3. Monte Carlo

경희대학교

기계공학과 RCI 연구실
박보형

2025-2 이동로봇

목차

Generalized Policy Iteration(GPI)

Policy Iteration

- Policy Evaluation과 Policy Improvement를 반복하는 것
  - Policy Evaluation : Current Policy를 기준으로 Value-Function을 수렴할 때 까지 계속 업데이트하는 것
  - Policy Improvement : 모든 State에 대해서 수렴하게 되면 그 값들을 이용해 greedy Method를 이용해 Policy를 Improve하는 것

Generalized Policy Iteration

- GPI는 Policy Iteration을 일반화한 개념
  - 실제 환경에서는 모든 State에 대해 정확한 value 계산이 불가능
  - Sample 기반 근사치를 이용해서 업데이트
    - Evaluation : Sample Data를 모아서 Value Function 추정
    - Improvement : Value를 기반으로 Policy 계산

Monte Carlo Method(MC)

Monte Carlo

- 반복적인 Random Sampling을 통해서 특정한 수칙 결과를 얻어내는 계산 방법
- DP에서는 State Transition Probability와 그에 따른 Reward를 모두 알고 있음
    - GPI를 통해 S에 대한 Value Function 근사 가능
- 현실에서는 불가능하기 때문에 직접 sample data를 얻고 이를 평균 내면서 V를 찾아야 한다.
    - MC는 이러한 방식의 가장 기본

MC Prediction(Policy Evaluation)

Monte Carlo

$$q_\pi(s,a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a] \quad \rightarrow \quad Q(s,a) = \frac{1}{N(s,a)} \sum_{i=1}^{N(s,a)} G_t^{(i)}$$

- 실제로 여러 번 State-Action Pair (s, a)를 경험해서 나온 Return의 평균을 Q-value의 추정치로 사용
- 이를 위해서는 S(s, a)와 N(s, a)를 기록하고 있어야 한다.
    - 이를 단순하고 효율적으로 처리하기 위한 것이 Incremental Update

Incremental Update

- Return을 모두 저장해서 평균내는 것은 비효율적이니 점진적으로 Update하자

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{1}{N(S_t, A_t)}[G_t - Q(S_t, A_t)]$$

- 방문 횟수가 많은 S일 수록 영향이 줄어든다 : 최근 Return 값이 Q에 미치는 영향이 줄어든다.
- 그런데 후반에는 Agent가 똑똑해져서 최근 Return이 더 중요할 수도 있지 않나?

MC Prediction(Policy Evaluation)

Constant-α MC

- 1/N 대신에 고정 학습률 α 도입
  - 이제 N(s, t)를 계산할 필요 없음
  - 후반의 좋은 Return도 일정 비율로 도입
  - 수학적으로는 Exponential Moving Average와 동일한 개념

$$Q_{t+1} = (1-\alpha)Q_t + \alpha G_t \;\rightarrow\; Q_{t+1} = (1-\alpha)^{t+1}Q_0 + \sum_{k=0}^{t}\alpha(1-\alpha)^k G_{t-k}$$

MC Control(Policy Improvement)

ε-greedy Policy

- Exploitation : 이미 알고 있는 정보 내에서 가장 최선의 결정을 내리는 것
- Exploration : 더 많은 정보를 모아서 새로운 결정들을 찾는 것

ε-greedy policy : 1- ε의 확률로 Exploitation, ε의 확률로 Exploration을 실행하는 것

MC Control(ε-greedy Policy Improvement)

- 전체 m개의 Action들은 $\frac{\varepsilon}{m}$의 random 확률을 가진다(Stochastic).

$$\pi'(a|s) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{m}, & \text{if } a = \arg\max_{a'} Q^{\pi}(s, a') \\ \frac{\epsilon}{m}, & \text{otherwise} \end{cases}$$

- A가 최선의 Action일 때
  - Exploitation : 1- ε
  - Exploration : $\frac{\varepsilon}{m}$
- a가 최선의 Action이 아닐 때
  - 전체 ε을 Action 수 만큼 균등 분배
  - 모든 Action이 Non-zero Property 만족

    (모든 Action이 선택될 확률을 가진다)

MC Control(Policy Improvement)

Greedy in Limit with Infinite Exploration(GLIE)

- 무한히 탐험하면서도 최종적으로는 greedy하게 수렴한다.
- 조건 1 : Infinite Exploration

$$\lim_{k\to\infty} n_k(s,a) = \infty$$

  - 학습이 진행될수록 모든 (s, a)를 무한히 방문해야 한다 : 탐험을 끝없이 보장해야 한다.
  - $\varepsilon > 0$ 이면 만족
- 조건 2 : Greedy in the Limit

$$\lim_{k\to\infty} \pi_k(a|s) = \begin{cases} 1, & \text{if } a = \arg\max_{a'} Q(s,a') \\ 0, & \text{otherwise} \end{cases}$$

  - 학습이 무한히 진행되면 최종적으로 greedy 정책에 수렴해야 한다.
  - ε을 조절 : $\varepsilon = \frac{1}{k}$

- MC는 GLIE의 조건 1, 2를 모두 만족하므로 MC를 따른다면 Optimal로 수렴한다.

Temporal Difference Learning(TD)

- Bellman Equation으로 Return이 없어도 One-Step으로 계산할 수 있게 되었다.
    - 그런데 Transition Probability를 몰라서 Value-Function을 직접 계산할 수 없다.

- Value-Function을 모른다면 MC를 이용해서 실제로 받는 Return의 평균으로 추정하자.

- BE와 MC의 장점을 합친 것이 TD
    - Value-Function은 모르지만 MC처럼 현재 추정치를 이용해서 BE처럼 한 단계만 보고 업데이트 해보자.

감사합니다