Trust Region Policy Optimization (TRPO)

- Hard to choose stepsizes
  - Input data is nonstationary due to changing policy: observation and reward distributions change
  - Bad step is more damaging than in supervised learning, since it affects visitation distribution
    - Step too far $\rightarrow$ bad policy
    - Next batch: collected under bad policy
    - Can't recover—collapse in performance
- Sample efficiency
  - Only one gradient step per environment sample
  - Dependent on scaling of coordinates

안정적인 학습과 성능의 향상을 뒷받침할만한 이론적인 접근/근거가 필요하다.

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

$P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ is the transition probability

$r : \mathcal{S} \rightarrow \mathbb{R}$ is the reward function

$\boxed{\rho_0 : \mathcal{S} \rightarrow \mathbb{R} \text{ is the distribution of the initial state } s_0}$ → Something new
시작 state의 분포

$\gamma \in (0, 1)$ is the discount factor

Policy

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

Expected Discounted Reward (Sum)

$$\eta(\pi) = \underline{\mathbb{E}_{s_0, a_0, \ldots}}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t)\right], \text{ where}$$
$$s_0 \sim \rho_0(s_0),\ a_t \sim \pi(a_t|s_t),\ s_{t+1} \sim P(s_{t+1}|s_t, a_t).$$

Objective Function

Advantage Function

*David Silver 7강 Policy Gradient 참고

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

$$A_\pi(s, a) = Q_\pi(s, a) - V_\pi(s), \text{ where}$$
$$a_t \sim \pi(a_t|s_t), s_{t+1} \sim P(s_{t+1}|s_t, a_t) \text{ for } t \geq 0.$$

해당 부분에서 차이를 보이니 유의하세요

Schulman, John, et al. "Trust Region Policy Optimization." *Icml*. Vol. 37. 2015.

Kakade & Langford (2002)

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

Action selection하는 policy와 Advantage를 구하는 policy를 나누겠다.

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

(Unnormalized) Discounted Visitation Frequencies

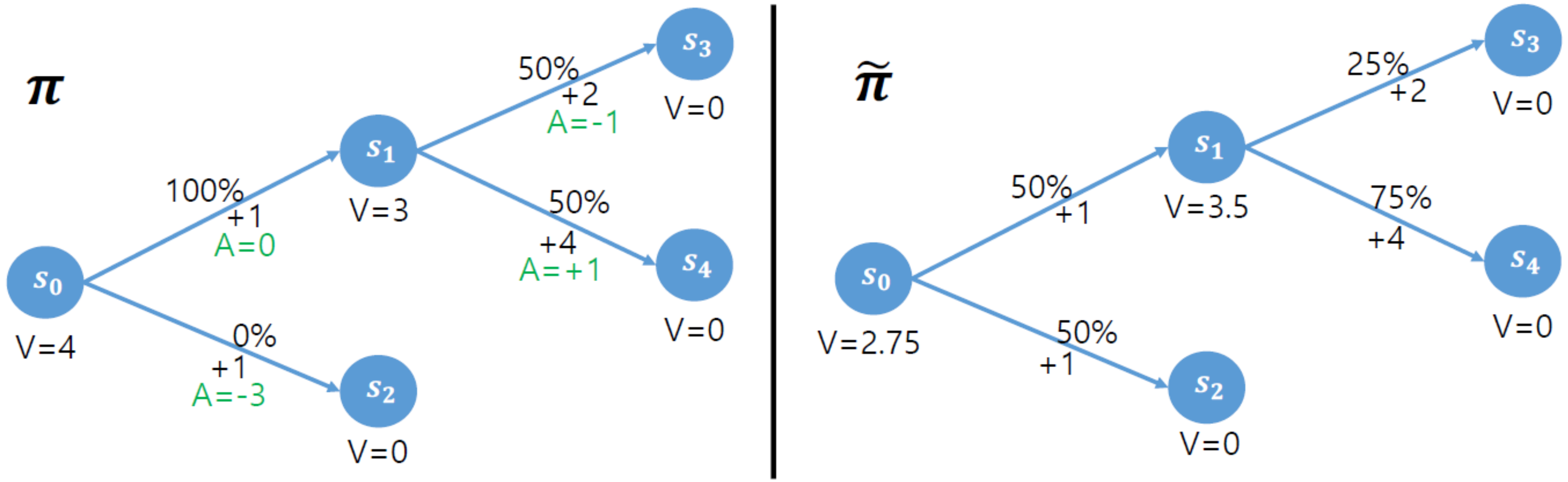
$$\boxed{\rho_\pi(s) = P(s_0 = s) + \gamma P(s_1 = s) + \gamma^2 P(s_2 = s) + \ldots,}$$

