2021-2 한양대학교 HAI

강화학습 부트캠프

Chris Ohk

utilForever@gmail.com

오늘 배울 내용

- Trust Regions #1
    - Trust Region Policy Optimization (TRPO)

TRPO – Introduction

- 강화학습의 목표

$$\max_{\pi} \eta(\pi)$$

  - 앞으로 받을 보상의 합을 최대로 하는 정책을 찾는 것

- Conservative Policy Iteration
  - $\eta(\pi)$를 바로 최대로 하는 정책을 찾는 건 대부분 어렵다. Kakade와 Langford (2002)는 $\eta(\pi)$의 성능 향상을 보장하면서 정책을 갱신하는 Conservative Policy Iteration 기법을 제안했다.

$$\max L_{\pi}(\tilde{\pi}) = \sum_{s} \rho_{\pi}(s) \sum_{a} \tilde{\pi}(a|s) A_{\pi}(s, a)$$

  - 이 기법을 사용하면 정책을 갱신함으로 인해 성능을 향상시키지 못하더라도 최소한 성능을 악화시키지 않는다는 게 이론적으로 보장된다.

TRPO – Introduction

- Theorem 1

$$\max L_{\pi_{\text{old}}}(\pi_{\text{new}}) - \frac{4\epsilon\gamma}{(1-\gamma)^2}\alpha^2, \qquad \left(\alpha = D_{TV}^{\max}(\pi_{\text{old}}, \pi_{\text{new}})\right)$$

  - 기존의 Conservative Policy Iteration은 과거 정책과 새 정책을 섞어서 사용하기에 실용적이지 않다는 단점이 있었는데 이를 보완해 온전히 새로운 정책만으로 갱신할 수 있는 기법을 제안한다.

- KL Divergence Version of Theorem 1

$$\max L_{\pi}(\tilde{\pi}) - C \cdot D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta)$$

  - Distance Metric을 KL Divergence로 변경할 수 있다.

TRPO – Introduction

- Using Parameterized Policy

$$\max_{\theta} L_{\pi_{\text{old}}}(\theta) - C \cdot D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta)$$

  - 최적화 문제를 더욱 편리하게 풀 수 있도록 낮은 차원을 갖는 매개 변수들로 변환한 정책을 사용할 수 있다.

- Trust Region Constraint

$$\max_{\theta} L_{\theta_{\text{old}}}(\theta)$$
$$\text{s.t} \quad D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta) \le \delta$$

  - 정책을 갱신할 때 지나치게 많이 변하는 걸 방지하기 위해 Trust Region을 제약 조건으로 설정할 수 있다. 이 아이디어로 인해 TRPO라는 명칭을 갖게 된다.

TRPO – Introduction

- Heuristic Approximation

$$\max_{\theta} L_{\theta_{\text{old}}}(\theta)$$
$$\text{s.t} \quad \overline{D}_{\text{KL}}^{\rho_{\text{old}}}(\theta_{\text{old}}, \theta) \le \delta$$

  - 앞에서 본 식에서 제약 조건은 모든 상태에 대해서 성립해야 하기 때문에 문제를 매우 어렵게 만든다. 이를 좀 더 다루기 쉽게 상태 분포에 대한 평균을 취하는 식으로 변형한다.

- Monte Carlo Simulation

$$\max_{\theta} E_{s\sim\rho_{\theta_{\text{old}}}, a\sim q}\left[\frac{\pi_{\theta}(a|s)}{q(a|s)} Q_{\theta_{\text{old}}}(s,a)\right]$$
$$\text{s.t} \quad E_{s\sim\rho_{\theta_{\text{old}}}}\left[D_{\text{KL}}\left(\pi_{\theta_{\text{old}}}(\cdot|s) \parallel \pi_{\theta}(\cdot|s)\right)\right] \le \delta$$

  - 샘플링을 통한 계산이 가능하도록 식을 다시 표현할 수 있다.

TRPO – Introduction

- Efficiently Solving TRPO

$$\max_{\theta} \nabla_\theta L_{\theta_{\text{old}}}(\theta)\Big|_{\theta-\theta_{\text{old}}} (\theta - \theta_{\text{old}})$$

$$\text{s.t} \quad \frac{1}{2}(\theta_{\text{old}} - \theta)^T A(\theta_{\text{old}})(\theta_{\text{old}} - \theta) \leq \delta$$

$$A_{ij} = \frac{\partial}{\partial \theta_i}\frac{\partial}{\partial \theta_j}\overline{D}_{\text{KL}}^{\rho_{\text{old}}}(\theta_{\text{old}}, \theta)$$

- 문제를 효율적으로 풀기 위해 근사를 적용할 수 있다. 목표 함수는 1차 근사, 제약 조건은 2차 근사를 취하면 효율적으로 문제를 풀 수 있는 형태로 바뀌는데 이를 자연 기울기법(Natural Gradient)으로 풀 수도 있고 켤레 기울기법(Conjugate Gradient)으로 풀 수도 있다.

Preliminaries

- Markov Decision Process (MDP) : $(\mathcal{S}, \mathcal{A}, P, r, \rho_0, \gamma)$
  - $\mathcal{S}$ is a finite set of states
  - $\mathcal{A}$ is a finite set of actions
  - $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the transition probability
  - $r : \mathcal{S} \to \mathbb{R}$ is the reward function
  - $\rho_0 : \mathcal{S} \to \mathbb{R}$ is the distribution of the initial state s_0
  - $\gamma \in (0, 1)$ is the discount factor
  - $\eta(\pi) = E_{s_0, a_0, \ldots}[\sum_{t=0}^{\infty} \gamma^t r(s_t)]$, where $s_0 \sim \rho_0(s_0)$, $a_0 \sim \pi(a_t|s_t)$, $s_{t+1} \sim P(s_{t+1}|s_t, a_t)$
  - $Q_\pi(s_t, a_t) = E_{s_{t+1}, a_{t+1}, \ldots}[\sum_{l=0}^{\infty} \gamma^l r(s_{t+l})]$ is action value function
  - $V_\pi(s_t) = E_{a_t, s_{t+1}, a_{t+1}, \ldots}[\sum_{l=0}^{\infty} \gamma^l r(s_{t+l})]$ is value function
  - $A_\pi(s_t, a_t) = Q_\pi(s_t, a_t) - V_\pi(s_t)$ is advantage function

Useful Identity

- Kakade & Langford (2002)
  - 우리의 목표는 $\eta(\pi)$를 최대가 되도록 만드는 거다.
    하지만 π가 변함에 따라 η이 어떻게 변하는지 알아내기는 쉽지 않다.
  - π를 기존 정책, $\tilde{\pi}$를 새 정책이라고 하자. 이때 η과 정책 업데이트 사이에 다음 관계가 있다는 게 증명되었다.
  - Lemma 1

$$\eta(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s_0, a_0, \cdots \sim \tilde{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t A_\pi(s_t, a_t) \right]$$

Useful Identity

- Kakade & Langford (2002)
  - 증명

$$
\begin{aligned}
&\mathbb{E}_{\tau|\tilde{\pi}}\left[\sum_{t=0}^{\infty}\gamma^t A_\pi(s_t,a_t)\right] \\
&= \mathbb{E}_{\tau|\tilde{\pi}}\left[\sum_{t=0}^{\infty}\gamma^t\left(r(s_t)+\gamma V_\pi(s_{t+1})-V_\pi(s_t)\right)\right] \\
&= \mathbb{E}_{\tau|\tilde{\pi}}\left[-V_\pi(s_0)+\sum_{t=0}^{\infty}\gamma^t r(s_t)\right] \\
&= -\mathbb{E}_{s_0}\left[V_\pi(s_0)\right]+\mathbb{E}_{\tau|\tilde{\pi}}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t)\right] \\
&= -\eta(\pi)+\eta(\tilde{\pi})
\end{aligned}
$$

$$
\begin{aligned}
&E_{\tau|\tilde{\pi}}\left[\sum_{t=0}^{\infty}\gamma^t A_\pi(s_t,a_t)\right] \\
&= E_{\tau|\tilde{\pi}}\left[\sum_{t=0}^{\infty}\gamma^t\left(r(s_t)+\gamma V_\pi(s_{t+1})-V_\pi(s_t)\right)\right] \\
&= E_{\tau|\tilde{\pi}}\left[\left(\sum_{t=0}^{\infty}\gamma^t r(s_t)\right)+\gamma V_\pi(s_1)-V_\pi(s_0)+\gamma^2 V_\pi(s_2)-\gamma V_\pi(s_1)+\gamma^3 V_\pi(s_3)-\gamma^2 V_\pi(s_2)+\cdots\right] \\
&= E_{\tau|\tilde{\pi}}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t)+\gamma V_\pi(s_1)-V_\pi(s_0)+\gamma^2 V_\pi(s_2)-\gamma V_\pi(s_1)+\cdots\right] \\
&= E_{\tau|\tilde{\pi}}\left[-V_\pi(s_0)+\sum_{t=0}^{\infty}\gamma^t r(s_t)\right] \\
&\overset{(a)}{=} -E_{s_0}\left[V_\pi(s_0)\right]+E_{\tau|\tilde{\pi}}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t)\right] \\
&= -\eta(\pi)+\eta(\tilde{\pi}) \\
&\quad\therefore \eta(\tilde{\pi})=\eta(\pi)+E_{s_0,a_0,\ldots\sim\tilde{\pi}}\left[\sum_{t=0}^{\infty}\gamma^t A_\pi(s_t,a_t)\right]
\end{aligned}
$$

