Unsupervised Meta-Learning for Reinforcement Learning

田鸿龙

LAMDA, Nanjing University

November 9, 2020

南京大學 NANJING UNIVERSITY

LAMDA Learning And Mining from DatA

Table of Contents

Table of Contents

Terminology

- task: a problem needs RL Algorithm to solve
- MDP = CMP + Reward Mechanisms
  - one-to-one correspondence between MDP and task
- CMP: controlled Markov process
  - namely the dynamics of the environments
  - consist of **state space, action space, initial state distribution, transition dynamics**...
- Reward Mechanisms: $r(s, a, s', t)$

Terminology(cont.)

- skill: a latent-conditioned policy that alters that state of the environment in a consistent way
- there is a fixed latent variable distribution $p(z)$
- $Z \sim p(z)$ is a latent variable, policy conditioned on a fixed Z as a "skill"
- policy(skill) $=$ parameter θ $+$ latent variable Z

Mutual Information

- mutual information (MI) of two random variables is a measure of the mutual dependence between the two variables
- $\mathrm{I}(x, y) = \mathrm{KL}[p(x, y) \| p(x)p(y)] = -\iint p(x, y) \ln \frac{p(x)p(y)}{p(x,y)} \mathrm{d}x\mathrm{d}y$
  - Kullback–Leibler divergence: a directed divergence between two distributions
  - the **larger** of MI, the **more divergent** between $P(x, y)$ and $P(x)P(y)$, the **more dependent** between $P(x)$ and $P(y)$
- or $\mathrm{I}(x, y) = \mathrm{H}(x) - \mathrm{H}(x \mid y)$
  - $\mathrm{H}(y \mid x) = -\iint p(x, y) \ln p(y \mid x) \mathrm{d}y\mathrm{d}x$

Table of Contents

Motivation

- Autonomous acquisition of useful skills without any reward signal.
- Why without any reward signal?
  - for sparse rewards setting, learning useful skills without supervision may help address challenges in exploration
  - serve as primitives for hierarchical RL, effectively shortening the episode length
  - in many practical settings, interacting with the environment is essentially free, but evaluating the reward requires human feedback.
  - it is challenging to design a reward function that elicits the desired behaviors from the agent(without imitation sample, hard to design a reward funciton)
  - when given an unfamiliar environment, it is challenging to determine what tasks an agent should be able to learn

Motivation(cont.)

- Autonomous acquisition of useful skills without any reward signal.
- How to define "useful skills"?
  - consider the setting where the reward function is unknown, so we want to learn a set of skills by **maximizing the utility of this set**
- How to maximize the utility of this set?
  - each skill individually is distinct
  - the skills collectively explore large parts of the state space

Key Idea: Using discriminability between skills as an objective

- design a reward function which only depends on CMP
- skills are just distinguishable ✗
- skills diverse in a semantically meaningful way ✓
  - action distributions ✗(actions that do not affect the environment are not visible to an outside observer)
  - state distributions ✓

How It Works

1. skill to dictate the states that the agent visits
   - one-to-one correspondence between skill and Z(for any certain time, parameters θ is fixed)
   - $Z \sim p(z)$, which means Z is different with each other
   - make state distributions depend on Z(vice versa.), then state distributions become diverse
2. ensure that states, not actions, are used to distinguish skills
   - given state, action is not related to skill
   - make action directly depends on skill is a **trivial** method, we better avoid it
3. viewing all skills together with p(z) as a mixture of policies, we maximize the entropy $\mathcal{H}[A \mid S]$

- Attention: 2 maybe causes the network don't care input Z, but 1 avoids it; maybe causes output(action) become same one, but 3 avoids it

$$
\begin{aligned}
\mathcal{F}(\theta) &\triangleq I(S;Z) + \mathcal{H}[A \mid S] - I(A;Z \mid S) \\
&= (\mathcal{H}[Z] - \mathcal{H}[Z \mid S]) + \mathcal{H}[A \mid S] - (\mathcal{H}[A \mid S] - \mathcal{H}[A \mid S, Z]) \\
&= \mathcal{H}[Z] - \mathcal{H}[Z \mid S] + \mathcal{H}[A \mid S, Z]
\end{aligned}
$$