time step에 걸쳐 해당 state일 확률을 Discounted Sum을 해준다.

where $s_0 \sim \rho_0$ and the actions are chosen according to π.

Schulman, John, et al. "Trust Region Policy Optimization." *Icml*. Vol. 37. 2015.

Kakade & Langford 예시



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

Action에 대한 확률(policy)는 π~를 쓰고, Advantage는 π로 구한 걸 쓰겠다는 의미.

Kakade & Langford 증명

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

t=0부터 대입해 나열하면 규칙적으로 소거된다.

Sum over states관점으로 식을 바꿔보자

Sum of timesteps → Sum over states

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

$$\begin{aligned} \eta(\tilde{\pi}) &= \eta(\pi) + \sum_{t=0}^{\infty} \sum_s P(s_t = s|\tilde{\pi}) \sum_a \tilde{\pi}(a|s) \gamma^t A_\pi(s, a) \\ &= \eta(\pi) + \sum_s \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\tilde{\pi}) \sum_a \tilde{\pi}(a|s) A_\pi(s, a) \\ &= \eta(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) A_\pi(s, a). \quad (2) \end{aligned}$$

Timestep에서 state의 관점으로 바꾼 이유가 뭘까?(Motivation)

Schulman, John, et al. "Trust Region Policy Optimization." *Icml*. Vol. 37. 2015.

Sum over states관점으로 바꾼 식의 의미

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

1. 만일 모든 state s 에 대해 빨간 부분이 양수라면, Policy의 성능인 η가 증가하는게 보장된다는 의미
2. 그러나 approximation error가 있기때문에 저 부분이 항상 양수라는 보장이 없다.

https://github.com/minyoungjun/Pang-yo/blob/master/TRPO.pdf

Discounted Visitation Frequencies 수정 (1)

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

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

ρ를 구하려면 policy를 돌려 trajectory를 먼저 만들어야 하는데,
위 식에선 업데이트할 policy의 trajectory를 구하는 꼴이므로 최적화하기 어렵다.

따라서, 현재 policy의 ρ로 대체한다. (바꾸면서 생기는 density변화는 무시한다.)

Discounted Visitation Frequencies 수정 (2)

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

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

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

θ_0는 old parameter를 의미.

First-order(1차 미분)으로 근사하면 같다는 의미인데,

이는 충분히 작은 step으로 업데이트를 하면 L을 최대화하는 것이 η를 최대화하는 것과 같다는 말.

step이 얼마나 작아야 하는데? Conservative Policy Iteration!

$$\pi_{\text{new}}(a|s) = (1-\alpha)\underbrace{\pi_{\text{old}}}_{\text{Current policy}}(a|s) + \alpha\pi'(a|s). \qquad (5)$$

$$\pi' = \arg\max_{\pi'} L_{\pi_{\text{old}}}(\pi').$$

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

$$\text{where } \epsilon = \max_s \left| \mathbb{E}_{a\sim\pi'(a|s)} [A_\pi(s,a)] \right|. \qquad (6)$$

하지만 이 식은 mixture policy에서만 쓸 수 있어서 실질적으로 별 도움이 못된다.

좀 더 general하게 통용되는 방법론이 필요하다.

https://github.com/minyoungjun/Pang-yo/blob/master/TRPO.pdf
Schulman, John, et al. "Trust Region Policy Optimization." *Icml*. Vol. 37. 2015.