Useful Identity

- Kakade & Langford (2002)
  - Lemma 1은 새로운 정책과 기존 정책 사이의 관계를 규정한다.
  - 다음과 같은 식을 정의하고 이를 이용해 Lemma 1을 변형해 보자.
  - (Unnormalized) Discounted Visitation Frequencies

$$\rho_\pi(s) = P(s_0 = s) + \gamma P(s_1 = s) + \gamma^2 P(s_2 = s) + \cdots$$

$$\begin{aligned}
\eta(\tilde{\pi}) &= \eta(\pi) + \sum_{t=0}^{\infty} \sum_{s} P(s_t = s | \tilde{\pi}) \sum_a \tilde{\pi}(a|s) \gamma^t A_\pi(s,a) \\
&= \eta(\pi) + \sum_s \sum_{t=0}^{\infty} \gamma^t P(s_t = s | \tilde{\pi}) \sum_a \tilde{\pi}(a|s) A_\pi(s,a) \\
&= \eta(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) A_\pi(s,a)
\end{aligned}$$

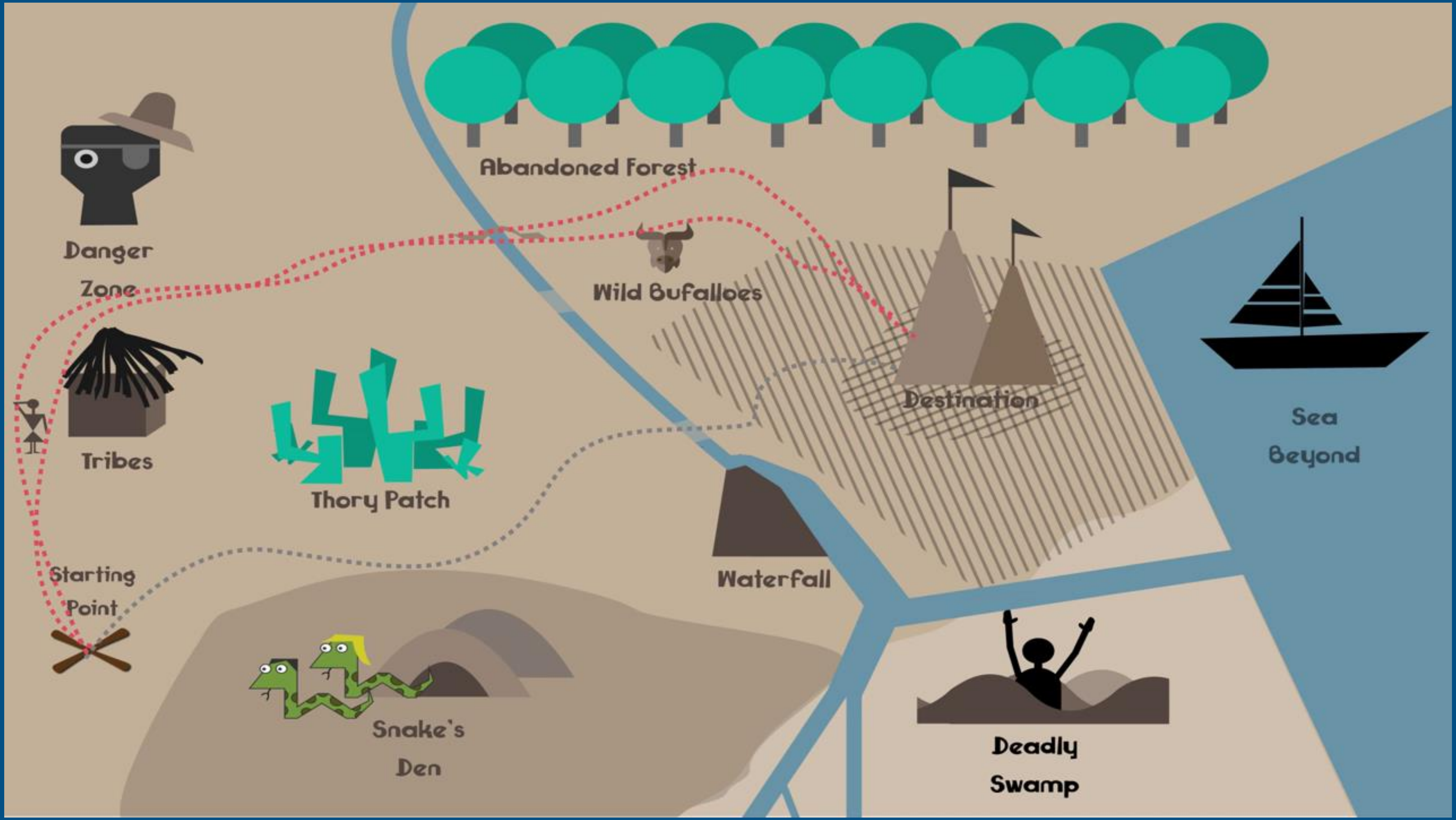
Useful Identity

- Kakade & Langford (2002)
  - 이 수식의 의미는 무엇일까? 만약 $\sum_a \tilde{\pi}(a|s)A_\pi(s,a) \geq 0$이라면 $\eta(\tilde{\pi})$는 항상 $\eta(\pi)$보다 크다.
  - 즉, 정책을 업데이트하면 항상 개선된다.
    → 항상 더 좋은 성능을 내는 정책 업데이트가 가져야 할 특징을 알 수 있다.
  - 다음과 같은 결정적 정책이 있다고 하자.

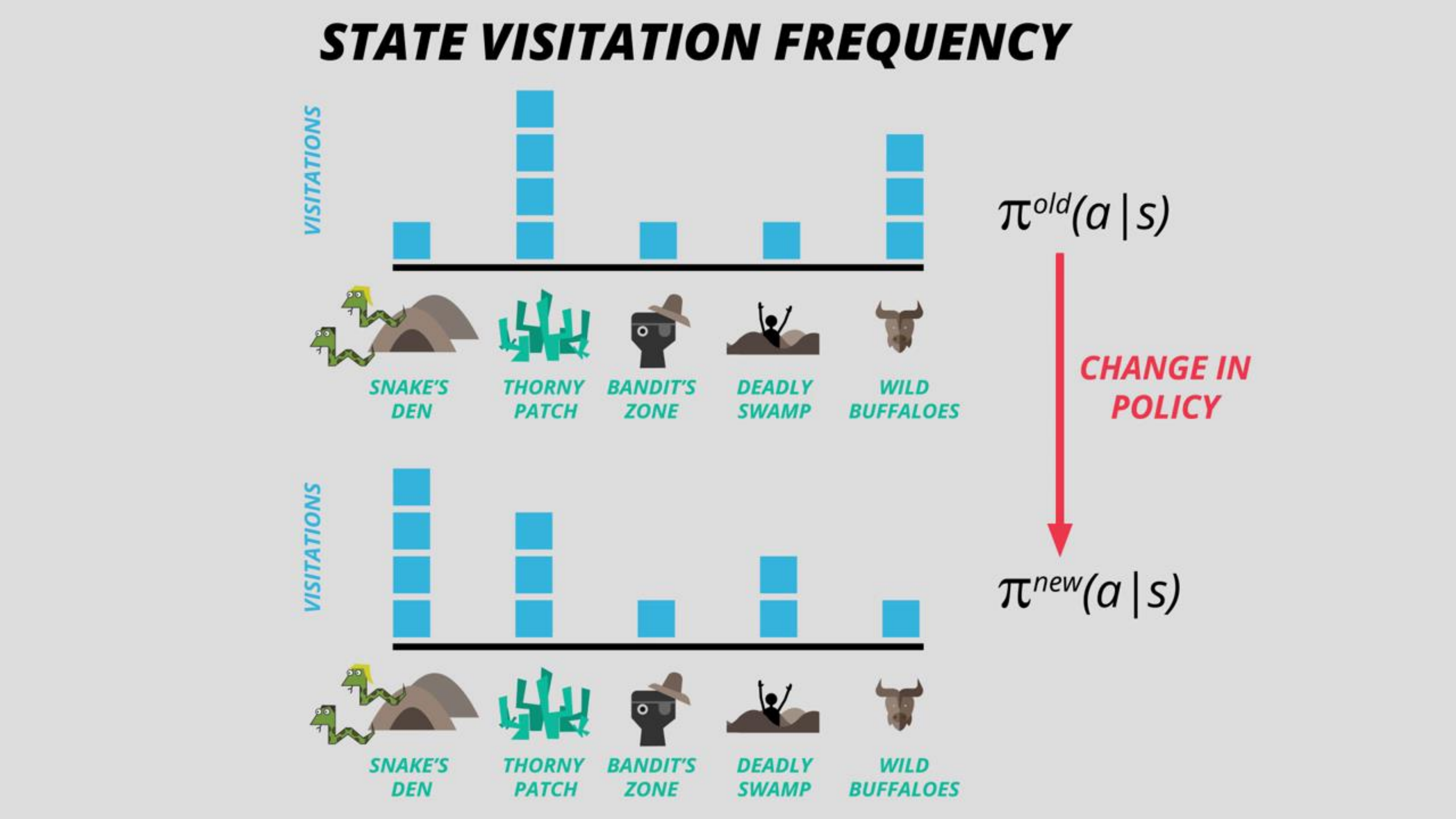
$$\tilde{\pi}(s) = \underset{a}{\operatorname{argmax}} A_\pi(s,a)$$