How It Works(cont.)

$$\begin{aligned}\mathcal{F}(\theta) &\triangleq I(S;Z) + \mathcal{H}[A \mid S] - I(A;Z \mid S) \\ &= (\mathcal{H}[Z] - \mathcal{H}[Z \mid S]) + \mathcal{H}[A \mid S] - (\mathcal{H}[A \mid S] - \mathcal{H}[A \mid S, Z]) \\ &= \mathcal{H}[Z] - \mathcal{H}[Z \mid S] + \mathcal{H}[A \mid S, Z]\end{aligned}$$

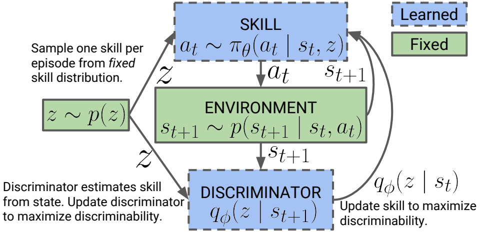
1. fix p(z) to be uniform in our approach, guaranteeing that is has maximum entropy
2. it should be easy to infer the skill z from the current state
3. each skill should act as randomly as possible

How It Works(cont.)

$$
\begin{aligned}
\mathcal{F}(\theta) &= \mathcal{H}[A \mid S, Z] - \mathcal{H}[Z \mid S] + \mathcal{H}[Z] \\
&= \mathcal{H}[A \mid S, Z] + \mathbb{E}_{z \sim p(z), s \sim \pi(z)}[\log p(z \mid s)] - \mathbb{E}_{z \sim p(z)}[\log p(z)] \\
&\geq \mathcal{H}[A \mid S, Z] + \mathbb{E}_{z \sim p(z), s \sim \pi(z)}\left[\log q_\phi(z \mid s) - \log p(z)\right] \triangleq \mathcal{G}(\theta, \phi)
\end{aligned}
$$

- $\mathcal{G}(\theta, \phi)$ is a variational lower bound

Implementation



- maxize a cumulative pseudo-reward by SAC
- pseudo-reward: $r_z(s, a) \triangleq \log q_\phi(z \mid s) - \log p(z)$

NANJING UNIVERSITY

Learning And Mining from DatA

Algorithm

Algorithm 1: DIAYN

while *not converged* **do**
- Sample skill $z \sim p(z)$ and initial state $s_0 \sim p_0(s)$
- **for** $t \leftarrow 1$ **to** $steps\_per\_episode$ **do**
  - Sample action $a_t \sim \pi_\theta(a_t \mid s_t, z)$ from skill.
  - Step environment: $s_{t+1} \sim p(s_{t+1} \mid s_t, a_t)$.
  - Compute $q_\phi(z \mid s_{t+1})$ with discriminator.
  - Set skill reward $r_t = \log q_\phi(z \mid s_{t+1}) - \log p(z)$
  - Update policy (θ) to maximize r_t with SAC.
  - Update discriminator (ϕ) with SGD.

Applications

- adapting skills to maximize a reward
- hierarchical RL
- imitation learning
- **unsupervised meta RL**

Table of Contents

Motivation

- aim to do so without depending on any human supervision or information about the tasks that will be provided for meta-testing
- assumptions of prior work ✗
  - a fixed tasks distribution
  - tasks of meta-train and meta-test are sample from this distribution
- Why not pre-specified task distribution?
  - specifying a task distribution is tedious and requires a significant amount of supervision
  - the performance of meta-learning algorithms critically depends on the meta-training task distribution, and meta-learning algorithms generalize best to new tasks which are drawn from the same distribution as the meta-training tasks
- assumptions of this work: the environment dynamics(CMP) remain the same
- **”environment-specific learning procedure”**

Attention

- this paper have been rejected(maybe twice)
- this paper make some vary strong assumption when analysising:
  - deterministic dynamics(the "future work" of 2018, but authors maybe forget it...)
  - only get a reward when the end state(two case have been concerned)
- the expriment may be not enough and convincing
- there are something wrong (at least ambiguous) in the paper...

