TPRO

1. 준비운동

- 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

- Policy

  $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$

- ρ_0 (로) 시작 State의 분포(예 : 스타크)
- Policy : State에서 Action을 선택할 확률
- 나머지는 전에 이야기 했던것
- P : transition probability
  어떤 State에서 Action을 하고 다음 State로 어디로 가야할 확률이 얼마냐
- r : State를 주면 값이 나오는 reward 펑션
- γ : 0에서 1에서의 값. 미래의 불확실성을 표현
- Policy : State에서 어떤 행동을 할 확률

1. 준비운동(에타 파이를 정의함)

Let π denote a stochastic policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, and let $\eta(\pi)$ denote its expected discounted reward:

$$\eta(\pi) = \mathbb{E}_{s_0, a_0, \ldots}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t)\right], \text{ where}$$

$$s_0 \sim \rho_0(s_0),\ a_t \sim \pi(a_t|s_t),\ s_{t+1} \sim P(s_{t+1}|s_t, a_t).$$

- $\eta(\pi)$ (에타 파이) : Policy를 던져주면 policy가 얼마나 좋은 policy인지 알려주는 함수
- 에타는 끝날때까지 discount sum of reward를 한 것의 기대값
- 결국 에타 파이는 최적화 할려고 하는 목적함수다
- policy가 얼마나 좋은지 알기위해 policy가 받을 return의 기대값이 에타 파이임
- So는 로0를 따르고, Action t는 Policy 파이를 따르고, S t+1은 Transition Matrix에 의해서 정해지는 샘플이다.

1. 준비운동

We will use the following standard definitions of the state-action value function Q_π, the value function V_π, and the advantage function A_π:

$$Q_\pi(s_t, a_t) = \mathbb{E}_{s_{t+1}, a_{t+1}, \ldots}\left[\sum_{l=0}^{\infty} \gamma^l r(s_{t+l})\right],$$

$$V_\pi(s_t) = \mathbb{E}_{a_t, s_{t+1}, \ldots}\left[\sum_{l=0}^{\infty} \gamma^l r(s_{t+l})\right],$$

$$A_\pi(s, a) = Q_\pi(s, a) - V_\pi(s), \text{ where}$$

$$a_t \sim \pi(a_t|s_t), s_{t+1} \sim P(s_{t+1}|s_t, a_t) \text{ for } t \geq 0.$$

- Q는 Action value : Q는 어떤 상태에서 어떤 행동을 했을때 그 상태 이후로 받을 reward의 합의 기대값
- V는 어떤 state로 부터 게임 끝날때가지 받을 reward의 합의 기대값.(Action 불필요)
- Advantage는 Q-V이다.
- 에타는 s0에서 Q는 st에서 정의하는것이 다름

1. 준비운동(KAKADE & Langford)

The following useful identity expresses the expected return of another policy $\tilde{\pi}$ in terms of the advantage over π, accumulated over timesteps (see Kakade & Langford (2002) or Appendix A for proof):

$$\eta(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s_0, a_0, \cdots \sim \tilde{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t A_\pi(s_t, a_t) \right] \quad (1)$$

where the notation $\mathbb{E}_{s_0, a_0, \cdots \sim \tilde{\pi}}[\ldots]$ indicates that actions are sampled $a_t \sim \tilde{\pi}(\cdot|s_t)$. Let ρ_π be the (unnormalized) discounted visitation frequencies

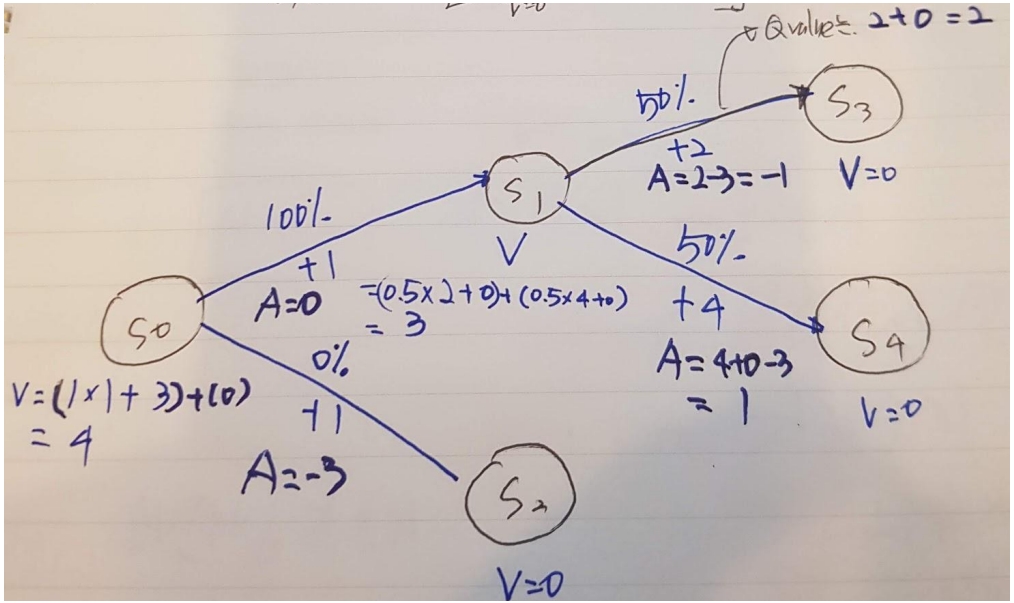
$$\rho_\pi(s) = P(s_0 = s) + \gamma P(s_1 = s) + \gamma^2 P(s_2 = s) + \ldots,$$