  - 이 정책은 적어도 하나의 상태-행동 쌍에서 0보다 큰 값을 갖는 Advantage가 있고
    그 때의 확률이 0이 아니라면 항상 성능을 개선시킨다.
  - 문제는 정책을 바꾸면 ρ도 바뀐다는 점이다.

Useful Identity



Useful Identity



Useful Identity

- Kakade & Langford (2002)
  - 이로 인해 정책을 최적화하기 어려워진다. 그래서 정책에 따른 변화를 무시하도록 근삿값을 사용한다.

$$L_\pi(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_\pi(s) \sum_a \tilde{\pi}(a|s) A_\pi(s,a)$$

  - 정책을 바꾸더라도 이전의 정책 분포를 계속 이용하는 방식이다.
    이 방식은 정책의 변화가 크지 않다면 어느 정도 허용할 수 있을 거다.
  - 그렇지만 얼마나 많은 변화까지 허용할 것인가? 이를 정하기 위해 이용하는 게 바로 Trust Region이다.

Conservative Policy Iteration

- 정책의 변화를 쉽게 다룰 수 있도록 다음과 같은 매개 변수를 이용해서 정책을 표현하자.

$\pi_\theta(a|s)$: Parameterized Policy

  - 여기서 π_θ는 θ에 대해 미분 가능한 함수다.
    $L_\pi(\tilde{\pi})$를 θ_0에서 $\eta(\pi)$에 대한 1차 근사라고 하면 다음 식이 성립한다.

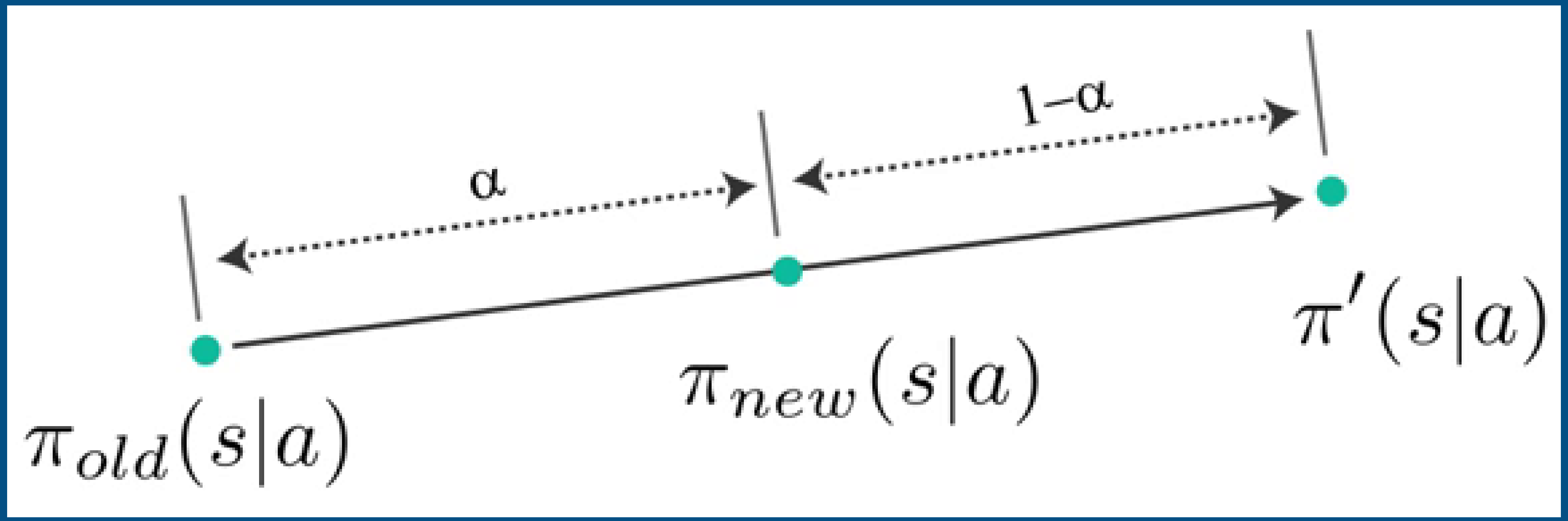
$$L_{\pi_{\theta_0}}(\pi_{\theta_0}) = \eta(\pi_{\theta_0}),$$
$$\nabla_\theta L_{\pi_{\theta_0}}(\pi_\theta)\big|_{\theta=\theta_0} = \nabla_\theta \eta(\pi_\theta)\big|_{\theta=\theta_0}.$$

  - 위 식의 의미는 π_{θ_0}가 매우 작게 변한다면 $L_{\pi_{\theta_0}}$를 개선하는 게 η를 개선한다는 거다.
    그러나 지금까지의 설명만으로는 π_{θ_0}를 얼마나 작게 변화시켜야 할 지에 대해서는 알 수 없다.

Conservative Policy Iteration

- Kakade & Langford는 Conservative Policy Iteration이라는 기법을 제안한다.
  - η 개선의 Lower Bound를 제공
  - 기존의 정책을 π_{old}라고 하고 π'를 $\pi' = \underset{\pi'}{\operatorname{argmax}} L_{\pi_{\text{old}}}(\pi')$과 같이 정의할 때,
    새로운 Mixture Policy π_{new}를 다음과 같이 제안

$$\pi_{\text{new}}(a|s) = (1-\alpha)\pi_{\text{old}}(a|s) + \alpha\pi'(a|s).$$



Conservative Policy Iteration

- Kakade & Langford는 Conservative Policy Iteration이라는 기법을 제안한다.
  - 다음과 같은 Lower Bound를 정의

$$\eta(\pi_{\text{new}}) \geq L_{\pi_{\text{old}}}(\pi_{\text{new}}) - \frac{2\epsilon\gamma}{(1-\gamma)^2}\alpha^2$$

$$\text{where } \epsilon = \max_s \left| \mathbb{E}_{a\sim\pi'(a|s)} \left[A_\pi(s,a) \right] \right|.$$

  - 하지만 Mixture된 정책은 실용적이지 않다.

General Stochastic Policy

- 이전에 설명한 Lower Bound는 오직 Mixture Policy에 대해서만 성립하고 더 많이 사용되는 Stochastic Policy에는 적용되지 않는다. 따라서 Stochastic Policy를 이용할 수 있도록 개선할 필요가 있다.
  - $\alpha \rightarrow$ Distance Measure between π and $\tilde{\pi}$
  - Constant $\epsilon \rightarrow \max_{s,a} |A_\pi(s,a)|$
- 여기서 Distance Measure로 Total Variation Divergence를 이용한다. 이산 확률 분포 p와 q에 대해 다음와 같이 정의된다.