General한 방법론 : Total Variance Divergence (1)

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

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

*가장 차이가 많이 나는 state

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

α값이 정해졌다!

$$\eta(\pi_{\mathrm{new}}) \geq L_{\pi_{\mathrm{old}}}(\pi_{\mathrm{new}}) - \frac{4\epsilon\gamma}{(1-\gamma)^2}\alpha^2$$

$$\text{where } \epsilon = \max_{s,a} |A_\pi(s,a)| \qquad (8)$$

https://github.com/minyoungjun/Pang-yo/blob/master/TRPO.pdf
Schulman, John, et al. "Trust Region Policy Optimization." *Icml*. Vol. 37. 2015.

General한 방법론 : Total Variance Divergence (2)

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

α가 정의되었다.

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

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

$$D_{\text{KL}}^{\max}(\pi, \tilde{\pi}) = \max_s D_{\text{KL}}(\pi(\cdot|s) \parallel \tilde{\pi}(\cdot|s))$$

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

https://github.com/minyoungjun/Pang-yo/blob/master/TRPO.pdf
Schulman, John, et al. "Trust Region Policy Optimization." *Icml*. Vol. 37. 2015.

진짜 Performance Improvement를 보장하는가?

$\pi_{i+1} = \arg\max_{\pi} \left[L_{\pi_i}(\pi) - C D_{\mathrm{KL}}^{\max}(\pi_i, \pi)\right]$ 일 때, $\eta(\pi_0) \le \eta(\pi_1) \le \eta(\pi_2) \le \ldots$ 인가?

$M_i(\pi) = L_{\pi_i}(\pi) - C D_{\mathrm{KL}}^{\max}(\pi_i, \pi)$ 라고 할 때, (*M을 η의 surrogate function이라 한다.)

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

Argmax policy의 M에서 뺐기때문에 최소한 같거나 크다

결국, M을 최대로 만드는 것이 η의 non-decreasing을 보장해주는 셈이다.
*Minorization Maximization (a.k.a MM algorithm)

https://github.com/minyoungjun/Pang-yo/blob/master/TRPO.pdf
Schulman, John, et al. "Trust Region Policy Optimization." *Icml*. Vol. 37. 2015.

알고리즘을 좀 더 Practical하게 바꿔보자.

Practical하게 만들기 위해 아래 문제를 먼저 해결하자

1. 앞서 정의한 C의 값이 매우 큰 값이라 이를 줄이기 위해 step size가 매우 작아지는 문제가 있다.
2. KL Divergence의 Max값을 구하기 굉장히 까다롭다. (모든 state 평가가 사실상 불가능)
3. David Silver 7강을 보면, function approximation하기위해 expectation꼴로 맞춰줬었다.

https://github.com/minyoungjun/Pang-yo/blob/master/TRPO.pdf
Schulman, John, et al. "Trust Region Policy Optimization." *Icml*. Vol. 37. 2015.

들어가기 전에...

바뀐 Notation 확인하자.

$$\eta(\theta) := \eta(\pi_\theta),$$

$$L_\theta(\tilde{\theta}) := L_{\pi_\theta}(\pi_{\tilde{\theta}}),$$

$$D_{\mathrm{KL}}(\theta \,\|\, \tilde{\theta}) := D_{\mathrm{KL}}(\pi_\theta \,\|\, \pi_{\tilde{\theta}})$$

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

$$\theta = \theta_{\text{old}}.$$

Old policy를 θ_{old}로 바꾸겠다.

https://github.com/minyoungjun/Pang-yo/blob/master/TRPO.pdf
Schulman, John, et al. "Trust Region Policy Optimization." *Icml*. Vol. 37. 2015.