- kakade & Langford
- $\eta(\tilde{\pi})$(에타 파이 틸드) 는 $\eta(\pi)$(에타 파이)

  더하기 Advatage의 기대값이다. 이 논문은 이 식에서부터 시작함
- 파이랑, 파이 틸드는 다른 Policy다.
- 파이 틸드의 성능을 구하고 싶으면 우선 파이의 성능을 구하고, 파이 틸드로 부터 샘플링한다.
- 파이틸드를 따라다니면서(파이 틸드로 게임을 하면서) episode를 구하고,
- 파이 틸드를 따라다니는 경로의 파이의 Adventage를 구해서 더하면 파이틸드의 성능이다.(내가 아는 폴리시가 있으면 다른 폴리시의 성능을 구할수 있다라는 뜻)
- 다른 policy의 성능을 알고 싶으면 내가 아는 policy 하나랑 다른 폴리시를 따라다니면서 트랜지토리(S,A)를 뽑고, 이것의 내가 아는 파이의 Adventage를 더하면 나온다. ㅋㅋ

1. 준비운동(Kakade langford 풀어보기)

π

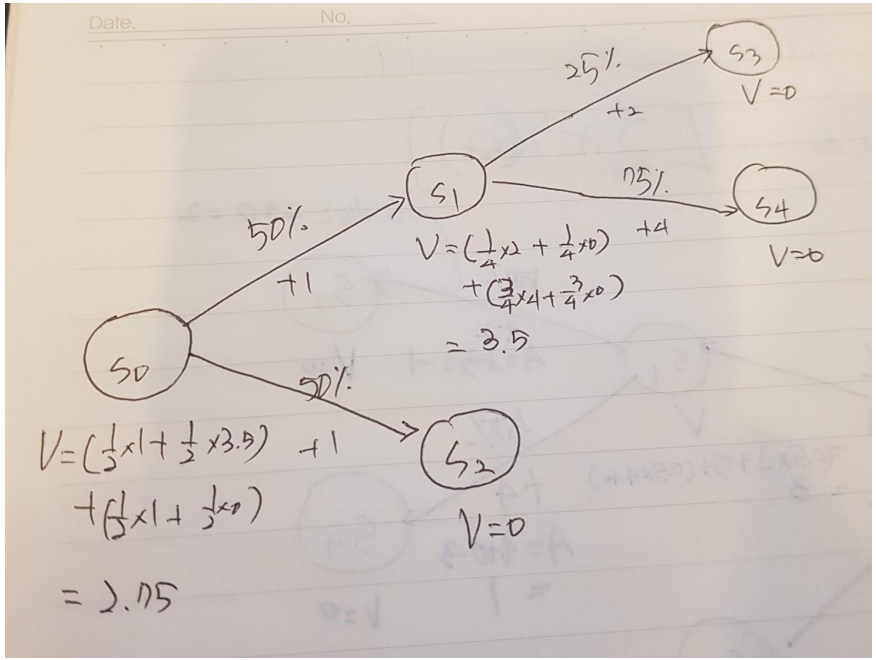


- Policy 파이랑, Policy 파이 틸드가 있음

- State가 5개 있고, 각자 2개의 Action을 함
- Policy만 다르기 때문에 Action의 확률만 다르고 나머지는 같음
- 그때의 Value랑 Advantage를 구해보았음
- Value는 bellman equation을 통해서 구하고, Advantage는 Q-V로 구한다.
- S2, S3, S4의 Value는 0이다. value는 해당 State이후에 받은 reward의 합이기 때문
- Q value는 어떤 상태에서 다음에 받을 보상에 대한 기대값
- A는 Q-V

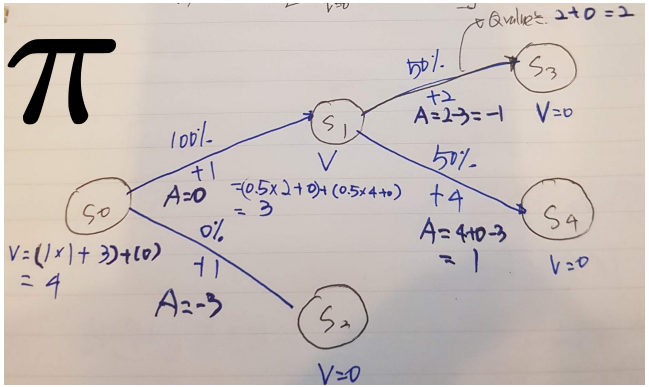
1. 준비운동(Kakade langford 풀어보기)