$$D_{\mathrm{TV}}(p \parallel q) = \frac{1}{2} \sum_i |p_i - q_i|$$

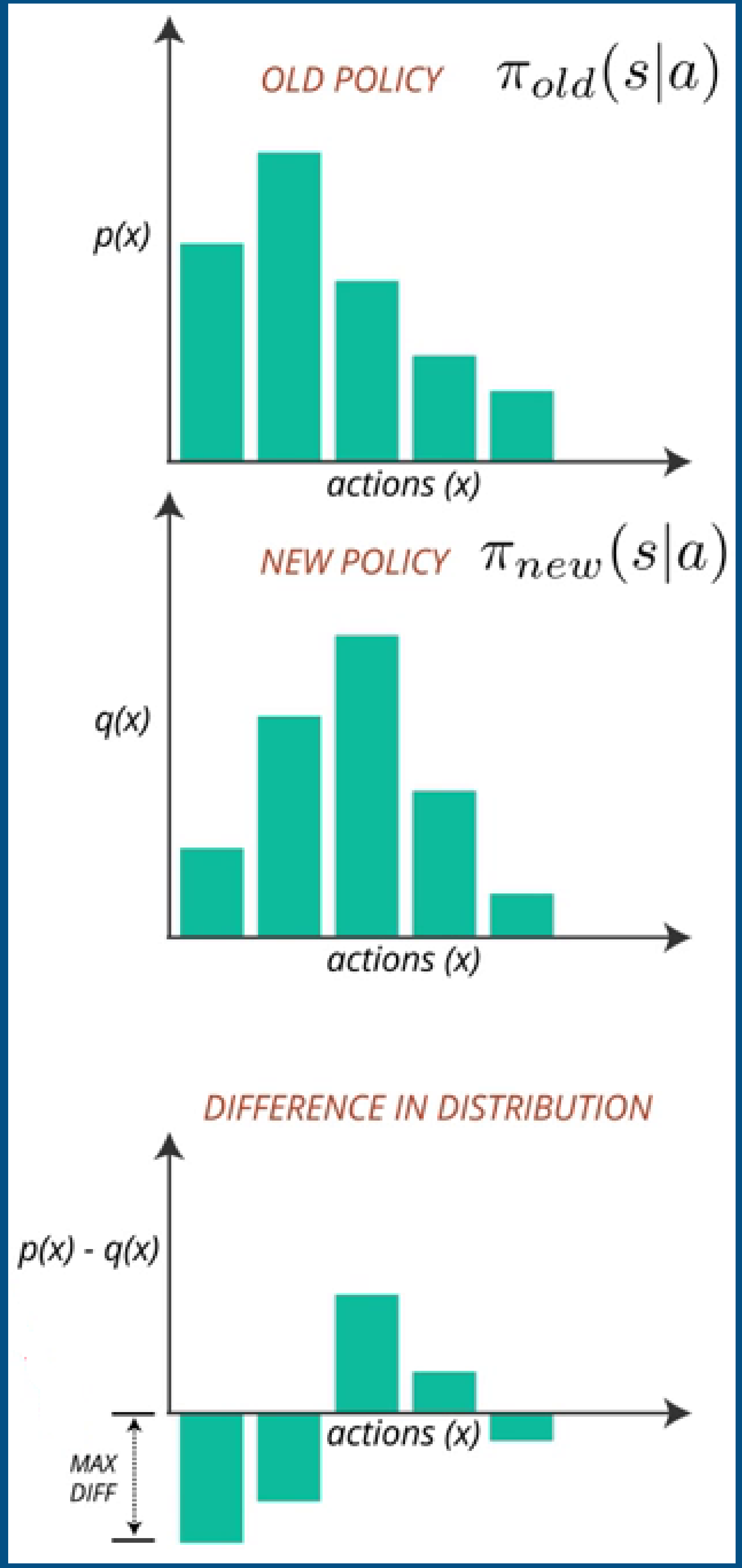
General Stochastic Policy

- 이를 이용해 $D_{\mathrm{TV}}^{\max}$를 다음과 같이 정의한다.

$$D_{\mathrm{TV}}^{\max}(\pi, \tilde{\pi}) = \max_s D_{TV}(\pi(\cdot|s) \parallel \tilde{\pi}(\cdot|s)).$$

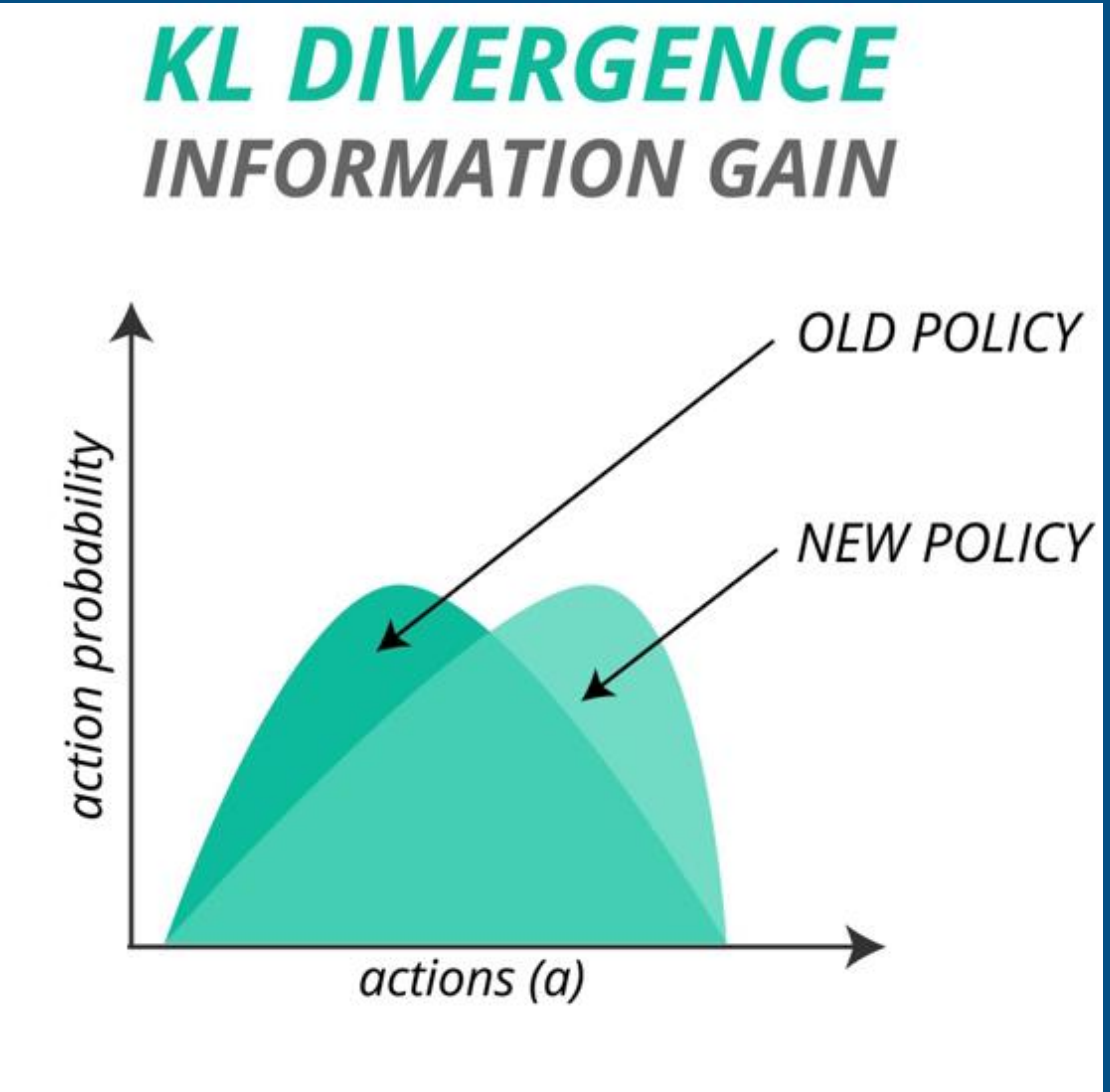
- 이제 다음과 같은 관계식을 얻을 수 있다.
- Theorem 1. Let $\alpha = D_{\mathrm{TV}}^{\max}(\pi_{\mathrm{old}}, \pi_{\mathrm{new}})$.
  Then the following bound holds:

$$\eta(\pi_{\mathrm{new}}) \geq L_{\pi_{\mathrm{old}}}(\pi_{\mathrm{new}}) - \frac{4\epsilon\gamma}{(1-\gamma)^2}\alpha^2$$
$$\textit{where } \epsilon = \max_{s,a} |A_\pi(s,a)|$$



General Stochastic Policy

- 다른 Distance Metric으로 KL Divergence가 있다.
  (왜 KL Divergence로 바꿀까? 확실하진 않지만,
  계산 효율을 위해 Conjugate Gradient Method를
  이용하는데, 이를 위해 바꾸지 않았을까 생각한다.)



$$D_{KL}(\pi||\pi') = \pi(a)log\frac{\pi(a)}{\pi'(a)}$$

General Stochastic Policy

- Total Variation Divergence와 KL Divergence 사이에는 다음과 같은 관계가 있다.

$$D_{\mathrm{TV}}(p \parallel q)^2 \leq D_{\mathrm{KL}}(p \parallel q)$$

- 다음 수식을 정의하자.

$$D_{\mathrm{KL}}^{\max}(\pi, \tilde{\pi}) = \max_s D_{\mathrm{KL}}\left(\pi(\cdot \mid s) \parallel \tilde{\pi}(\cdot \mid s)\right)$$

- Theorem 1을 이용해 다음과 같은 수식이 성립함을 알 수 있다.

$$\eta(\tilde{\pi}) \geq L_\pi(\tilde{\pi}) - C D_{\mathrm{KL}}^{\max}(\pi, \tilde{\pi}),$$
$$\text{where } C = \frac{4\epsilon\gamma}{(1-\gamma)^2}.$$

General Stochastic Policy

- 이러한 Policy Improvement Bound를 기반으로
  다음과 같은 Approximate Policy Iteration 알고리즘을 고안할 수 있다.

