04. 모델 프리 프리딕션

김호철

Contents

1 몬테카를로(Monte-Carlo) 학습

1.1 몬테카를로 강화학습

- MC는 경험의 에피소드들로부터 직접 학습한다.
- MC는 모델 프리 : MDP의 트랜지션이나 보상에 대한 지식이 없다.
- 종료된 에피소드로부터 배운다 : no 부트스트래핑
- MC는 $value = meanreturn$ 라는 가장 단순한 아이디어를 사용한다.
- 에피소드형 MDP에만 MC를 적용할 수 있고, 모든 에피소드는 종료되어야 한다.

1.2 몬테카를로 정책 평가(Policy Evaluation)

- 목표 : 정책 π 기반의 실제 경험한 에피소드들에서 v_π를 학습

$$S_1, A_1, R_2, ..., S_k \sim \pi \tag{1}$$

- 리턴은 할인된 보상의 합이다 :

$$G_t = R_{t+1} + \gamma R_{t+2} + ... + \gamma^{T-1} R_T \tag{2}$$

- 가치(value) 함수는 기대되는 리턴이다 :

$$v_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s] \tag{3}$$

- 몬테카를로 정책 평가는 기대(expected) 리턴 대신 경험적 평균(empirical mean) 리턴을 사용
- 한번의 에피소드를 마치면 그 에피소드 동안 진행해왔던 상태들 마다 리턴값이 존재하는데, 하나의 상태에 여러번의 중복 방문시 리턴값을 처리하는 방식에 따라 first-visit MC와 every-visit MC로 나누어짐
- first-visit MC는 상태에서 첫 번째 방문에만 리턴값을 저장하고 두 번째 이후는 고려하지 않는 것이고, every-visit MC는 방문할 때마다 리턴값으로 갱신하는 것
- **알고리즘** : 에피소드에서 상태 s를 처음(항상) 방문하는 t번째,
  - 카운터 증가 : $N(s) \leftarrow N(s) + 1$
  - 리턴의 합 증가 : $S(s) \leftarrow S(s) + G_t$
  - 가치함수는 리턴의 평균 : $V(s) = S(s)/N(s)$
  - 반복적으로 수행하면 최적 정책으로 수렴 : $V(s) \rightarrow v_\pi(s)\ as\ N(s) \rightarrow \infty$
- **증감적 평균(Incremental Mean)** : $x_1, x_2, ..$의 평균 $\mu_1, \mu_2, ...$는 증가하면서 계산되어 질 수 있다.

$$\begin{aligned}
\mu_k &= \frac{1}{K} \sum_{j=1}^{k} x_j \\
&= \frac{1}{K}\Big(x_k + \sum_{j=1}^{k-1} x_j\Big) \\
&= \frac{1}{K}(x_k + (k-1)\mu_{k-1}) \\
&= \mu_{k-1} + \frac{1}{K}(x_k - \mu_{k-1})
\end{aligned} \tag{4}$$

- **증감적 몬테카를로 업데이트**
  - 한번의 에피소드 $S_1, A_1, R_2, ..., S_T$를 종료 후 $V(s)$를 점진적으로 업데이트
  - 리턴 G_t가 있는 각 상태 S_t마다,

$$\begin{aligned}
N(S) &\leftarrow N(S_t) + 1 \\
V(S_t) &\leftarrow V(S_t) + \frac{1}{N(S_t)}(G_t - V(S_t))
\end{aligned} \tag{5}$$

  - non-stationary problems(stationary:고정적) : 고정적이지 않고 변하는 MDP에서는 상수값을 α로 고정 할 수 있다. 즉, 오래된 기억은 버린다.

$$V(S_t) \leftarrow V(S_t) + \alpha(G_t - V(S_t)) \tag{6}$$

2 TD(Temporal-Difference) 학습

2.1 TD 개요

- TD 방법은 경험의 에피소드에서 바로 학습
- TD는 모델 프리 : MDP 트랜지션, 보상에 대한 지식이 없음
- TD는 부트스트래핑을 통해 끝나지 않은 에피소드에서 학습
- TD는 추측(guess)으로 추측(guess)을 업데이트

2.2 MC와 TD

- 정책 π를 따르는 경험에서, 온라인으로 가치함수 v_π를 학습하는 것이 목표
- 증감적 every-visit 몬테카를로에서는 실제(actual) 리턴 G_t 방향으로 $V(S_t)$를 업데이트 한다.

