Offline Meta-Reinforcement Learning with Online Self-Supervision

💡 요약: Unsupervised(Self-Supervision)을 사용한 Online Meta-RL의 일반화 성능 향상

💡 주제: Offline Meta-RL에 Unsupervised Online data(Self-Supervision)를 사용해 Meta RL시에 그냥 Online-RL하는 것만큼 일반화 성능이 향상된다.
⇒ 즉, 오프라인을 기반으로 하되, 온라인 자기 학습을 사용해, 전체 온라인 RL 하는 것 만큼의 성능을 낼 수 있다.

Glossary (용어 정리)

- 메타러닝: 머신러닝의 일반화
- self-supervised Learning: 스스로 학습 데이터에 대한 분류를 수행한다.
- 준지도학습(semi-supervised learning: 레이블이 달려 있는 데이터와 레이블이 달려 있지 않은 데이터를 동시에 사용해서 더 좋은 모델을 만드는 것
- Actor-critic 알고리즘: 상태가 주어졌을 때 Actor는 행동을 결정하고, Critic은 상태의 가치를 평가한다.
- supervision: 모델 학습에 사용되는 지도 신호 또는 레이블으로, 지도 신호를 통해, 모델이 원하는 출력을 예측할 수 있다.
- On policy: 정책 업데이트에 실제로 행동 하고 있는 최신 policy로 수집된 데이터만 사용하는 방식
- Off policy: 정책 업데이트에 어떤 데이터를 써도 관계 없고, 최근에 업데이트한 정책에서 수집된 데이터 아니여도 상관없음.
    - 정책에서 데이터를 수집하는군, 같은 데이터로 데이터는 고정되어 있고, 정책을 계속 업데이트하는구나.

⇒ 관련 참고 블로그: https://lifeisenjoyable.tistory.com/15

단어 Check Point

- denote: 나타내다.
- treat 명사: 특별한 것, 대접, 간식의 뜻
- which 해석: 이는, 어떤 것이냐면~
- bootstrap: 자기 스스로 하는, 자력으로 성공하다
- manner: 방식
- variance: 분산
- invariant: 불변
- diagnal: 대각선
- multivariate: 다변량
- critic loss: 비판적 손실
- annotate, annotation: AI에서 해석 정보(메타 데이터)를 추가하는 작업
- Parse Check - Compare A and B, Compare A to B와 같이 비교격이 오는 경우 다음에 두 문장이 옴.
    - We **compare** (our method to prior work on offline meta-RL on simulated robot locomotion) **and** (manipulation tasks and find that using additional unsupervised online data collection leads to a dramatic improvement in the adaptive capabilities of the meta-trained policies,

▼ Abstract

> 💡 **기본 Online Meta-RL은 성능이 좋지만 고비용**의 문제점이 있음,
> Offline meta-RL에 추가로 **Labell 되지 않은 Online Unsupervised data로** 방법을 적용하면 **정책을 Meta-학습**하면, 그냥 **Online Meta-RL만큼 성능이 좋아짐** !!

> 💡 그리고 재밌는 점은 여기 Abstract에서 나온, Unsupervised data가 제목의 Self-Supervision으로 추측됨.

> 💡 (궁금) 분포의 이동 유발이 왜 나올까

- Meta-RL은 기존 RL보다 더 적은 데이터로 새로운 Task에 적응하는 메타 훈련 정책
- **(문제) 온라인 메타 강화학습(Meta-RL)은 비용과 시간이 많이 듬**(비효율적)
- **(핵심) Hybrid Offline meta-RL 알고리즘** 을 제안한다. 우리는 적응 Policy을 메타 학습(훈련)하기 위해 reward와 함께 오프라인 데이터를 사용한다. ⇒ Reward와 Offline data로 학습시킨다
  - 오프라인 데이터에서 메타 훈련을 다른 Task에 대해 한 번 Reward로 Labelling된 동일한 **데이터셋을 Policy를 meta-train하는데 재사용 가능**.
  - Meta_RL은 새로운 task에 빠르게 적응할 수 있도록 데이터를 수집하는 탐색 전략(온라인)을 학습함.
  - **(문제점)** 이 정책은 고정된 오프라인 데이터 세트에 메타 학습되어서, 탐색적 전략으로 수집된 데이터에 적응할 때 행동을 예측할 수 없다. 오프라인 데이터와 체계적으로 다르다. 그러므로 분포의 이동을 유발한다.