Definition of Terminology and Symbol

- MDP: $M = (S, A, P, \gamma, \rho, r)$
- CMP: $C = (S, A, P, \gamma, \rho)$
- S: state space
- A: action space
- P: transition dynamics
- γ: discount factor
- ρ: initial state distribution
- dataset of experience(for MDP): $\mathcal{D} = \{(s_i, a_i, r_i, s_i')\} \sim M$
- learning algorithm(for MDP): $f : \mathcal{D} \to \pi$

Definition of Terminology and Symbol(cont.)

- for CMP: $R(f, r_z) = \sum_i \mathbb{E}_{\substack{\pi = f(\{\tau_1, \cdots, \tau_{i-1}\}) \\ \tau \sim \pi}} \left[\sum_t r_z(s_t, a_t)\right]$
- evaluate the learning procedure f by summing its cumulative reward across iterations

Key Idea

- from the perspective of "no free lunch theorem":
  the assumption that the dynamics remain the same across tasks affords us an inductive bias with which we pay for our lunch
- our results are lower bounds for the performance of general learning procedures

Regret for certain Task Distribution(given CMP)

- For a task distribution $p(r_z)$, the optimal learning procedure f^* is given by
  $f^* \triangleq \arg\max_f \mathbb{E}_{p(r_z)}\left[R\left(f, r_z\right)\right]$
- regret of a certain learning procedure and task distribution:
  $\mathrm{REGRET}\left(f, p\left(r_z\right)\right) \triangleq \mathbb{E}_{p(r_z)}\left[R\left(f^*, r_z\right)\right] - \mathbb{E}_{p(r_z)}\left[R\left(f, r_z\right)\right]$
- Obviously
  $f^* \triangleq \arg\min_f \mathrm{REGRET}\left(f, p\left(r_z\right)\right)$
  and
  $\mathrm{REGRET}\left(f^*, p\left(r_z\right)\right) = 0$
- f^* should be the output of traditional "meta RL algorithm"

Regret for worst-case Task Distribution(given CMP)

- evaluate a learning procedure f based on its regret against the worst-case task distribution for CMP C:
  $\mathrm{REGRET}_{\mathrm{WC}}(f, C) = \max_{p(r_z)} \mathrm{REGRET}\left(f, p\left(r_z\right)\right)$
- by this way, we do not need any prior knowledge of $p(r_z)$
- Attention: CMP may lead to **inductive bias**

Optimal Unsupervised Learning Procedure

Definition

The optimal unsupervised learning procedure f_C^* for a CMP C is defined as

$$f_C^* \triangleq \arg\min_f \textsc{Regret}_{\text{WC}}(f, C).$$

- "unsupervised" means you do not need "reward"(like DIAYN)
- f_C^* should be the output of our "unsupervised meta RL algorithm"

Optimal Unsupervised Meta-learner

Definition

The optimal unsupervised meta-learner $\mathcal{F}^*(C) = f_C^*$ is a function that takes as input a CMP C and outputs the corresponding optimal unsupervised learning procedure f_C^*:

$$\mathcal{F}^* \triangleq \arg\min_{\mathcal{F}} \textsc{Regret}_{\text{WC}}(\mathcal{F}(C), C)$$

- the optimal unsupervised meta-learner $\mathcal{F}^*$ is universal, it does not depend on any particular task distribution, or any particular CMP

Min-Max

Learning procedure

Task distribution

Returns

$$\text{Regret}(f, p) = E_{\text{task} \sim p(T)} \sum_i R(\pi_i, \text{task}) - R(\pi_i^*, \text{task})$$

$$\pi_i = f(\pi_{i-1}, \text{task})$$ ← Update of a learning procedure.

