# Bellman Equation

Kihwan Lee

# Bellman Equation

**Bellman Equation이 무엇인가?**

컴퓨터가 계산하기 위해 수학적으로 잘 정리한 것이 MDP

=> 강화학습에서는 이 MDP를 기반

=> 이 MDP를 해결하기 위해 가장 기초적인 개념이 value ft

=> 이 value ft을 계산하기 위한 것이 Bellman Equation

# Bellman Equation

**Bellman Equation이 무엇인가?**

컴퓨터가 계산하기 위해 수학적으로 잘 정리한 것이 MDP

=> 강화학습에서는 이 MDP를 기반

=> 이 MDP를 해결하기 위해 가장 기초적인 개념이 value ft

=> 이 value ft을 계산하기 위한 것이 Bellman Equation

**즉! 강화학습에서 MDP를 해결하기 위해 Bellman Equation을 통해 value ft을 계산하여,**
**이를 통해 optimal한 policy를 찾는 것이 목표이다.**

# Bellman Equation

**Bellman Equation이 무엇인가?**

컴퓨터가 계산하기 위해 수학적으로 잘 정리한 것이 MDP

=> 강화학습에서는 이 MDP를 기반

=> 이 MDP를 해결하기 위해 가장 기초적인 개념이 value ft

=> 이 value ft을 계산하기 위한 것이 Bellman Equation

**State Value Function**

$v\pi(s) = E[Gt|St=s]$

**(State) Action Value Function**

$q\pi(s,a) = E[Gt|St=s,At=a]$

++ **Adv ft** = q - v

**즉! 강화학습에서 MDP를 해결하기 위해 Bellman Equation을 통해 value ft을 계산하여,**

**이를 통해 optimal한 policy를 찾는 것이 목표이다.**

# 확률 이론 두 가지

## 1) Law of total probability

- $P(A) = P(A \cap (\cup B_n)) = \sum_n P(A \cap B_n) = \sum_n P(A|B_n) \cdot P(B_n)$
- $E[X] = \sum_x x \cdot P(X=x) = \sum_y E[X|Y=y] \cdot P(Y=y).$
- $E[X|Z=z] = \sum_x x \cdot P(X=x|Z=z) = \sum_y E[X|Z=z, Y=y] \cdot P(Y=y|Z=z).$

## 2) Law of large numbers

표본의 크기 n이 커질 수록, 표본 평균이 모평균에 수렴

$$\bar{X}_n = \frac{X_1 + \cdots + X_n}{n} \quad , \quad \lim_{n\to\infty} \bar{X}_n = \mu$$

# Bellman Equation 계산

## 1. Bellman expected Equation

1) State value funtion

$$V_\pi(s) = E[G_t \mid S_t = s]$$ - 정의.

$$= \sum_a \underbrace{E[G_t \mid S_t = s, A_t = a]}_{q_\pi(s,a)} \cdot \underbrace{P(A_t = a \mid S_t = s)}_{\pi(a|s)}$$ - 확률이론 1. (action의 합)

$$= \sum_a \pi(a|s) \cdot q_\pi(s,a).$$

$$= \sum_a \pi(a|s) \cdot E_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a]$$ - $G_t = R_{t+1} + \gamma G_{t+1}$ (return의 정의)

$$= \sum_a \pi(a|s) \cdot \sum_{s',r} E_\pi[\underbrace{R_{t+1}}_{= r} + \gamma G_{t+1} \mid \underbrace{S_t = s, A_t = a}_{\text{past}}, S_{t+1} = s', R_{t+1} = r] \cdot P(S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t = a).$$ - 확률이론 1 (s', r의 합)

$$= \sum_a \pi(a|s) \cdot \sum_{s',r} p(s', r \mid s, a) \cdot [r + \gamma E_\pi[G_{t+1} \mid \underbrace{S_{t+1} = s'}_{\text{present}}]]$$ - MDP에서 past는 영향 x.

$$= \sum_a \pi(a|s) \cdot \sum_{s',r} p(s', r \mid s, a) \cdot [r + \gamma \cdot v_\pi(s')].$$ - value ft으로 치환.

현재 state s에서 policy와 transition probability에 의해 결정되는 a, r, s'이다. 즉 앞의 수식들은 a, r, s'이 나올 확률을 곱해준 것.

$$= E[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s].$$

# Bellman Equation 계산

## 1. Bellman expected Equation

1) State value funtion

$$
\begin{aligned}
V(S_t) &= E[G_t \mid S_t = s] \\
&= E[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \mid S_t = s] \\
&= E[R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \cdots) \mid S_t = s] \\
&= E[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\
&= E[R_{t+1} + \gamma V(S_{t+1}) \mid S_t = s]
\end{aligned}
$$

# Bellman Equation 계산

## 1. Bellman expected Equation 을 통해 변경함에 따른 장점?

1) State value funtion

Bellman expectation Equation은 State-value function을 점화식으로 바꿔주는데 immediate reward Rt+1과 discounted next state value γvπ(St+1)의 합으로 분해한 것

=> 이제는 더 이상 return이 필요하지 않습니다! 대신 Rt+1과 St+1만 알면 되는 것 입니다.