Algorithm 1 Policy iteration algorithm guaranteeing non-decreasing expected return η

Initialize π_0.
for $i = 0, 1, 2, \ldots$ until convergence **do**
  Compute all advantage values $A_{\pi_i}(s, a)$.
  Solve the constrained optimization problem

$$\pi_{i+1} = \arg\max_{\pi} \left[L_{\pi_i}(\pi) - C D_{\mathrm{KL}}^{\max}(\pi_i, \pi) \right]$$

  where $C = 4\epsilon\gamma/(1-\gamma)^2$

  and $L_{\pi_i}(\pi) = \eta(\pi_i) + \sum_s \rho_{\pi_i}(s) \sum_a \pi(a|s) A_{\pi_i}(s, a)$

end for

General Stochastic Policy

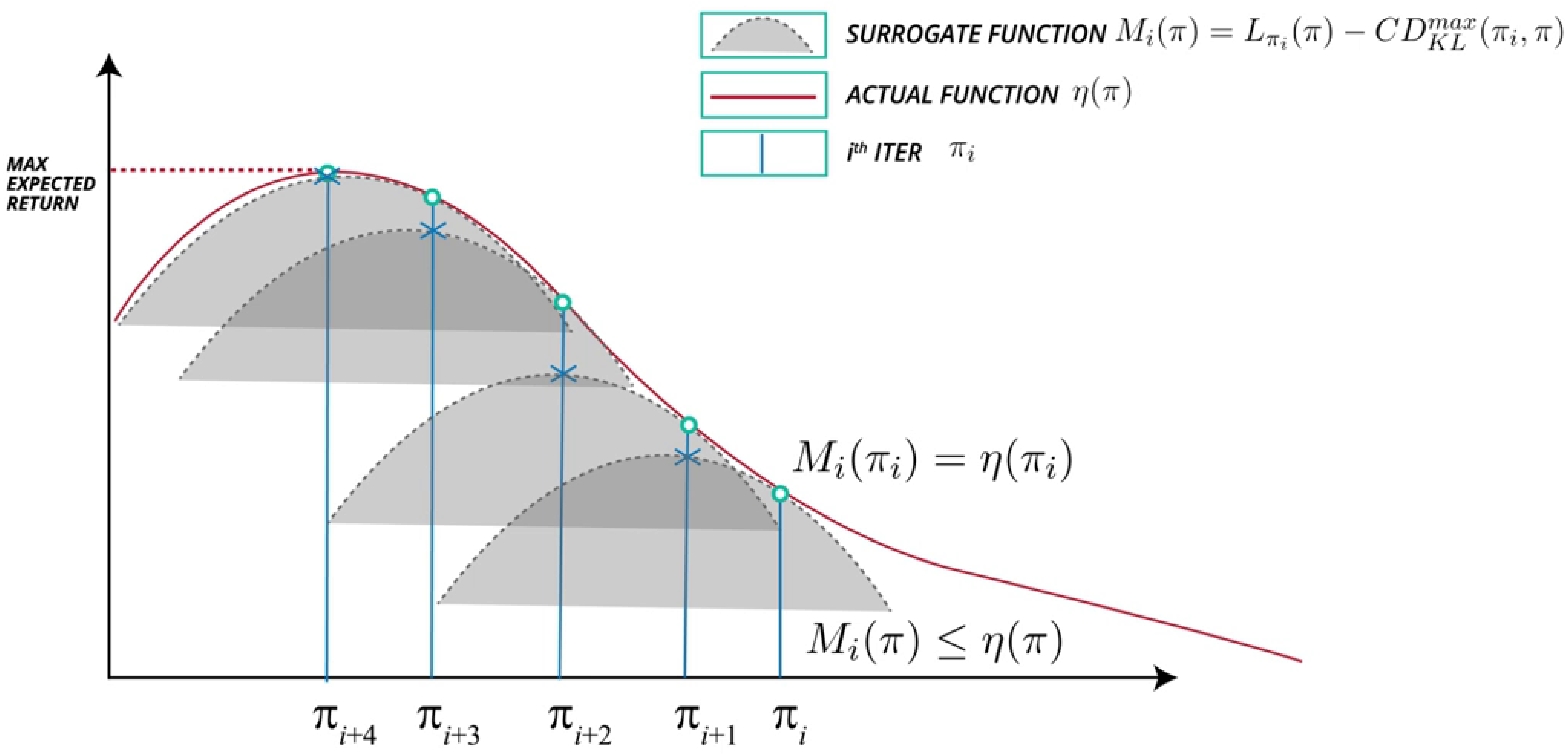
- Algorithm 1은 Advantage를 정확하게 계산할 수 있다고 가정하고 있다.
  이 알고리즘은 Monotonical하게 성능이 증가($\eta(\pi_0) \leq \eta(\pi_1) \leq \eta(\pi_2) \leq \cdots$)한다.
  - $M_i(\pi) = L_{\pi_i} - CD_{\mathrm{KL}}^{\max}(\pi_i, \pi)$라고 하자.

$$\begin{aligned}
&\eta(\pi_{i+1}) \geq M_i(\pi_{i+1}) \text{ by Equation (9)} \\
&\eta(\pi_i) = M_i(\pi_i), \text{ therefore,} \\
&\eta(\pi_{i+1}) - \eta(\pi_i) \geq M_i(\pi_{i+1}) - M(\pi_i).
\end{aligned}$$

  - 위 수식과 같이 반복할 때마다 M_i를 최대화함으로써 η이 감소하지 않는다는 걸 보장할 수 있다.
    이와 같은 타입의 알고리즘을 Minimization-Maximization (MM) 알고리즘이라고 한다.

General Stochastic Policy

- M_i는 π_i일 때 같아지는 η에 대한 Surrogate Function이다. TRPO는 이 함수를 최대화하고 KL Divergence를 패널티가 아닌 제약 조건으로 두는 알고리즘이다.



Optimization of Policies

- 표기의 편의를 위해 기호들을 다음과 같이 더 간략하게 정의한다.
  - $\eta(\theta) \coloneqq \eta(\pi_\theta)$
  - $L_\theta(\tilde{\theta}) \coloneqq L_{\pi_\theta}(\pi_{\tilde{\theta}})$
  - $D_{\mathrm{KL}}(\theta \parallel \tilde{\theta}) \coloneqq D_{\mathrm{KL}}(\pi_\theta \parallel \pi_{\tilde{\theta}})$
  - θ_{old} : previous policy parameters

Trust Region Policy Optimization

- 이전의 중요 결과를 앞서 소개한 표기법으로 다시 적어보자.

$$\eta(\theta) \geq L_{\theta_{\text{old}}}(\theta) - CD_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta)$$

- η의 성능 향상을 보장하기 위해 Lower Bound를 최대화할 수 있다.

$$\text{maximize}_{\theta}\left[L_{\theta_{\text{old}}}(\theta) - CD_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta)\right]$$

- 이 최적화 문제는 Step Size를 매우 작게 해야 올바른 동작을 한다. 위에서 살펴봤듯이 1차 근사이기 때문이다. 좀 더 큰 Step Size를 가질 수 있도록 이 최적화 문제를 Trust Region Constraint를 도입해 다음과 같이 바꾼다.

$$\begin{aligned} &\underset{\theta}{\text{maximize}}\ L_{\theta_{\text{old}}}(\theta) \\ &\text{subject to } D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta) \leq \delta. \end{aligned}$$

