1. MDP

경희대학교

기계공학과 RCI 연구실
박보형

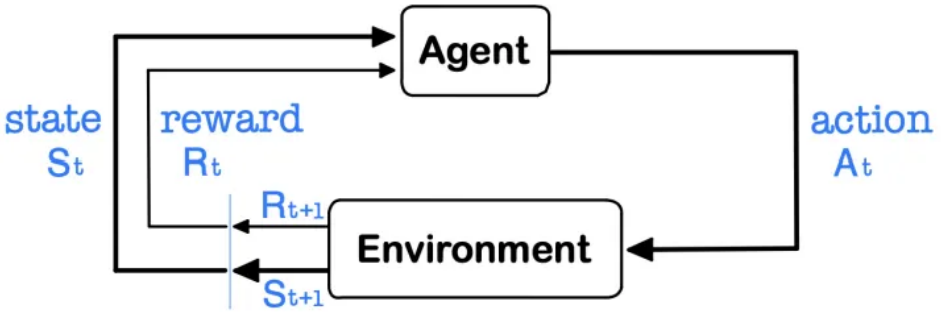
2025-2 이동로봇

목차

Introduction

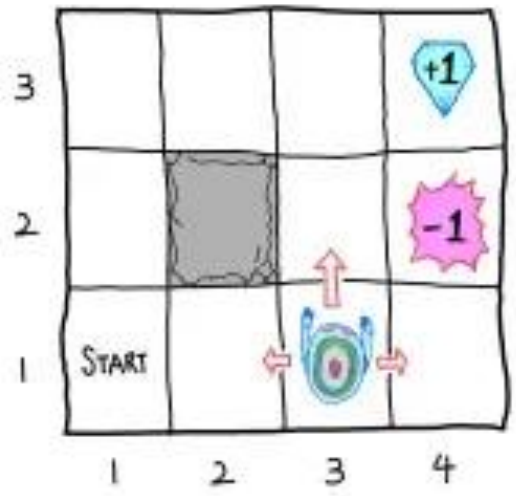
Reinforcement Learning(RL)

- Supervised Learning
  - Label이 있는 data, 즉 정답이 있는 데이터 사용
- Unsupervised Learning
  - Label이 없는 data를 사용하며, classification 수행 불가능
- Reinforcement Learning
  - 미리 준비된 training data가 없다
  - Agent가 환경 속에서 상호작용하며 직접 얻는 data를 가지고 학습



Grid World

Grid World?



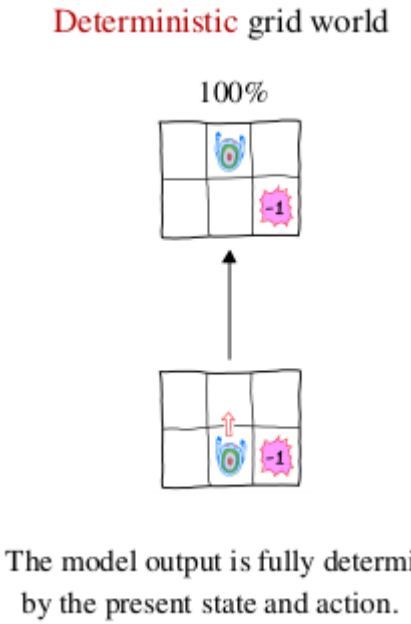
- Markov Decision Process가 잘 적용되는 모델

- Environment
  - State : S = {(1, 1), (1, 2), ⋯, (4, 2), (4, 3)}
  - Action : A = {north, south, east, west} 방향으로 한 칸씩 이동
    - Action이 벽에 막혀 이동할 수 없는 경우에는 이동하지 않고 제자리에 정지
  - Reward
    - Big Reward : (4, 3)에 도착하면 +1, (4, 2)에 도착하면 −1점을 부여하고 에피소드 종료
    - Small Reward : Action을 하나 취할 때 마다 작은 감점인 c(−0.1) 점을 부여
      - 단, Big Reward를 받아서 에피소드가 끝나는 경우에는 Big Reward만 부여
- Goal
  - 각 State 마다 Total Sum of Rewards를 최대화하는 Action을 찾는 것
  - 위 목표를 위해 각 State에 대해 Action을 선택하는 방법을 Policy라고 한다.

