동아대학교 DONG-A UNIVERSITY

[실증적SW개발프로젝트]

RLHF기반 로봇 팔 제어 프로그램 개발

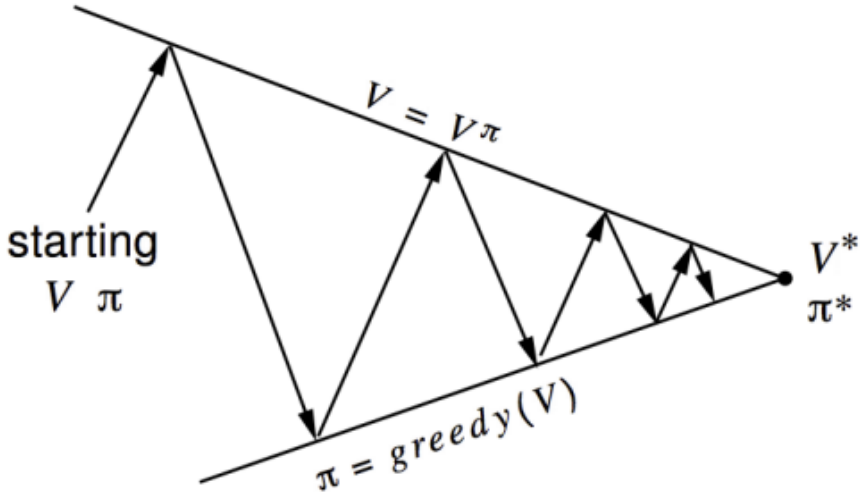
2143841 권은주
1824751 진현석
2051505 조현진

CONTENTS

A *Markov Decision Process* is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

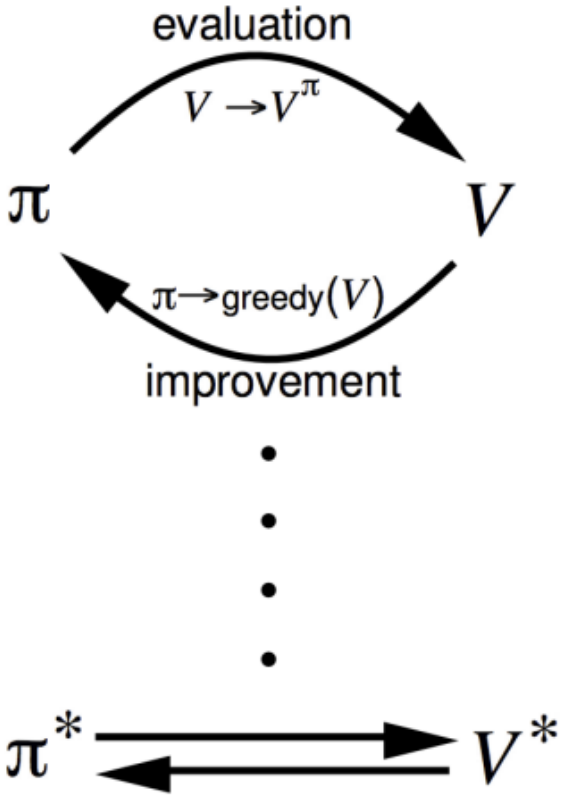
- Model Free
  - Policy and/or Value Function
  - No Model
- Model Based
  - Policy and/or Value Function
  - Model