Trust Region Policy Optimization

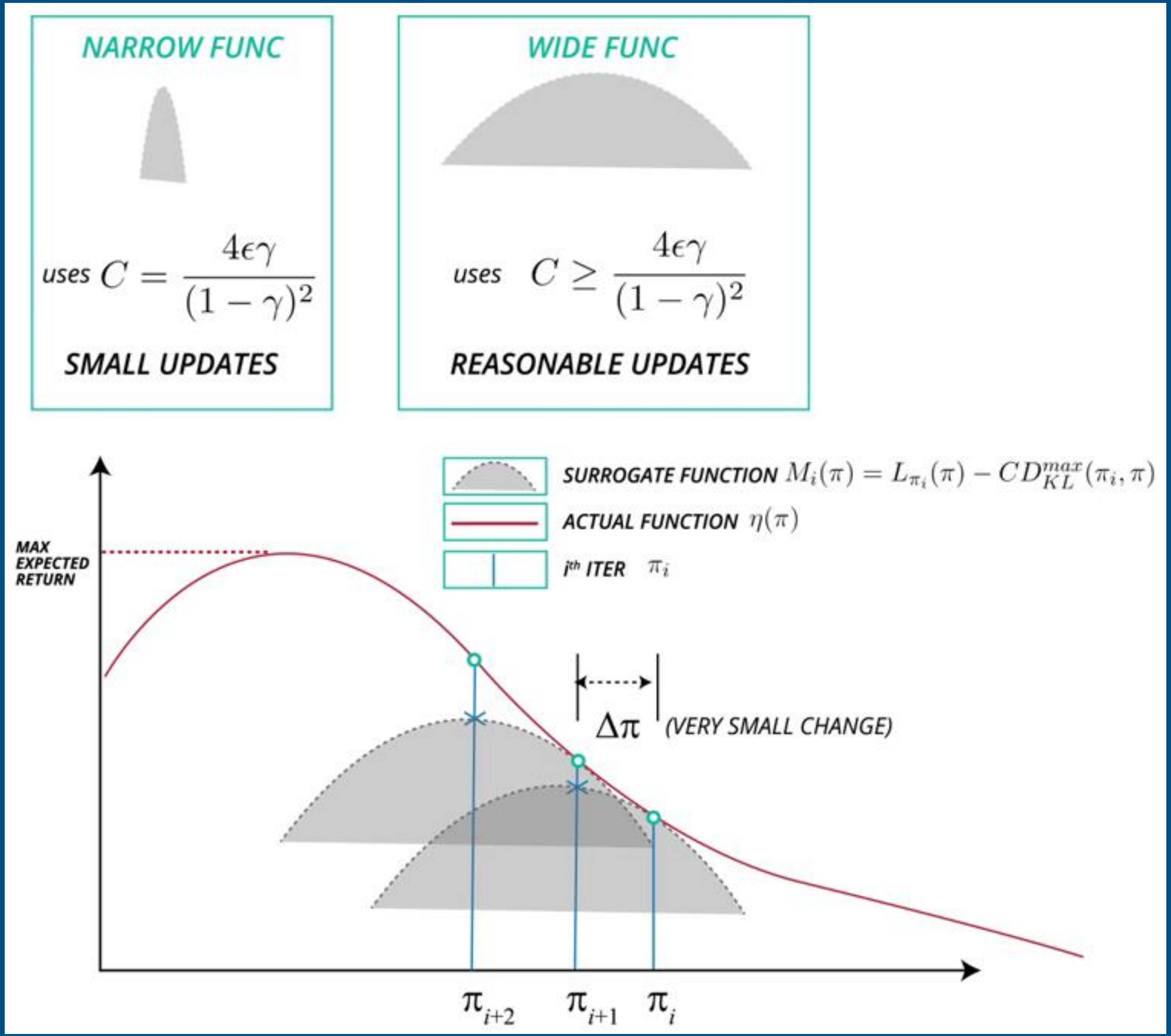
- 이 최적화 문제의 제약 조건은 모든 상태 공간에 대해서 성립해야 하며 최댓값을 매 번 찾아야 한다. 하지만 상태가 많은 경우 제약 조건의 수가 매우 많아져서 문제를 풀기 어렵게 만든다. 제약 조건의 수를 줄이기 위해 Average KL Divergence를 이용하는 휴리스틱 근사를 취한다. 최선의 방법은 아닐 수 있지만 실용적인 방법이다.

$$\overline{D}_{\mathrm{KL}}^{\rho}(\theta_1, \theta_2) := \mathbb{E}_{s\sim\rho}\left[D_{\mathrm{KL}}(\pi_{\theta_1}(\cdot|s) \,\|\, \pi_{\theta_2}(\cdot|s))\right].$$

- 이를 기반으로 다음 최적화 문제를 풀 수 있다.

$$\begin{aligned}&\underset{\theta}{\text{maximize}}\, L_{\theta_{\mathrm{old}}}(\theta)\\ &\text{subject to } \overline{D}_{\mathrm{KL}}^{\rho_{\theta_{\mathrm{old}}}}(\theta_{\mathrm{old}}, \theta) \le \delta.\end{aligned}$$

Trust Region Policy Optimization



Sample-based Estimation

- 실용적인 알고리즘을 만들기 위한 노력은 아직 끝나지 않았다.
  이제 앞의 알고리즘을 샘플 기반 예측, 즉 Monte Carlo Estimation할 수 있도록 바꿔보자.
- 샘플링을 편하게 할 수 있도록 아래와 같이 바꾼다.
  - $\sum_s \rho_{\theta_{\text{old}}}(s)[\dots] \rightarrow \frac{1}{1-\gamma} E_{s\sim\rho_{\theta_{\text{old}}}}[\dots]$
  - $A_{\theta_{\text{old}}} \rightarrow Q_{\theta_{\text{old}}}$
  - $\sum_a \pi_{\theta_{\text{old}}}(a|s) A_{\theta_{\text{old}}} \rightarrow E_{a\sim q}\left[\frac{\pi_\theta(a|s)}{q(a|s)} A_{\theta_{\text{old}}}\right]$

Sample-based Estimation

$$\underset{\theta}{\text{maximize}} \sum_{s} \rho_{\theta_{\text{old}}}(s) \sum_{a} \pi_\theta(a|s) A_{\theta_{\text{old}}}(s,a)$$

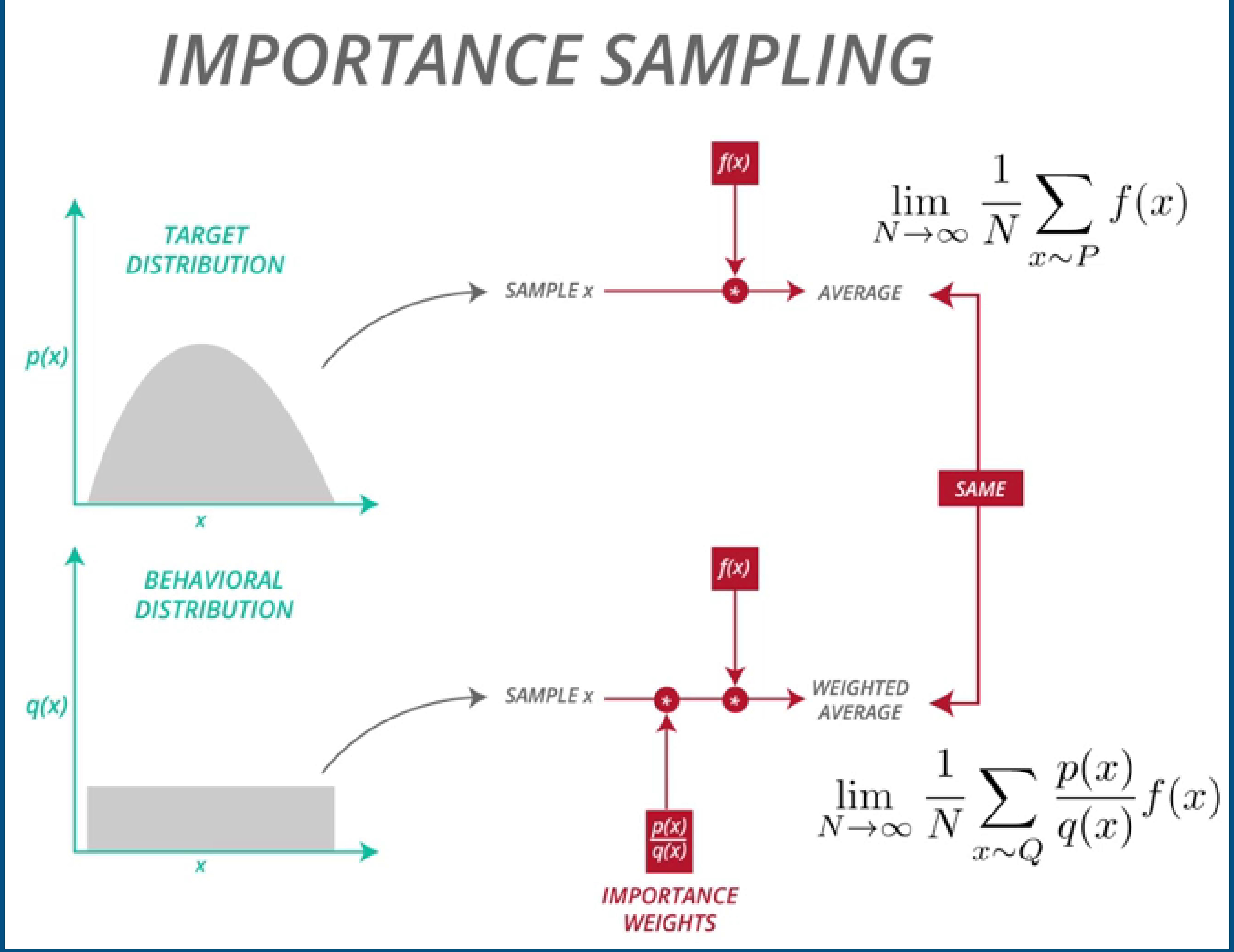
APPROXIMATE STATE VISITIATION
VIA SAMPLING

$$\underset{\theta}{\text{maximize}}\, \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}, a\sim q}\left[\frac{\pi_\theta(a|s)}{q(a|s)} Q_{\theta_{\text{old}}}(s,a)\right]$$

$$\sum_{a} \pi_\theta(a|s_n) A_{\theta_{\text{old}}}(s_n,a) = \mathbb{E}_{a\sim q}\left[\frac{\pi_\theta(a|s_n)}{q(a|s_n)} A_{\theta_{\text{old}}}(s_n,a)\right]$$

Sample-based Estimation

- 한 가지 짚고 넘어가자면, 행동을 샘플링할 때 Importance Sampling을 사용한다.