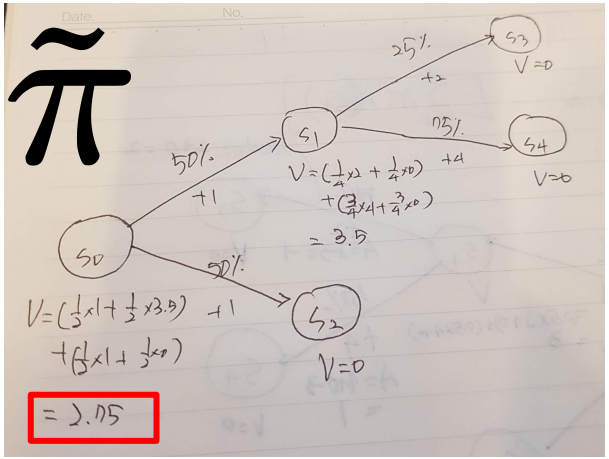
$\tilde{\pi}$



- Value를 구해봄

1. 준비운동(Kakade langford 풀어보기)





$$\eta(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s_0, a_0, \cdots \sim \tilde{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t A_{\pi}(s_t, a_t) \right] \qquad (1)$$

- So의 값이 policy의 성능이라고 볼수 있음
- 파이는 4, 파이 틸드는 2.75. 여기서는 파이가 더 좋은 policy임
- 위 식을 풀어보면 policy의 확률을 틸드파이를 쓰면서 Advance는 파이를 이용
- = 4 + 0.5* -3 + 0.5 * (0 + 0.25*-1 + 0.75 * 1)
  = 4-1.5 + 0.5*(0.5)
  = 2.5 + 0.25
  = 2.75
- 내가 아는 폴리시로부터 파이딜드의 성능을 알수 있음

1. 준비운동(Kakade langford 증명)

$$\eta(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s_0, a_0, \cdots \sim \tilde{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t A_\pi(s_t, a_t) \right] \quad (1)$$

Proof. First note that $A_\pi(s,a) = \mathbb{E}_{s' \sim P(s'|s,a)} [r(s) + \gamma V_\pi(s') - V_\pi(s)]$. Therefore,

$$\mathbb{E}_{\tau|\tilde{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t A_\pi(s_t, a_t) \right]$$

$$= \mathbb{E}_{\tau|\tilde{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t (r(s_t) + \gamma V_\pi(s_{t+1}) - V_\pi(s_t)) \right]$$

시그마를 풀면 상쇄되어 V s0만 남음

$$= \mathbb{E}_{\tau|\tilde{\pi}} \left[-V_\pi(s_0) + \sum_{t=0}^{\infty} \gamma^t r(s_t) \right]$$

reward합

Expectation 을 꺼내면

$$= -\mathbb{E}_{s_0} [V_\pi(s_0)] + \mathbb{E}_{\tau|\tilde{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t) \right]$$

$$= -\eta(\pi) + \eta(\tilde{\pi})$$

s0의 value의
기대값이
에타 파이 임

증명시작

- Advatage를 Q-V로 표현한다.
- 이것을 시그마를 취하면
- 파이에 대한 식을 자세히 보면 t가 0 일때와, t 1일때가 서로 상쇄되면서 결국 브이 s0만 남는다.(쭉쭉 사라짐)
- 그리고 리워드의 discount 합
- 브이 파이 S0는 에타 파이(앞장에 나옴)
- 리워드의 합을 에타파이 틸드를 따라가면서 기대값을 구한것이 에타 파이 틸드 이다.

1. 준비운동(Kakade langford를 시간기준에서 State 기준 변경)

$$\eta(\tilde{\pi}) = \eta(\pi) + \sum_{t=0}^{\infty} \sum_{s} P(s_t = s|\tilde{\pi}) \sum_{a} \tilde{\pi}(a|s)\gamma^t A_\pi(s,a)$$

$$= \eta(\pi) + \sum_{s} \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\tilde{\pi}) \sum_{a} \tilde{\pi}(a|s) A_\pi(s,a)$$

$$= \eta(\pi) + \sum_{s} \rho_{\tilde{\pi}}(s) \sum_{a} \tilde{\pi}(a|s) A_\pi(s,a). \quad (2)$$

- 아래의 로파이의 개념을 생각해서 kakade를 Sum over timestep에서 Sum over states로 변경해보자
- 시간순으로 하지 말고, 방문하는 스테이트 관점에서 더해서 보자
- 모든 스테이트에서 Sum을 하면 모든 timestep에서 한것이랑 같다.

- 시간과 상관없이 State 관점으로 변경

where the notation $\mathbb{E}_{s_0,a_0,\cdots\sim\tilde{\pi}}[\ldots]$ indicates that actions are sampled $a_t \sim \tilde{\pi}(\cdot|s_t)$. Let ρ_π be the (unnormalized) discounted visitation frequencies

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

- Discount된 방문 횟수 로 파이로 표현할수 있음
- 로 파이 S는
  S일 확률을 discount factor를 사용해서 다 더한것
  S0가 s일 확률 s1=s일 확률 s2가 s일 확률을 다 더한것

- episode동안 S에 있을 확률이 구해짐
- agent가 움직여 다니는데 1번 state에 있을 방문 빈도(?), 2번 state에 있을 방문 빈도에 대한 함수