Step size 문제 : Penalty → Constraint

$$\underset{\theta}{\text{maximize}} \left[L_{\theta_{\text{old}}}(\theta) - \underbrace{C D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta) \right]. \quad \text{where } C = \frac{4\epsilon\gamma}{(1-\gamma)^2}.}_{\text{Penalty}}$$

$$\underset{\theta}{\text{maximize}}\, L_{\theta_{\text{old}}}(\theta) \quad \underbrace{\text{subject to } D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta) \le \delta.}_{\text{Constraint}}$$

Trust Region

C가 커서 step size가 작아지는 문제를 해결하고 좀 더 안정적이고 robust한 방식인 constraint로 Trust Region을 설정한다.

https://github.com/minyoungjun/Pang-yo/blob/master/TRPO.pdf
Schulman, John, et al. "Trust Region Policy Optimization." *Icml*. Vol. 37. 2015.

KL Divergence Max → Mean

$$\underset{\theta}{\text{maximize}}\, L_{\theta_{\text{old}}}(\theta) \quad \text{subject to } \underline{D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta) \le \delta.}$$

$$\underset{\theta}{\text{maximize}}\, L_{\theta_{\text{old}}}(\theta) \quad \text{subject to } \underline{\overline{D}_{\text{KL}}^{\rho_{\theta_{\text{old}}}}(\theta_{\text{old}}, \theta) \le \delta.}$$

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

KL Divergence의 Max값은 policy가 “모든“ states 에서 평가될 수 있다는 가정에 기반한 것인데

이는 현실세계에서 불가능한 가정이다.

Sample mean은 True mean의 unbiased estimate이므로 mean으로 대체한다.

https://github.com/minyoungjun/Pang-yo/blob/master/TRPO.pdf
Schulman, John, et al. "Trust Region Policy Optimization." *Icml*. Vol. 37. 2015.

Objective Function with Importance Sampling

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

단순히 L을 풀어 쓴 것일 뿐

$$\underset{\theta}{\text{maximize}} \sum_s \rho_{\theta_{\text{old}}}(s) \boxed{\sum_a \pi_\theta(a|s) A_{\theta_{\text{old}}}(s,a)} \text{subject to } \overline{D}_{\text{KL}}^{\rho_{\theta_{\text{old}}}}(\theta_{\text{old}}, \theta) \le \delta.$$

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

A를 최대화하는 것과 Q를 최대화 하는 것은 같다.

$$\underset{\theta}{\text{maximize}}\, \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}, a\sim q}\left[\frac{\pi_\theta(a|s)}{\boxed{q(a|s)}} \boxed{Q_{\theta_{\text{old}}}(s,a)}\right] \qquad (14)$$

여기서 q는 $\pi_{\theta_{old}}$를 의미

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

https://github.com/minyoungjun/Pang-yo/blob/master/TRPO.pdf
Schulman, John, et al. "Trust Region Policy Optimization." *Icml*. Vol. 37. 2015.

Sampling Schemes

1. Single Path

2. Vine

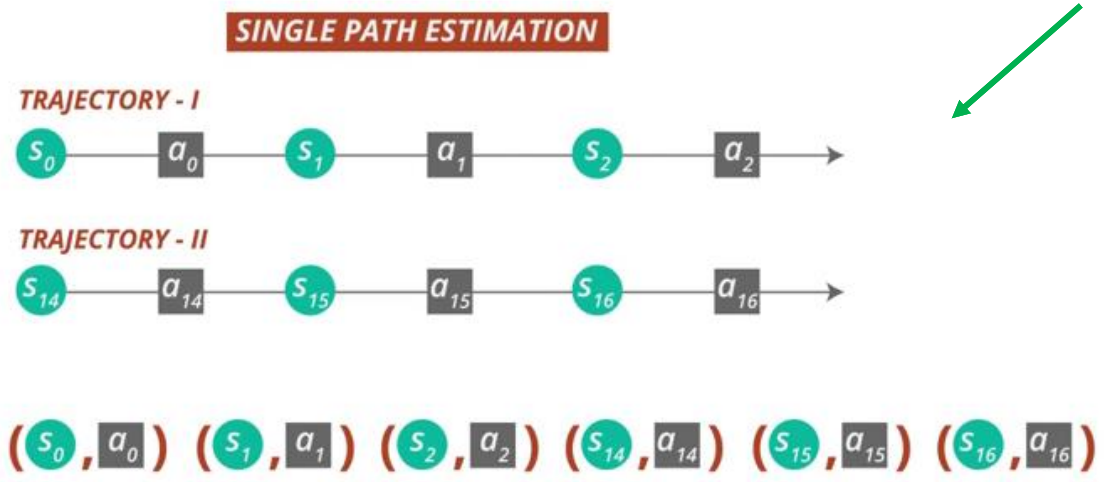
Practical하게 Estimation을 하기 위해 expectation꼴로 만들어줬고,
sampling을 통해 업데이트를 해보도록 하자.

