Markov

Geonhee Lee
gunhee6392@gmail.com

1. Markov Process(MP)

MDP(Markov Decision Process)에 비해 좀 더 간단한(기본이 되는) 모델.

MP problem을 Markov Chain(MC)이라고 부르기도 함.

MC는 이산 확률 프로세스(Discrete Stochastic Process), Continous Stochastic Process를 다루는 MC가 있긴하다.

- Stochastic Process: 확률 분포를 가진 랜덤변수(random variable)가 일정한 시간 간격(Time interval)으로 값을 발생시켜 모델링하는 것.
- 이러한 모델 중 현재의 상태가 오로지 *이전 상태에만 영향을 받는 확률 프로세스* $\Rightarrow$ **Markov Process**
- MP 모델은 두 가지 속성으로 표현 가능
    - X: (유한한) 상태 공간(state space)의 집합.
    - P: 전이 확률(Transition probability) = 모든 상태 X 사이의 전이 확률을 의미.

MP(X, P)

- **Step**
    - 각 state의 transition는 이산시간(Discrete time)에 이루어지며, 상태 집합 X에 속하는 어떤 임의의 상태에 머무를 때는 시간.
    - 현재 step이 n이라 하면, 다음 step은 $n+1$ 라고 기술.
- **상태 전이 확률(State Transition Probability)**
    - p_{ij}는 상태 i에서 j로 전이될 확률 값.
    - X_n은 step n 에서 머물러있는 상태(state)를 의미, 정확히는 해당 상태(state)에 대한 랜덤 변수(Random Variable)를 의미.

$$p_{ij} = p(X_{n+1} = j | X_n = i) \qquad \forall i, j \in X$$

- **(Remark)** 상태 i를 방문했을 때, 그 이전에 어떤 상태에 방문했는지 상관없이 상태 i에서의 다음 상태 j로의 전이 확률 값은 언제나 동일하다는 것.
    - MDP 의 기본 속성, **모델을 단순**하게 만들어준다.
    - 또한 이를 **무기억성 속성**이라고도 한다.
    - 추가적으로 transition probability의 조건은 다음과 같다.

$$p_{ij} \geq 0 \qquad \sum_{j \in X} p_{ij} = 1, \qquad \forall \in X$$

> RL과의 연관성
>
> > MP는 수동적인 모델. MP(X, P) 환경에서는 사용자가 개입할 수 없다. Time step이 진행되는 동안 알아서 상태가 전이된다 이후(MDP)에는 사용자가 action을 제어함으로써 전체 작업에 직접적인 개입을 하게 된다.

2. Dynamic Programming(DP)

- Dynamic Programming(DP)은 동적 프로그래밍이라고 한다.
- RL에서는 기본적으로 확률 동적 프로그래밍(Probabilistic Dynamic Programming) 기법을 사용한다.
    - 즉, 기본적인 DP 모델에 확률 이론이 포함

DP 기초

- 최적화 기법 중 하나로 재귀(recursion)를 이용하여 최적 솔루션을 얻어내는 방식.
    - Tree 형태의 탐색 방식을 가지고 있다.
- 모든 상태에 대해 최적의 결과를 '저장'하고 있는 특성.
    - 저장 비용 발생.

RL 과의 연관성

> DP는 계산 복잡도가 높은 일반적인 알고리즘.
>
> > 따라서 stochatic 모델이 아니다.

> 이런 이유로 최적 값을 얻을 수 있는 모델.
>
> > 결과가 확률 상태로 주어지지 않는다.

> MP와는 다르게 능동적인 모델, 사용자가 암묵적으로 상태 전이를 선택할 수 있다.
>
> > 물론 최대 보상을 가지는 deterministic한 상태 전이를 선택하기 때문에 MDP와는 차이가 있다. 이후에는 MP 모델을 바탕으로 DP 구조를 적용한 모델을 살펴볼 것.

3. Markov Decision Process(MDP)

확률 프로세스(Stochastic Process)의 확장

MP 모델에 Action을 추가

- 각 상태(state)에서 다른 상태(state)로 이동시, 발생하는 행위를 action이라고 정의.
    - 어떤 상태(state)에 있느냐에 따라 취할 수 있는 action이 다를 수 있다.
- 상태 x에서 취할 수 있는 action 은 $A(x) \in A$이다.