Sample-based Estimation

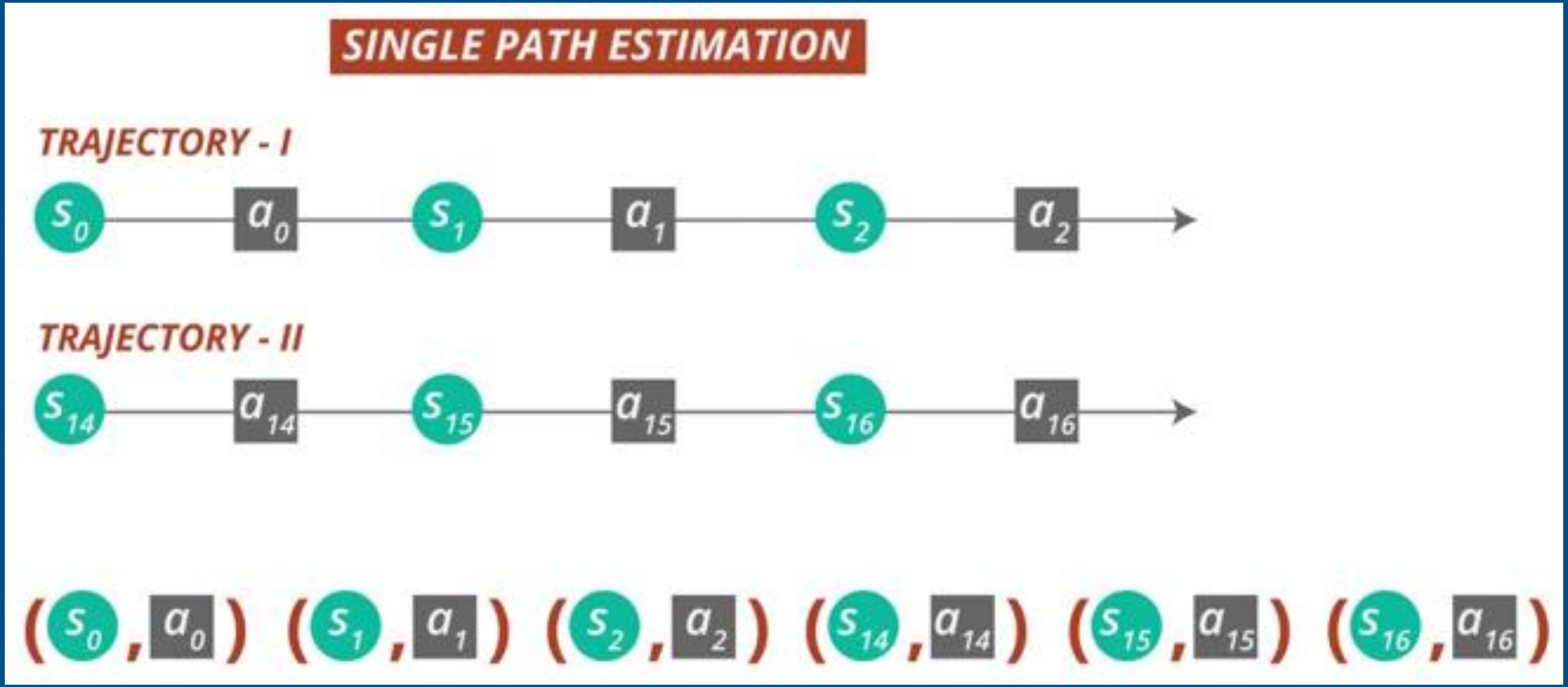
- 바뀐 최적화 문제는 다음과 같다.

$$\underset{\theta}{\text{maximize}}\, \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}, a\sim q}\left[\frac{\pi_\theta(a|s)}{q(a|s)} Q_{\theta_{\text{old}}}(s,a)\right]$$
$$\text{subject to } \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}}\left[D_{\text{KL}}(\pi_{\theta_{\text{old}}}(\cdot|s) \,\|\, \pi_\theta(\cdot|s))\right] \le \delta.$$

- 이 때 샘플링하는 방법은 두 가지가 있다.