$$V(S_t) \leftarrow V(S_t) + \alpha(G_t - V(S_t)) \tag{7}$$

- 가장 단순한 TD(0) 학습 알고리즘에서는 예상(estimated) 리턴 $R_{t+1} + \gamma V(S_{t+1})$ 방향으로 $V(S_t)$를 업데이트 한다.

$$V(S_t) \leftarrow V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1} - V(S_t)) \tag{8}$$

  - $R_{t+1} + \gamma V(S_{t+1})$ 를 TD 타겟이라 하고,
  - $\alpha(R_{t+1} + \gamma V(S_{t+1} - V(S_t))$를 TD 에러라고 한다.

2.3 퇴근길 남은 시간 예측 예제

상태	소요시간	예측된 남은 시간	예측된 전체 시간	MC로 예측	TD로 예측
사무실 출발	0	30	30	43	30
차에 도착, 비가 옴	5	35	40	43	40
고속도로 탈출	20	15	35	43	35
트럭 뒤	30	10	40	43	40
집앞 거리	40	3	43	43	43
집 도착	43	0	43	43	43

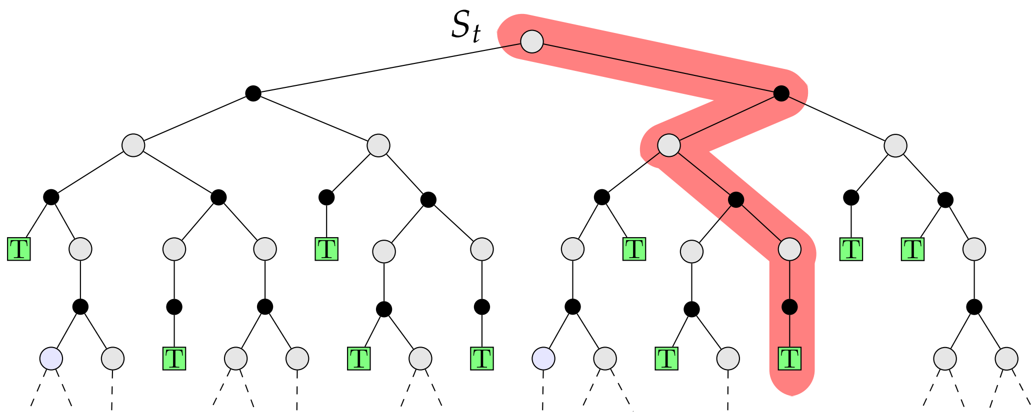
- **MC vs TD 의 장단점**
  - TD는 최종 결과를 알기 전에 학습 가능하고, 각 단계마다 온라인으로 학습 가능
  - MC는 리턴이 알려지기 전에 에피소드가 끝날 때까지 기다려야 한다.
  - TD는 최종 결과를 몰라도 학습 가능
  - MC는 완전한 시퀀스에서만 학습하고, TD는 불완전한 시퀀스에서도 학습 가능
  - MC는 종료가 있는 환경에서만 작동하고, TD는 지속적인 환경에서도 작동

- **편향/분산 트레이드오프**
  - MC는 분산이 높고 편향이 없다. : 우수한 수렴 속성, 초기 값에 민감하지 않고, 이해하고 사용하기 매우 간단
  - TD는 분산이 낮고 편향이 있다. : 일반적으로 MC보다 효율적, TD(0)는 $v_\pi(s)$로 수렴하고, 초기 값에 더 민감
- MC는 마르코프 속성을 이용(exploit)하지 않으므로 비 마르코프 환경에서 효과적이고,
- TD는 마르코프 속성을 이용(exploit)하므로 마르코프 환경에서 효과적이다.