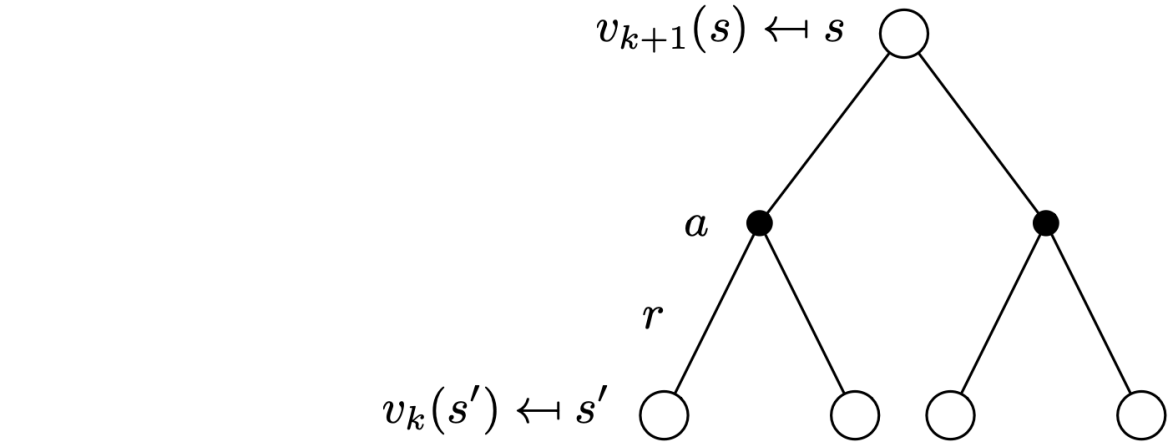
- Dynamic programming assumes full knowledge of the MDP
- It is used for *planning* in an MDP
- For prediction:
  - Input: MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ and policy π
  - or: MRP $\langle \mathcal{S}, \mathcal{P}^{\pi}, \mathcal{R}^{\pi}, \gamma \rangle$
  - Output: value function v_{π}
- Or for control:
  - Input: MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$
  - Output: optimal value function v_{*}
  - and: optimal policy π_{*}



Policy evaluation Estimate v_π
Iterative policy evaluation

Policy improvement Generate $\pi' \geq \pi$
Greedy policy improvement





$$v_{k+1}(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_k(s') \right)$$

$$\mathbf{v}^{k+1} = \boldsymbol{\mathcal{R}}^{\boldsymbol{\pi}} + \gamma \boldsymbol{\mathcal{P}}^{\boldsymbol{\pi}} \mathbf{v}^k$$

- Consider a deterministic policy, $a = \pi(s)$
- We can *improve* the policy by acting greedily

$$\pi'(s) = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, q_\pi(s, a)$$

- This improves the value from any state s over one step,

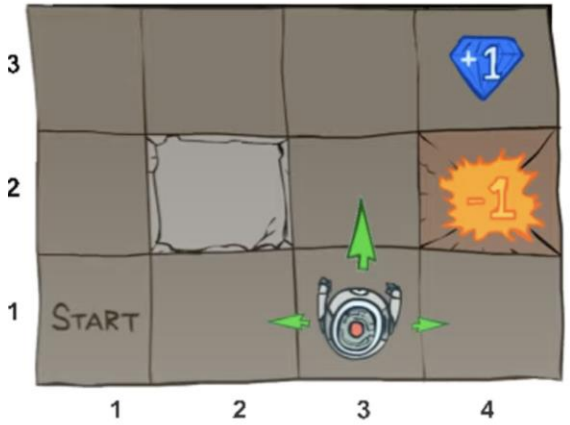
$$q_\pi(s, \pi'(s)) = \max_{a \in \mathcal{A}} q_\pi(s, a) \geq q_\pi(s, \pi(s)) = v_\pi(s)$$

- It therefore improves the value function, $v_{\pi'}(s) \geq v_\pi(s)$

$$\begin{aligned} v_\pi(s) &\leq q_\pi(s, \pi'(s)) = \mathbb{E}_{\pi'}\left[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s\right] \\ &\leq \mathbb{E}_{\pi'}\left[R_{t+1} + \gamma q_\pi(S_{t+1}, \pi'(S_{t+1})) \mid S_t = s\right] \\ &\leq \mathbb{E}_{\pi'}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 q_\pi(S_{t+2}, \pi'(S_{t+2})) \mid S_t = s\right] \\ &\leq \mathbb{E}_{\pi'}\left[R_{t+1} + \gamma R_{t+2} + \ldots \mid S_t = s\right] = v_{\pi'}(s) \end{aligned}$$