$$\pi_i^* = f^*(\pi_{i-1}^*, \text{task})$$ ← Update of the optimal learning procedure.

$$\min_f \max_p \quad \text{Regret}(f, p)$$

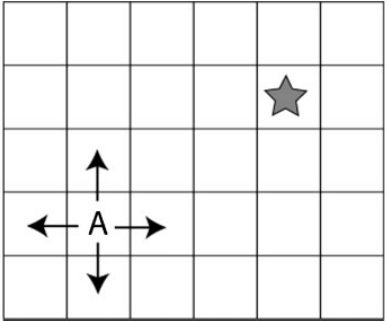
Analysis by Case Study

- Special Case: Goal-Reaching Tasks
- General Case: Trajectory-Matching Tasks
- in these case, we make some assumption such as deterministic dynamics, then generalize it

Special Case: Goal-Reaching Tasks

consider episodes with finite horizon T and a discount factor of $\gamma = 1$

reward: $r_g(s_t) \triangleq \mathbf{1}(t = T) \cdot \mathbf{1}(s_t = g)$



Optimal Learing Procedure for known $p(s_g)$

- Define f_π as the learning procedure that **uses policy π to explore until the goal is found**, and then always returns to the goal state(f is a learning procedure, which is something like SAC or PPO...)
- the goal of meta-RL (for known $p(s_g)$): find the best exploration policy π
- probability that policy π visits state s at time step $t = T$: $\rho_\pi^T(s)$
- expected hitting time of this goal state:

$$\text{HITTINGTIME}_\pi\left(s_g\right) = \frac{1}{\rho_\pi^T\left(s_g\right)}$$

- tips: "hitting time" means the **epected number of episodes** we need to make our end-state to be the goal-state(explore by the given policy π)

Optimal Learing Procedure for known $p(s_g)$(cont.)

- difinition of regret:

$$\mathrm{REGRET}\left(f, p\left(r_z\right)\right) \triangleq \mathbb{E}_{p\left(r_z\right)}\left[R\left(f^*, r_z\right)\right]-\mathbb{E}_{p\left(r_z\right)}\left[R\left(f, r_z\right)\right]$$

- regret of the learning procedure f_π:

$$\mathrm{REGRET}\left(f_\pi, p\left(r_g\right)\right)=\int \mathrm{HITTINGTIME}_\pi\left(s_g\right) p\left(s_g\right) d s_g=\int \frac{p\left(s_g\right)}{\rho_\pi^T\left(s_g\right)} d s_g$$

- exploration policy for the optimal meta-learner, π^*, satisfies:

$$\rho_{\pi^*}^T\left(s_g\right)=\frac{\sqrt{p\left(s_g\right)}}{\int \sqrt{p\left(s_g^{\prime}\right)} d s_g^{\prime}}$$

Optimal Learing Procedure for Unknown $p(s_g)$

Lemma

Let π be a policy for which $\rho_\pi^T(s)$ is uniform. Then f_π is has lowest worst-case regret among learning procedures in $\mathcal{F}_\pi$.

(proof is straight by disproval)

- finding such a policy π is challenging, especially in **high-dimensional** state spaces and in the absense of resets
- acquiring f_π directly without every computing π

Optimal Learing Procedure for Unknown $p(s_g)$(cont.)

- what we want: $\rho_\pi^T(s)$ is a uniform distribution
- how to do: define a latent variable z, make z and s_T, and sample z from a uniform distributions
- there exists a conditional distribution $\mu(s_T|z)$ (more detail later), change it to maximize the mutual information:

$$\max_{\mu(s_T|z)} I_\mu\left(s_T; z\right)$$

- still need to make sure maximize the mutual information can make s_T uniform

Optimal Learing Procedure for Unknown $p(s_g)$(cont.)