- 앞서 정의한 전이 확률 함수에 action에 대한 속성을 포함하면 다음과 같다.

$$p_{xy}^{a} = p(y|x,a)$$

- 즉, 상태 x에서 상태 y로 action a를 취한 뒤 이동할 확률을 의미
    - 이는 $x, y \in X$이고, $a \in A(x)$
    - 다르게 해석하면, 어떤 행동을 취하는 가에 따라 전이 확률, 다음 상태가 달라 질 수 있다는 말.
- **(Remark)** 행동의 존재 유무가 MP, MDP를 구분하는 핵심 요소.
    - **MP(Markov Process)**: $p(\acute{s}|s)$
        - 상태의 이동 제약 조건에 이전 상태만 영향을 받는다.
    - **MDP(Markov Decision Process)**: $p(\acute{s}|s,a)$
        - 상태의 이동 제약 조건에 이전 상태와 행해진 행동에 영향을 받는다.
- 그리고 이 행동을 취함으로써 얻어지는 보상(Reward)은 다음과 같다.

$$R(x,a)$$

- 즉, 어떤 상태 x에서 행동 a를 취했을 때 얻어지는 보상을 R(x, a로 기술)

Policy

- **Policy(π)**: "Step 시퀀스에서 각각의 state들과, 그 state들에 mapping된 action의 집합"
- 어떤 경우에는 π가 state들에 mapping된 고정된 action을 의미하는 것이 아니라,
    - State에 mapping된 'action에 대한 **확률 함수(probability function)**'로 정의할 수도 있다.
    - 즉, (s,a)에서 a가 특정 값이 아닌 p(a|s) 확률 함수를 가지는 랜덤 변수로 취급될 수 있다는 말.
        - 따라서, 특정 상태 x에 대한 출력 값은 **확률 함수가 반환**되게 된다.
- 결론적으로, policy π를 크게 두 가지로 나누어 생각할 수 있다
    - Deterministic policy
    - Stochastic policy

Deterministic policy

- Deterministic policy는 다음과 같이 정의할 수 있다.

$$\pi(x) = a$$

- 임의의 π를 하나의 함수로 생각하여, 특정 π를 따르는 시퀀스에서 특정 시점 t에 대해 상태 x를 대입하면 그때에 사용될 정확한(고정) 액션을 얻을 수 있다.

Stochastic policy

- Stochastic policy 모델에서는 π를 사용하는 형태가 조금 다르다.
  - Deterministic policy에서는 π 자체를 함수 식처럼 사용하게 되는데, Stochastic policy는 확률식처럼 표기하게 된다.
  - 즉, 행동도 확률적으로 취하게 된다. 이로인해 π는 조건부 확률함수가 된다.

$$\pi(a|x) = p(A = a|X = x) \qquad \forall x \in X$$

- 다음과 같이 표기하기도 한다.
  - 출력 결과로 행동에 대한 확률함수가 반환된다고 생각하면 된다.

$$\pi(a, x) = p(A = a|X = x)$$

Markov Decision Process(MDP)

X_t^π= 랜덤 변수, 상태 X가 특정 스텝 t에서 policy π를 따르는 상태 X에 대한 랜덤 변수로 취급.

- 어떤 policy π가 주어진 상태의 모델이라고 생각하면 된다.
- **(Remark)** 비용(Reward): 사전 정의된 policy π를 가진 프로세스에서 H 스텝(step) 발생된 이후에 발생된 비용을 계산하면 어떻게 될까?
- Discounte factor 적용.

$$\sum_{t=0}^{H-1} \gamma^t R(X_t^\pi, \pi_t(X_t^\pi))$$

- 어떤 policy π를 따르는 프로세스에서 각 단계의 상태에서의 행동을 수행한 뒤 얻어지는 보상의 총합을 의미.
- 사실 policy는 순차적으로 어떤 상태를 방문하여 특정 행동을 선택하는 방식처럼 보이기도 한다.
  - $h = s(1)a(1)s(2)a(2)\dots$
  - 이와 같은 실제 프로세스가 진행되어 얻어진 시퀀스를 History라고 한다.
  - 한 번의 시퀀스를 모두 끝내면 이를 한 개의 에피소드(episode)로 취급한다.

Value Function

- 어떤 임의의 policy를 선택하여 얻어지는 총 보상(Reward)을 V라는 함수로 정의하자.
- 이 식을 DP에 적용.
  - DP의 경우, 각 상태에 대해 특정 값들을 저장하는 형태를 취함으로써 space-complexity와 time-complexity를 교환하는 효과를 가졌었다.
  - 여기서도 마찬가지로 Value function가 이러한 정보를 담는 매개체로 사용되게 된다.
  - 결국 각 상태(state)에 대해 어떤 값들을 저장하게 될 것.

Reference

[1] norma3/rl [2] 대손의 스마트 웹