Grid World

Deterministic vs Stochastic Action

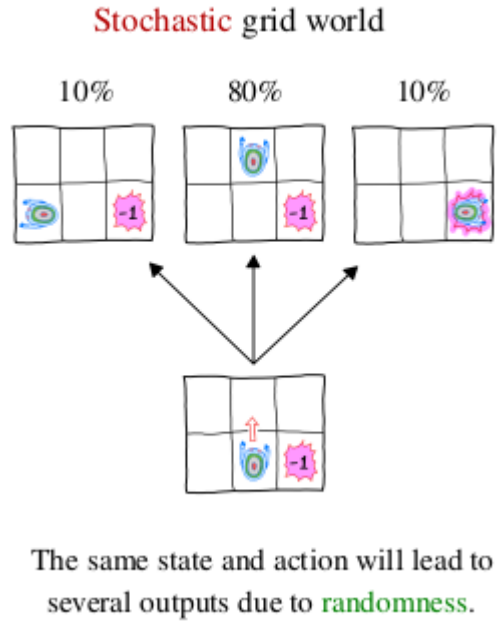
Deterministic grid world



The model output is fully determined by the present state and action.

- North로 Input이 주어지면 그대로 Action을 취할 확률이 100% : Noise가 없다.
- 하나의 Policy에 대해서 Episode는 정확히 하나만 생성된다.

Stochastic grid world



The same state and action will lead to several outputs due to randomness.

- 확률을 가지고 움직이기 때문에 동일 Input에 대해서도 random한 output 가능
- Policy가 정해져서 각 State 별 Action이 정해졌다고 하더라도 Noise 때문에 여러 Episode가 생성될 수 있다.

Markov Property

- 어떤 환경이 Present State만으로 Feature State를 예측할 수 있는 성질
  - Feature State로 이동할 때 Past State에 영향을 받지 않는다.

$$P(S_{t+1} = s' \mid S_t = s) = P(S_{t+1} = s' \mid S_0 = s_0,\ S_1 = s_1,\ \ldots,\ S_t = s)$$

  - 따라서 Markov Property를 만족한다면 Past State를 기록할 필요가 없다.
    - Memoryless Property라고도 한다.
- State Transition Probability
  - State가 S에서 S'로 전이하므로 State Transition이라고 하며, 이때 사용된 확률 P는 State Transition Probability라고 한다.

$$P(S_{t+1} = s' \mid S_t = s)$$

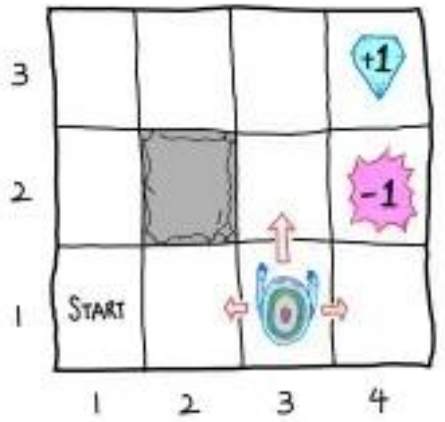
Markov Decision Process(MDP)

- MDP는 Markov Property가 성립하는 환경을 수학적으로 모델링한 것
  - Tuple(S, A, P, R, γ)로 구성되며 모든 State에서 Markov Property를 만족해야 한다.
  - Decision은 State에 의존하는 Markov Property에서 추가된 Action을 의미한다.
- MDP Elements
  - S : State Space
  - A : Action Space
    - Agent가 각 State 마다 취할 수 action을 모아둔 것이다.
  - P : State Transition Probability from S to S' given A

$$P_{ss'}^{a} = p(s' \mid s, a) = P\left(S_{t+1} = s' \mid S_t = s,\ A_t = a\right)$$