Lemma

Assume there exists a conditional distribution $\mu(s_T \mid z)$ satisfying the following two properties:

1. *The marginal distribution over terminal states is uniform:* $\mu(s_T) = \int \mu(s_T \mid z)\mu(z)\,dz = \text{Unif}(\mathcal{S})$; *and*
2. *The conditional distribution $\mu(s_T \mid z)$ is a Dirac:* $\forall z, s_T\ \exists s_z$ s.t. $\mu(s_T \mid z) = \mathbf{1}(s_T = s_z)$.

Then any solution $\mu(s_T \mid z)$ to the mutual information objective satisfies the following:

$$\mu(s_T) = \text{Unif}(\mathcal{S}) \quad \textit{and} \quad \mu(s_T \mid z) = \mathbf{1}(s_T = s_z).$$

Optimal Learing Procedure for Unknown $p(s_g)$(cont.)

- how to get $\mu(s_T|z)$?
- define a latent-conditioned policy $\mu(a \mid s, z)$
- then we have

$$\mu(\tau, z) = \mu(z) p\left(s_1\right) \prod_t p\left(s_{t+1} \mid s_t, a_t\right) \mu\left(a_t \mid s_t, z\right)$$

- get marginal likelihood by integrate the trajectory except s_T

$$\mu\left(s_T, z\right) = \int \mu(\tau, z) d s_1 a_1 \cdots a_{T-1}$$

- divide by $\mu(z)$ (which is a uniform distribution): $\mu(s_T \mid z) = \frac{\mu(s_T, z)}{\mu(z)}$
- then make $r_z\left(s_T, a_T\right) \triangleq \log p\left(s_T \mid z\right)$

南京大學 NANJING UNIVERSITY

LAMDA Learning And Mining from DatA

Optimal Learing Procedure for Unknown $p(s_g)$(cont.)

what wrong with it?

$$
\begin{aligned}
I_\mu(s_T; z) &= \mathcal{H}[S_T] - \mathcal{H}[S_T \mid Z] \\
&= \mathbb{E}_{z\sim p(z), s_T\sim\mu(s_T|z)}[\log \mu(s_T \mid z) - \log \mu(s_T)]
\end{aligned}
$$

but... how to get $\log \mu(s_T)$?

$$
\begin{aligned}
I_\mu(s_T; z) &= \mathcal{H}[Z] - \mathcal{H}[Z \mid S_T] \\
&= \mathbb{E}_{z\sim p(z), s_T\sim\mu(s_T|z)}[\log \mu(z \mid s_T) - \log \mu(z)]
\end{aligned}
$$

$\log \mu(z \mid s_T)$ is also difficult to get(because we do not have $\mu(s_T)$), but we can learn $\mu(z \mid s_T)$ directedly, just like DIAYN

General Case: Trajectory-Matching Tasks

- "trajectory-matching" tasks: only provide a positive reward when the policy executes the **optimal trajectory**

$$r_\tau^*(\tau) \triangleq \mathbf{1}\left(\tau = \tau^*\right)$$

- trajectory-matching case is actually a generalization of the typical reinforcement learning case with Markovian rewards
- hitting time and regret (for known $p(\tau^*)$)

$$\text{HITTINGTIME}_\pi\left(\tau^*\right) = \frac{1}{\pi\left(\tau^*\right)}$$

$$\text{REGRET}\left(f_\pi, p\left(r_\tau\right)\right) = \int \text{HITTING TIME}_\pi(\tau) p(\tau) d\tau) = \int \frac{p(\tau)}{\pi(\tau)} d\tau$$

General Case: Trajectory-Matching Tasks(cont.)

for unknow $p(\tau^*)$, we have lemma, again

Lemma

Let π be a policy for which $\pi(\tau)$ is uniform. Then f_π has lowest worst-case regret among learning procedures in $\mathcal{F}_\pi$.

and we maxize the object just the same as last time

$$I(\tau; z) = \mathcal{H}[\tau] - \mathcal{H}[\tau \mid z]$$

General Reward Maximizing Tasks

- that trajectory-matching is a super-set of the problem of optimizing any possible Markovian reward function at test-time
- bounding the worst-case regret on R_π minimizes an upper bound on the worst-case regret on $R_{s,a}$:

$$\min_{r_\tau \in R_\tau} \mathbb{E}_\pi \left[r_\tau(\tau) \right] \leq \min_{r \in R_{s,a}} \mathbb{E}_\pi \left[\sum_t r\left(s_t, a_t\right) \right]$$

- (bound is too loose, is it realy work?)