1. 준비운동(Kakade langford 식의 해설)

$$\eta(\tilde{\pi}) = \eta(\pi) + \sum_{t=0}^{\infty} \sum_{s} P(s_t = s|\tilde{\pi}) \sum_{a} \tilde{\pi}(a|s)\gamma^t A_\pi(s,a)$$

$$= \eta(\pi) + \sum_{s} \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\tilde{\pi}) \sum_{a} \tilde{\pi}(a|s) A_\pi(s,a)$$

$$= \eta(\pi) + \sum_{s} \rho_{\tilde{\pi}}(s) \sum_{a} \tilde{\pi}(a|s) A_\pi(s,a). \qquad (2)$$

- 모든 state s에 대해 파이틸다를 따르는 Advantage가 0보다 크면, policy성능인 에타가 증가하는게 보장됨
- policy iteration 에서도 s,a쌍에 대해서 Advantage가 하나라도 양수면 policy가 개선된다는것을 알수 있음
- 그러나 뉴럴넷 근사에러 때문에 실전에서는 항상 0 보다 크다는 것을 보장못함
- 어쨋든 policy가 개선됨을 보장되는 식임

1. 준비운동(Kakade langford L과 에타)

$$= \eta(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) A_\pi(s,a). \qquad (2)$$

of $\rho_{\tilde{\pi}}(s)$ on $\tilde{\pi}$ makes Equation (2) difficult to optimize directly. Instead, we introduce the following local approximation to η:

$$L_\pi(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_\pi(s) \sum_a \tilde{\pi}(a|s) A_\pi(s,a). \qquad (3)$$

$$L_{\pi_{\theta_0}}(\pi_{\theta_0}) = \eta(\pi_{\theta_0}),$$
$$\nabla_\theta L_{\pi_{\theta_0}}(\pi_\theta)\big|_{\theta=\theta_0} = \nabla_\theta \eta(\pi_\theta)\big|_{\theta=\theta_0}. \qquad (4)$$

Equation (4) implies that a sufficiently small step $\pi_{\theta_0} \to \tilde{\pi}$ that improves $L_{\pi_{\theta_{\text{old}}}}$ will also improve η, but does not give us any guidance on how big of a step to take.

- 로 파이 틸다 optimize하기 어려워서 그냥 로파이로 바꾼다!!!
- 세타 0에서 에타 파이 틸트랑 엘 파이랑 같다.
- 왜???
- 식 (4)의 이유때문에
- policy를 세타로 파마미터라이즈 할수 있다면, 세타 제로와에서는 같고, 1차 미분에서도 세타0에서는 같다.(?)

- 근데, 세타 0에서만 안전하다.
- 그래서 뭔가 세타0 즉, 어떤 작은 구역에서, policy를 증가 시킬수 있다라는 의미(작은 구역, Trust region, 믿을 만한 구역에서 policy 업데이트)
- 세타0는 원래 policy

1. 준비운동(Kakade langford L과 에타 해설)

$$L_{\pi_{\theta_0}}(\pi_{\theta_0}) = \eta(\pi_{\theta_0}),$$
$$\nabla_\theta L_{\pi_{\theta_0}}(\pi_\theta)\big|_{\theta=\theta_0} = \nabla_\theta \eta(\pi_\theta)\big|_{\theta=\theta_0}. \qquad (4)$$

Equation (4) implies that a sufficiently small step $\pi_{\theta_0} \rightarrow \tilde{\pi}$ that improves $L_{\pi_{\theta_{\text{old}}}}$ will also improve η, but does not give us any guidance on how big of a step to take.

- 엘 파이와, 에타가 같다.
- 충분히 작은 step만큼 policy를 업데이트 하면

$$\pi_{\theta_0} \rightarrow \tilde{\pi}$$

- L을 증가 시키는것에 에타를 증가 시키는것이 같음
-
- 그러나 얼마나 작은 step이어야 하는것은 알려주지 않음

- 그래서 kakade & langford가 Conservative policy update를 제시함(보수적인 policy iteration)

1. 준비운동(Kakade langford Conservative policy iteration)

To address this issue, Kakade & Langford (2002) proposed a policy updating scheme called conservative policy iteration, for which they could provide explicit lower bounds on the improvement of η. To define the conservative policy iteration update, let π_{old} denote the current policy, and let $\pi' = \arg\max_{\pi'} L_{\pi_{\text{old}}}(\pi')$. The new policy π_{new} was defined to be the following mixture:

$$\pi_{\text{new}}(a|s) = (1-\alpha)\pi_{\text{old}}(a|s) + \alpha\pi'(a|s). \quad (5)$$

Kakade and Langford derived the following lower bound:

$$\eta(\pi_{\text{new}}) \geq L_{\pi_{\text{old}}}(\pi_{\text{new}}) - \frac{2\epsilon\gamma}{(1-\gamma)^2}\alpha^2$$
$$\text{where } \epsilon = \max_s \left|\mathbb{E}_{a\sim\pi'(a|s)}\left[A_\pi(s,a)\right]\right|. \quad (6)$$

- 기본 policy랑 새로운 policy랑 1-a, a만큼 섞어씀
- 섞어 쓴 policy의 성능은 lower bound가 있다.

