Review of “Data-Efficient Hierarchical Reinforcement Learning”

28th October, 2019

Outlines

- Motivation
- Method Overview
- Experimental Results
- Q&A and Discussion

Motivation

What is HRL?

Hierarchical Reinforcement Learning will train multiple layers of policies, in which the policy at higher layer generates small goals to policy at lower layer. In particular, the policy at the lowest layer directly interacts with the environment.

Motivation

What is HRL?

Hierarchical Reinforcement Learning will train multiple layers of policies, in which the policy at higher layer generates small goals to policy at lower layer. In particular, the policy at the lowest layer directly interacts with the environment.

Problems of Current HRL Algorithms

1. They are mostly on-policy algorithms, which are data-inefficient.
2. They mostly need a lot of task-specific designs, which are not general.

Motivation

What is HRL?

Hierarchical Reinforcement Learning will train multiple layers of policies, in which the policy at higher layer generates small goals to policy at lower layer. In particular, the policy at the lowest layer directly interacts with the environment.

Problems of Current HRL Algorithms

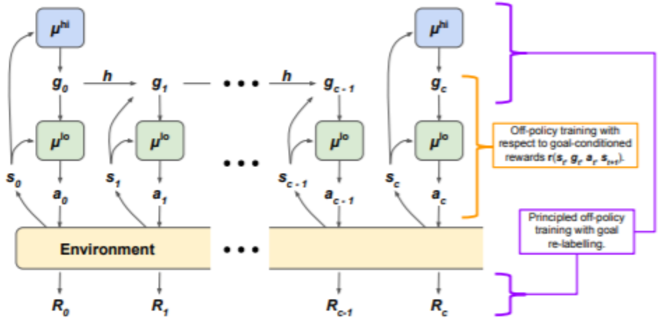
1. They are mostly on-policy algorithms, which are data-inefficient.
2. They mostly need a lot of task-specific designs, which are not general.

Challenge for Designing Off-policy HRL Algorithm

1. Past high-level goals do not correspond to current low-level agent.
2. Training can be unstable.

Model and Method Overview

A two-layer HRL algorithm.



μ^{hi}
g_0
h
g_1
h
g_{c-1}
g_c
μ^{lo}
s_0
a_0
s_1
a_1
s_{c-1}
a_{c-1}
s_c
a_c
Environment
R_0
R_1
R_{c-1}
R_c
Off-policy training with respect to goal-conditioned rewards $r(s_t, g_t, a_t, s_{t+1})$.
Principled off-policy training with goal re-labelling.

1. Collect experience $s_t, g_t, a_t, R_t, \ldots$.
2. Train μ^{lo} with experience transitions $(s_t, g_t, a_t, r_t, s_{t+1}, g_{t+1})$ using g_t as additional state observation and reward given by goal-conditioned function $r_t = r(s_t, g_t, a_t, s_{t+1}) = -||s_t + g_t - s_{t+1}||_2$.
3. Train μ^{hi} on temporally-extended experience $(s_t, \tilde{g}_t, \sum R_{t:t+c-1}, s_{t+c})$, where $\tilde{g}_t$ is re-labelled high-level action to maximize probability of past low-level actions $a_{t:t+c-1}$.
4. Repeat.

Model and Method Overview (Continued)

The training for high-level and low-level policies can be seen as training on two different but related MDPs.

High-level MDP

Data buffer content: $(s_t, \tilde{g}_t, \sum R_{t:t+c-1}, s_{t+c})$

- s_t: MDP state, which is the same as environment state
- s_{t+c}: state after transition, which is environment state after c steps of transition
- g_t: action, functioning as the goal of low-level policy

$$g_t = \begin{cases} \mu^{high}(s_t) & \text{if } t \equiv 0 \mod c \\ s_{t-1} + g_{t-1} - s_t & \text{otherwise} \end{cases}$$

- $\tilde{g}_t$: action used for high-level training, modified by off-policy correction (more details later)
- $\sum R_{t:t+c-1}$: MDP reward, summation of c steps' environment reward.

Model and Method Overview (Continued)

Low-level MDP

Data Buffer Content: $((s_t, g_t), a_t, r_t, (s_{t+1}, g_{t+1}))$

- (s_t, g_t): MDP state, where s_t is environment state and g_t is the action of high-level policy
- a_t: action, which interacts directly with the environment
- r_t: MDP reward, which is goal-conditioned, defined as $r_t = -\|s_t + g_t - s_{t+1}\|_2$
- (s_{t+1}, g_{t+1}): state after transition

Model and Method Overview (Continued)

Low-level MDP

Data Buffer Content: $((s_t, g_t), a_t, r_t, (s_{t+1}, g_{t+1}))$

- (s_t, g_t): MDP state, where s_t is environment state and g_t is the action of high-level policy
- a_t: action, which interacts directly with the environment
- r_t: MDP reward, which is goal-conditioned, defined as $r_t = -\|s_t + g_t - s_{t+1}\|_2$
- (s_{t+1}, g_{t+1}): state after transition

Instantiation Algorithm

Separately train high-level and low-level policies by using an off-policy algorithm. Specifically, the authors used TD3 algorithm, which is a variant of DDPG.