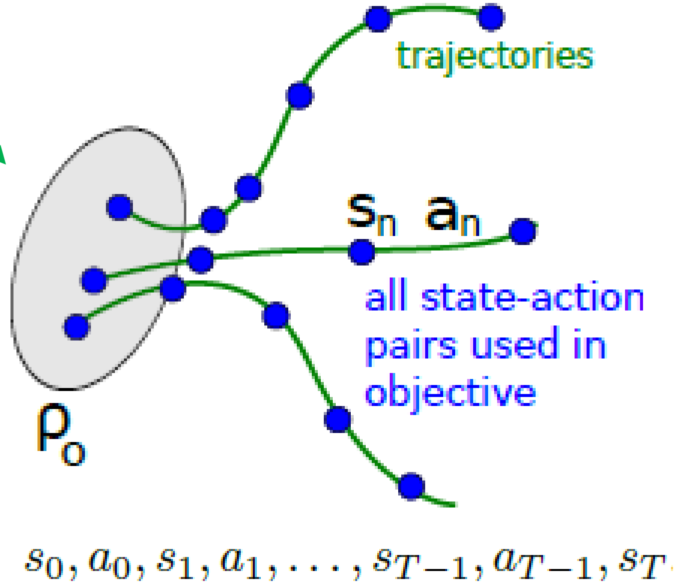
https://github.com/minyoungjun/Pang-yo/blob/master/TRPO.pdf
Schulman, John, et al. "Trust Region Policy Optimization." *Icml*. Vol. 37. 2015.

Sampling Schemes - 1. Single Path

둘 다 같은 의미

Action은 q, 즉 $\pi_{\theta_{old}}$에서 뽑는 거 잊지마세요!



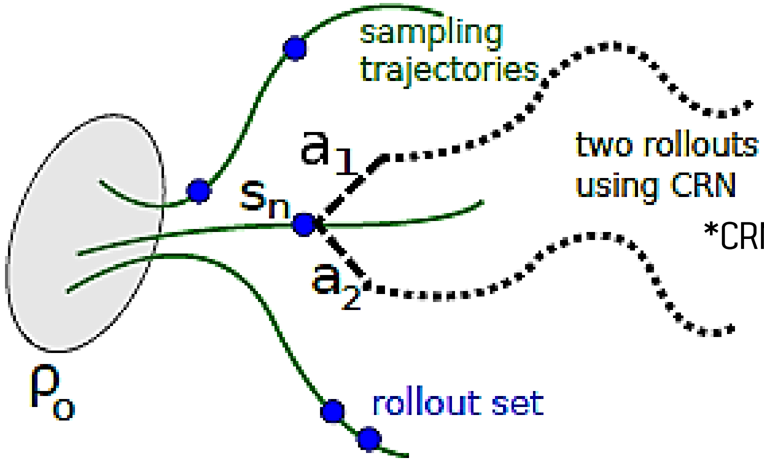


Monte-Carlo처럼 한번 쭉~가서 trajectory를 쌓고, trajectory가 끝나면
(s_t, a_t) pair별로 discounted sum of future reward를 계산하여 Q를 구한다.

https://reinforcement-learning-kr.github.io/2018/06/24/5_trpo/

Schulman, John, et al. "Trust Region Policy Optimization." *Icml*. Vol. 37. 2015.

Sampling Schemes – 2. Vine (1)



*CRN : Common Random Number

- Single Path와 같이 여러 trajectories를 만들어냅니다.
- 이 trajectories에서 N개의 state(s_1 , s_2 , …. s_n)를 뽑습니다. (rollout set)
- 각 s_n마다 K개의 action을 q에 따라 sampling합니다.
- …뭐 이런 내용인데..