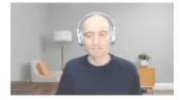
Value Iteration

$$V_2(s) \leftarrow \max_a \sum_{s'} P(s'|s,a)(R(s,a,s') + \gamma V_1(s'))$$



Noise = 0.2
Discount = 0.9

k = 1

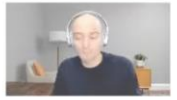


Value Iteration

$$V_2(s) \leftarrow \max_a \sum_{s'} P(s'|s,a)(R(s,a,s') + \gamma V_1(s'))$$

k = 2

Noise = 0.2
Discount = 0.9

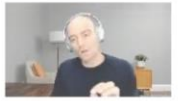


Value Iteration

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} P(s'|s,a)(R(s,a,s') + \gamma V_k(s'))$$

k = 4

Noise = 0.2
Discount = 0.9



k = 10


0.64
0.74
0.85
1.00
0.56
0.57
-1.00
0.48
0.41
0.47
0.27
VALUES AFTER 10 ITERATIONS

k = 100


0.64
0.74
0.85
1.00
0.57
0.57
-1.00
0.49
0.43
0.48
0.28
VALUES AFTER 100 ITERATIONS

- **Monte-Carlo (MC)**

$$V(S_t) \leftarrow V(S_t) + \alpha\left(G_t - V(S_t)\right)$$

agent가 episode들을 끝까지 진행하도록 만든 후 그 결과에 따라 $V(s)$를 평가

- **Temporal-Difference Learning (TD)**

$$V(S_t) := V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$$

현재 시점 t에서 상태 St의 가치 $V(St)$를 바로 다음 시점 $t+1$에 수정하는 방식

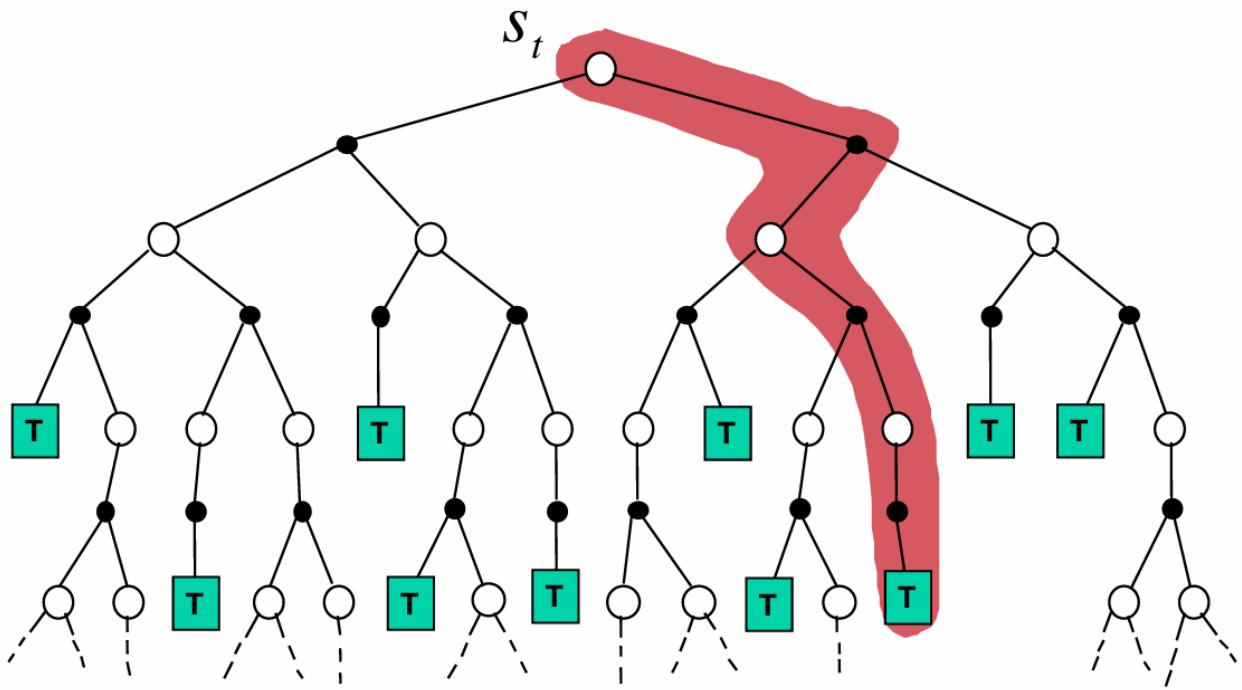
- **Bias-Variance Trade-off**

- MC has high variance, zero bias
  - Good convergence properties
  - (even with function approximation)
  - Not very sensitive to initial value
  - Very simple to understand and use
- TD has low variance, some bias
  - Usually more efficient than MC
  - TD(0) converges to $v_\pi(s)$
  - (but not always with function approximation)
  - More sensitive to initial value