  - R : Reward
    - State 기반 보상 R_s , Action 기반 보상 R_s^a, Transition 기반 보상 : $R_{s,s'}^a$
  - γ : Discount Factor $\in$ [0, 1]
    - 미래 보상을 얼마나 중요하게 볼지 결정하는 계수

MDP in Grid World



- State Set
  - Grid 위에서 Agent가 위치할 수 있는 모든 좌표(벽 (2, 2)제외)
  - S = (1, 1), (1, 2), ⋯ ,(4, 2), (4, 3) : 11 States
- Action Set
  - Agent가 취할 수 있는 행동 4가지
  - A = north, south, east, west : 4 Actions
- State Transition Probability
  - 특정 State에서 특정 Action을 했을 때 다음 State로 이동할 확률
  - $P^{east}_{(3,1)(3,2)} = 0.8$, $P^{north}_{(3,1)(3,2)} = 0.1$, ⋯
  - $9 \times 4 \times 11 = 369$개의 State Probability가 존재
- Reward
  - Big Rewards : $R^{east}_{(3,3)(4,3)} = +1$, $R^{north}_{(3,2)(4,2)} = R^{east}_{(4\,1)(4,2)} = -1$, ⋯
  - Small Rewards : $R^{a}_{ss'} = c$ for the other cases where c is a small negative value.

Return

- Return G_t는 현재 Time Step t에서 이후에 얻어지는 Total Discounted Reward

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

- Discount Factor $\gamma \in [0, 1]$
  - Time Step이 먼 Reward일 수록 γ가 더 많이 곱해져 더해진다.
    - 더 먼 미래의 Reward일 수록 약하게 반영하겠다.
    - Present State에서 내가 Action을 취했을 때 얻어지는 Reward는 명확하지만, Stochastic Action이기 때문에 더 먼 미래의 Reward일 수록 불명확하다는 점을 반영
  - 끝나지 않는 게임이라면 Return의 값이 무한대가 될 수 있다.
    - Discount Factor를 사용해서 수렴시키는 역할
  - 관습적으로 가까이 있는 Reward가 멀리 있는 Reward보다 더 중요하다.

Policy

- (Stochastic) Policy π is a probability distribution over Actions for given States.

$$\pi(a \mid s) = P(A_t = a \mid S_t = s)$$

  - Deterministic Policy일 때는 State가 주어지면 Action이 딱 하나만 정해진다.
  - 확률이 아니라 State에서의 Action 값이 Policy가 된다.

- Policy π는 Return인 Total Discounted Reward를 Maximize하는 것이 목표
  - Policy는 이를 위해 각 State마다 Optimal Action이 무엇인지 말해주는 가이드라인
  - MDP Policy는 Past State에 의존하지 않기 때문에 Present State에만 좌우된다.

Unknown MDP

- Known MDP
  - Transition Probability를 모두 알고 있는 MDP
  - Return G_t에 대한 Expectation을 모두 계산 가능
  - 이를 이용해 Deterministic Optimal Policy를 직접 계산할 수 있고, 이는 항상 존재
  - Dynamic Programming을 통해 계산 가능
- Unknown MDP
  - Transition Probability를 모르는 MDP. 대부분의 환경이 이에 해당
  - Sample Data만 가지고 있기 때문에 Return G_t의 Expectation을 계산 불가능
- ε-greedy policy
  - Stochastic Policy
  - Sample Data를 통해 찾은 Optimal Policy : $P = 1 - \epsilon + \frac{\epsilon}{n}$
  - 나머지 Policy : $P = \frac{\epsilon}{n}$
  - Sample Data로 찾은 Optimal Policy를 어느 정도 신뢰하되, 다른 Policy에도 가능성을 열어둔다는 의미.

Next

Bellman Equation

- MDP를 통해 State, Action, Transition, Reward 구조와 Policy의 목적을 정의
- Optimal Policy를 만들기 위해서는 각 State가 장기적으로 얼마나 큰 Reward를 주는지 알아야 함
- 따라서 어떤 State가 얼마나 좋은 지 계산하는데 필요한 것이 Bellman Equation.

감사합니다