    ⇒ 고정 오프라인 데이터로 학습하고, **탐색 데이터에 적응할 때 행동 예측이 불가능**해서 (예측하는) **분포의 이동이 발생**한다.
- 분포를 이동하기 위해 **reward-label이 되지 않은 Unsupervised Online 데이터를 추가로 수집**한다. Online data는 데이터 라벨X, 데이터를 수집이 더 저렴
- 우리 방법론을 이전에 존재하는 Offline meta-RL의 로봇 이동 및 조작과 비교해서, 추가적으로 라벨링되지 않은 Unsupervised Online data를 사용한 결과, 메타 학습 정책의 적응 능력이 매우 향상되었다.
- 우리의 방법은추가로 수집된 Online Unsupervised data는 **Meta-Train Policy**의 적응 능력이 매우 향상됐다. (일반화 성능이 향상)
- **(결론/결과)** 그리고 제안한 방법론은 새로운 작업에 대해 일반화가 필요한 도전적인 RL 도메인에서 Online Meta RL 성능 수준으로 높였다.

▼ 1. Introduction(서론)

> 💡 문장 구조가 완벽에 가까움. 본인들이 제안하려는 방법론이 어디에 기반했는지, 말하고 근거도 잘 제시하고, 방법론이 나오기 위한 “배경”을 아주 잘 설명했음

> 💡 핵심: 제안한 방법론-오프라인 보상 데이터 + Online Non Lable, unsupervised data로 Onlie raward 기반 meta-RL 만큼의 성능을 낸다.

- RL Agent는 동물과 훈련 받듯이 보상과 처벌을 통해 학습한다.
- **(문제)** 기존의 실제 deep-RL agent는 보상을 통해서 task를 학습 시키기에는 **너무 많은 시도**(시련, 보상과 처벌로 학습하는 시도)가 **필요해서** 이는 실용적이지 않다 !
- **(해결, 장점)** 그런데 **메타-RL을 사용**하면 meta-학습 테스트를 사용하고, **meta-test 때**에만, **새로운 행동에 대해서만 몇몇 시도(trials, 학습)**을 진행하면 돼서 문제를 완화시키고 소수의 궤적만 남긴다. (== reward 학습의 trials을 최소화해 효율 높임)
- 그런데 Meta-RL의 meta-train은 멀티 태스크로 수행되어서, **기존 RL보다 많은 온라인 샘플을 필요**로 하는 문제가 있다. 이는 오프라인-RL을 사용해서 해결 가능하다.
  - 오프라인-RL에서는 한 번 reward 받은 오프라인 멀티 태스크 데이터에 대해서만 처리하면 되기 때문에, 많은 온라인 샘플을 필요로 하지 않는다..
  - 그리고 라벨링된 오프라인 멀티태스크 데이터는 많은 훈련 실행을 위해 반복적으로 **재사용될 수 있다.**

- 본 논문에서는 오프라인 meta-RL에서 새로운 task에서 테스트될 때 분포의 이동이 발생하는 문제를 찾았다.
  - 문제의 원인은 정적인 오프라인 데이터에서 학습하고, meta-test 때 수집한 탐색 정책에서는 잘 동작하지 않을 것이다. 왜냐하면 **오프라인 메타-RL은 데이터를 탐색적 정책에 의해 생성해서 학습하지 않기 때문**이다. 그래서 **새로운 task에 대해 성능 저하가 발생**한다.

    ⇒ 오프라인 meta-RL으로 학습한 정책은, test 데이터, 새로운 task에 대해서 성능 저하가 발생한다.
- 그리고 우리는 보수적인 탐색 전략에서 간단하게 채택해서, 분포를 제거하고 싶지 않다. 왜냐하면, 탐색 전략은 Agent가 빠른 적응을 위해 더 나은 데이터를 선택하는 것이 불가능하기 때문이다.
- (핵심) 그래서 우리는 이러한 문제에 추가적으로 reward로 지도되지 않은 온라인 데이터를 추가함으로써 이 문제를 해결하려고 한다. 이것은 **semi-supervised meta-RL**이다.
  - 왜냐하면 라벨링되지 않은 온라인 데이터는 수집하기 더 저렴하다. 그리고 분포 이동 문제를 완화시켜 줄 수 있다. (분포 이동 문제 == 학습 정책에서 test시의

차이로 인해 잘못된 분포로 학습해, 분포 이동이 발생하는 것)

    - 메타 학습에 사용 가능하도록 만드려면 **오프라인 데이터의 라벨에 기반한 인조 보상 라벨이 필요**하다. (**Semi-Supervised**)