2.4 MC, TD, DP 백업 비교

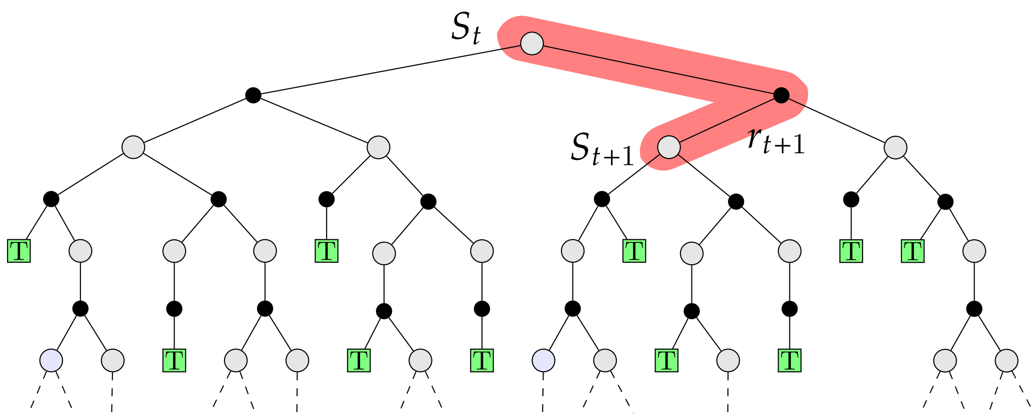
- **몬테카를로 백업**

$$V(S_t) \leftarrow V(S_t) + \alpha({\color{red}G_t} - V(S_t)) \tag{9}$$



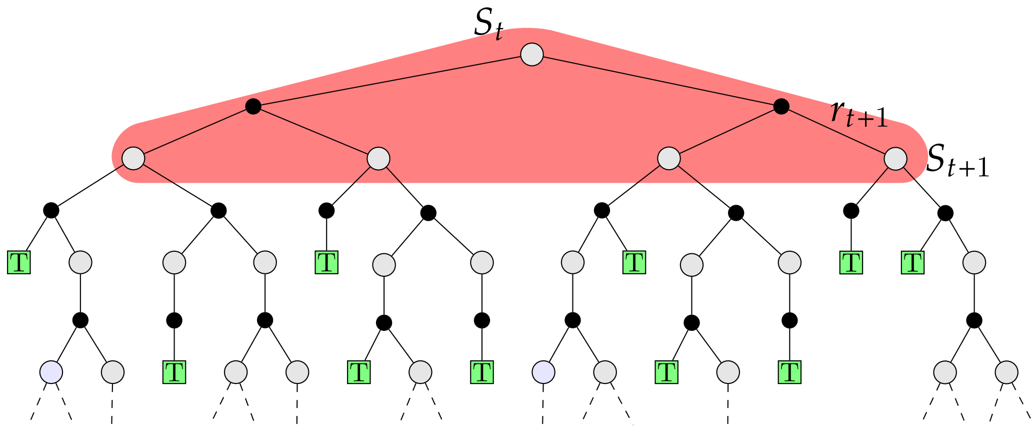
- **TD 백업**

$$V(S_t) \leftarrow V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)) \tag{10}$$



- **다이나믹 프로그래밍 백업**

$$V(S_t) \leftarrow \mathbb{E}[R_{t+1} + \gamma V(S_{t+1})] \tag{11}$$



2.5 부트스트래핑(Bootstrapping)과 샘플링(Sampling)

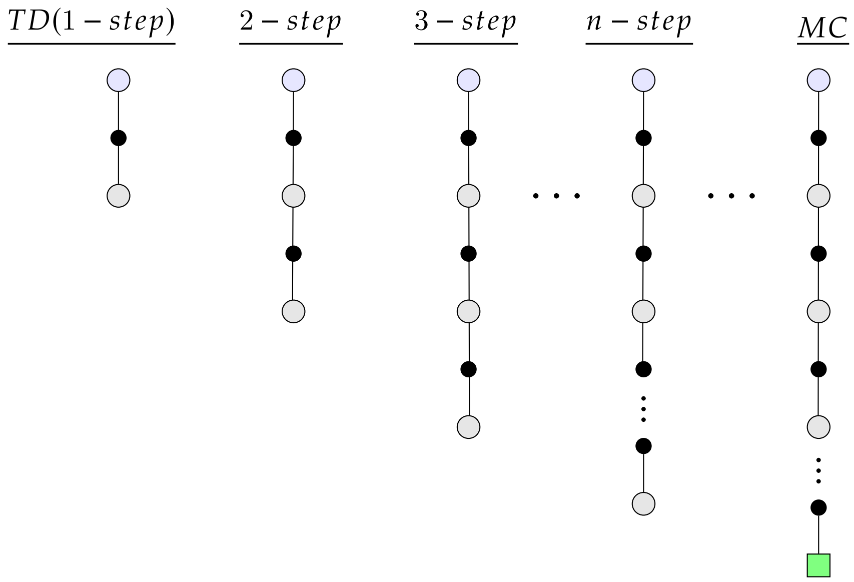
- **부트스트래핑** : 추정치(estimate:어림잡아)가 포함된 업데이트
- MC는 부트스래핑하지 않는다. DP와 TD는 부트스트래핑한다.
- **샘플링** : 예측치(expectation:확신을 가진)의 샘플링으로 업데이트
- MC는 샘플링한다. DP는 샘플링하지 않는다. TD는 샘플링한다.

