Time Different learning

TD 예측

Monte Carlo

$$V(S_t) \leftarrow V(S_t) + \alpha\,[G_t - V(S_t)]$$

G_t : 시각 t 이후의 실제 이득

- 하나의 에피소드가 끝날 때에 $V(S_t)$의 증가량을 결정하기 위한 실제 이득을 알 수 있다.

Time-Difference

$$V(S_t) \leftarrow V(S_t) + \alpha[\boldsymbol{R_{t+1} + \gamma V(S_{t+1})} - \boldsymbol{V(S_t)}]$$

- 다음 시간 단계(t+1)까지 기다리면 실제이득을 알 수 있다.
- T+1 시각에서 목표를 형성하고 관측된 보상 R_{t+1} 과 추정값 $V(St_{+1})$ 으로 갱신(update) 한다

TD error

- Update : 이후 상태가치와 그 때까지의 이득을 이용하여 이후 상태에 대한 표본을 예측, 원래 상태의 가치를 그에 따라 update하는 것을 포함
- 매 시각 t에서 TD error는 그 시각에 만들어진 추정값의 오차
- 다음 상태와 다음 보상에 의존하기 때문에 한 시간 단계가 지나기 전에는 알 수 없다

$$\delta_t = \boldsymbol{R}_{t+1} + \boldsymbol{\gamma V}(\boldsymbol{S}_{t+1}) - \boldsymbol{V}(\boldsymbol{S}_t)$$

TD 예측 방법의 좋은 점

추측으로부터 추측을 학습한다

– $environment, model, reward,$
$probability\ distribution\ for\ next\ state$ 가 필요 없다.

Online 에서 사용할 수 있다.

- 에피소드가 매우 길거나 연속적인 작업이라 에피소드가 없는 경우, TD는 전이(t -> t+1)로부터 학습하기 때문에 에피소드에 덜 영향을 받는다.

실제 결과를 기다리지 않고 다음 추측으로부터 지금 추측을 학습해도 올바른 결과로 수렴한다

- 확률론적인 문제에서 TD 방법이 고정 α 몬테카를로보다 빠르다.

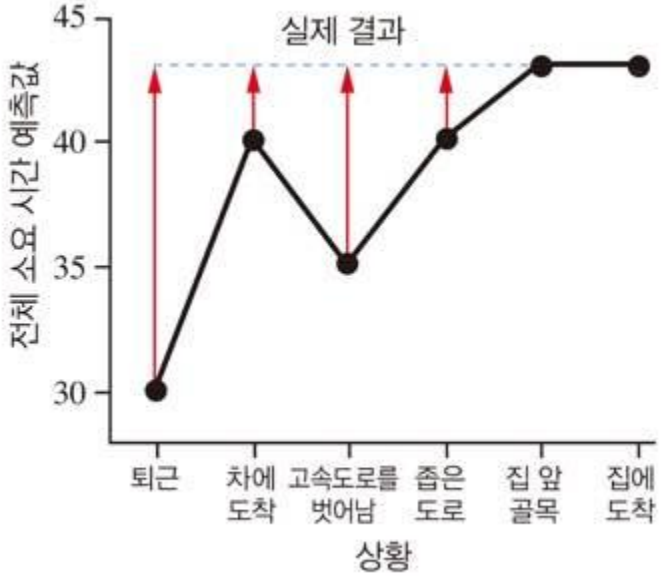
운전해서 집에 가기

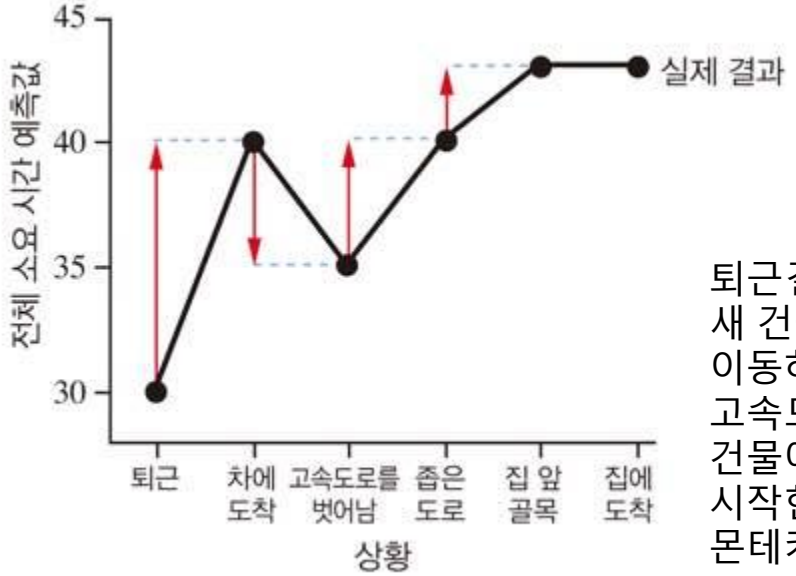
보상 : 퇴근길의 각 구간에서 소요된 시간.
각 상태에서 보상 : 앞으로 더 가야하는 실제 시간
상태의 가치 : 가야할 시간의 기댓값