- 우리는 semi-supervised meta actor-critic(SMAC)를 제안한다. 이는 오프라인 데이터의 reward-label을 사용해서 자기 스스로 semi-supervised meta-RL 절차를 가능하게 한다.
- SMAC는 새로운 reward 함수를 만들기 위해 오프라인 데이터로부터 reward supervision(지도 신호/레이블)를 사용해서 학습한다.
- 이 새로운 보상함수는 자동적으로 rewards에 annotate하여 새로운 데이터에 리워드가 없는 상호작용과 메타 학습을 한다.
- 우리 논문에서는 2가지 기여를 한다. **첫 번째는 오프라인 meta-RL에서 앞서 언급한 분포 이동 문제의 증거를 특정하고 제공한다.**
- **(논문 목표)** 두 번째는 Offline meta-RL과 reward 없는 self-supervised online 데이터 finetuning을 통해 이러한 분포 변화 문제를 완화시키는 새로운 방법을 제안한다.
- **(평가)** 우리의 알고리즘(방법론)에 대해 이전에 벤치마크된 오프라인 meta-RL 방법을 평가하고, 새로운 task에 일반화가 필요한 도전적인 로봇 조작 영역에서 meta-test 때, 적은 reward-labeles trials 에 대해서 평가했다.
- 그 결과 기존 meta-RL은 training task에 대해서는 잘 적응해서 동작하지만, 새로운 태스크에 적응할 때 데이터 분포 변화를 겪는 문제가 있다. 반대로 우리의 방법은 fully labelled된 기존 온라인 meta-RL 만큼의 눈에 띄게 더 나은 성능을 나타냈다. (제안 방법에서 분포 변화가 없는지에 대한 결과는 제시하지 않았다. 아마 있었을 듯)

▼ 2. Related Work

기존의 meta-RL은 모두 Online기반에 reward-label을 요구했음. 그런데, 우리는 오프라인 reward-label 기반에, 추가적인 online non reward-label 데이터를 사용해서 성능을 개선했음.

관련 연구 파트에서는 제안한 방법론이 기존의 방법론과 어떤 차이가 있고, 특정 논문의 어떤 부분과 유사하고, 어떤 부분은 차이가 있고, 어떤 부분에서 더 뛰어난지를 적는 곳.

- **기존의 많은 meta-RL 알고리즘**은 각 **Online interaction의 episode(행동)마다, reward-label이 주어졌다**고 **가정**했음.
- **(차이점)** 이런 이전의 방법론과는 반대로, 우리의 방법은 **오프라인 과거 데이터의 보상만을 요구**하고 **추가**적인 **Online Interaction data(실시간 데이터)의 reward-signal 요구 X** (semi-supervised learning)
- 이전의 연구에서는, 라벨링된 것과 라벨링되지 않은 것들을 결합하는 다른 제형에 대해서도 연구했었다.
    - 예를 들어 모방과 역 **강화학습 방법은 오프라인 데모를 사용하여 보상 가능을 배우거나 한다.**
- **semi-supervised(준지도)나 positive-unlabeled(긍정에 대해 라벨X) reward 학습**은 RL의 보상 기능을 훈련하기 위해 **일부 상호작용에 대해 reward-label을 제공**한다.
    - 그러나 모든 이전의 방법은 single-task에 대해서만 학습이 이뤄졌다.
- 이와 반대로 우리는 **RL에서 Reward 기능을 제공할 수 있는 meta-learning 절차에 주목**했다. (== 즉, 기본 준지도 학습도, 보상 기능 훈련 위해, reward-label을 사용했는데,우리는 **reward-label을 쓰지 않고 meta-learning을 통해, 보상 기능을 학습**했다)

  ⇒ 우리는 reward 나 task에 대해 single test time이 없기 때문에 단일 보상 기능 복구에 집중하지 않아도 된다.
- **SMAC(Semi-Supervised meta actor-critic)**는 컨텍스트 기반의 적응 절차를 사용하고, 이는 기존의 레퍼런스 논문과 비슷하고 이는 목표 조건부 RL(Goal-conditional RL)과 같은 Gocontextual policies와 관련 있다.
- 대조적으로 SMAC는 reward가 **단일 목표 상태에 의해 정의되지 않거나 reward가 기본 기능을 고치지 않는다고 가정**하는 **모든 RL 문제에 적용**할 수 있다.
- 우리 방법론은 이전 Offline meta-RL 방법과 유사한 문제를 다룬다.
- 우리의 방법은 이 기존의 방법과 비슷하지만, 우리는 추가적인 Self-supervised Online fine-tuning을 추가하였고, , 우리는 앞서 언급한 분포의 변화 문제(잘못된 모델의 학습 분포 이동)를 완화시키는데서, 제안한 방법론이 뛰어난 성능을 냈다.

  ⇒ 정리하면, 기존에는 meta-RL만 제안되었는데 본 논문에서는 추가로 reward 지도 없이 Self-Supervised Online fine-tuning을 추가하고, 분포 변화 문제를 완화시키는데 뛰어난 성능을 냈다.
- 실험에서 SMAC는 학습과 보류 작업(held-out task)에서 모두 우수하게 성능이 향상되었다. 또한 **SMAC는 Unsupervised meta-learning method와 관련이 있**

다.