Off-policy Correction

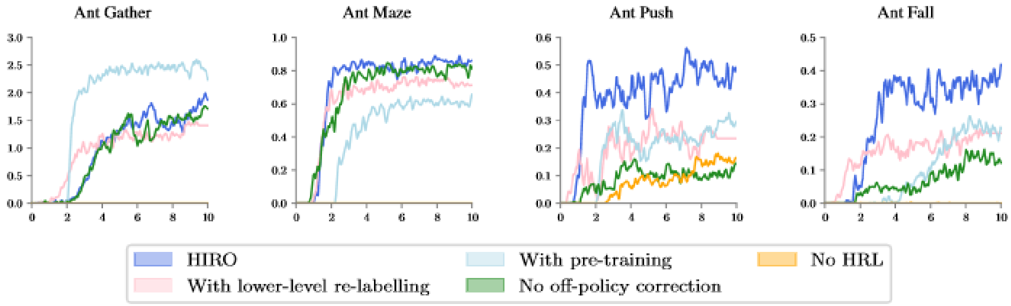
As the lower-level policy evolves, the lower-level actions taken to achieve the same goal varies, which in turns results in different states and rewards collected along the way. This invalidates old high-level transitions $(s_t, g_t, \sum R_{t:t+c-1}, s_{t+c})$.

- Need to relabel the goal for high-level training
- Choose $\tilde{g}_t$ such that
  - The probability $\mu^{\text{low}}(a_{t:t+c-1}|s_{t:t+c-1}, \tilde{g}_{t:t+c-1})$is maximized
  - The intermediate goals are calculated by $\tilde{g}_{t+j+1} = s_{t+j} + \tilde{g}_{t+j} - s_{t+j+1}, \; \forall j : 0 \leq j \leq c-2$
- Practically, in authors' implementation, the maximization is taken over 10 candidate goals, in which eight are sampled Gaussian centered at $s_{t+c} - s_t$, one is the original g_t and the last one is $s_{t+c} - s_t$

Video Illustration

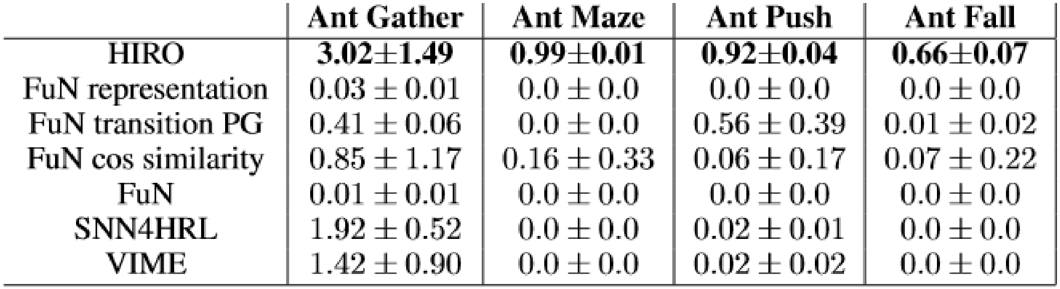
Video Link

Comparison with variants of the HIRO model



Comparison with other models

- Comparison with FuN, SNN4HRL VIME, on rewards of the best policy obtained in 10M steps of training,averaged over 10 randomly seeded trials with standard error



	Ant Gather	**Ant Maze**	**Ant Push**	**Ant Fall**
HIRO	**3.02±1.49**	**0.99±0.01**	**0.92±0.04**	**0.66±0.07**
FuN representation	0.03 ± 0.01	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
FuN transition PG	0.41 ± 0.06	0.0 ± 0.0	0.56 ± 0.39	0.01 ± 0.02
FuN cos similarity	0.85 ± 1.17	0.16 ± 0.33	0.06 ± 0.17	0.07 ± 0.22
FuN	0.01 ± 0.01	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
SNN4HRL	1.92 ± 0.52	0.0 ± 0.0	0.02 ± 0.01	0.0 ± 0.0
VIME	1.42 ± 0.90	0.0 ± 0.0	0.02 ± 0.02	0.0 ± 0.0

Summary and Critiques

Summary of Strengths

- This paper propose a noval way to do off-policy HRL high-level training, by correcting goals used in high-level training.
- Data-Efficient -off-policy
- Generality -goals are learned and proposed automatically by the higher-level controllers

Summary and Critiques

Summary of Strengths

- This paper propose a noval way to do off-policy HRL high-level training, by correcting goals used in high-level training.
- Data-Efficient -off-policy
- Generality -goals are learned and proposed automatically by the higher-level controllers

Critiques of Weakness

- In the Ant Gather case, with the high uncertainty, it is less sure that HIRO performs much better.
- Would be more convincing with more experiments on other settings
- Lack of discussion on the aspects of the models in the experiment section
- Discussion on the settings where this model is particularly helpful and where it might fail or not behave as expected

Discussion and Q&A

Thank you very much!