Schulman, John, et al. "Trust Region Policy Optimization." *Icml*. Vol. 37. 2015.

Sampling Schemes – 2. Vine (2)

In small, finite action spaces, we can generate a rollout for every possible action from a given state. The contribution to $L_{\theta_{\text{old}}}$ from a single state s_n is as follows:

$$\boxed{L_n(\theta) = \sum_{k=1}^{K} \pi_\theta(a_k|s_n)\hat{Q}(s_n, a_k),} \quad (15)$$

얘는 왜 ratio를 곱하지 않는가…

where the action space is $\mathcal{A} = \{a_1, a_2, \ldots, a_K\}$. In large or continuous state spaces, we can construct an estimator of the surrogate objective using importance sampling. The self-normalized estimator (Owen (2013), Chapter 9) of $L_{\theta_{\text{old}}}$ obtained at a single state s_n is

$$L_n(\theta) = \frac{\sum_{k=1}^{K} \frac{\pi_\theta(a_{n,k}|s_n)}{\pi_{\theta_{\text{old}}}(a_{n,k}|s_n)}\hat{Q}(s_n, a_{n,k})}{\sum_{k=1}^{K} \frac{\pi_\theta(a_{n,k}|s_n)}{\pi_{\theta_{\text{old}}}(a_{n,k}|s_n)}}, \quad (16)$$

https://reinforcement-learning-kr.github.io/2018/06/24/5_trpo/

Schulman, John, et al. "Trust Region Policy Optimization." *Icml*. Vol. 37. 2015.

Sampling Schemes – 2. Vine (3)

정리하면, Vine이 Single Path보다 randomness가 감소해 estimate의 variance가 줄어들고
결국 더 좋은 성능을 내도록 한다.
하지만, 계산량과 메모리 사용량 등이 훨씬 많이 들기 때문에 실용적이지 못하다.

https://reinforcement-learning-kr.github.io/2018/06/24/5_trpo/
Schulman, John, et al. "Trust Region Policy Optimization." *Icml*. Vol. 37. 2015.

Estimation 과정을 요약해보자면..

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 $\boxed{14}$.

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 $\boxed{\text{C}}$ for details.

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

Single path나 Vine을 써서 trajectory를 모으고 Q value를 각 state-action pair에 대해 구한다.

구한 Q value를 평균내서 objective function과 constraint를 구한다.

제약조건이 있는 최적화 문제를 푸는데
Conjugate Gradient, Line Search 개념이 들어감.

https://github.com/minyoungjun/Pang-yo/blob/master/TRPO.pdf
Schulman, John, et al. "Trust Region Policy Optimization." *Icml*. Vol. 37. 2015.

Summary

- 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.

$M_i(\pi) = L_{\pi_i}(\pi) - C D_{\mathrm{KL}}^{\max}(\pi_i, \pi)$ *M : Surrogate Function

이론적으로 KL divergence penalty 를 사용한 surrogate objective Function을 Maximize 해도 된다는 것을 확인.

$$\underset{\theta}{\text{maximize}}\, L_{\theta_{\mathrm{old}}}(\theta) \quad \text{subject to } D_{\mathrm{KL}}^{\max}(\theta_{\mathrm{old}}, \theta) \le \delta.$$

하지만, penalty를 사용하면 C가 커지게 되고 이에 따라 step이 너무 작아지게 된다. penalty 대신 constraint 를 사용함.

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

KL Divergence Max값은 구하기 어려워 Mean으로 대체.

Advantage 함수의 측정 오차는 무시하였음.

https://github.com/minyoungjun/Pang-yo/blob/master/TRPO.pdf
Schulman, John, et al. "Trust Region Policy Optimization." *Icml*. Vol. 37. 2015.

추가 공부해야 할 사항/개념들

1. Natural Policy Gradient
2. Conjugate Gradient Algorithm
3. Line Search
4. 기타 여러 의문들