  - 자신의 보상으로 데이터에 주석을 다는 점에서 관련 있다. (self reward 학습 작용)

- 그래서 우리는 이런 비지도 메타학습과 대조되게, 우리는 존재하는 **리워드 있는 오프라인 데이터를 학습해서 유사한 보상을 생성**할 수 있다고 추정한다. (유사 auto-labelling)

▼ 3. Preliminaries

- Meta-Reinforcement Learning 에서 **정의된 기호**
  - meta-RL에서는 분배된 **task pt()**가 있다고 추정한다.
  - task pt()는 **Tuple Task T**로 정의되는 MDP(Markov Decision Process)이다.
    - **T=(S, A, r, γ, p0, pd)**
    - S: state 의 공간
    - A: Action의 공간
    - r: reward 기능
    - γ: discount factor (할인 요인)
    - p0: 초기 state 분포
    - pd(S(t+1)|S(t), a(t)): 상태 전이 분포 (state transition distribution)
  - **Replay buffer D**: state, action, reward, next-states 의 집합
    - **D={S(i), a(i), r(i), s'(i)}^(Nsize)** 에서 모든 보상은 같은 task에서 옴
  - h: 미니 배치 히스토리
  - h~D: h는 reply buffer D에서 샘플링된 것. (일부로 나온 것)
  - 라벨 없는 state와 action의 궤도를 나타내는 τ = (s1, a1, s2, . . .)
  - meta-episode는 Task T의 샘플과 policy π(θ)와 T 궤도의 집합으로 구성된다.
    - θ는 task와 궤도 사이에서 적응하는 정책
  - meta-episode는 마지막 궤적의 성능을 측정한다.
  - 궤적 사이에서 적응 절차는 state와 action을 변환한다.
  - 현재 메타 에피소드의 미니 배치 히스토리(h)가 컨텍스트(z = Aφ(h))로 들어간
  - 이 컨텍스트는 정책(πθ(a, | s, z)에게 적응을 위해 주어진다.

    - 정책(πθ),컨텍스트(Aφ), 컨텍스트 모음(z)의 정확한 표현은 meta-RL 사용법에 의존
        - **context z**는 기울기 업데이트에 의한 뉴럴 네트워크 **출력 가중치**가될 수 있다.
        - RNN에 의해 출력되는 숨겨진 활성화나 **확률적 인코더에 의해 출력되는 잠재 변**수가 될 수 있다.
    - meta-RL의 목표는 적응 파라미터 φ와 정책 파라미터를 학습해서 새롭게 주어진 task T에서 meta-episode의 성능을 최대화하는 것이다.
- PEARL (논문에서 정의한 용어, **actor-critic RL위한 확률적 임베딩**)
    - 우리는 offline, off-policy 기반 meta-RL 절차를 요구했기 때문에, actor-critic RL위한 확률적 임베딩을 만듬 (PEARL==Probabilistic embeddings for actor-critic RL)

      ⇒ offline meta-RL을 위해 actor-critic RL을 위한 확률적 임베딩을 만들었다.
    - z(컨텍스트 모음)은 벡터이자 적응 절차이다.
    - 분포 qφe는 파라미터 φe와 인코더로 생성되었다. 인코더는 뉴럴 네트워크의 프로세스 h이다.
        - 이는 순열 불변 방식으로 대각선 다변량 평균과 분산을 생성한다.
    - 정책은 상황에 맞는 정책 ㅠ으로 z와 state s를 합쳐서 z를 조건으로 하는 정책이다.
    - 정책 파라미터 θ는 soft actor-critic을 통해 훈련되고 이는 Qw(s,a,z)함수의 파라미터 w에 관여 한다. w는 (현재 상태, 행동, 다음 상태)에 따르는 미래 감소(할인) 보상의 합계을 측정한다. (estimates the sum of future discounted rewards)
    - actor, critic, encoder loasses는 각 작업에 대해 별도의 재생 버퍼에서 샘플링된 미니 배치를 사용하여gradient-descent로 최소화된다.
- Offline reinforcement learning

▼ 4. The Problem with Naive Offline Meta-Reinforcement Learning
▼ 5. Semi-Supervised Meta Actor-Critic
▼ 6. Experiments
▼ 7. Conclusion

8. Supplementary Material (Discussion)

느낀점

- 논문이 재밌는게, 참 잘 정리된 글처럼, 또 관련 연구 파트 소개에서는 다음 논문을 리딩하기 위한 너무나 근거 있고, 좋은 자료가 되어주네.

  ⇒ 그럼 마찬가지로 나도 좋은 논문 쓸려면 그런 근거 있고, 잘 정리된 좋은 자료를 제공해야함.