- **Monte-Carlo 시각화**

Monte-Carlo Backup

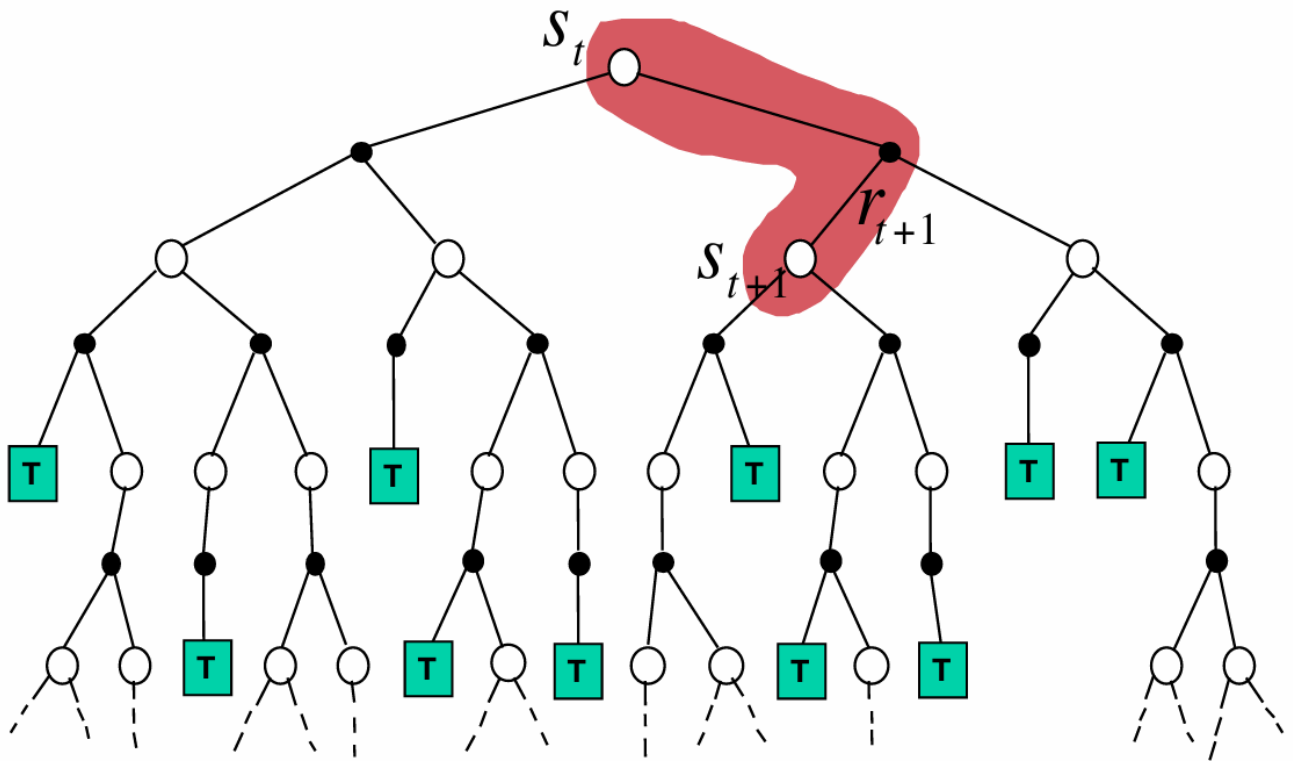
$$V(S_t) \leftarrow V(S_t) + \alpha\left(G_t - V(S_t)\right)$$