- 입실론은 파이 세타의 adventaged의 max값

- 알파에 따라서 새로운 에타 파이에 대한 lower bound를 제시함
- 보통 우리는 policy를 섞어 쓰지 않음(Mixture policy)
- 이 식은 Mixture policy에서만 통용됨
- 그래서 일반적인 방법을 제시함. 뭔가 stochastic policy에서 통용되는 general 한 방법론 필요

준비운동 끝

2. General Stochastic Policies Improvement (Total variation Divergence)

provement guarantee to practical problems. The particular distance measure we use is the total variation divergence, which is defined by $D_{TV}(p \parallel q) = \frac{1}{2}\sum_i |p_i - q_i|$ for discrete probability distributions p, q.[1] Define $D_{\mathrm{TV}}^{\max}(\pi, \tilde{\pi})$ as

$$D_{\mathrm{TV}}^{\max}(\pi, \tilde{\pi}) = \max_s D_{TV}(\pi(\cdot|s) \parallel \tilde{\pi}(\cdot|s)). \quad (7)$$

Theorem 1. *Let* $\alpha = D_{\mathrm{TV}}^{\max}(\pi_{\mathrm{old}}, \pi_{\mathrm{new}})$. *Then the following bound holds:*

$$\eta(\pi_{\mathrm{new}}) \geq L_{\pi_{\mathrm{old}}}(\pi_{\mathrm{new}}) - \frac{4\epsilon\gamma}{(1-\gamma)^2}\alpha^2$$
$$\text{where } \epsilon = \max_{s,a} |A_\pi(s,a)| \quad (8)$$

- Policy Improve를 보장하기 전에 Total variation devergence를 먼저 보자
- 두 확률 분포가 얼마나 다른지를 표현하는 방법
- D_{TV} 가 Total variation devergence이고 구하는 방법은 두 확률을 뺀것의 합의 1/2이다.
- p라는 확률 분포가 action 3개에 대해서 (⅓,⅓,⅓) 이고 q는(1,0,0)이면 Total variation devergence는 ½(⅔+⅓+⅓) = ⅔ 이다.
- D_{TV}^{max} 는 두 policy에 대해서 모든 state에 대해서 Dtv값을 다 계산함(State별로 가장 크게 다른것은 무엇인가?)
- 그래서 DTV max를 알파로 놓으면 아까 식과 비슷한 8번 식이 만들어짐
- 그러나 이식은 mixture policy에 대한 식이 아닌 Dtvmax로 lower bound를 계산한 일반적인 식이다.

2. General Stochastic Policies Improvement (다시 정리)

$$\eta(\pi_{\text{new}}) \geq L_{\pi_{\text{old}}}(\pi_{\text{new}}) - \frac{2\epsilon\gamma}{(1-\gamma)^2}\alpha^2$$
$$\text{where } \epsilon = \max_s \left|\mathbb{E}_{a\sim\pi'(a|s)}\left[A_\pi(s,a)\right]\right|. \quad (6)$$

Theorem 1. *Let* $\alpha = D_{\text{TV}}^{\max}(\pi_{\text{old}}, \pi_{\text{new}})$. *Then the following bound holds:*

$$\eta(\pi_{\text{new}}) \geq L_{\pi_{\text{old}}}(\pi_{\text{new}}) - \frac{4\epsilon\gamma}{(1-\gamma)^2}\alpha^2$$
$$\textit{where } \epsilon = \max_{s,a} |A_\pi(s,a)| \quad (8)$$

Next, we note the following relationship between the total variation divergence and the KL divergence (Pollard (2000), Ch. 3): $D_{TV}(p \parallel q)^2 \leq D_{\text{KL}}(p \parallel q)$. Let $D_{\text{KL}}^{\max}(\pi, \tilde{\pi}) = \max_s D_{\text{KL}}(\pi(\cdot|s) \parallel \tilde{\pi}(\cdot|s))$. The following bound then follows directly from Theorem 1:

- Kakade & langford의 방법
- TRPO에서 Total variation Divergence로 제안한 방법
- KL divergence를 사용해서 한번 더 변환
- D TV Divergence 제곱은 D KL Divergence보다 작거나 같다를 Pollard가 증명
- D KL max는 모든 state에 대해 policy두개의 차이중에 가장 큰것

2. General Stochastic Policies Improvement (논문의 흐름 정리)

$$\eta(\pi_{\text{new}}) \geq L_{\pi_{\text{old}}}(\pi_{\text{new}}) - \frac{2\epsilon\gamma}{(1-\gamma)^2}\alpha^2$$
$$\text{where } \epsilon = \max_s \left|\mathbb{E}_{a\sim\pi'(a|s)}\left[A_\pi(s,a)\right]\right|. \quad (6)$$

- Kakade & langford의 방법, Conversative policy iteration

$$\eta(\pi_{\text{new}}) \geq L_{\pi_{\text{old}}}(\pi_{\text{new}}) - \frac{4\epsilon\gamma}{(1-\gamma)^2}\alpha^2$$
$$\textit{where } \epsilon = \max_{s,a} |A_\pi(s,a)| \quad (8)$$

- TRPO에서 Total variation Divergence로 제안한 방법

$$\eta(\tilde{\pi}) \geq L_\pi(\tilde{\pi}) - CD_{\text{KL}}^{\max}(\pi,\tilde{\pi}),$$
$$\text{where } C = \frac{4\epsilon\gamma}{(1-\gamma)^2}. \quad (9)$$

- TRPO에서 KL Divergence 로 치환
- L-CD
- 이 부등식으로 무엇을 할것인가? 빼기 CD인데.
- 이제 기본공식 끝!! 토나옴

2. General Stochastic Policies Improvement (알고리즘)

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

- policy iteration 성능지표 에타가 줄어들지 않음을 보장하는 알고리즘(iteration 할때마다, 같거나 늘어남을 보장한다.)