=> 이는 에피소드가 전체가 다 끝나지 않아도 계산할 수 있다는 장점이 존재

# Bellman Equation 계산

## 1. Bellman expected Equation

2) State action value funtion

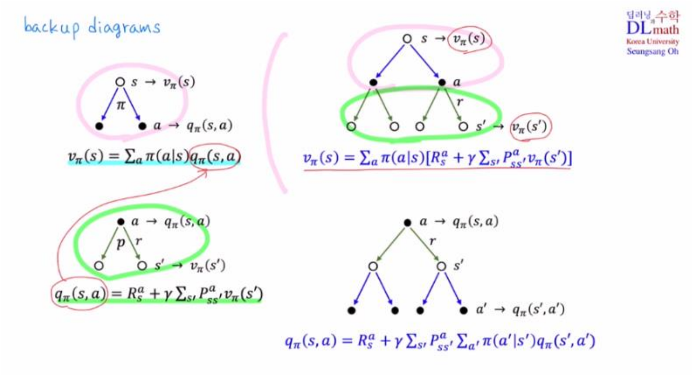
$$q_\pi(s,a) = E_\pi[G_t \mid S_t = s, A_t = a].$$

$$= \sum_{s',r} p(s',r \mid s,a)\left[r + \gamma \sum_{a'} \pi(a' \mid s')\, q_\pi(s',a')\right].$$ – action이 정해져 있기에 policy X.

$$= E_\pi[R_{t+1} + \gamma\, q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a].$$

# Bellman Equation 계산

## 1. Bellman expected Equation



# Bellman Equation 계산

## 2. Bellman optimality Equation

value ft을 찾는 것도 중요하지만, 우리의 최종 목표는 reward를 최대화 시키는 policy 자체를 찾는 것!

=> 이를 optimal policy라고 하며, 이를 찾기 위해 optimal state value ft과 optimal action value ft을 이용

=> bellman optimality equation

# Bellman Equation 계산

## 2. Bellman optimality Equation

value ft을 찾는 것도 중요하지만, 우리의 최종 목표는 reward를 최대화 시키는 policy 자체를 찾는 것!

=> 이를 optimal policy라고 하며, 이를 찾기 위해 optimal state value ft과 optimal action value ft을 이용

=> bellman optimality equation

**value ft들 중에서 maximum이 되는 것이 optimal value ft. 최적 가치 함수.**

**이 optimal value ft으로 optimal policy를 찾게 되며,**

**이를 찾게 되는 것이 Markov Decision process를 해결한 것**

# Bellman Equation 계산

## 2. Bellman optimality Equation

value ft을 찾는 것도 중요하지만, 우리의 최종 목표는 reward를 최대화 시키는 policy 자체를 찾는 것!

=> 이를 optimal policy라고 하며, 이를 찾기 위해 optimal state value ft과 optimal action value ft을 이용

=> bellman optimality equation

**value ft들 중에서 maximum이 되는 것이 optimal value ft. 최적 가치 함수.**

**이 optimal value ft으로 optimal policy를 찾게 되며,**

**이를 찾게 되는 것이 Markov Decision process를 해결한 것**

**=> MDP는 항상 적어도 하나의 optimal policy가 존재!!**

# Bellman Equation 계산

## 2. Bellman optimality Equation

1) Optimal State value funtion

$$V_*(s) = \max_{a \in A(s)} q_*(s,a) = \max_a E_{\pi_*}[G_t \mid S_t = s, A_t = a]$$

$$= \max_a E_{\pi_*}[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a]$$ – return이 축용되기에 episode 끝날 때까지 고려

$$= \max_a E[R_{t+1} + \gamma V_*(S_{t+1}) \mid S_t = s, A_t = a]$$ – immediate reward와 next state만 고려

$$\max_a \sum_{s',r} p(s',r \mid s,a)[r + \gamma V_*(s')]$$ – Sum으로 분해 $\iff V_\pi(s) = \sum_a \pi(a \mid s) \sum_{s',a} P(s',r \mid s,a)[r + \gamma V_\pi(s')]$.

immediate reward, next state 의 추출

# Bellman Equation 계산

## 2. Bellman optimality Equation

1) Optimal state action value funtion

$$q_*(s,a) = E_{\pi^*}[G_t \mid S_t = s, A_t = a] = E[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') \mid S_t = s, A_t = a].$$

$$= \sum_{s',r} p(s', r \mid s, a)[r + \gamma \max_{a'} q_*(s', a')].$$

# Bellman Equation 계산

## 2. Bellman optimality Equation