택할지 정해주는 함수이다.

Value Function

MRP에서의 가치함수와 MDP에서의 가치함수는 약간의 차이가 존재한다. MRP 에서의 state value 는 현재 state 에서 미래에 얻을 수 있는 reward 의 평균이다. 하지만 MDP 에서는 action이 추가되었기 때문에 Policy 에 따라서 return이 달라진다. 또한 State value 이외에 Action의 value 에 대해서도 정의할 수 있다. 정리하자면 MDP 에서는 2개의 value function 이 존재하고 다음과 같이 정의된다.

State value function

The state value function $V_\pi(s)$ of an MDP is the expected return starting from state s , and then following policy π

$$V_\pi(S) = E_\pi[G_t|S_t = s]$$

State-Action value function

The action-value function $q_\pi(s,a)$ is the expected return starting from state s, taking action a, and then following policy π

$$q_\pi(s,a) = E_\pi[G_t|S_t = s,\ A_t = a]$$

2 가지 value function 중 state value function은 state s 부터 끝까지 policy π 를 따라 갔을 때 얻는 리턴의 기댓값을 의미한다. MRP에서의 가치함수에서 policy π 만 추가 되었을 뿐 나머지는 같다. MDP 의 state value function 에서 가장 중요한 점은 value 가 policy에 의존해서 움직인다는 점이다. MDP 에서는 state value function 외에도 state-action value function이라는 것이 존재한다. State value function이 state의 value에 대해 평가한 것이라면 state-action value function은 action 의 value에 대해서 평가한 것이다. State – action value function을 수학적으로 표현한 $q_\pi(s,a) = E_\pi[G_t|S_t = s, A_t = a]$ 다음 식의 의미는 state s 에서 action a를 선택하고 그 후 policy π 를 따라서 움직였을 때 얻는 리턴의 기댓값이다. 앞서 말한 2가지 value function의 유일한 차이는 state 에서 어떤 액션을 선택하는 가이다. State value function 에서는 주어진 policy에 따라서 액션을 선택하게 되지만 state – action value function 에서는 첫번째 action은 미리 정해지고 그 후 policy에 따라서 액션을 선택하게 된다.

Bellman Expectation Equation

앞서 말한 State의 Value 와 Action의 value를 구하기 위해서는 Bellman Expectation Equation을 이용해야 한다. V^π 와 Q^π 의 Bellman Expectation Equation 은 다음과 같다.

Bellman Expectation Equation for V^π

$V_\pi(S) = \sum_{a \in A} \pi(a|s)(R_a^s + \gamma \sum_{s' \in S} P_{ss'}^a V_\pi(s'))$

Bellman Expectation Equation for Q^π

$q_\pi(s,a) = R_a^s + \gamma \sum_{s' \in S} P_{ss'}^a \sum_{a' \in A} \pi(a'|s') q_\pi(s',a')$

Bellman Expectation Equation에 대해서 이해하기 위해서는 방정식을 단계별로 살펴볼 필요가 있다. 우선 $q_\pi(s,a)$ 는 state s 에서 action a를 선택하고 그 후 policy π 를 따라서 움직였을 때 얻는 리턴의 기댓값이다. 이는 다음과 같은 식으로 표현 될 수 있다.

$$q_\pi(s,a) = R_a^s + \gamma \sum_{s' \in S} P_{ss'}^a V_\pi(s') \quad (1)$$

또한 $V_\pi(S)$ 는 현재 state에서 policy π 를 따라 움직였을 때 얻는 리턴의 기댓값이기 때문에 policy와 state-action value 의 곱의 합으로 표현될 수 있다. 이를 식으로 나타내면 다음과 같다.

$$V_\pi(S) = \sum_{a \in A} \pi(a|s) q_\pi(s,a) \quad (2)$$

식(2)의 $q_\pi(s,a)$에 식(1)을 대입하면 $V_\pi(S)$를 도출할 수 있다. 또한 식(1)에서 $V_\pi(s')$에 식(2)를 대입한다면 $q_\pi(s,a)$를 도출할 수 있다.