Algorithm

Algorithm 1 Unsupervised Meta-RL Pseudocode

Input: $\mathcal{M} \setminus R$, an MDP without a reward function

$D_\phi \leftarrow$ DIAYN() or $D_\phi \leftarrow random$

while not converged **do**

  Sample latent task variables $z \sim p(z)$

  Define task reward $r_z(s)$ using $D_\phi(z|s)$

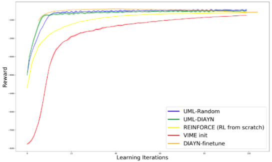
  Update f using MAML with reward $r_z(s)$

end while

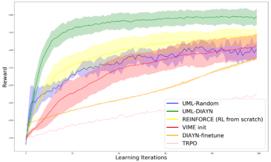
Return: a learning algorithm $f : D_\phi \rightarrow \pi$

Performance

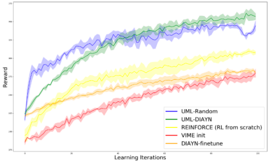
Unsupervised meta-learning accelerates learning



2D navigation



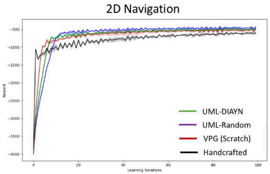
Half-Cheetah



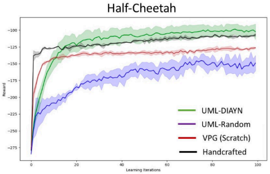
Ant

Performance(cont.)

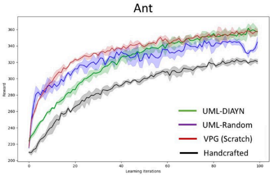
Comparison with handcrafted tasks



2D Navigation



Half-Cheetah



Ant Navigation

Discussion: 能不能再"无监督"一点？

- 文中强调了他们的算法是基于给定 CMP 的情况下的，也就是说算法对 reward mechanism 不作要求，但是要求所有的 task 都有相同的 CMP。
- 能否直接去掉"固定 CMP"的约束？✗
- 能否使用其他 meta-RL 的方法，例如 PEARL，得到关于 CMP 的 context，再根据这个 context 做 unsupervised meta-RL?

Discussion: 能不能再“有监督”一点？

- 文中一直再强调 task distribution 设计很困难，试图直接放弃设计 task distribution，直接从 CMP 中获得 prori knowledge。但是这样的方式完全抛弃了加入 expert knowledge 的可能性。
- 有没有更好的融合 expert knowledge 和 environment dynamics 的方式？
- 在 Goal-Reaching Tasks 中，如果到达 goal state 的奖赏不同，满足 min-max 的探索策略则将不再是均匀分布，而是和最终的奖赏有关。

Discussion: 结合上面两点，能不能显式的使用对抗策略，在无监督 meta-RL 和监督 meta-RL 中寻找平衡？

- 可以理解为，无监督 meta-RL 的精髓就是在给定某个特性（文中是 CMP）后，根据对抗的思想得到一个“能在最差的情况下都表现的足够好的 learning procedure”
- 文章中经过分析认为对抗的思想蕴含在“每个状态出现的频率相同”这一假设上。
- 是否可以结合前面的讨论，显式的对抗，使用更弱一点的假设，从而引入 expert knowledge。

南京大学 NANJING UNIVERSITY

LAMDA Learning And Mining from DatA

Discussion: 关于 stochastic dynamics

- 被作者遗忘的"future work"
- 同样使用 context-based 表示 dynamics
- 其实现在的方法可以直接应用在 stochastic dynamics，但是需要更多的理论证明

Thank You!