상태	경과 시간(분)	주행 시간 예측값	전체 소요 시간 예측값
퇴근, 금요일엔 6시	0	30	30
차에 도착, 비가 옴	5	35	40
고속도로를 벗어남	20	15	35
좁은 도로, 트럭 뒤	30	10	40
집 앞 골목 진입	40	3	43
집에 도착	43	0	43

운전해서 집에 가기

MC : 초기 상태에 대해 추정값을 증가시키려면 집에 도착할 때까지 기다려야한다.
TD : 오차는 시간에 따른 예측값의 변화량-시간차에 비례





퇴근길에 운전한 경험이 많다. 새 건물과 새 주차장으로 이동하는데, 같은 장소에서 고속도로로 들어선다. 새 건물에 대해 예측을 학습하기 시작한다. TD 갱신이 몬테카를로보다 좋은가?

TD 최적성

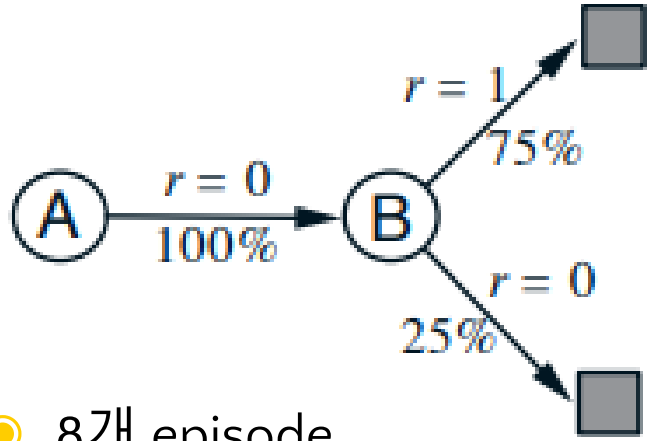
- 데이터 묶음(batch), MRP
- V(A) 의 optimal estimation
- 1. V(A) = ¾ ?

- A 상태에서 100% B로 감

- 2. V(A) = 0 ?

- RMS를 도출하는 답변

- V(B)의 최적값 = 3/4



- 8개 episode

TD 최적성

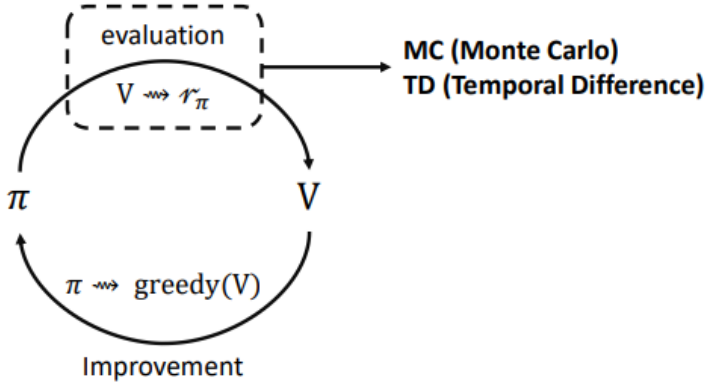
- TD <- MP의 최대 공산 모델에 대해 올바른 추정값
- **최대 공산 추정값** : 데이터를 생성할 확률이 가장 큰 파라미터 값
- 관측된 에피소드로부터 분명한 방식으로 형성된 마르코프 과정의 모델
- I로부터 관측된 전이 중 j로 이동하는 비율은 i로부터 j로의 전이 확률 추정값, 보상의 기댓값은 이 전이에서 관측된 보상의 평균
- 올바른 모델이면 올바른 가치 함수의 추정값을 구할 수 있다.

- MC <- 훈련 데이터에 대해 MSE 최소화하는 추정값 찾기
- 최적의 추정값에 도달하지 못하겠지만, 더 좋은 추정값을 향해 이동하기 때문에 MC보다 빠르게 최적의 추정값에 가까워진다.
- 추정값 근사에 메모리가 상태개수= n수준에서 구할 수 있다.