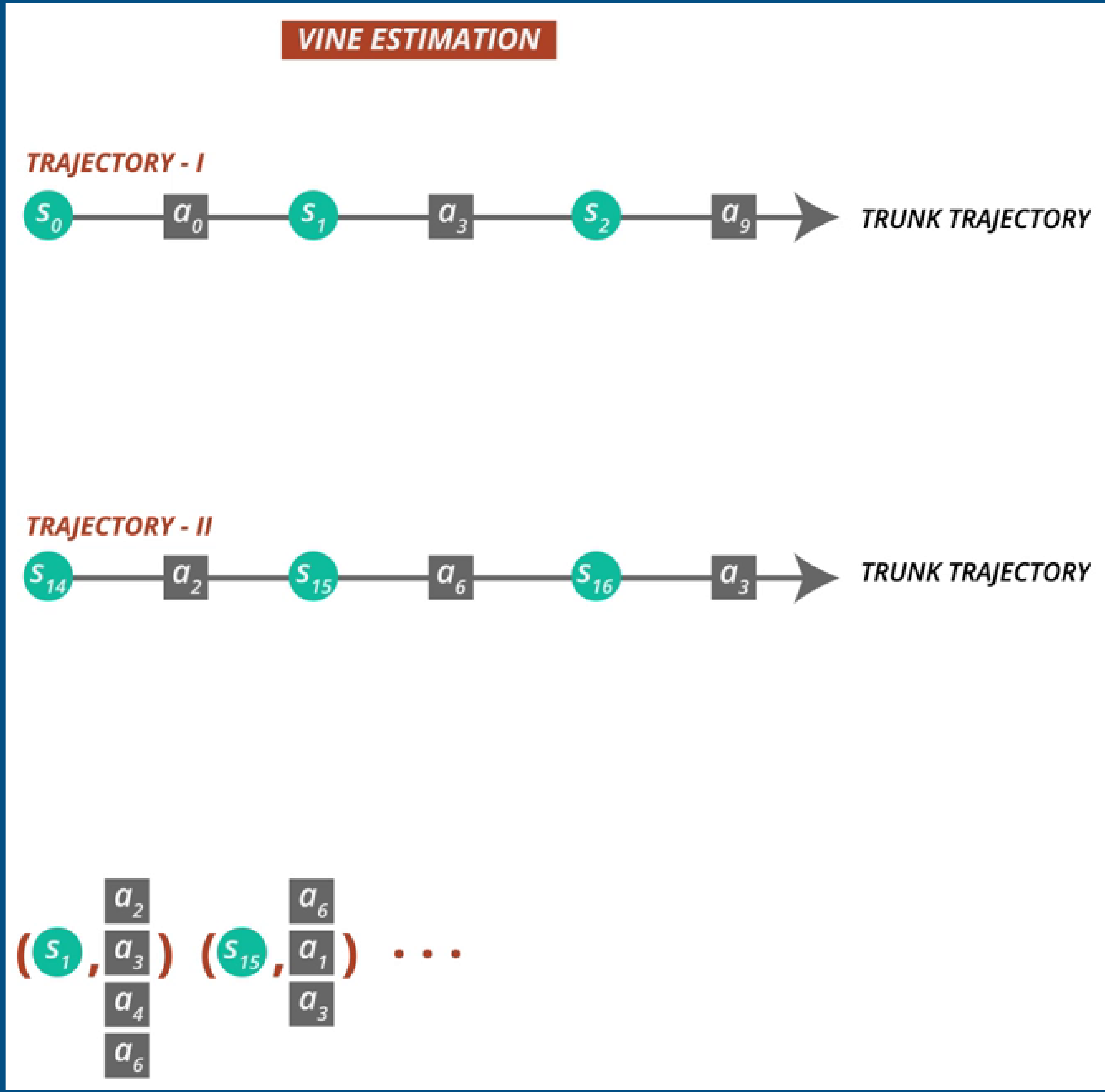
Single Path

- 개별 Trajectory들을 이용하는 방법



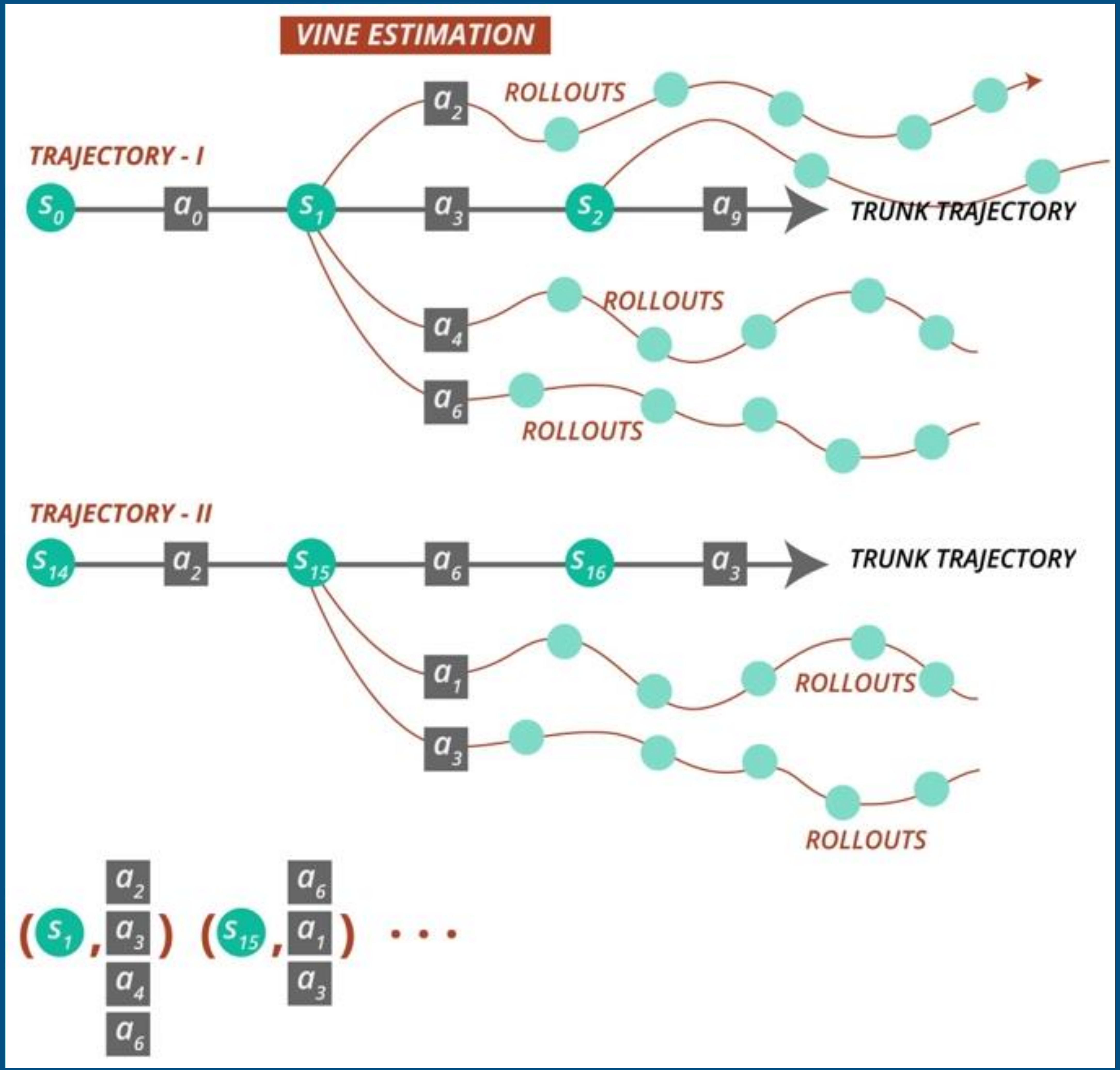
Vine

- 한 State에서 Rollout을 이용해 여러 Action을 수행하는 방법



Vine

- 한 State에서 Rollout을 이용해 여러 Action을 수행하는 방법



Practical Algorithm

- Single Path, Vine 샘플링을 사용하는 두 가지의 방식의 정책 최적화 알고리즘을 살펴봤다. 실용적인 알고리즘은 다음 과정을 반복해서 수행한다.
  - 1) Q 값의 몬테 카를로 추정을 통해 상태-행동 쌍 집합을 Single Path 또는 Vine 과정을 통해 수집한다.
  - 2) 샘플 평균으로, 다음 식의 목적 함수와 제약 조건 식을 추정한다.

$$\underset{\theta}{\text{maximize}}\, \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}},a\sim q}\left[\frac{\pi_\theta(a|s)}{q(a|s)}Q_{\theta_{\text{old}}}(s,a)\right]$$
$$\text{subject to } \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}}\left[D_{\text{KL}}(\pi_{\theta_{\text{old}}}(\cdot|s)\,\|\,\pi_\theta(\cdot|s))\right]\le\delta.$$

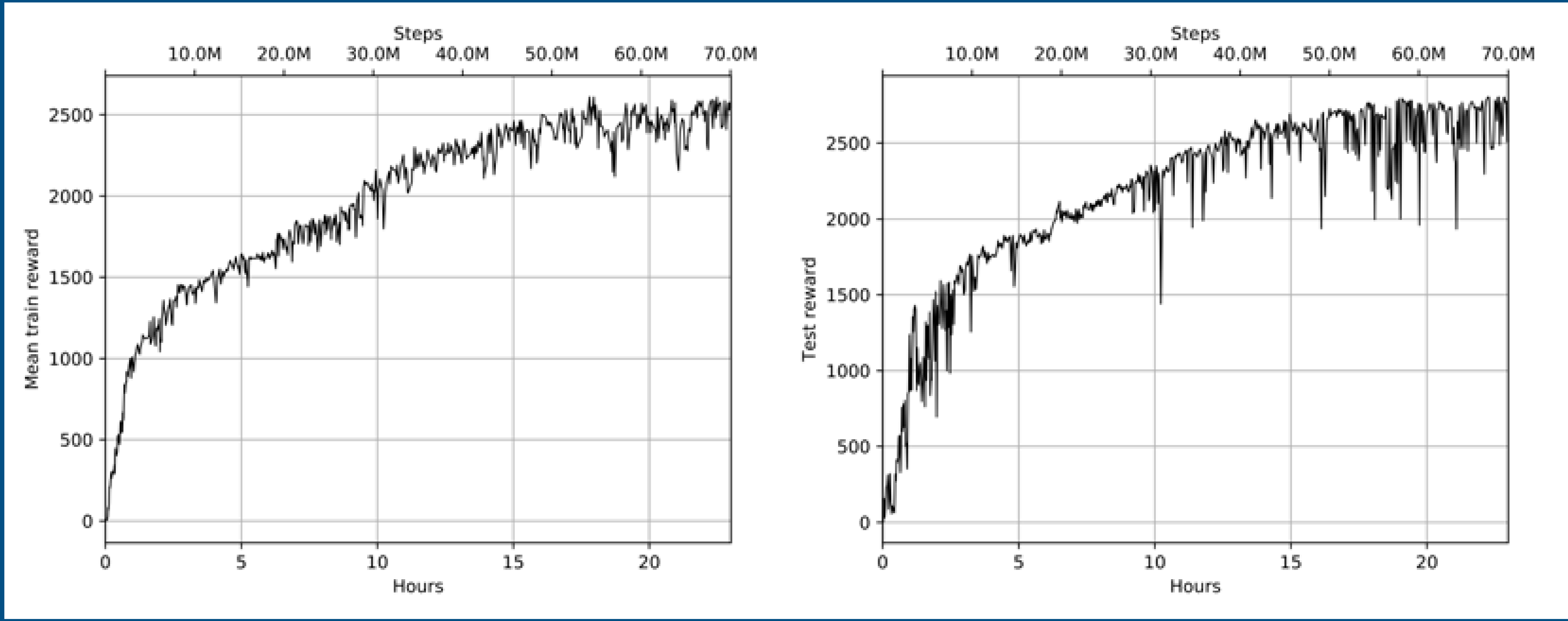
  - 3) Policy Parameter Vector인 θ를 업데이트하면서 제약 조건이 있는 최적화 문제를 근사적으로 푼다. 논문에서는 그레디언트를 직접 계산하지 않고 약간 더 계산량이 있는 Line Search와 Conjugate Gradient Algorithm을 사용했다.

Practical Algorithm

- 3)에 대해서 Gradient의 Covariance Matrix를 사용하지 않고 KL Divergence의 Hessian을 해석적으로 계산해 Fisher Information Matrix를 구성했다.
- 즉, $\frac{1}{N}\sum_{n=1}^{N} \frac{\partial}{\partial\theta_i} \log \pi_\theta(a_n|s_n) \frac{\partial}{\partial\theta_j} \log \pi_\theta(a_n|s_n)$ 대신 $\frac{1}{N}\sum_{n=1}^{N} \frac{\partial^2}{\partial\theta_i \partial\theta_j} D_{\mathrm{KL}}(\pi_{\theta_{\mathrm{old}}}(\cdot|s_n) \parallel \pi_\theta(\cdot|s_n))$ 을 계산한다.
- 이 Analytic Estimator는 Hessian이나 Trajectories의 모든 Gradient를 저장하지 않아도 되기 때문에 대규모 환경을 고려할 경우 계산상 이점이 있다.

TRPO

- Training Reward and Test Reward



감사합니다!

스터디 듣느라 고생 많았습니다.