- **Temporal-Difference 시각화**

Temporal-Difference Backup

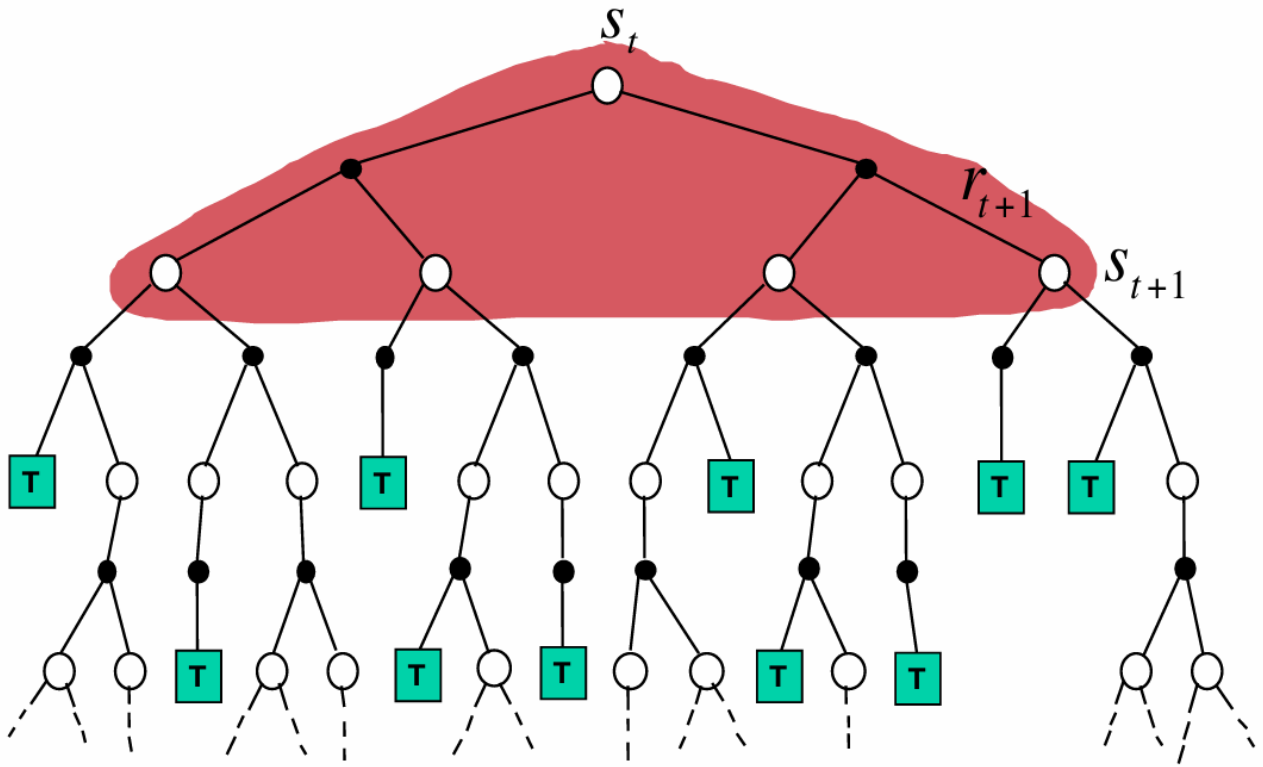
$$V(S_t) \leftarrow V(S_t) + \alpha\left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right)$$