Generalized policy iteration

GPI

정책 평가와 정책 개선(update, improvement)를 번갈아 한번씩
수행하면서 가치함수가 최적 optimal 가치함수에 수렴할 때까지 연산

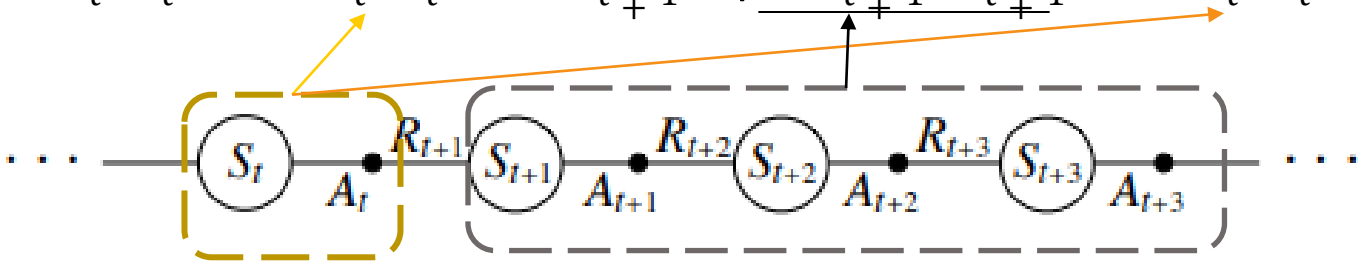


SARSA

행동 가치함수(Q value)를 TD 기반으로 학습한다.

정책 평가와 정책 개선(update, improvement)를 번갈아 한번씩
수행하면서 가치함수가 최적 optimal 가치함수에 수렴할 때까지 연산

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha [R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)]$$



• TD(0) SARSA pseudo code

모든 $s \in S, a \in A(s)$ 에 대해 $Q(s,a)$는 임의의 값으로 초기화 및 $Q(terminal-state,\cdot) = 0$으로 초기화

각 에피소드에 대해 반복:

s를 초기화

현재 상태에서 행동을 선택하는 정책 (=행동정책)

s에서 Q 로 부터 도출된 정책 (예: ϵ-탐욕)으로 행동 a 를 선택

에피소드의 각 단계에서 반복:

행동 a를 취한 후 보상 r과 다음 상태 s'를 관측

행동가치함수를 학습하기 위해 다음 상태 s'에서 취할 행동 a'을 선택하는 정책 (=타겟정책)