	모델 프리(백업)	부트스트래핑	업데이트 시점
DP	× (full-width)	○	각 단계마다
MC	○ (샘플링)	×	에피소드의 끝
TD	○ (샘플링)	○	각 단계마다

3 TD 람다(λ)

3.1 n-스텝 TD

- 미래의 n-단계 만큼 계산하는 TD



- n이 1, 2, ∞ 에 대하여 n-스텝 리턴을 고려해보면,

$$
\begin{aligned}
&\color{red}{n=1\ (TD)} && G_t^{(1)} = R_{t+1} + \gamma V(S_{t+1}) \\
&\color{red}{n=2} && G_t^{(2)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 V(S_{t+2}) \\
& && \vdots \\
&\color{red}{n=\infty\ (MC)} && G_t^{(\infty)} = R_{t+1} + \gamma R_{t+2} + \dots \gamma^{T-1} R_T
\end{aligned}
\tag{12}
$$

- n-스텝 리턴을 정의하면,

$$G_t^{(n)} = R_{t+1} + \gamma R_{t+2} + \dots + \gamma^{n-1} R_{t+n} + \gamma^n V(S_{t+n}) \tag{13}$$

- n-스텝 TD 학습

$$V(S_t) \leftarrow V(S_t) + \alpha(G_t^{(n)} - V(S_t)) \tag{14}$$

- **평균 n-스텝 리턴** n-스텝을 m번 진행하고, 그 평균을 사용
- 2-step, 4-step 으로 구성된 평균 n-스텝 리턴은 다음과 같다.

$$\frac{1}{2}G^{(2)} + \frac{1}{2}G^{(4)} \tag{15}$$

3.2 전방보기(Forward-view) TD 람다(λ)

- λ 리턴 G_t^{λ}는 모든 n-스텝(에피소드가 종료할 때까지) 리턴들($G_t^{(n)}$)을 조합한다.
- 가중치 $(1-\lambda)\lambda^{n-1}$을 사용

$$G_t^{\lambda} = (1-\lambda)\sum_{n=1}^{\infty} \lambda^{n-1} G_t^{(n)} \tag{16}$$

- 전방보기 TD 람다

$$V(S_t) \leftarrow V(S) + \alpha(G_t^\lambda - V(S_t)) \tag{17}$$

- 미래를 보고 업데이트 하는 방식으로 에피소드가 끝나야 할 수 있는 방식
- TD의 장점이 사라지고, MC와 비슷해진다.

3.3 후방보기(Backward-view) TD 람다(λ)

- 전방보기는 이론을 제공하고, 후방보기는 메커니즘을 제공한다.
- 종료되지 않은 시퀀스에서 모든 단계를 온라인으로 업데이트한다.
- 과거를 보고 업데이트하는 방식으로, TD(0)와 TD(λ) 모두의 장점을 가진다.
- TD(λ) 는 대부분 후방보기 TD(λ)를 사용
- **Eligibility(적격, 적임) Traces** : 자주 방문한 상태와 최근에 방문한 상태에 가중치를 높이는 방식

$$\begin{aligned} E_0(s) &= 0 \\ E_t(s) &= \gamma\lambda E_{t`1}(s) + 1(S_t = s) \end{aligned} \tag{18}$$

- 상태 s에 방문 할 때마다 eligibility trace를 유지 한다.
- 상태 s에 방문 할 때마다 $V(s)$를 업데이트
- TD 에러 δ_t(델타)와 eligibility trace $E_t(s)$는 비례하므로,

$$\begin{aligned} \delta_t &= R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \\ V(s) &\leftarrow V(s) + \alpha\delta_t E_t(s) \end{aligned} \tag{19}$$