- 일단 for loop 수렴할때까지
- 모든 s,a에 대해서 Advatage를 구한다(말이 안됨) 왜냐면 앞에서 입실론을 구해야 하기 때문(식8)

- 다음 식을 최적화 하는데 L -CD를 최대화 시키는 Policy 파이를 구하고, C와 L은 앞에 나온 식을 구한다. 파이가 나오면 그것이 다음 policy다.

- 진짜 이러면 policy 파이가 개선될까?

2. General Stochastic Policies Improvement (알고리즘이 진짜 개선될까?)

$$\eta(\tilde{\pi}) \geq L_{\pi}(\tilde{\pi}) - CD_{\mathrm{KL}}^{\max}(\pi, \tilde{\pi}),$$
$$\text{where } C = \frac{4\epsilon\gamma}{(1-\gamma)^2}. \qquad (9)$$

- :L-CD를 Mi로 치환

It follows from Equation (9) that Algorithm 1 is guaranteed to generate a monotonically improving sequence of policies $\eta(\pi_0) \leq \eta(\pi_1) \leq \eta(\pi_2) \leq \ldots$. To see this, let $M_i(\pi) = L_{\pi_i}(\pi) - CD_{\mathrm{KL}}^{\max}(\pi_i, \pi)$. Then

$$\eta(\pi_{i+1}) \geq M_i(\pi_{i+1}) \text{ by Equation (9)}$$
$$\eta(\pi_i) = M_i(\pi_i), \text{ therefore,}$$
$$\eta(\pi_{i+1}) - \eta(\pi_i) \geq M_i(\pi_{i+1}) - M(\pi_i). \qquad (10)$$

- (9)번 식에 의해서 부등식이 성립
- 에타 파이 아이랑 mi 파이 i는 왜 같느냐?
- CD KM max(파이아이, 파이아이)는 0이기 때문에 L파이 아이만 남고, 그것은 앞의 (3)번식의 L정의로 인해 같게 된다.

- M을 극대화 시키는것이 에타가 줄어들지 않음을 보장함

- M을 surrogate fucntion 이라고도 함

3. Optimization Parameterized Polices (이제 뭔가 실제로 쓸수 있는 것으로)

Since we consider parameterized policies $\pi_\theta(a|s)$ with parameter vector θ, we will overload our previous notation to use functions of θ rather than π, e.g. $\eta(\theta) := \eta(\pi_\theta)$, $L_\theta(\tilde{\theta}) := L_{\pi_\theta}(\pi_{\tilde{\theta}})$, and $D_{\mathrm{KL}}(\theta \,\|\, \tilde{\theta}) := D_{\mathrm{KL}}(\pi_\theta \,\|\, \pi_{\tilde{\theta}})$. We will use θ_{old} to denote the previous policy parameters that we want to improve upon.

The preceding section showed that $\eta(\theta) \geq L_{\theta_{\mathrm{old}}}(\theta) - CD^{\max}_{\mathrm{KL}}(\theta_{\mathrm{old}}, \theta)$, with equality at $\theta = \theta_{\mathrm{old}}$. Thus, by performing the following maximization, we are guaranteed to improve the true objective η:

$$\underset{\theta}{\mathrm{maximize}} \left[L_{\theta_{\mathrm{old}}}(\theta) - CD^{\max}_{\mathrm{KL}}(\theta_{\mathrm{old}}, \theta) \right].$$

- 파이를 세타를 이용하여 parameterize를 함
- 세타로 치환하여 notation을 변경함

- 이것을 maximize하는 세타를 찾는 문제로 변경

3. Optimization Parameterized Polices (이제 뭔가 실제로 쓸수 있는 것으로)

$$\underset{\theta}{\text{maximize}} \left[L_{\theta_{\text{old}}}(\theta) - C D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta) \right].$$

$$\text{where } C = \frac{4\epsilon\gamma}{(1-\gamma)^2}.$$

In practice, if we used the penalty coefficient C recommended by the theory above, the step sizes would be very small. One way to take larger steps in a robust way is to use a constraint on the KL divergence between the new policy and the old policy, i.e., a trust region constraint:

$$\underset{\theta}{\text{maximize}} \; L_{\theta_{\text{old}}}(\theta) \tag{11}$$

$$\text{subject to } D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta) \leq \delta.$$

- L - CD형태인데 L을 Maximize하고 싶은데 CD라는 Penalty가 있는 형태임
- 세타 old랑 세타가 달라질수록 KL divergence가 커지는 구조임
- C도 잘 보면 엄청 큰 수임. 감마에 0.99만 있다고 하면 0.01의 제곱이 분모에 있으므로 수가 커짐 입실론이 1이면 감마가 0.4일경우 4만임

- C가 너무 커서 세타 올드랑, 세타랑 거의 안바뀜
- C는 이론적인 숫자라 이상태로 쓸수 없음

- 그래서 다른 방법으로 Maximize를 하도록 함 penalty 형태에서 contraint Optimization형태로 바꾸게 함(제한조건이 있는 최적화 문제로 변환)