- **Dynamic Programming 시각화**

Dynamic Programming Backup

$$V(S_t) \leftarrow \mathbb{E}_\pi \left[R_{t+1} + \gamma V(S_{t+1}) \right]$$



- **Bootstrapping and Sampling**

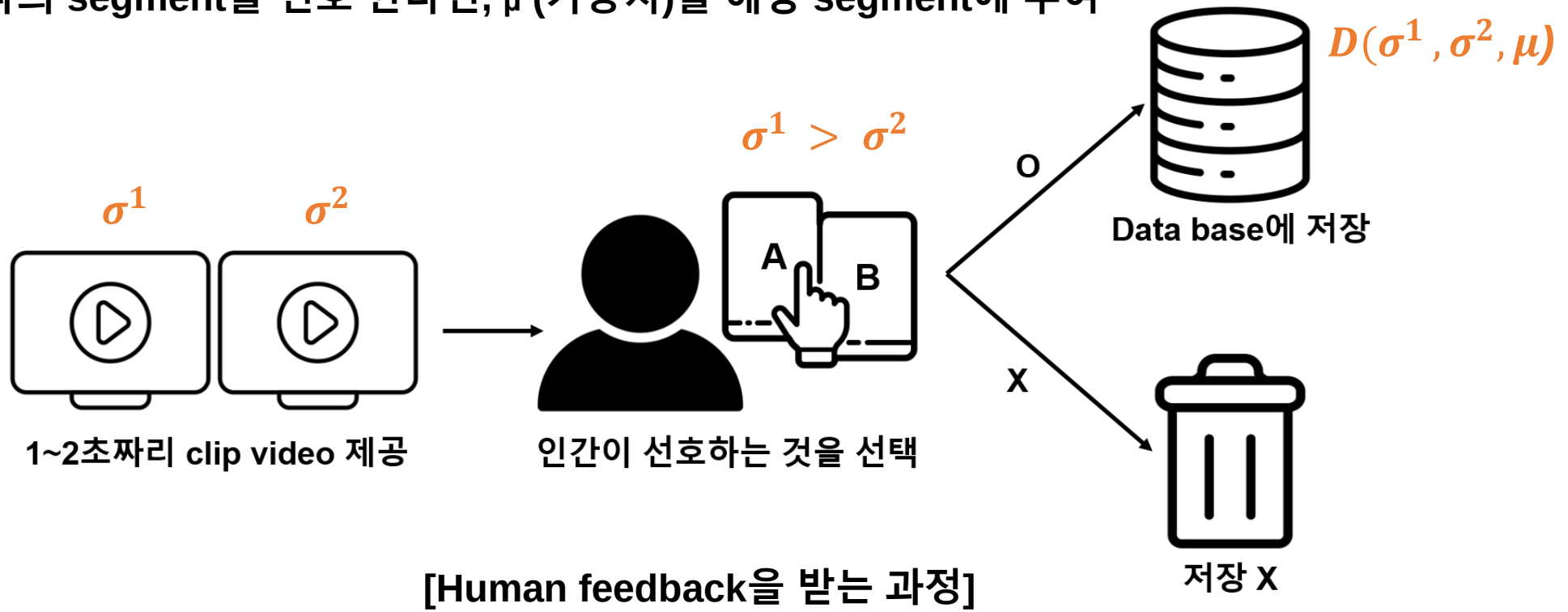
- Bootstrapping: update involves an estimate
  - MC does not bootstrap
  - DP bootstraps
  - TD bootstraps
- Sampling: update samples an expectation
  - MC samples
  - DP does not sample
  - TD samples

- Introduction - RL의 단점
  - RL이 좋은 성과를 보여준 사례는 reward function이 잘 정의된 영역에서 이루어짐.
  - 많은 task는 복잡하거나, 제대로 정의되어 있지 않거나 어려운 목표를 포함하고 있음.
  - 의도한 행동을 포착하는 간단한 reward function reward hacking으로 이어질 수 있음.
  - 이러한 단점을 극복하는 것은 RL의 사용 가능한 분야를 확대 시킬 수 있음.

- Introduction - 인간의 의도 또는 선호도를 충족 시키는 방법
  - 인간의 직접적인 개입이 필요한 방법들은 다양한 환경에 적용하기 어려움
  - Behavior Cloning
  - Interactive Imitation learning (Dagger)

- Introduction – 논문에서 제시하는 RL의 단점을 해결하기 위한 조건
  - 인간이 원하는 행동을 인식만 할 수 있고, 시연할 수 없는 작업을 해결할 수 있어야 함.
  - 비전문가인 사용자도 에이전트를 교육할 수 있어야 함.
  - 큰 규모의 문제까지 확장 가능해야 함.
  - 사용자 피드백을 효율적으로 활용해야 함.