s'에서 Q 로 부터 도출된 정책 (예: ϵ-탐욕)으로 행동 a' 를 선택

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma Q(s',a') - Q(s,a)]$$

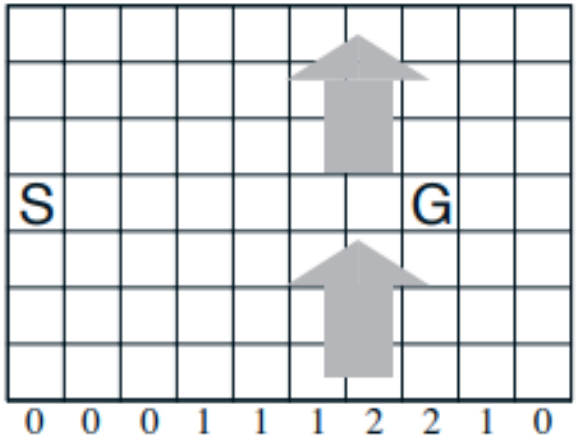
$$s \leftarrow s'; a \leftarrow a';$$

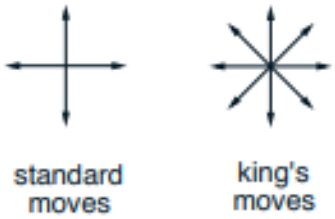
s 가 마지막 상태라면 종료

출처: 강화학습과 수학적 알고리듬

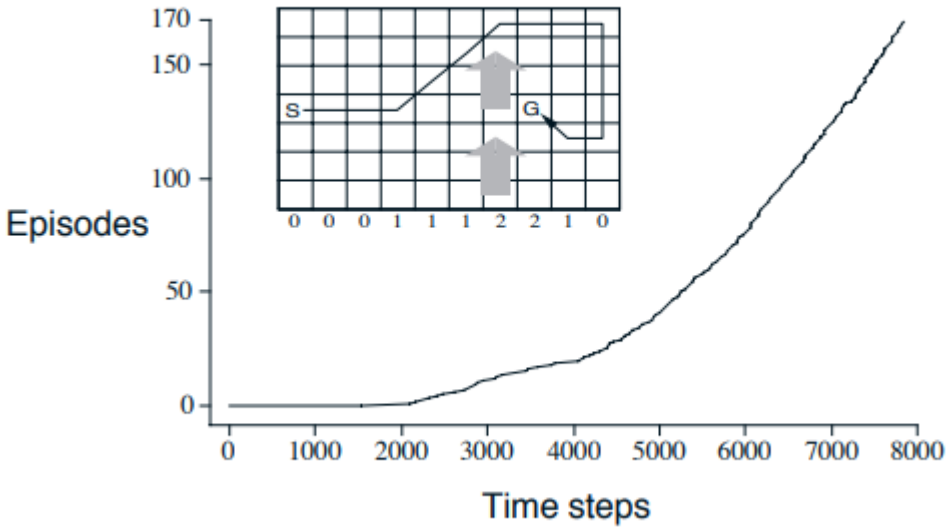
왕이 이동하는 바람이 부는 격자

- 아래에 적힌 수는 아래에서 위로 부는 바람의 세기
- 1인 열에서 오른쪽으로 이동하면, 오른쪽 위 1칸 이동





왕이 이동하는 바람이 부는 격자



- Monte Carlo (MC)를 사용할 수 없음
- MC는 종결 상태까지 도달한 종료된 에피소드만 사용가능
- 일반적인 grid world와 달리 바람에 의하여 종결 상태 G에 도달하지 못하는 에피소드 존재 가능

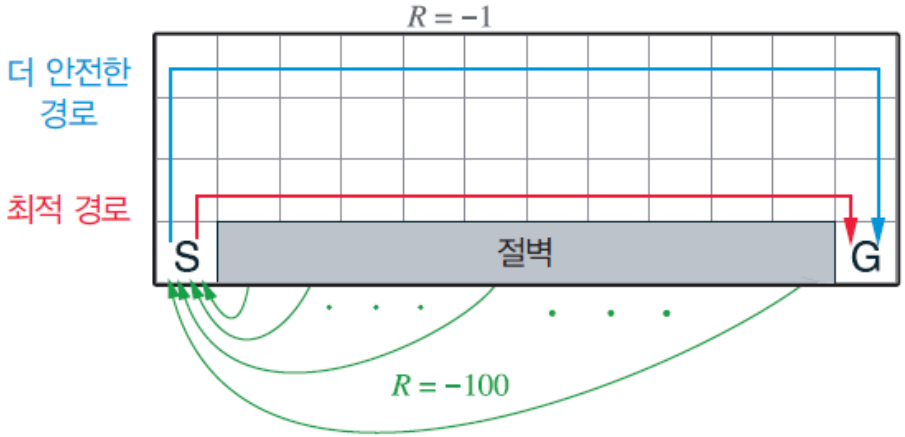
Q-learning

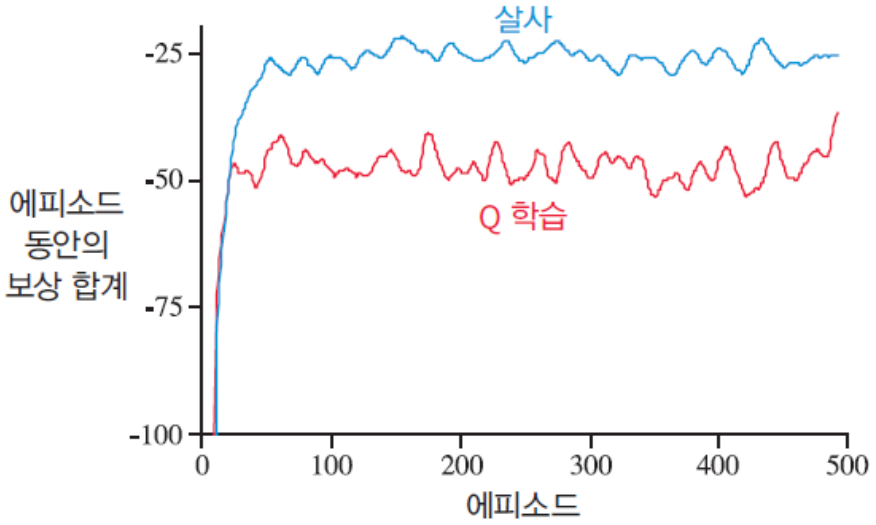
- 가치 반복(value iteration)
- SARSA와 달리 매번 다음 상태에서 행동은 행동가치함수를 극대화하는 행동을 선택

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left(r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right)$$

- SARSA : $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha [R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)]$

절벽 걷기


R = −1
더 안전한
경로
최적 경로
S
절벽
G
R = −100


살사
Q 학습
에피소드
동안의
보상 합계
-25
-50
-75
-100
0
100
200
300
400
500
에피소드