- 즉 L을 최대화 하는데 D kl을 델타보다 작게 하면서 Maximize해라로 바꿈

3. Optimization Parameterized Polices (이제 뭔가 실제로 쓸수 있는 것으로)

$$\underset{\theta}{\text{maximize}}\, L_{\theta_{\text{old}}}(\theta) \qquad (11)$$
$$\text{subject to } D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta) \leq \delta.$$

a heuristic approximation which considers the average KL divergence:

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

- 제한조건이 있는 최적화 문제로 변환했음
- 모든 state에 대해서 D KL을 구해야 한다는 뜻인데 D KL max를 구할수가 없음. 불가능함
- 그래서 휴리스틱하게 평균으로 근사함.
- s가 로를 따라서 샘플링이 되었을때 그것의 KL diver를 구하고, 그것의 기대값을 구한다. 기대값은 평균과 같다.
- 이것을 D kl max로 쓰겠다.
- Max를 평균으로 쓴다? 휴리스틱으로 된다고함.
- 결국 1억개의 state에서 대충 한 1000개의 state에 대한 샘플로 평균을 구하기로 함.
- 1억개중 1000개의 모집단의 평균은 unbiased 된 estimator라고 함. 1000개를 계속 뽑다 보면 1억개의 집단의 평균으로 근사함

4. Sample Based Estimation of the Objective and Constraint

date. This section describes how the objective and constraint functions can be approximated using Monte Carlo simulation.

We seek to solve the following optimization problem, obtained by expanding $L_{\theta_{\text{old}}}$ in Equation (12):

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

$$\text{subject to } \overline{D}_{\text{KL}}^{\rho_{\theta_{\text{old}}}}(\theta_{\text{old}}, \theta) \leq \delta. \qquad (13)$$

We first replace $\sum_s \rho_{\theta_{\text{old}}}(s)[\dots]$ in the objective by the expectation $\frac{1}{1-\gamma}\mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}}[\dots]$. Next, we replace the advantage values $A_{\theta_{\text{old}}}$ by the Q-values $Q_{\theta_{\text{old}}}$ in Equation (13),

- 어떻게 Object and contraint 기반의 함수를 Monte Carlo simulation으로 바꿔서 풀것인가?
- L은 에타 세타가 있지만 상수여서 여기서는 제외함
- L을 풀어쓰고
- 식안에 Expectation 형태가 있어야 Sample 기반으로 문제를 풀수 있음. Expection안에 있는 값들을 여러 Sampling을 통해 값을 구해고 평균을 내야 값이 같아짐
- 로 old를 state가 로 세타 올드를 따를때 기대값으로 변경함
- 그리고 A를 Q로 바꿈 A는 Q-V인데, V는 상수기 때문에 그냥 Q로 바꿈.
- 결국 maximize문제는 x를 맥시마이즈하나, x-3를 맥시마이즈 하나 x는 동일함

4. Sample Based Estimation of the Objective and Constraint

replace the sum over the actions by an importance sampling estimator. Using q to denote the sampling distribution, the contribution of a single s_n to the loss function is

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

- Expectation 형태를 만들기 위해 Important Sampling을 사용함

- 아래의 방법을 사용함 q가 있는데 q를 분모분자에 곱해서 expection으로 변경함
- q는 old policy의 sampling distribution 임

Importance Sampling

- Estimate the expectation of a different distribution

$$\begin{aligned}\mathbb{E}_{X\sim P}[f(X)] &= \sum P(X) f(X) \\ &= \sum Q(X) \frac{P(X)}{Q(X)} f(X) \\ &= \mathbb{E}_{X\sim Q}\left[\frac{P(X)}{Q(X)} f(X)\right]\end{aligned}$$

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

- f(x)의 기대값을 구하고 싶은데 X가 p로 부터 샘플링
- 시그마 P(x)f(x)로 표현
- 근데 실제 x는 Q로 부터 샘플링 되고 있음
- 그래서Q(x)를 분모, 분자에 곱해줌
- 시스마 Q는 x가 Q로 부터 샘플링 되었을때 기대값

- 이것을 사용함

4. Sample Based Estimation of the Objective and Constraint

$$\underset{\theta}{\text{maximize}} \sum_{s} \rho_{\theta_{\text{old}}}(s) \sum_{a} \pi_\theta(a|s) A_{\theta_{\text{old}}}(s,a)$$
$$\text{subject to } \overline{D}_{\text{KL}}^{\rho_{\theta_{\text{old}}}}(\theta_{\text{old}}, \theta) \leq \delta. \qquad (13)$$

$$\underset{\theta}{\text{maximize}}\, \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}, a\sim q}\left[\frac{\pi_\theta(a|s)}{q(a|s)} Q_{\theta_{\text{old}}}(s,a)\right] \qquad (14)$$
$$\text{subject to } \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}}\left[D_{\text{KL}}(\pi_{\theta_{\text{old}}}(\cdot|s) \,\|\, \pi_\theta(\cdot|s))\right] \leq \delta.$$

