강화학습 1 장 정리 **-김지원-**

Dynamic Programming (동적계획법)

-Programming은 최적화 planning의 의미를 가짐

-시간에 따라 변하는 시스템에서 sequential decision을 내림

Reinforcement learning is based on DP

Markov process

**Markov process의 구성요소

- State
- Transition probability matrix

**마코브체인

목적: DP를 모델링하는 툴

-이전과 현재의 state간 관계를 모델링하기 위해 마코브체인 도입

** 마코브체인의 Assumption:

vs iid (Independent identical distribution): 이전 step과 다음 step의 dependency가 존재하지 않음

ex) 삼성전자 주가처럼 dependency가 있는 모델에는 적합하지 않음

예제

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

365일동안 sunny or rainy day의 possible sequence는 2^{365}개가 존재한다. 이때 Limiting probability π_{ij}를 다음과 같이 정의할 수 있다.

π_{ij}: *t=0에서 state가 i일 때 t=∞에서 j일 확률*

$$\pi_{ij} = \lim_{n\to\infty} P\{X_n = j \mid X_0 = i\}$$

π_{ij}가 i에 independent하다고 가정할 경우,

$$\pi_j = \lim_{n\to\infty} P\{X_n = j \mid X_0 = i\}$$

$$\pi P = \pi \ldots \ Balance\ equation$$

$\alpha = \frac{3}{4}$, $\beta = \frac{1}{2}$ 일때,

$$\pi_1 = \frac{3}{4}\pi_1 + \frac{1}{2}\pi_2 \qquad \pi_2 = \frac{1}{4}\pi_1 + \frac{1}{2}\pi_2$$

$$\pi_1 = \frac{2}{3}, \qquad \pi_2 = \frac{1}{3}$$

Markov decision process는 action이 추가됨

Ex) sunny day에 우산을 팔것인지 짚신을 팔것인지

Ex) 현재 바둑판이 비어있을 때 어디서 시작할지

Markov Reward Process

-Markov chain with reward process

:앞서 Markov processs에 reward와 discount factor 가 추가된 형태로

State s의 Reward는 현재 t시점에 state가 s일 때 t+1시점에 얻는 reward의 평균으로 구할 수 있다.

Definition

A *Markov Reward Process* is a tuple $\langle \mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

- $\mathcal{S}$ is a finite set of states
- $\mathcal{P}$ is a state transition probability matrix, $\mathcal{P}_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s]$
- $\mathcal{R}$ is a reward function, $\mathcal{R}_s = \mathbb{E}[R_{t+1} \mid S_t = s]$
- γ is a discount factor, $\gamma \in [0, 1]$

ex) 우산장수와 짚신장수

S={sunny, rainy}

State transition probability matrix $\mathrm{P} = \begin{bmatrix} 3/4 & 1/4 \\ 1/2 & 1/2 \end{bmatrix}$

Sunny day에 짚신장수는 5만원의 이익을 우산장수는 2만원의 손해를 보고, rainyday에 짚신장수는 1만원의 손해를 우산장수는 10만원의 이익을 본다고 가정하면 다음과 같이 reward를 표현할 수 있다.

우산장수의 reward $\mathrm{R} = \begin{bmatrix} -2 \\ 10 \end{bmatrix}$

짚신장수의 reward $\mathrm{R} = \begin{bmatrix} 5 \\ -1 \end{bmatrix}$

****Return**

: 현재 t일 때 미래에 얻을 수익은 얼마나 될 것인가

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

$\gamma \epsilon [0,1]$: 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.

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

****Value function** ($state\ s$*에서 시작했을 때 앞으로* G_t의 average 값)

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

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

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

$\therefore v(s) = R(s) + \gamma \sum_{s' \epsilon S} P_{ss'} v(s')$ -> (현재 state의 reward)+(다음 state가 s′일때 모든 s′에 대한 v(s′)의 평균)

Matrices로 표현

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

Markov Decision Process

MDP 는 MRP 에서 Action을 더해준 형태

MRP 구성요소	MDP 구성요소
State, Transition probability, Reward, Discount factor	State, Action Transition probability, Reward, discount factor

Policy

Given state에서 어떤 action을 취할 확률

*policy 개수: (Action 개수)^(state 개수)

A policy π is a distribution over actions given states

π(a|s) = P[$A_t = a$ | S_t = s]

Value Function

MDP의 state-value function: state s에서 시작했을 때 policy π 를 따랐을 경우, expected return

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 s에서 action a를 하고 이후에는 policy π를 따를때의 expected return

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

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

➔ 둘의 차이: 첫번째 state에서 action이 specify되는가

Bellman Expectation Equation

Bellman Expectation Equation을 이용해 state-value fuction과 action value function을 풀수 있다. $V_\pi(S)$ 는 현재 state에서 policy π 를 따라 움직였을 때의 expected return으로 bellman equation 은 다음과같으며 $q_\pi(s,a)$ 는 현재 state s 에서 action a를 취하고 그 후 policy π 를 따라서 State s′에서의 a′를 취했을때의 expected return으로 다음과 같이 나타낼 수 있다.

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

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