0.Abstract

본 논문에서는 반복적 절차(Iterative Procedure)를 통한 정책 향상 알고리즘을 소개하고 있습니다. TRPO라고 불리우는 이 알고리즘은 이론적으로 정책향상을 보장하는 알고리즘을 실용적으로 적용 가능하게 근사한 것입니다. 이 알고리즘은 신경망과 같이 비선형 정책을 최적화하는데 효과적인 Natural Policy Gradient 방법과 비슷합니다. TRPO는 robotic swimming, hopping 등과 같은 다양한 Task에서 좋은 성능을 보이며 연속적인 행동 공간 제어와 관련해서 Policy Gradient의 가능성을 보여준 알고리즘입니다. 또한 TRPO는 연속적인 행동 공간 뿐만 아니라 이산적인 행동 공간 환경에서도 사용이 가능합니다.

TRPO는 Trust Region Policy Optimization의 약자로, 정책에 의해 행동해야 할 행동이 확률적으로 표현되는 "확률론적 정책"의 향상을 보장하기 위해 Trust Region이라는 개념을 도입하였기에 이러한 이름이 붙여졌습니다. TRPO는 on-policy 알고리즘이기에 local-optima에 빠질 위험이 존재합니다. 그러나 local-optima에 빠질 지언정 이론적으로 정책 향상이 보장되는 장점을 가지고 있습니다. TRPO의 또다른 장점은 타 알고리즘에 비해 비교적 하이퍼-파라미터의 수가 적어 다양한 환경에서의 튜닝에 의한 노력이 크게 준다는 점입니다. 이는 TRPO가 상당히 일반화된 알고리즘임을 말하기도 합니다.

1.Introduction

최정 정책을 찾는 방법은 크게 3가지 카테고리로 묶을 수 있습니다.

1. **정책 반복 방법 (Policy Iteration Methods)**
    - 현재 정책 상에서 가치 함수를 평가하고, 이를 바탕으로 정책을 향상시키는 것을 반복하는 방법
2. **정책 경사도 방법 (Policy Gradient Methods)**
    - 샘플 궤적(Sample Trajectories)하에 얻은 보상의 합의 기댓값의 기울기를 추정량으로 하여 정책을 학습하는 방법
    - 본 논문에서는 이후 정책 경사도 방법과 정책 반복 방법과의 연결점을 이야기하고 있습니다.
3. **Derivative-free 최적화 방법**
    - Cross-Entropy Method(CEM)이나 Covariance Matrix Adaptation(CMA) Method와 같이 보상의 합을 black box 함수로두고 정책을 향상시키는 방법

본 논문의 TRPO는 **정책 경사도 방법**을 이용한 알고리즘입니다. 정책 경사도 방법은 경사도를 이용하여 보상의 합(object function)을 증가시키는 방향으로 정책을 향상시키는데, first-order 경사도를 이용한다면, 곡선이 있는 영역에서 부정확 할 수 밖에 없습니다. 예를 들어 경사도 정책에서의 step-size(경사도를 바탕으로 얼마나 많이 갱신할 지의 지표) 가 크다면, 과장된 정책 향상의 방향을 그대로 믿고 정책 업데이트를 할 수도 있게 됩니다. 이렇게 되면 정책 향상이 올바르게 수렴하지 않을 가능성이 있습니다. 그렇다고 step-size를 너무 작게하면 학습의 속도가 너무 느려진다는 단점이 있습니다.

앞으로 자세히 살펴보겠지만 TRPO는 이론적인 알고리즘을 실용적으로 사용하기 위해 근사한 알고리즘입니다. 이 근사된 알고리즘이 잘 통하기 위해서는 정책 갱신의 step-size가 충분이 작아야 합니다. 그렇다면 이 step-size가 얼마나 작아야할까요? 무턱대고 step-size를 매우 줄이면 학습 효율이 매우 떨어질 겁니다.

요약하자면 본 논문은,

1. 먼저 정책 향상이 보장되는 이론적인 알고리즘을 제시합니다.
2. 이 이론적인 알고리즘을 실용적으로 적용하기 위해 근사합니다.
3. 이렇게 근사된 알고리즘을 통해 정책을 최적정책으로 수렴시키고 싶습니다.
4. 그러나 갱신을 할 때 step-size가 너무 크면 정책이 잘 수렴하지 않습니다.
    - 이유1: first-order 경사도는 step-size가 너무 크면 overconfidence를 가지고 object function을 갱신하기 때문입니다.
    - 이유2: step-size가 너무 크면 이론적인 알고리즘을 실용적인 알고리즘으로 근사하기 어렵기 때문입니다.
5. 따라서 TRPO라는 step-size의 크기를 제한하면서, 정책 향상이 보장된 근사된 알고리즘을 제안합니다.

2.Preliminaries

1. **할인이 포함된 MDP는 다음과 같이 정의 됩니다.**

$$할인이\ 포함된\ MDP는\ 튜플\ (\mathcal{S}, \mathcal{A}, P, r, \rho_0, \gamma)로\ 정의되며,$$

$$\mathcal{S}: \ finite\ set\ of\ states$$
$$\mathcal{A}: \ finite\ set\ of\ actions$$
$$P: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}\ is\ transition\ probability\ distribution$$
$$r: \mathcal{S} \to \mathbb{R}\ is\ the\ reward\ function$$
$$\rho_0: \mathcal{S} \to \mathbb{R}\ is\ the\ distribution\ of\ the\ initial\ state\ s_0$$
$$\gamma \in (0,1)\ is\ the\ discount\ factor$$

2. **할인된 누적 보상의 기댓값(Expected Discounted Reward)**

    현재 정책을 π이고, 초기 상태를 s_0 라고 한다면 이 정책을 따랐을 때 예상되는 보상의 합은 다음의 수식과 같습니다.

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

3. 행동가치 함수(state-action value function) Q_π, 가치 함수(value function) V_π, 그리고 Advantage function A_π의 표현

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

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

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

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