- 그래서 식이 이렇게 변경됨

- Sampling을 위한 Expectation이 들어간 식으로 변경
- 두 식은 완벽히 같음

- expection으로 바꾸면 monte carlo로 풀수 있음
  계속 샘플링 해서 값을 찾을수 있음

5. Sample Based Estimation of the Objective and Constraint

5.1 Single Path

In this estimation procedure, we collect a sequence of states by sampling $s_0 \sim \rho_0$ and then simulating the policy $\pi_{\theta_{\text{old}}}$ for some number of timesteps to generate a trajectory $s_0, a_0, s_1, a_1, \ldots, s_{T-1}, a_{T-1}, s_T$. Hence, $q(a|s) = \pi_{\theta_{\text{old}}}(a|s)$. $Q_{\theta_{\text{old}}}(s,a)$ is computed at each state-action pair (s_t, a_t) by taking the discounted sum of future rewards along the trajectory.

$$\underset{\theta}{\text{maximize}}\ \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}},a\sim q}\left[\frac{\pi_\theta(a|s)}{q(a|s)}Q_{\theta_{\text{old}}}(s,a)\right] \qquad (14)$$
$$\text{subject to}\ \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}}\left[D_{\text{KL}}(\pi_{\theta_{\text{old}}}(\cdot|s)\ \|\ \pi_\theta(\cdot|s))\right] \le \delta.$$

- 샘플을 이제 뽑으면 된다

- 샘플을 뽑는 방법을 2개로 제안함 single path, vine
-
- 로0를 따르는 s0부터 state를 샘플링하는데, action은 파이 세타 올드를 사용한다.

- 파이세타 올드는 앞에서 expectation에 사용한 q(a|s)이다.(분모 분자 곱한것)

- q가 현재 policy기 때문에 값 샘플링을 구할수 있음
-
- Expectation에 현재 policy가 아니면 샘플링을 못함 만약 파이 세타면 업데이트 되기 전이기 때문에 업데이트 안된 policy로 샘플링을 할수 없음

- s와 a를 계속 모으고, 14식을 대입해서 구한다.

- DKL은 expection이니깐 s,a에 대해서 평균을 내고,
- 파이세타는 알고 있는것이고, Q는 discounted sum of future rewards로 계산한다

5. Pratical Algorithm

1. Use the *single path* or *vine* procedures to collect a set of state-action pairs along with Monte Carlo estimates of their Q-values.

2. By averaging over samples, construct the estimated objective and constraint in Equation (14).

3. Approximately solve this constrained optimization problem to update the policy's parameter vector θ. We use the conjugate gradient algorithm followed by a line search, which is altogether only slightly more expensive than computing the gradient itself. See Appendix C for details.

- Single Path나 vine방법으로 s,a를 모은다.
  몬테카를로 estimates로 Q value도 모은다.

- 모든 값의 평균을 구한다. 그게 기대값이다. 그래서 objective, constraint의 estimate값을 구한다.

- object랑 constraint를 구했으면 이것을 optimization문제로 푼다.
- conjugate gradient, line search를 사용해서 세타를 구한다. appendix에 나옴

5. Pratical Algorithm(요약)

- The theory justifies optimizing a surrogate objective with a penalty on KL divergence. However, the large penalty coefficient C leads to prohibitively small steps, so we would like to decrease this coefficient. Empirically, it is hard to robustly choose the penalty coefficient, so we use a hard constraint instead of a penalty, with parameter δ (the bound on KL divergence).
- The constraint on $D_{\mathrm{KL}}^{\max}(\theta_{\mathrm{old}}, \theta)$ is hard for numerical optimization and estimation, so instead we constrain $\overline{D}_{\mathrm{KL}}(\theta_{\mathrm{old}}, \theta)$.
- Our theory ignores estimation error for the advantage function. Kakade & Langford (2002) consider this error in their derivation, and the same arguments would hold in the setting of this paper, but we omit them for simplicity.

- KL divergence를 사용한 surrogate object함수를 이론적으로 만듬
- 근데 C가 너무 커서 penalty대신 constraint optimization 문제로 바꿈

- D KL max는 측정하기 어려워서 평균 기반의 D KL로 바꿔서 사용함

- 그래서 contraint optimization 문제를 만들고 매 이터레이션마다 optimization 문제를 풀고 그 값으로 policy를 업데이트 해나간다.

6. 결과

	B. Rider	*Breakout*	*Enduro*	*Pong*	*Q*bert*	*Seaquest*	*S. Invaders*
Random	354	1.2	0	−20.4	157	110	179
Human (Mnih et al., 2013)	7456	31.0	368	−3.0	18900	28010	3690
Deep Q Learning (Mnih et al., 2013)	4092	168.0	470	20.0	1952	1705	581
UCC-I (Guo et al., 2014)	5702	380	741	21	20025	2995	692
TRPO - single path	1425.2	10.8	534.6	20.9	1973.5	1908.6	568.4
TRPO - vine	859.5	34.2	430.8	20.9	7732.5	788.4	450.2

Table 1. Performance comparison for vision-based RL algorithms on the Atari domain. Our algorithms (bottom rows) were run once on each task, with the same architecture and parameters. Performance varies substantially from run to run (with different random initializations of the policy), but we could not obtain error statistics due to time constraints.

- 별로 안좋음 ㅋㅋ

- 완전히 새로운 방식의 policy update방식을 이론적으로 증명하고, 실제로 가능한지를 보여줌