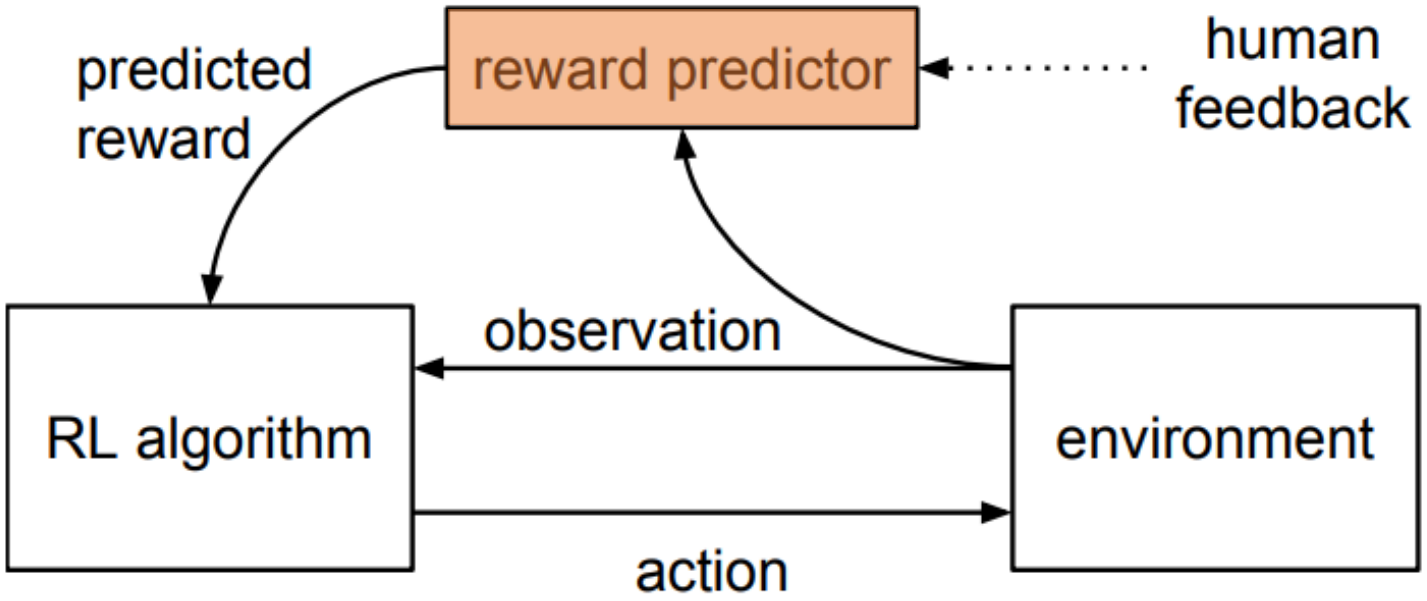
- **Method - Preference Elicitation**
  - 두 개의 비디오 클립을 비교하여 Feedback 제공
  - Trajectory segments: $\sigma = ((o_0, a_0), (o_1, a_1), \ldots (o_{k-1}, a_{k-1})) \in (O \times A)^k$
  - observation: $o_t \in O$, action: $a_t \in A$
  - $\sigma^1 > \sigma^2$ 는 사람이 σ^1 를 선호 한다는 것을 의미
  - 둘 중 하나의 segment를 선호 한다면, μ (가중치)를 해당 segment에 부여



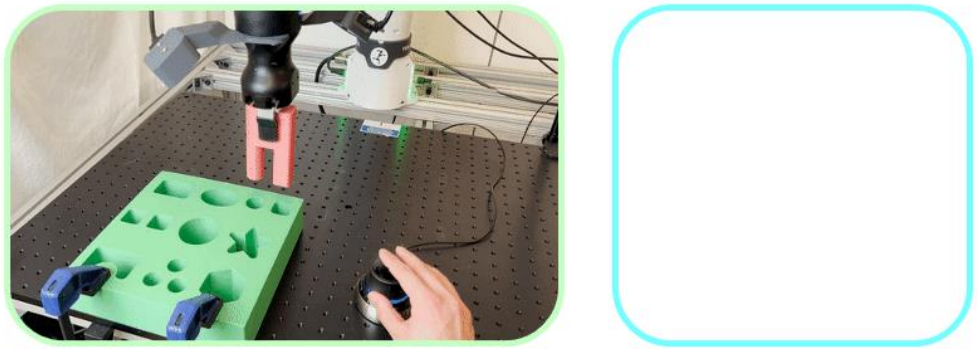
[Human feedback을 받는 과정]

- **Method - Fitting the Reward Function**

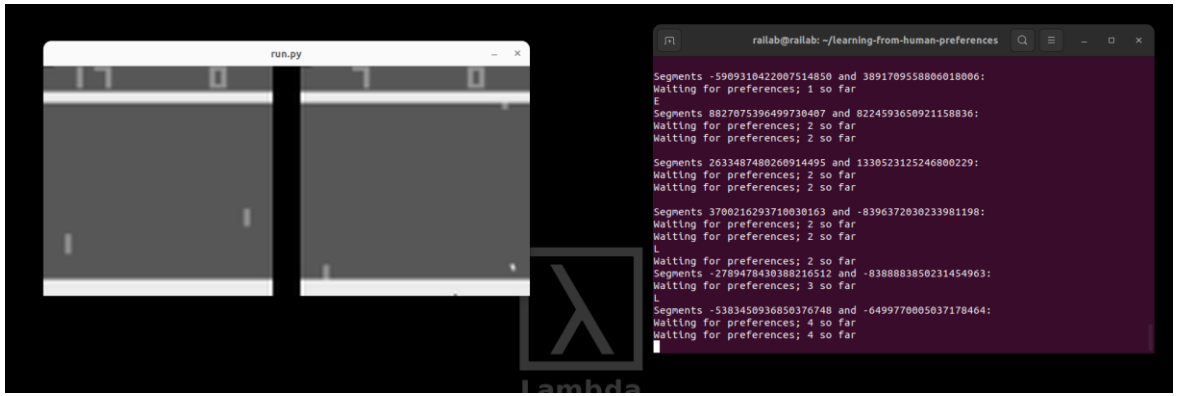
  - $\hat{P}[\sigma^1 > \sigma^2] = \dfrac{exp(\sum \hat{r}(o_t^1, a_t^1)}{exp(\sum \hat{r}(o_t^1, a_t^1)) + exp(\sum \hat{r}(o_t^2, a_t^2))}$
  - $L(\hat{r}) = -\sum \mu(1)\, log\, \hat{P}[\sigma^1 > \sigma^2] + \sum \mu(2)\, log\, \hat{P}[\sigma^2 > \sigma^1]$
  - **preference predictor:** $\hat{r}$



- 동일한 고차원 환경 RLIF, RLHF 알고리즘 비교
  - RLIF는 인간의 개입을 받아 policy에 대한 피드백을 받는 방법
  - RLHF는 두 개의 trajectory에 대한 선호도를 받는 방법
  - RLIF는 인간이 최적의 policy를 알지 못하는 경우 제대로 된 학습 불가능
  - RLHF는 최적의 행동을 알지 못하더라도 둘 중 더 나은 것에 대한 피드백 가능
  - 복잡한 작업을 수행해야하는 환경에선 RLHF가 유리할 것이라는 가정
  - 동일한 High-dim 환경에서 비교 실험을 통해
  - 복잡한 작업을 수행해야하는 환경에선 RLHF가 유리하다는 것을 증명할 예정



[RLIF 학습 과정]



[RLHF 학습 과정]

실증적AI프로젝트 금주 활동내역

주제: RLHF를 이용한 협동 로봇 제어 프로그램 개발

금주 활동계획	**1. Prof. David Silver 강화학습 3~4주차 정리 후 스터디** **2. RLHF 논문 리뷰**		
	팀장 (권은주)	팀원 1 (조현진)	팀원 2 (진현석)
금주 개인별 활동내역	1. 3~4주차 개념정리 2. RLHF 논문 리뷰 3. 활동내역을 바탕으로 한 스터디 진행	1. 3~4주차 개념정리 2. RLHF 논문 리뷰 3. 활동내역을 바탕으로 한 스터디 진행	1. 3~4주차 개념정리 2. RLHF 논문 리뷰 3. 활동내역을 바탕으로 한 스터디 진행
차주 활동계획	1. Prof. David Silver 강화학습 5주차 강의 정리 후 스터디 2. Atari Game 실습 3. 실험용 가상환경 구성		

QUESTIONS & ANSWERS

Dept. of AI, Dong-A University

권은주 (kkkoj4284@donga.ac.kr)

진현석 (cpu132465@donga.ac.kr)

조현진 (gkfkgkdh@naver.com)

Github (https://github.com/eunjuyummy/AI_Project_CoRLHF)